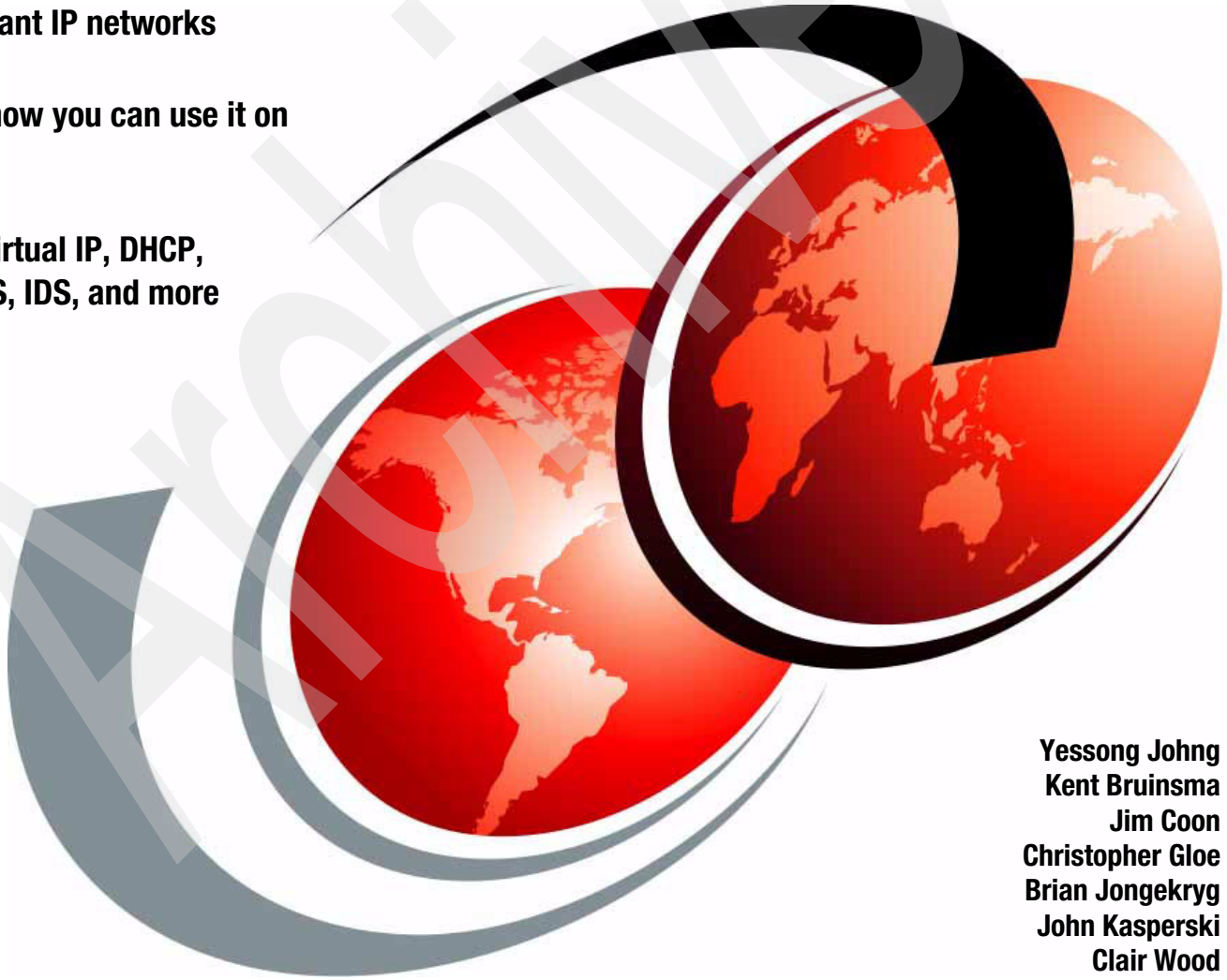
IBM

IBM i5/OS IP Networks: Dynamic

- Hints for secure, self-configuring, fault-tolerant IP networks
- IPv6 and how you can use it on i5/OS
- How-to: Virtual IP, DHCP, DDNS, QoS, IDS, and more



Yessong Johng
Kent Bruinsma
Jim Coon
Christopher Gloe
Brian Jongekryg
John Kasperski
Clair Wood

Redbooks

ibm.com/redbooks

International Technical Support Organization

IBM i5/OS IP Networks: Dynamic

June 2007

SG24-6718-02

Note: Before using this information and the product it supports, read the information in “Notices” on page ix.

Third Edition (June 2007)

This edition applies to IBM i5/OS V5R4 (product number 5722-SS1).

Contents

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

Redbooks (logo)  ®
eServer™
iSeries®
i5/OS®
pSeries®
AIX 5L™
AIX®
AS/400e™
AS/400®
Domino®
DRDA®
Electronic Service Agent™
Integrated Language Environment®
IBM®
Language Environment®
Network Station®
NetServer™
OS/400®
PAL™
POWER™
Redbooks®
System i™
System i5™
System/36™
SAA®
WebSphere®

The following terms are trademarks of other companies:

IPX, Java, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Active Directory, Windows NT, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Pentium, Intel logo, Intel Inside logo, and Intel Centrino logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

Preface

Over the course of many years, the developers in both the Endicott and Rochester labs have been working very hard adding functions to each release of OS/400® and IBM® i5/OS® to make the configuration and use of the IBM System i™ platform in a TCP/IP network easier and more powerful. If you need to design an IP network that is self-configuring, fault-tolerant, secure, and efficient in its operation, then this IBM Redbooks® publication is for you.

We start low with the details of IP interface and route implementation on an i5/OS server on a System i platform. Through the study of these building blocks, we show how to create IP networks that are easier to configure, tolerant of faults, and can perform both inbound and outbound load balancing. The topics we study are virtual IP addresses (VIPA), Network Address Translation (NAT), Classless InterDomain Routing (CIDR), Virtual Ethernet, and RIPv2.

i5/OS now has the capability to configure and test a new generation of IPv6 applications. This book gives you an introduction to IPv6 including benefits, features, and configuration. IPv6 promises to truly make IP Networks dynamic.

At the link layer, we take advantage of numerous enhancements in the SLIC-based TCP/IP stack. One such enhancement called Multilink Protocol enables a single logical connection to share multiple links providing dynamic bandwidth allocation when you need it and increased fault tolerance in the event of single-link failures. In addition, the i5/OS Point-to-Point Protocol (PPP) has been enhanced to provide additional services such as dynamically assigning IP addresses from DHCP and RADIUS servers in your network and providing PPP over Ethernet (PPPoE). No longer must you manually manage these remote clients that connect to your IP network.

i5/OS has always had many built-in Network Security features. These features have been enhanced to include an Intrusion Detection System (IDS). This allows you to be notified of attempts to hack into, disrupt, or deny service to the system.

Moving up to the application layer, this book demonstrates the dynamic power of IP by having the DHCP server assigning IP addresses and automatically updating the i5/OS Dynamic DNS. Now clients and servers can be added dynamically to the IP network and assigned a name automatically. In addition, we have Quality of Service (QoS), which enables you to define policies to regulate the flow of inbound and outbound TCP/IP connections and data.

This revision of the book includes many new and updated topics that are critical to know if you are currently using or planning to use a System i platform in your network.

Note: Throughout this book, the following terms are used accordingly:

- System i platform: System i platform includes iSeries® servers and System i5™ servers unless it is specifically defined in a particular context. As a hardware platform, you can install multiple operating systems such as i5/OS, Linux®, and AIX® 5L™ on a single System i platform.
- i5/OS: i5/OS includes OS/400 and i5/OS operating systems unless it is specifically defined in a particular context. As operating systems services, most of TCP/IP implementation discussed in this book is pertinent to i5/OS rather than the System i platform.

The team that wrote this book

This book was produced by a team of specialists from around the world working at the International Technical Support Organization, Rochester Center.

Kent Bruinsma is a Senior Software Engineer at IBM Rochester. He started with IBM in 1984 on PC Support/36. He has since worked on several SNA communications projects including 5494 Remote Workstation Controller and Display Station Passthru. He spent six years on the HTTP Server for i5 and was the technical lead on the team that brought Apache to the i5 system. He worked on WebSphere® Business Integration products on i5 for two years, then on the Directory Server (LDAP) for another two years. In 2007, he became the team leader for the TCP/IP Applications on i5, which includes FTP, Telnet, DNS, DHCP, Mail, QoS, and LDAP.

Jim Coon has been an IBM Software Engineer for 22 years. He started out working on the IBM System 36 in networking and communications. He then migrated to the AS/400®, and now the iSeries. In addition to networking and communications, he has worked on retail support, Digital Certificate Manager, and now Intrusion Detection.

Christopher Gloe (Chris) is a Senior Software Engineer at IBM in Rochester Minnesota where he has worked for 20 years. He is currently the Technical Chief Engineering Manager (TCEM) and Chief Architect for TCP/IP Networking on i5/OS. During the last 10 years he has designed and implemented various portions of the System i TCP/ IP Communications stack and associated components.

Brian Jongekryg is an Advisory Software Engineer at IBM Rochester. He has worked as a developer on the IBM System/36™ in networking and communications and subsequently on the IBM AS/400 and now System i platform, where he is currently a Developer on the i5/OS TCP/IP stack team.

John Kasperski is an Advisory Software Engineer at IBM Rochester. He started his IBM career in 1991 and spent five years designing/developing various networking-related projects in the Networking Software and Networking Hardware Divisions at the IBM Research Triangle Park site in NC. He transferred to IBM Rochester in 1996 and joined the OS/400 Sockets/SSL development team. He designed and implemented various socket enhancements for the V4R2 through V5R3 releases. He became the team leader of the Sockets/SSL development team in 2001. John has presented numerous Sockets and TCP/IP related topics at COMMON conferences for the past five years. John is currently the team leader of the i5/OS TCP/IP stack development team and the Project Lead for IPv6 on i5/OS.

Clair Wood graduated with a Bachelor of Science degree in Computer Science from Brigham Young University in 1988. After graduation, he started working at IBM as a Software Engineer on the AS/400 Device Configuration team. Other past work assignments at IBM have included working on the Source/Sink and Work Management teams. His current job is Team Leader of the i5/OS TCP/IP Configuration Team.

Yessong Johng is an IBM Certified IT Specialist at the IBM International Technical Support Organization, Rochester Center. He started his IT career at IBM as a S/38 Systems Engineer in 1982 and has been with S/38, AS/400, and iSeries for 20 years. He writes extensively and develops and teaches IBM classes worldwide on the areas of IT Optimization, whose topics include Linux, AIX, and Windows® implementations on System i platform. His other coverage areas include TCP/IP, networking security, HA, and SAN.

Thanks to the following people for their contributions to this project:

Brian R. Smith
Pallav Agrawala

Mihai Badea
Jay Johnson
Gary Lakner
Allyn Walsh

Dave Christenson
Rick Hemmer
Craig Jacquez
Scott McCreadie
K.C. Meng
Paul Rieth
Tim Seeger
Jeff Stevens
John C. Wingertsman
The previous authors of the second edition of this book

James G. Johnson
Makoto Kikuchi
Axel Lachmann
Rian Lemmer
The previous authors of the first edition of this book

Mark Bauman
Dave Boutcher
Ken Brown
Joe Cors
John Hall
Garrett Lanzy
Kyle Lucke
John McGinn
Dave Murray
Tom Murphy
Kristine Ryan
Jeff Stevens
Mark Vanderwiel
Dan Vega
IBM Rochester Development Lab

Marcela Adan
Selwyn Dickey
Vess Natchev
Lloyd R. Perera
iSeries Technology Center (iTC) Rochester

Wayne Bowers
Johnnie Talamantes
James Johnson
Tom Kelly
Craig Nordmoe
Mervyn Venter
Russ Young
IBM Rochester Support Center

Yoshio Kaneko
Makoto Kikuchi
IBM Japan

Bryan Dietz
3X Corporation, USA

Richard Morley
Support Manager - IBM iSeries and AS/400, Rebus Insurance Systems Limited, UK

John Taylor,
Technical Director, Typex Group plc, UK

Sections of this IBM Redbooks publication were prepared with assistance by Information Development at IBM Rochester.

Become a published author

Join us for a two- to six-week residency program to help write an IBM Redbooks publication dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You will team with IBM technical professionals, Business Partners, or customers.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you will develop a network of contacts in IBM development labs and increase your productivity and marketability.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us because we want our Redbooks publication to be as helpful as possible. Send us your comments about this or other Redbooks in one of the following ways:

- Use the online **Contact us** review redbooks form found at:

  ibm.com/redbooks

- Send your comments in an e-mail to:

  redbook@us.ibm.com

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400

Part 1

Dynamic IP

Over the course of many years the developers in both the Endicott and Rochester IBM labs have been working very hard for you. They have been adding functions to each release of OS/400 and i5/OS to make the configuration and use of the IBM System i in a TCP/IP network easier and easier.

One of the most obvious ways that they have done this is to design a GUI that redefines how you can configure and administer the TCP/IP protocol suite on your System i. Every release of OS/400 and i5/OS added more function to iSeries Navigator (in V5R1 and previous releases this function was named Operations Navigator) to the point where the traditional 5250 *green screen* is no longer an option for configuration.

Each release of OS/400 and i5/OS also saw a more insidious growth in the functionality of the TCP/IP protocol suite that in turn directly increases your ability to create and manage secure, self-configuring, and self-healing networks. In this part we detail a wide range of function and protocols built into the IBM System i, which can help you automate your own TCP/IP network. We study these functions one at a time to:

- Tell you what the function is and why you would use it.
- Define the characteristics of the function.
- Provide you with basic examples of how to configure this function using iSeries Navigator.

In Part 2, “Scenarios” on page 177, we define problem scenarios and their solutions. We then demonstrate how to apply these functions to solve your networking problems. In some cases our scenarios are simple and encompass just a single function. As the scenarios become more and more complex, the solutions start to include a mixture of functions that interoperate and rely on each other.

1

An introduction to the protocols at hand

Transmission Control Protocol/Internet Protocol (TCP/IP) is the protocol suite that was developed to enable communication between computers. This set of protocols is used to allow a connection between heterogeneous systems and enable them to communicate meaningfully. The TCP/IP protocol suite is made up of many components, including applications, transmission protocols, routing protocols, and so on. Many times TCP/IP applications are also a protocol, such as File Transfer Protocol (FTP). Basic FTP, viewed as an application, copies files from one computer to another. But FTP is also a protocol, and there is an agreed-upon set of rules that the clients and servers use to communicate with each other. In this book we focus on the service below the application, that is, the protocols that control the routing and transmission of data.

We recommend that you read the IBM Redbooks publication *TCP/IP Tutorial and Technical Overview*, GG24-3376, to learn more about the components of TCP/IP and the functions provided by each standard TCP/IP application. Also, *V4 TCP/IP for AS/400: More Cool Things Than Ever*, SG24-5190, and *V5 TCP/IP Applications on the IBM eServer iSeries Server*, SG24-6321, provide a wide range of hints and tips for installing and configuring your IBM i5/OS server TCP/IP protocol suite of applications.

You can offer a wide range of services to your users with TCP/IP as the basis for your network, such as:

- Create fault-tolerant networks capable of routing around failures and keeping your business-critical applications up and running longer.
- Create a network that has defined performance objects so that business-critical applications enjoy good performance.
- Balance the load across many interfaces to improve throughput and in many cases increase the fault-tolerant nature of the network.
- Take advantage of numerous advances in security and network management. Just because your network is dynamic does not mean it is unsecured.

The i5/OS server implements many TCP/IP functions in V4 and V5 of OS/400 and i5/OS. Depending on the version you are running, some functions may not be available. Refer to Table 1-1 to determine which release is required for the function that you want to perform.

Note: Table 1-1 is not meant to be an all-inclusive list of TCP/IP functions available in OS/400 and i5/OS.

Table 1-1 i5/OS TCP/IP function availability reference

Function	Short description	V4 R1	V4 R2	V4 R3	V4 R4	V4 R5	V5 R1	V5 R2	V5 R3	V5 R4
Interfaces, routes, APIs, and related enhancements										
Sockets API	A socket is a communications connection point (endpoint) that you can name and address in a network. Sockets enable you to exchange information between processes on the same machine or across a network, distribute work to the most efficient machine, and allow easy access to centralized data. Socket APIs are the network standard for TCP/IP. i5/OS sockets support multiple transport and networking protocols.	X	X	X	X	X	X	X	X	X
	IP multicast and the sendmsg() and recvmsg() updates that enable descriptors to be passed between jobs.		X	X	X	X	X	X	X	X
	Thread safe socket system functions; socket network functions; accept_and_recv() / send_file() APIs. Also, new SSL sockets APIs to create secure socket applications on i5/OS server.			X	X	X	X	X	X	X
	Socket API performance enhancements; new send_file64() API, and Performance Explorer (PEX) trace points for API entry/exit.				X	X	X	X	X	X
	Several new APIs related to socket programming: asynchronous I/O; Global Secure ToolKit (GSKit) API; serviceability enhancements (set internal trace points useful for debug).					X	X	X	X	X
	QoS IntServ support for the sockets Resource Reservation Protocol (RSVP) API. Also, AF_UNIX_CCSID address family support; Berkeley Internet Name Domain (BIND) API support; and REXEC API support.						X	X	X	X
	IPv6 sockets API support for the AF_INET6 address family. Also X/Open Single UNIX® Specification compatibility; SSL performance improvements; Xsocket tool updates; and the removal of support for IPX/SPX and AF_NS address family. Also, new qtoq QoS sockets APIs to simplify the work required to use IntServ on the i5/OS server.							X	X	X
ICMP	Enables the generation of error messages, test packets, and information messages related to IP.	X	X	X	X	X	X	X	X	X
IGMP	Internet Group Management Protocol.	X	X	X	X	X	X	X	X	X

Function	Short description	V4 R1	V4 R2	V4 R3	V4 R4	V4 R5	V5 R1	V5 R2	V5 R3	V5 R4
Virtual IP addresses (VIPA)	Enables the i5/OS server to be known by a single IP address, even when it is attached to many different networks. Some of the more common uses of VIPA are load balancing, fault tolerance, and as anchors for unnumbered interfaces.			X	X	X	X	X	X	X
	A VIPA is now directly routable (responds to ARP) and is bound to an interface. Excellent fault-tolerant solution.							X	X	X
	A preferred interface list can now be specified so that you can control the list of interfaces the VIPA can be bound to (the interface that will respond to ARP). This also allows interfaces within a different subnet to respond to ARP on behalf of a VIPA address. These enhancements provide an even more flexible fault-tolerant solution.									X
ARP	Address Resolution Protocol (ARP) is used to map the IP address of the destination host to the physical MAC address that is ultimately needed for communication between two LAN connected systems.	X	X	X	X	X	X	X	X	X
	Proxy ARP enables physically distinct networks to appear as though they are a single, logical network. The advantage of this technique is that it provides connectivity between these physically separate networks without creating any new logical networks and, more important, without updating any route tables. Automatic (or *transparent)* connectivity is provided between the two networks.	X	X	X	X	X	X	X	X	X
	Duplicate IP address detection. This is done at both interface start time and for active interfaces at every ARP cache time out.						X	X	X	X
	ARP Cache can be viewed and cleared via iSeries Navigator.						X	X	X	X
IP over an Opticonnect bus	The Opticonnect bus may either be a true, physical opticonnect bus or it may be an internal *virtual opticonnect* bus used to connect partitions in an LPAR system. The TCP/IP interfaces can be configured so that the opticonnect bus appears to TCP/IP as an emulated LAN or as a series of point-to-point connections.				X	X	X	X	X	X
IP over Virtual Ethernet	A virtual Ethernet is an internal Ethernet network used to connect partitions in an LPAR environment. Similar to virtual opticonnect, virtual Ethernet is an alternative when running a non-i5/OS operating system in other partitions. In such cases, virtual Ethernet should be used to communicate between partitions because the opticonnect device driver is only supported on i5/OS. TCP/IP over virtual Ethernet must be configured as a LAN interface.						X	X	X	X

Function	Short description	V4 R1	V4 R2	V4 R3	V4 R4	V4 R5	V5 R1	V5 R2	V5 R3	V5 R4
PPP	Point-to-Point Protocol (PPP) enables system-to-system connection and data exchange through a modem and over a leased or telephone line. iSeries Navigator supports this point-to-point connection as a part of its wide area network (WAN) connectivity. Also, PPP dial-on-demand for WAN links, including frame relay.		X	X	X	X	X	X	X	X
	Multilink support that enables multiple PPP links to be grouped together to form a single virtual link or bundle.						X	X	X	X
	Remote Authentication Dial-In Service (RADIUS) is a distributed security system developed by Lucent Technologies InterNetworking Systems. RADIUS is the de facto industry standard for user authentication, authorization, and accounting.						X	X	X	X
	Dynamic Resource Sharing for analog connections.							X	X	X
	PPP over Ethernet (PPPoE) provides the ability to connect a network of hosts over a simple bridging access device to a remote access concentrator. i5/OS acts as client only.							X	X	X
	Remote Dial Capability allows the sharing of a modem between partitions or systems (using L2TP and PPP).								X	X
SLIP	Serial Line Internet Protocol (SLIP) enables system-to-system connection and data exchange over serial lines. Use iSeries Navigator to configure and administer SLIP connections. We recommend using PPP rather than SLIP. Both are supported.	X	X	X	X	X	X	X	X	X
Explicit Route Binding	By explicitly binding a route to an interface you can distribute traffic (load balancing) across multiple interfaces so that all routes do not use the same interface to reach the network. The two parameters on the Add TCP/IP Route (ADDTCPRTE) command are BINDIFC and DUPRTEPTY.		X	X	X	X	X	X	X	X
	With the addition of Schowler routes, this same method of load balancing now works equally well with local or remote clients.			X	X	X	X	X	X	X
CIDR	Classless Inter-Domain Routing (CIDR) (also sometimes called supernetting) is a way to combine several Class C network address ranges into a single network or route.			X	X	X	X	X	X	X

Function	Short description	V4 R1	V4 R2	V4 R3	V4 R4	V4 R5	V5 R1	V5 R2	V5 R3	V5 R4
RouteD (RIP)	Change routing paths on the TCP/IP servers. RouteD is the i5/OS RIP server. Enables the i5/OS to send and accept notifications of network changes in order to dynamically update host or gateway route tables. RIP version 1 was introduced in V4R1. Use iSeries Navigator to monitor and manage the RouteD server.	X	X	X	X	X	X	X	X	X
	RIP Version 2 (RIPv2) with CIDR and VLSM. RIPv2 is the only application that supports multicasting.		X	X	X	X	X	X	X	X
Tools and supporting protocols										
PING	Packet InterNet Groper (PING) is a tool that can be used to check connectivity to a system. Type `PING` or `VFYTCPCNN` on the command line.	X	X	X	X	X	X	X	X	X
	PING also available via iSeries Navigator GUI.						X	X	X	X
Trace Route	Tool to trace the route of IP packets to a user-specified destination system. The route can involve many different systems along the way. Each system along the route is referred to as a hop. You can trace all hops along the route or specify the starting and ending hops to be traced. Available via iSeries Navigator GUI and 5250 command line as Trace TCP/IP Route (TRACEROUTE).						X	X	X	X
NSLOOKUP	Tool to query DNS servers for information. Called via an application program in V4R2. Use the command NSLOOKUP for V4R3 and beyond.		X	X	X	X	X	X	X	X
	Look Up Host added to iSeries Navigator as a simple GUI tool to map host name to IP address and IP address to host name.						X	X	X	X
NETSTAT	Enables a system administrator to monitor and control the network status of an i5/OS system running TCP/IP or APPC over TCP/IP applications by displaying information about interfaces, routes, and application connections.	X	X	X	X	X	X	X	X	X
	NETSTAT also available via iSeries Navigator GUI.						X	X	X	X
	Enhanced to include IPv6 interfaces, routes, and connections.							X	X	X

Function	Short description	V4 R1	V4 R2	V4 R3	V4 R4	V4 R5	V5 R1	V5 R2	V5 R3	V5 R4
Applications										
Telnet	Enables login from one system to another. i5/OS TCP/IP supports both the Telnet client and server. Use iSeries Navigator to manage Telnet sessions, sign-on, and configuration.	X	X	X	X	X	X	X	X	X
	Telnet 5250 extensions for printing, device name selection, and automatic sign-on.		X	X	X	X	X	X	X	X
	Telnet server enhancements to enable: support for 128 byte passphrase; client certificate authentication enabled from the Digital Certificate Manager (DCM); SHA1 password encryption. Telnet client also now supports the 128-byte passphrase and other security enhancements.						X	X	X	X
	Enterprise Identity Mapping (EIM) support by Telnet server.							X	X	X
FTP	Transfers files between servers and clients across a TCP/IP network. i5/OS TCP/IP supports both the FTP client and server. Use iSeries Navigator to administer FTP formats, mappings, and startups.	X	X	X	X	X	X	X	X	X
	Anonymous FTP implemented via exit programs.	X	X	X	X	X	X	X	X	X
	Passive FTP (firewall-friendly FTP).		X	X	X	X	X	X	X	X
	iSeries Navigator's Application Administration can be used to control specific operations for both the FTP client and server based on i5/OS user profile.						X	X	X	X
	SSL/TLS support for the FTP server. This enables an FTP client to connect to the i5/OS FTP server via an encrypted session for both the control and data connections. Also several other FTP server enhancements related to performance and security.						X	X	X	X
	SSL/TLS support is now available for the i5/OS FTP client, too.							X	X	X
	Secure FTP through Network Address Translation (NAT) firewalls support.									X
DHCP DHCP Relay	The Dynamic Host Configuration Protocol (DHCP) server can automatically assign IP addresses. The i5/OS supports this work-saving protocol that dynamically provides client and server configuration information. iSeries Navigator includes a complete interface for DHCP configuration and administration. The i5/OS provides only a DHCP server, not a DHCP client.		X	X	X	X	X	X	X	X
	DHCP WAN client support for PPP Receiver profiles. This is not the same as a DHCP client.						X	X	X	X
	DHCP server can now automatically update DNS server with A and PTR records for hosts in network. Also, DHCP Monitor to show active leases.						X	X	X	X

Function	Short description	V4 R1	V4 R2	V4 R3	V4 R4	V4 R5	V5 R1	V5 R2	V5 R3	V5 R4
DNS	The Domain Name System (DNS) translates domain names to IP address. iSeries Navigator contains an easy-to-use DNS server configuration display that enables you to work with primary and secondary DNS names for your network. Based on BIND 4.		X	X	X	X	X	X	X	X
	Update to BIND 8.2.3 (V5R2 is 8.2.5) brings a full-featured DNS server to the i5/OS system including multiple DNS servers on one i5/OS system, secure dynamic updates to the DNS server (for example, DHCP), and incremental zone transfers (IXFR).						X	X	X	X
LPR and LPD	LPR sends files to print to any i5/OS server over TCP/IP. LPD places print files on a print queue on any i5/OS server in your network. i5/OS server supports these network printing protocols for ASCII and LAN-attached printers. Use iSeries Navigator to display and manage LPR and LPD properties.	X	X	X	X	X	X	X	X	X
IPP	Internet Printing Protocol (IPP) is a client-server protocol through which an IPP-enabled client can submit, over the Internet, an intranet, or a LAN, a print request to an IPP enabled printer or to an IPP-enabled server, such as an i5/OS server. i5/OS server is an IPP server only.						X	X	X	X
	i5/OS server can also be an Internet Printing Protocol (IPP) client.							X	X	X
SMTP	Simple Mail Transfer Protocol (SMTP) sends and receives e-mail on the i5/OS server. Use the SMTP capabilities to set up an extensive e-mail system with your i5/OS server as the e-mail server. iSeries Navigator enables easy and intuitive SMTP configuration and management.	X	X	X	X	X	X	X	X	X
	Dual stack support (primarily for Domino® co-existence). This support was added via PTF in V4R2 and it enabled the SMTP server to bind to a single interface. In V5R1 this support was extended to enable you to bind to more than one interface. Also: controlling unwanted mail traffic (spam) via accept and reject lists was made available via PTF.		X	X	X	X	X	X	X	X
	Mail filtering to prevent virus proliferation. This support was added via PTF in V4R4.				X	X	X	X	X	X
	New SMTP extensions to support ETRN-dialup retrieval, Delivery Status Notification (DSN), and 8-bit MIME. Also: selectable subsystem for SMTP; multiple domain support; and support for Realtime Black List (RBL) to further defend against spammers.						X	X	X	X

Function	Short description	V4 R1	V4 R2	V4 R3	V4 R4	V4 R5	V5 R1	V5 R2	V5 R3	V5 R4
POP3 server	Stores and forwards e-mail for e-mail client program connected to the i5/OS server. i5/OS server supports POP3 by creating mailboxes for all enrolled e-mail users. i5/OS server POP3 server enables users to access their mailboxes from third-party e-mail client programs.	X	X	X	X	X	X	X	X	X
LDAP	Lightweight Directory Access Protocol (LDAP) provides a directory service over TCP/IP. i5/OS server Directory Services provides an LDAP server. LDAP runs over TCP/IP and is a popular choice as a directory service for both Internet and non-Internet applications. Use iSeries Navigator to perform most setup and administration tasks of the LDAP directory server on the i5/OS server.			X	X	X	X	X	X	X
NetServer™	Enables Windows clients using Network Neighborhood to access i5/OS server shared directory paths and output queues, such as printers and file systems. i5/OS server supports PC clients using a network file and print sharing functions that is included in their operating systems. This means that you do not need to install any additional software to benefit from i5/OS server NetServer. You can administer and manage i5/OS server NetServer with iSeries Navigator.		X	X	X	X	X	X	X	X
	Enterprise Identity Mapping (EIM) support.							X	X	X
NFS	Network File System (NFS) client and server support.	X	X	X	X	X	X	X	X	X
HTTP server	Enables a i5/OS server attached to the Internet or intranet to provide objects (such as Web pages) at the request of any Web client (browser). This line refers to the HTTP Server (original) support.	X	X	X	X	X	X	X	X	X
	HTTP Server (powered by Apache) support at Version 2.0.18 (beta).					X				
	HTTP Server (powered by Apache) support of the GA version of the Apache server.						X	X	X	X
TCM	The Triggered Cache Manager (TCM) is not a cache, but as the name implies, a cache manager. TCM can proactively update the contents of the IFS based on application triggers (or events).					X	X	X	X	X
FRCA	Fast Response Cache Accelerator (FRCA) is an SLIC server that can serve files cached below the machine interface (MI). For an HTTP server this can reduce both the amount of time to serve a file and the amount of CPU required.						X	X	X	X
	Fast Response Cache Accelerator (FRCA) also provides support for QoS inbound rate control based on URI path information.							X	X	X

Function	Short description	V4 R1	V4 R2	V4 R3	V4 R4	V4 R5	V5 R1	V5 R2	V5 R3	V5 R4
DDM and DRDA®	Access data distributed across multiple machines. DRDA is part of the DDM architecture. The i5/OS server supports it by including DDM as part of OS/400 and i5/OS. SQL, an IBM database programming language, also supports DRDA implementations.									
	DRDA Level 1 client and server.		X	X	X	X	X	X	X	X
	DDM server.		X	X	X	X	X	X	X	X
	DDM client.				X	X	X	X	X	X
WSG	The WorkStation Gateway (WSG) transforms OS/400 5250 applications to HTML, enabling users to run OS/400 applications from a PC that has a Web browser. i5/OS server supports this simplified way to incorporate a point-and-click interface for your end users. Use iSeries Navigator to control WSG properties. Support for WSG removed in V5R2.	X	X	X	X	X	X			
TFTP	Transfers files. After the client asks for and receives an IP address from the BOOTP server, the client initiates a TFTP request to the TFTP server for the file. i5/OS server supports TFTP file transfers by leveraging the BOOTP server. Use iSeries Navigator to start, stop, configure, and manage TFTP.	X	X	X	X	X	X	X	X	X
BOOTP	Provides a means to notify a host of its assigned IP address, the IP address of a boot server host, and the name of a file to be loaded into memory and executed. Works in conjunction with TFTP. i5/OS server supports both BOOTP clients and server. Use iSeries Navigator to configure the local subnet mask, the local time offset, the addresses of default routers, and the addresses of various Internet servers.	X	X	X	X	X	X	X	X	X
REXEC (remote execution)	Issues OS/400 and i5/OS commands from other systems across TCP/IP networks. The i5/OS server supports the REXEC client (via Run Remote Command [RUNRMTCMD] command and rexec() APIs) and server (via the REXEC daemon). Use iSeries Navigator to start and stop the Remote Execution (REXEC) server.	X	X	X	X	X	X	X	X	X
SNMP	Simple Network Management Protocol (SNMP) provides a means of managing an Internet environment over UDP. The i5/OS system supports SNMP at Version 1. Use iSeries Navigator to manage SNMP.	X	X	X	X	X	X	X	X	X
SNTP	Simple Network Time Protocol (SNTP). This is a client (only) implementation.						X	X	X	X

Function	Short description	V4 R1	V4 R2	V4 R3	V4 R4	V4 R5	V5 R1	V5 R2	V5 R3	V5 R4
QoS	Quality of Service (QoS) is a collection of functions that enable the user to define what kind of network priority or bandwidth to assign to a TCP/IP application program. IntServ (via sockets RSVP [Resource Reservation Protocol] API) and DiffServ (policy-driven control of outbound data rates) are two of these functions.						X	X	X	X
	New qtoq QoS sockets APIs to simplify the work required to use IntServ on the i5/OS server. Also, new policies to control the inbound rate of connections and data to the i5/OS server, including by URI.							X	X	X
Security-related protocols and support										
VPN	A virtual private network (VPN) enables creation of secure end-to-end paths between any combination of hosts and gateways across an untrusted network. VPNs use authentication methods, encryption algorithms, and other precautions to ensure that data sent between the two endpoints of its connection remains secure.				X	X	X	X	X	X
	Native VPN support including remote access, branch office, and extranet scenarios. Includes L2TP support.				X	X	X	X	X	X
	Client-side Network Address Translation (NAT) support enables VPN connectivity through a NAT firewall when i5/OS is the initiator of the connection.							X	X	X
	Server-side NAT support enables VPN connectivity through a NAT firewall when i5/OS is the initiator or responder for the connection.									X
IP Packet Security	Protects the i5/OS server on the Internet. It contains two components, IP Packet Filtering and Network Address Translation (NAT), that act as a firewall of protection for a system. iSeries Navigator provides extensive access to IP Packet Security functions.			X	X	X	X	X	X	X
	IP Packet auditing and journaling.		X	X	X	X	X	X	X	X
	IP Network Address Translation (NAT) provides access to a remote network, usually the Internet, while protecting the private network by masking the IP addresses that are used inside your firewall. You can use Masquerade NAT, Dynamic NAT, and Static NAT for routing with your i5/OS server.			X	X	X	X	X	X	X
	IPCS firewall with IP security and VPN support.			X	X	X				
	IP filtering.				X	X	X	X	X	X
SOCKS	Acts as a gatekeeper for firewall systems. i5/OS server TCP/IP includes SOCKS client support to provide the i5/OS server communications through a firewall running SOCKS server. Use iSeries Navigator to configure SOCKS.		X	X	X	X	X	X	X	X

Function	Short description	V4 R1	V4 R2	V4 R3	V4 R4	V4 R5	V5 R1	V5 R2	V5 R3	V5 R4
SSL	Secure Sockets Layer (SSL) provides secure transmission of information over TCP/IP. It is an integral part of securing an internal network. The i5/OS server supports SSL on its HTTP server and LDAP server. Use the i5/OS Task page of the *Admin instance of the HTTP server to set up and manage SSL.	X	X	X	X	X	X	X	X	X
	SSL-enabled Telnet proxy.		X	X						
	SSL for Client Access and Telnet server.				X	X	X	X	X	X
EIM	Enterprise Identity Mapping (EIM) support. This support is based on both LDAP and a client implementation of Kerberos Version 5.							X	X	X
IDS	Intrusion Detection System (IDS) support. This enables a notification system of attempts to hack into, disrupt, or deny service to the system.									X

Other applications, such as Domino, WebSphere Commerce, Payment Manager, and user-written applications, can use TCP/IP.

2

Interfaces, routes, and Virtual IP

Nothing is more central to the efficiency of your TCP/IP network than the interfaces and routes that are your network. For example, no amount of Quality of Service (QoS) configuration (see Chapter 10, “Quality of Service (QoS)” on page 141) will help you when your one and only interface fails.

This chapter teaches you the basics of how the i5/OS server uses its interfaces and routes to communicate to other TCP/IP hosts and routers. Every step of the way, we explain the configuration building blocks upon which a fault-tolerant and load-balanced network is built.

In Chapter 12, “Defining adaptable TCP/IP interfaces and routes” on page 179, and in Chapter 13, “Virtual Ethernet within an LPAR environment” on page 211, we show how to put those building blocks together.

2.1 Interfaces

The interface is TCP/IP's connection to the external world. It is often defined as the Internet Protocol (IP) address.

Here is a simple question: What is the IP address of your i5/OS server?

Your answer might be, "192.168.2.2" or "It has several IP addresses."

An exception to this is the virtual IP address. See 2.1.3, "Virtual IP address" on page 38.

But really, your i5/OS server does not have an IP address. It is only the interfaces on your i5/OS server that have an IP address. An interface consists of a line description name, the Internet Protocol (IP) address, and an automatic start parameter.

The i5/OS server interface implementation supports multi-homing. This enables you to specify either a single interface or multiple interfaces per line description. The following combinations can be configured:

- A single host on a network over a communications line
- Multiple host on the same network over the same communications line
- Multiple hosts on different networks over the same communications line
- Multiple hosts on the same network over multiple communications lines
- Multiple hosts on different networks over multiple communications lines

The i5/OS server has three types of interfaces that you can configure, and these sections describe them in greater detail:

- "Local area network (LAN) interfaces" on page 16
- "Wide area network (WAN) interface wizard for frame relay" on page 38
- "Virtual IP address" on page 38

2.1.1 Local area network (LAN) interfaces

Figure 2-1 on page 19 shows the creation of a LAN interface. A LAN interface allows for a TCP/IP connection via an Ethernet or a Token-ring line description. This is the type most people will use. The installation of the core TCP/IP application and the configuration of the LAN interface are described in detail in the following section.

Installing TCP/IP core applications on the i5/OS server

Installing Transmission Control Protocol/Internet Protocol (TCP/IP) on your i5/OS server allows you to connect an i5/OS server to a network.

If you do not have TCP/IP Connectivity Utilities for i5/OS installed on your system already, follow the *i5/OS Software Installation Version 5 Release 4* and install the license program 5722-TC1. The TCP/IP Utilities Licensed Program is shipped with i5/OS at no additional charge and it must be installed separately.

Other licensed programs that you may want to install are listed in Table 2-1.

Table 2-1 List of licensed programs

Licensed program	Description
5722-XE1	Server iSeries Access for Windows: provides iSeries Navigator support that is used to configure some of the TCP/IP components.
5722-SS1 option 3	Extended Base Directory Support: provides directory support for functions such as DHCP.

Licensed program	Description
5722-SS1 option 31	Domain Name System: maps host names to IP addresses.
5722-SS1 option 34	Digital Certificate Manager: use if you plan to use Secure Socket Layer (SSL) support.
5722-CR1	Cryptographic Support for AS/400: provides SSL encryption support to many functions on the system.
5722-DG1	IBM HTTP Server for i5/OS: provides an i5/OS server Web server.
5722-DG1 option 1	Triggered Cache Manager: provides a mechanism for managing dynamically generated Web pages.

Configuring the TCP/IP

In this section we discuss the steps and prerequisites for configuring the TCP/IP on the i5/OS server.

Before you configure TCP/IP

Before you configure TCP/IP on a system, you must determine the values of the network information needed for TCP/IP. There are six values that may be used during the configuration of TCP/IP. These values should be given to you by the network administrator. If you are the network administrator, and you are setting up TCP/IP in your environment for the first time, you should determine these values before you continue.

Table 2-2 TCP/IP configuration parameters

Parameter	Value
IP address	If your system is going to be accessed by the public, it may require a publicly registered IP address. These addresses are obtained from your Internet Service Provider (ISP). If you are setting up an intranet, you or the network administrator assigns the address.
Subnet mask	The subnet mask is used to define the range of addresses that directly connect to the network segment where this interface is attached. The subnet mask and the IP address of the interface together determine the network address of the network segment.
Host name and domain name	The default complete name of the system is defined by the host name and domain name parameters. The host name is the portion that identifies this system by name. The domain name is the name of the domain of the network in which this system exists. If your company does not have a registered domain name, you should get one even if do not plan to have an Internet presence. The domain name is used for addressing e-mail as well as Web serving.
DNS server	A DNS server is used for name-to-address translation and address-to-name translation. Some applications expect the DNS to know all clients and servers in the network. One example of this is a Line Printer Daemon (LPD) server that uses the IP address to create a banner page.

Parameter	Value
Next hop gateway	The next hop gateway or router is how this system gets to other systems in other parts of the network. If your network consists of a single segment, you will not have any entries for next hop gateway. If you have multiple routers on this segment, you may have multiple next hop gateway entries. If you have multiple interfaces on your i5/OS server and you have a router in your network, you should have one default route defined for each interface. Each route entry should have Preferred binding Interface specified as a different interface.

The required parameters to configure TCP/IP on the i5/OS server are IP address and subnet mask. Some functions require additional parameters. One example is SMTP, which requires a host name and domain name.

To learn more about these parameters and how they are used in TCP/IP, refer to the book *TCP/IP Tutorial and Technical Overview*, GG24-3376.

Configuring TCP/IP using iSeries Navigator

You can configure and work with TCP/IP via iSeries Navigator (the graphical user interface) or the command line interface. For some functions, the entire configuration must be done using iSeries Navigator, while other functions can only be configured using the command line interface. You may need to create a TCP/IP interface and start TCP/IP using the i5/OS command-line interface before you can access iSeries Navigator the first time.

This section covers how to perform the following tasks for TCP/IP using iSeries Navigator:

- Creating a interface using iSeries Navigator
- Changing TCP/IP properties
- Configuring host table entries
- Changing IPV4 Properties
- Verifying a TCP/IP connection

Creating a interface using iSeries Navigator

iSeries Navigator is a powerful graphical interface for Windows clients. To use iSeries Navigator, you must have iSeries Access installed on your Windows PC with the latest Service Pack and have a connection defined to the i5/OS server that you want to configure.

The creation of interfaces is made easy through the use of a wizard. iSeries Navigator provides a wizard for both IPv4 and IPv6. For more information about IPv6, see Chapter 3, “IPv6: the next generation of the Internet” on page 61.

To create a new TCP/IP V4 interface using the iSeries Navigator:

1. Select **New Interface** → **Local Area Network** from the Interfaces context menu, as shown in Figure 2-1.

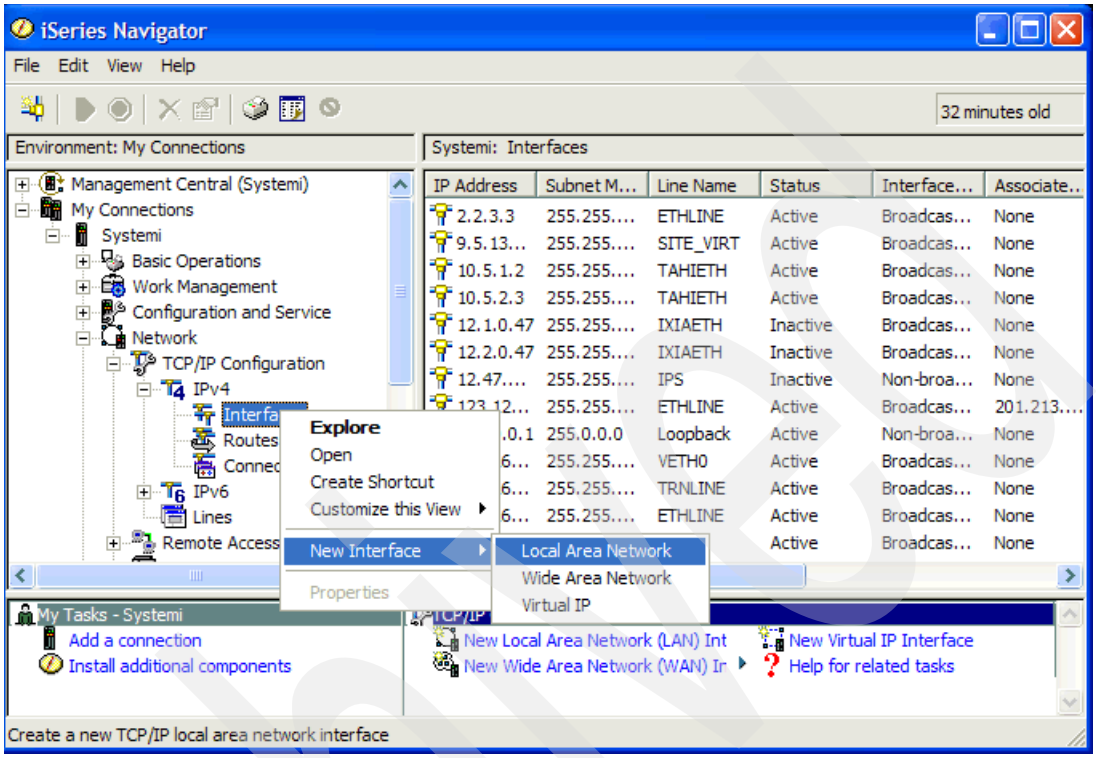


Figure 2-1 iSeries Navigator: new IPV4 interface

2. You now see the first window of the TCP/IP wizard interface. Click **Next** to continue.
3. As shown in Figure 2-2 select the type of connection you want to define for TCP/IP (Ethernet or Token-ring). In our example, we selected Ethernet. Click **Next** to continue.

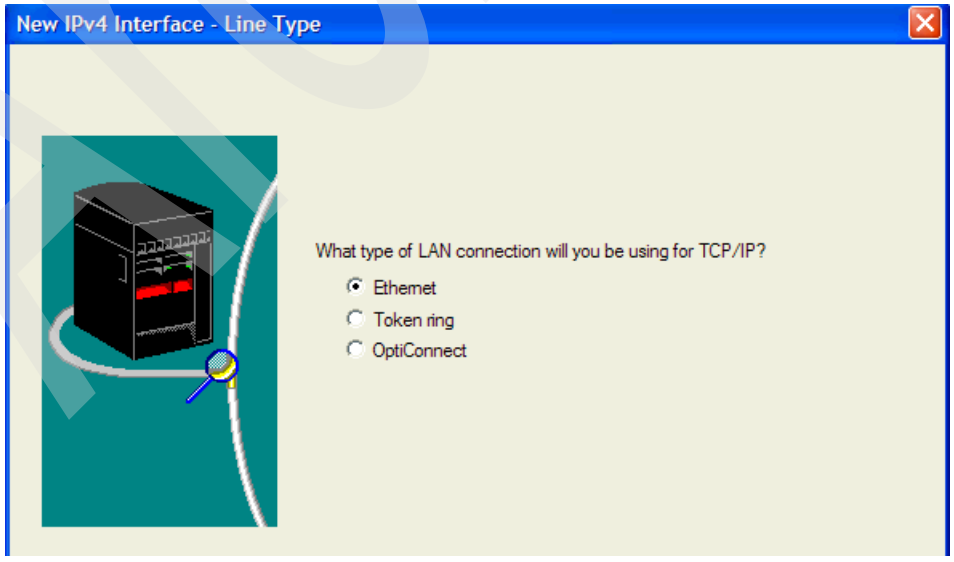


Figure 2-2 New IPv4 Interface: Line type

4. Figure 2-3, the New IPV4 Interface Resource window, shows all of the hardware on your system that matches your type selection. Use the buttons on the window to determine the location of the adapter. You can also use the buttons to list communication lines that are currently defined. Click the hardware resource you want to configure. Click **Next** to continue.

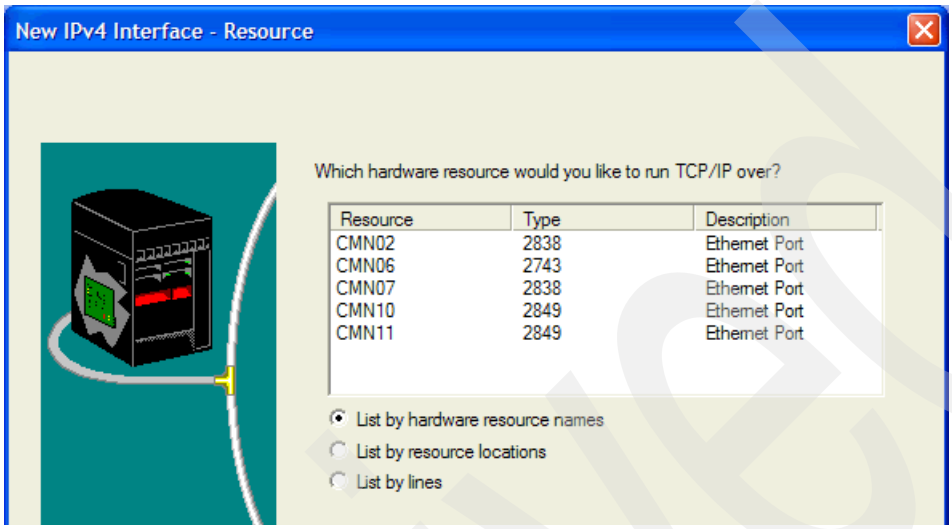


Figure 2-3 New IP V4 Interface: Select hardware resource

5. The Choosing a Line window presents a list of line descriptions that already exist on the system (Figure 2-4).
   - If you need to create a new line description, select **Create a new line** and click **Next** to continue with step 6 on page 21.
   - If a line is already defined and configured for the hardware resource, click **Use an existing line** and select the correct line description from the list. Click **Next** and skip to step 8 on page 22.

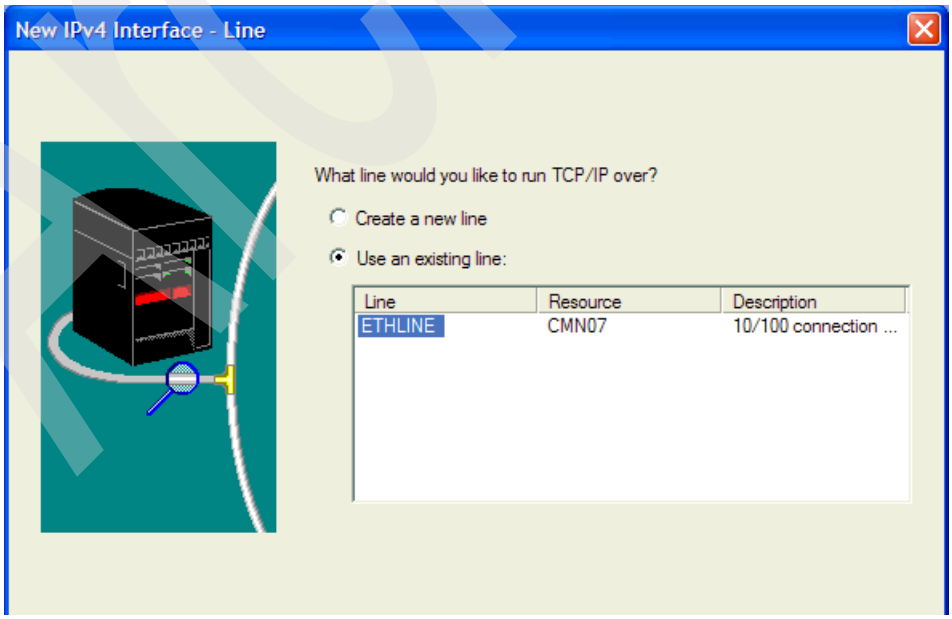


Figure 2-4 New IPV4 Interface: Choosing a line

6. You are prompted to create a new line. Enter a name and a description for the new line as shown in Figure 2-5, and select the appropriate values for **Authority** and **Duplex** based on your environment. Click **Next** to continue.

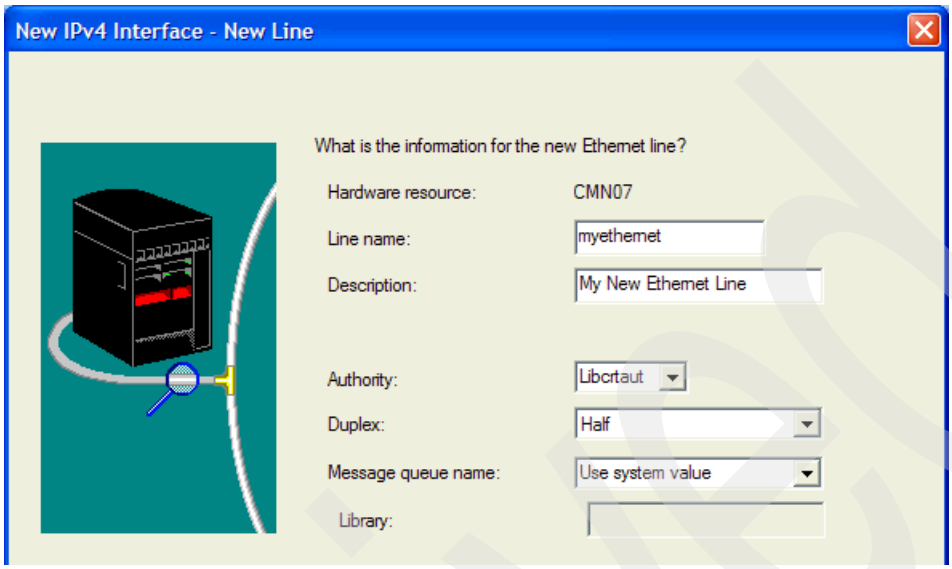


Figure 2-5 Creating a new line description

7. In the Ethernet Line Characteristics window (Figure 2-6), select the speed at which your LAN is running, and choose the protocol standards that you want to support on this adapter. Click **Next** to continue.

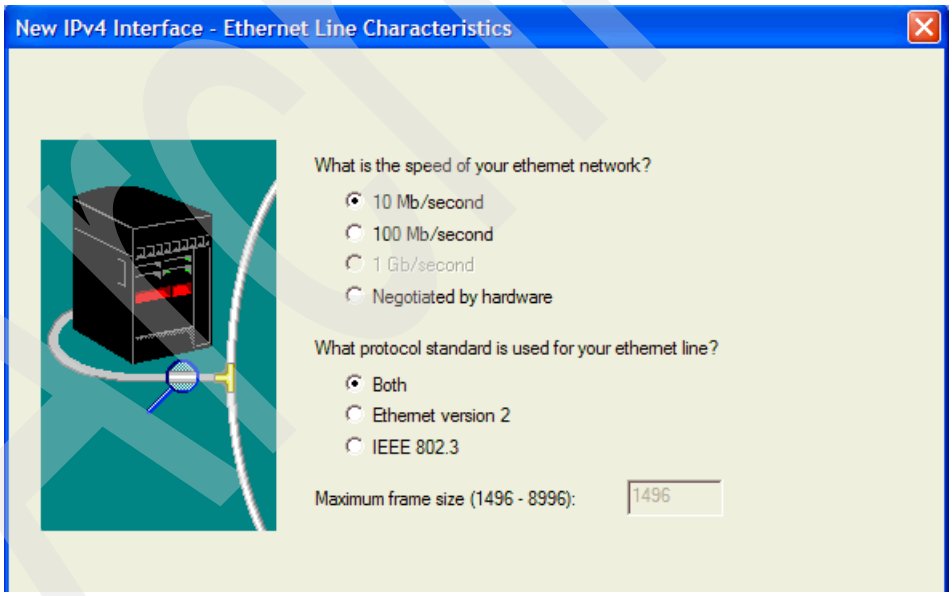


Figure 2-6 New IPv4 Interface: Line characteristics

8. In the TCP/IP Interface Settings window (Figure 2-7), enter the IP address and the interface name (we used line name), and the subnet mask for this IP address.

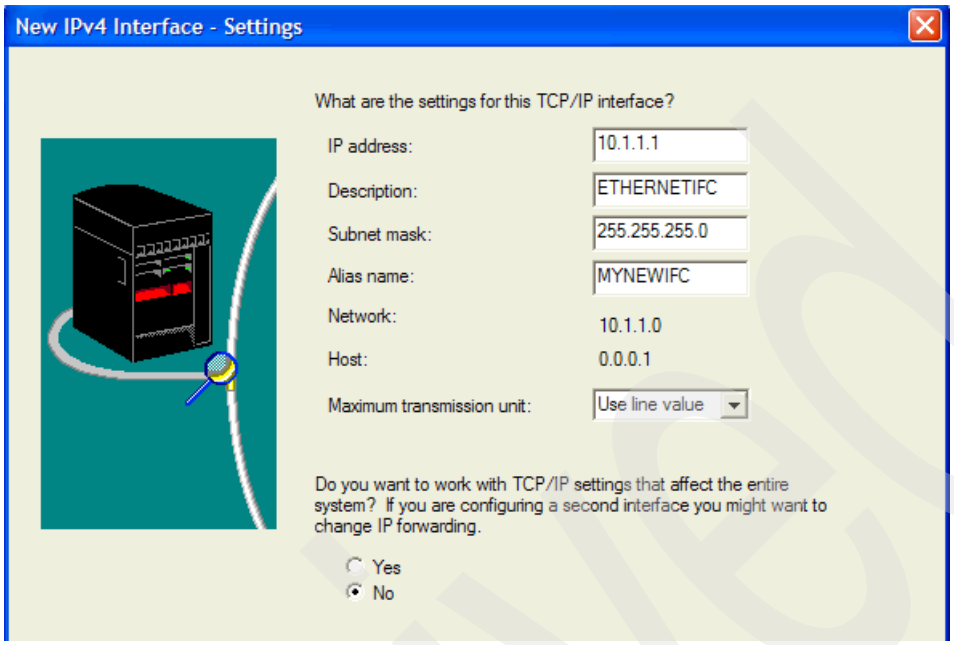


Figure 2-7 TCP/IP interface settings

For the IP address and Subnet mask parameter, specify the value provided by the LAN administrator or Internet Service Provider (ISP). The system takes the IP Address and Subnet Mask and performs a *logical AND* to determine the Network and Host values displayed in the window. The subnet mask and the IP address enable IP protocol to determine where to send the data it receives.

The Maximum Transmission Unit (MTU) specifies the maximum size (in bytes) of the IP datagram that you can send on this interface. The maximum size specified for a particular route should not be larger than the smallest MTU that is supported by any router or gateway in that route. If the MTU size is larger than the smallest MTU in the route, the router with the small MTU will fragment the packet. This can increase the traffic on the segment and lead to performance degradation. The Help Button provides additional information about MTU. (We have used the line defaults here.)

The Alias name specifies a name that can be used programmatically to refer to this interface rather than using the IP address. Click **Next** to continue.

9. As shown in Figure 2-8, enter the value for the gateway address. A gateway is a piece of hardware that connects two are more network segments. It is often called a router. You can define up to three gateway addresses. If your i5/OS server is attached to only a single network, you do not need to specify any gateway addresses. If you do not wish to enter any routing information, leave the defaults, click **Next**, and skip to step 14.

   This is also where you specify additional routing information for this interface. This may be used for load balancing or to define multiple routes for backup purposes. To add routes, click **Yes** to configure additional route information. Click **Next** to continue to step 10 on page 24.

Figure 2-8 New IPv4 Interface: TCP/IP routing information

10. The window shown in Figure 2-9 enables you to specify whether these routes should be published to the network using RIP1 or RIP2. You can also define default routes, network routes, and routes to a specific host. Click the appropriate button to add the required routes. In this example, we clicked **Add default route**.

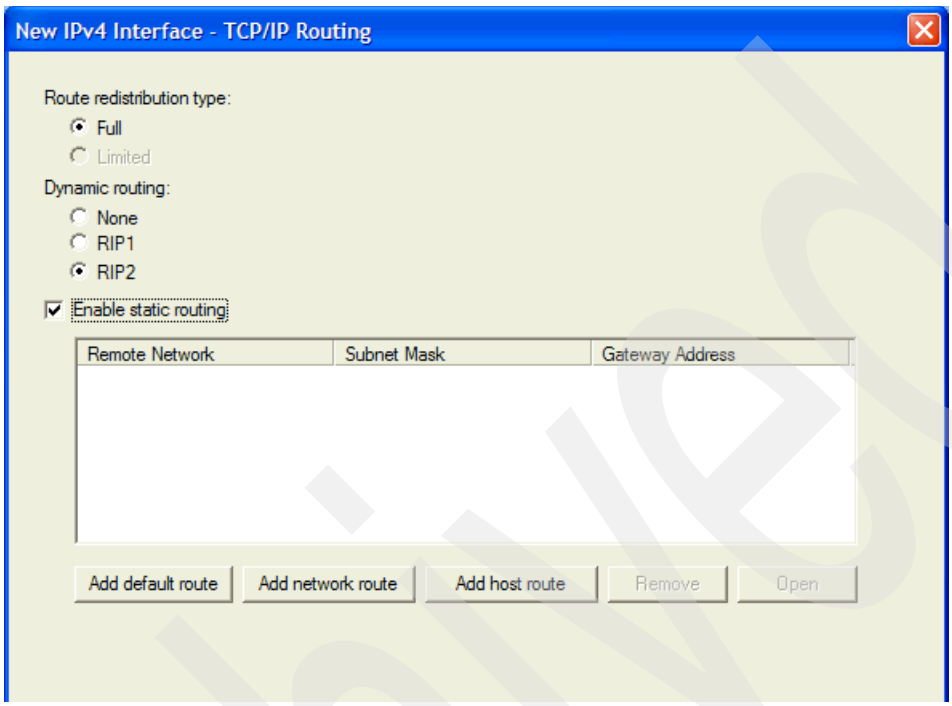


Figure 2-9 TCP/IP Routing additional information

11. Each of the Add Route windows has an Advanced button. Specify the gateway address, and click **Advanced** (Figure 2-10).

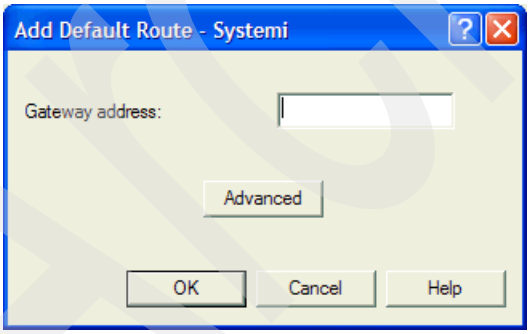


Figure 2-10 Add Default Route

12. The Advanced Routing Settings window (Figure 2-11) enables you to specify information about the route. If you leave Route precedence set to 5, the route selection works as usual. If you set Route precedence to a value less than 5, this route will not be a preferred route to the destination network. If Route precedence is set to a value greater than 5, the route will be considered a preferred route to the destination network.

    You may have multiple interfaces defined to the same network, multiple routes defined using the interfaces, and the route precedence of these routes set to the same value greater than 5. In this case, the TCP/IP traffic will be balanced across all interfaces with routes defined. Set the values that you need, and click **OK**.

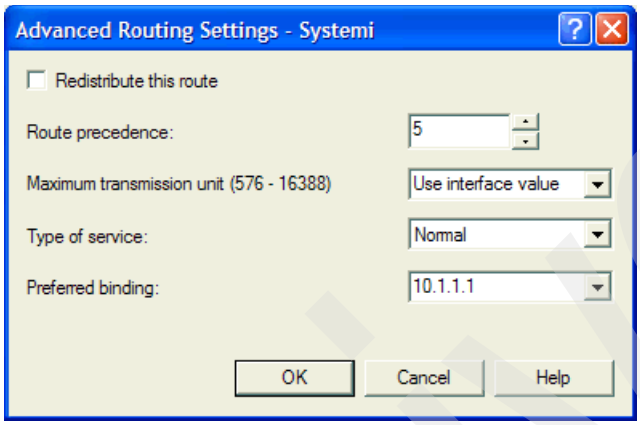


Figure 2-11 Advanced Routing Settings

13. When you have added all of the route information you need, click **OK** until you have returned to the TCP/IP Routing window (Figure 2-9 on page 24). Click **Next** to continue.
14. On the Start window (Figure 2-12), identify whether you want this TCP/IP interface started whenever you start TCP/IP and whether you want it to start now. If you choose to start the TCP/IP interface here, it will be tested when you click **Next**.

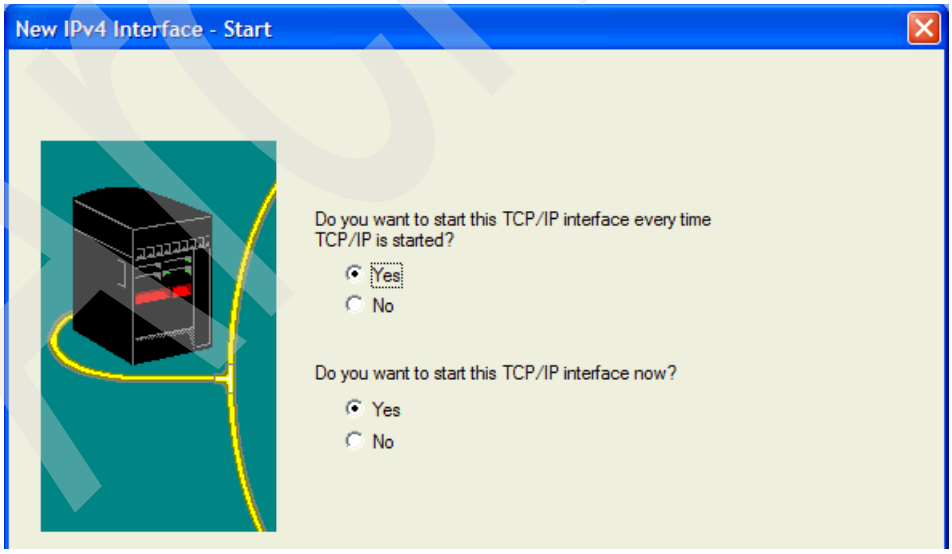


Figure 2-12 Interface start options

15. After a successful test, the New TCP/IP Interface Summary window appears.
16. Verify that all of the information displayed is correct. If all values are correct, click **Finish**.

Changing TCP/IP properties

The TCP/IP attributes of the i5/OS server are accessible from iSeries Navigator using the properties selection of the context menu, as shown in Figure 2-13:

1. Expand <system name> → **Network** → **TCP/IP Configuration**. Select **Properties** from the context menu to make detailed changes to the configuration of your TCP/IP interface.

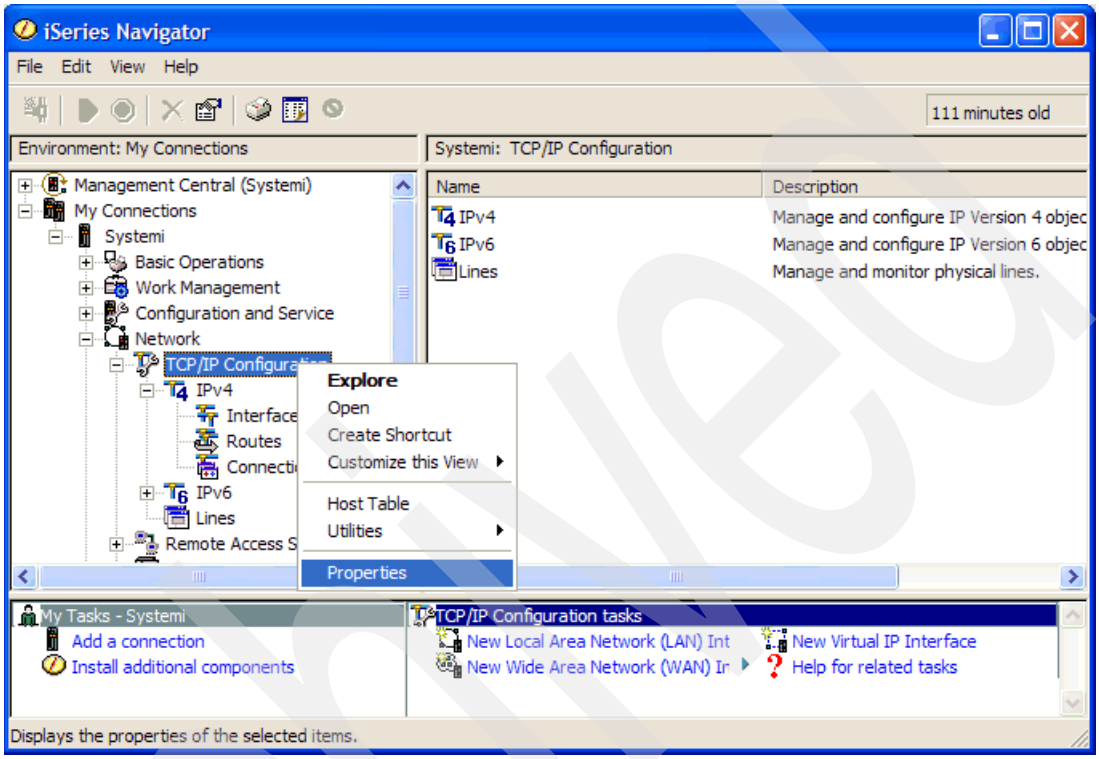


Figure 2-13 TCP/IP properties

2. Figure 2-14 shows the TCP/IP Configuration Properties window. Click the **Host Domain Information** tab to specify this information for your i5/OS server TCP/IP communication. Enter the host name, the domain name, and information such as:
   - Domain name servers: List up to three domain server IP address. The system uses the domain servers in the order that you list them. The domain name servers perform host name resolution by translating the host name into an IP address.
   - Search order: Specifies whether you want the local host table searched before or after the domain name server.

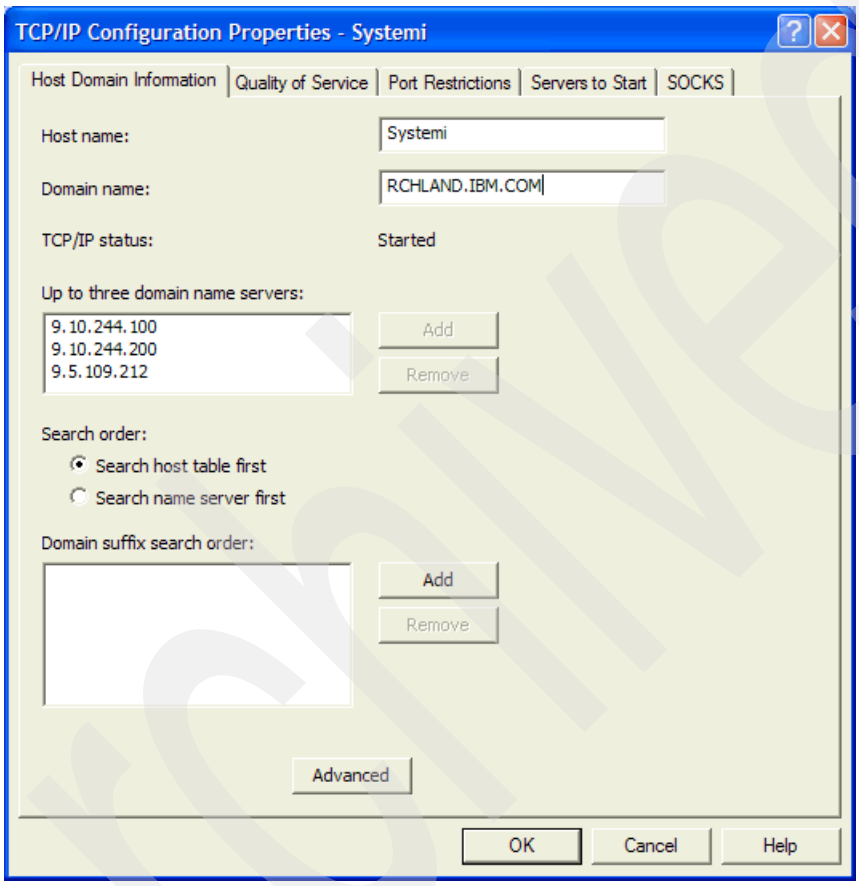


Figure 2-14 TCP/IP Configuration Properties: Host Domain Information

3. Click **Advanced** to set additional DNS values. The TCP/IP Advanced Host Domain Information window (Figure 2-15) appears. The default values shown work in most environments. If you have intermittent trouble resolving names to IP addresses, you may want to increase the number of attempts and the interval between attempts. If these values are set too high, you may experience a long wait time before an unknown host message is displayed.

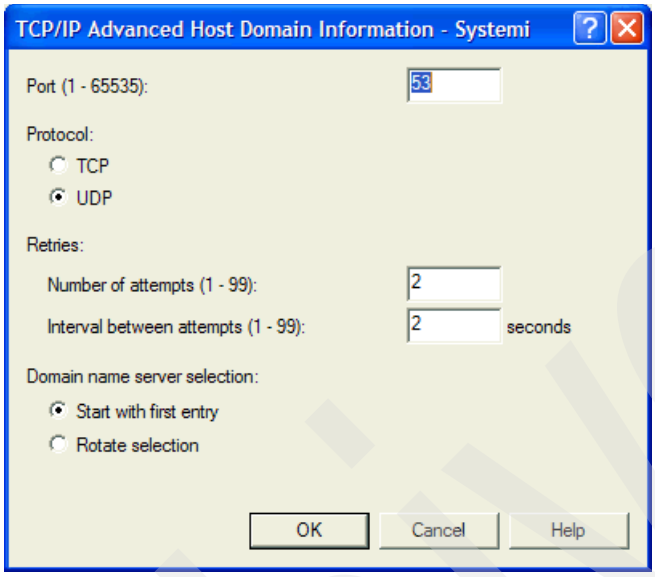


Figure 2-15 TCP/IP Advanced Host Domain Information

4. Click the **Quality of Service** tab (Figure 2-16). To initiate QoS on the i5/OS server you must enable it here. (For more information see Help.)

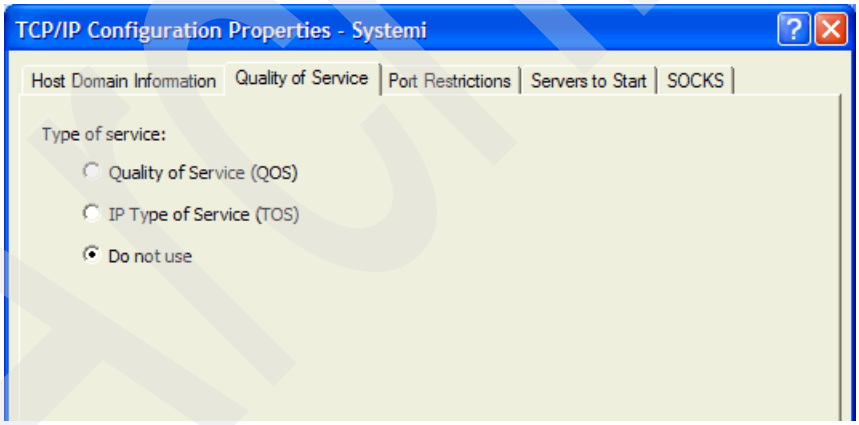


Figure 2-16 TCP/IP Configuration: Quality of Service

5. Click the **Port Restrictions** tab (Figure 2-17) to limit port use to a user profile name. If you want to restrict a single port, you must specify the same starting and ending port number.

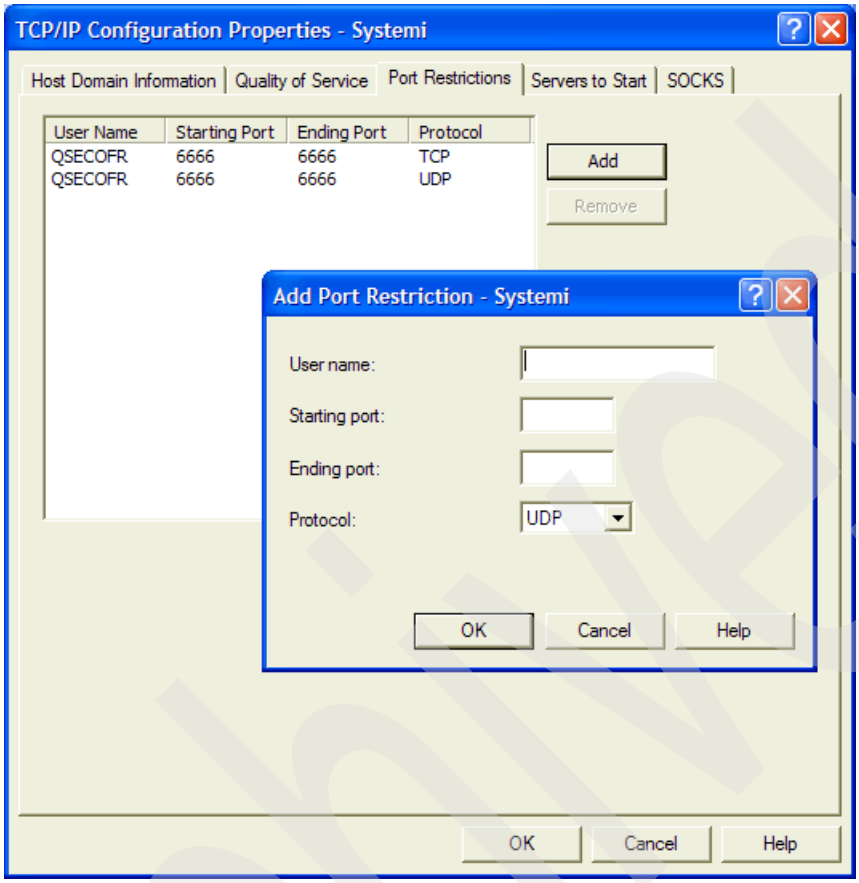


Figure 2-17 TCP/IP Configuration: Port Restrictions

6. Click the **Servers to Start** tab (Figure 2-18) to select the currently installed servers that you want to start automatically when TCP/IP starts.

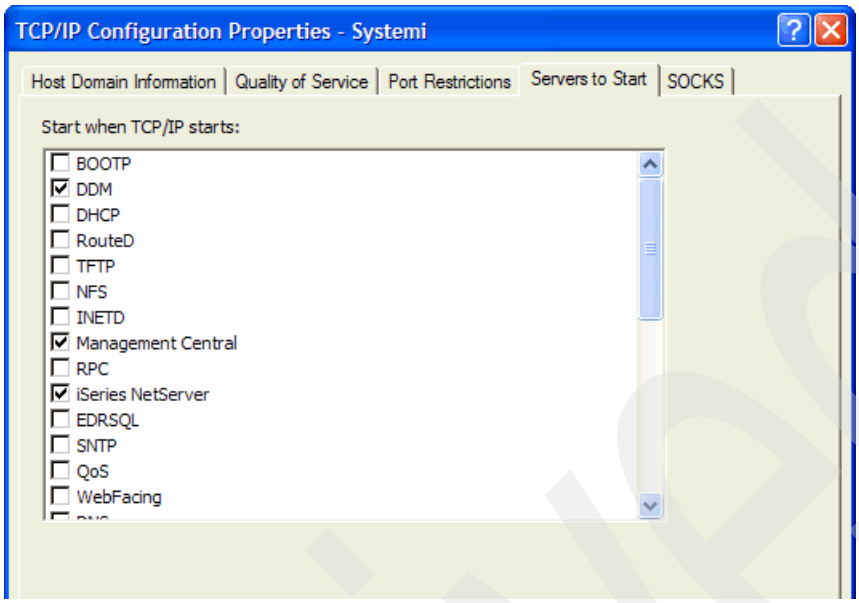


Figure 2-18 TCP/IP Configuration: Servers to Start

7. Click the **SOCKS** tab (Figure 2-19) to define the TCP client connection to internal secure networks and to less secure networks. You can define a direct connection to servers in the internal secure network. Users must have IOSYSCFG special authority to change information on this dialog.

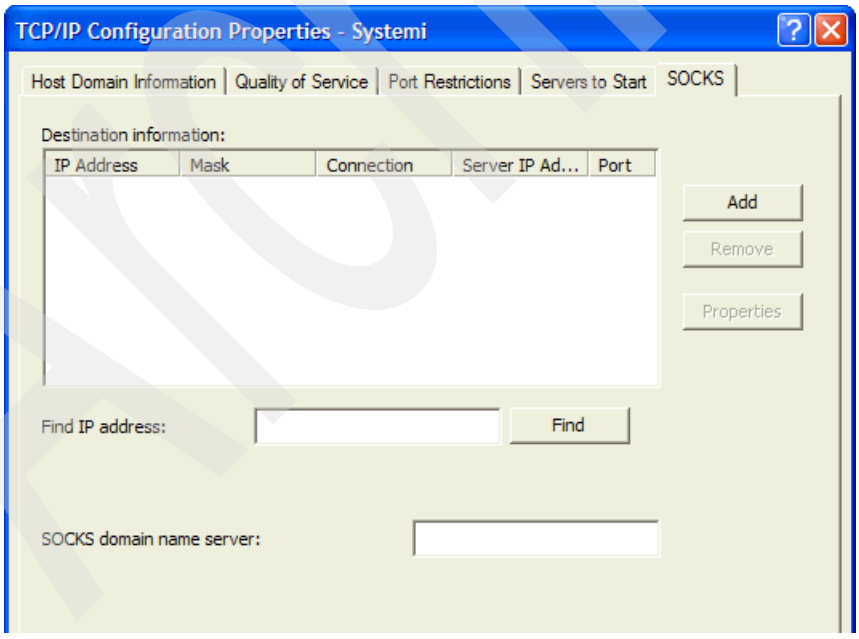


Figure 2-19 TCP/IP Configuration properties: SOCKS

8. After completing changes to the TCP/IP Properties dialog, click **OK** to save the configuration file and close the window.

Configuring host table entries

You must configure host table entries for TCP/IP if you want the users of your i5/OS server to use easily remembered names rather than IP addresses. If you are using the Domain Name System (DNS), you do not have to configure host table entries.

The host table provides the advantage of not having to remember actual Internet addresses for systems in the network. A host table accomplishes this task by mapping Internet addresses to TCP/IP host names. The local host table on your i5/OS server contains a list of the Internet addresses and related host names for your network.

Before you begin configuring your host table entries for TCP/IP, you should know the IP addresses of your hosts. You also need to know the host names and descriptions of the hosts that you want to include in the host table.

To configure host table entries for TCP/IP using Operations Navigator:

1. In the iSeries Navigator, expand **Network** → **TCP/IP Configuration**. Right-click **TCP/IP configuration** and select **Host Table**.
2. Click **Add** to open the TCP/IP Host Table Entry dialog (Figure 2-20). After specifying the IP address, the host name, and the description of the hosts that you want to include in the host table, click **OK** to save the configuration file and close the window.

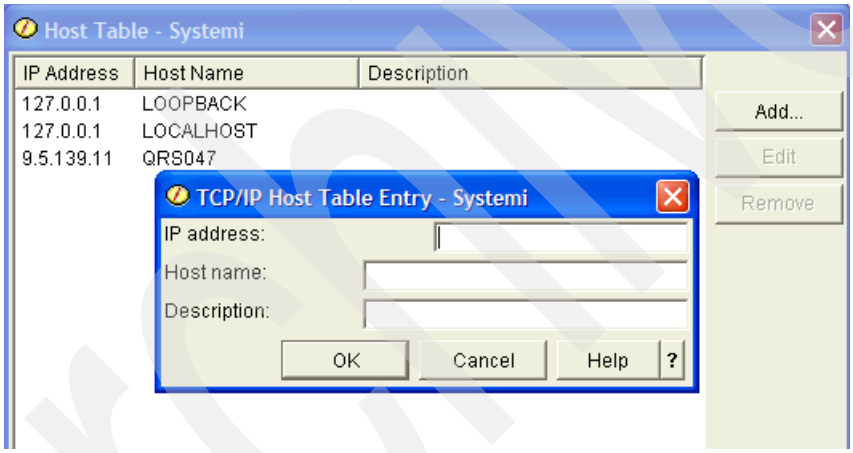


Figure 2-20 TCP/IP Host Table Entry: Add host

Changing IPV4 Properties

The general properties of the TCP/IP V4 can be altered using the iSeries Navigator.

1. In iSeries Navigator, expand **Network** → **TCP/IP Configuration**. Right-click **IPV4** and select **Properties** as shown in Figure 2-21.

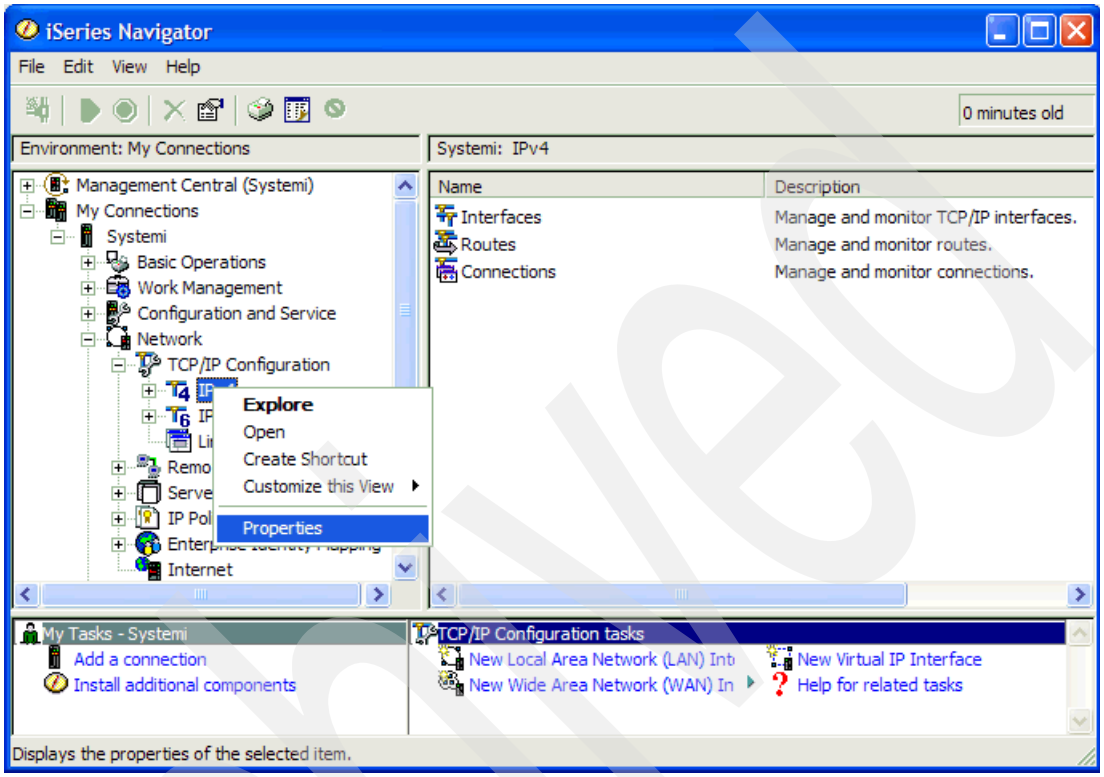


Figure 2-21 IPV4 properties

2. Click the **General** tab (Figure 2-22) to specify IP datagram forwarding, enable IP source routing, enable dead gateway detection, and other values:
   - IP forwarding specifies whether you want the IP layer to forward IP datagrams between different networks. The i5/OS server implementation of TCP/IP does not include full gateway function. Rather, a subset of the gateway functions is supported. One of the supported gateway functions is IP datagram-forwarding capabilities. IP does not forward datagrams between interfaces on the same subnet.
   - The IP reassembly time-out and IP time-to-live parameters can be increased if clients connected over a very slow WAN interface have communication problems.
   - Enable dead gateway detection detects the dead gateways on this system. Gateways marked as dead will continue to be polled and when they respond again all routes using that gateway will be reactivated. Detection interval specifies how often dead gateway detection should be performed.

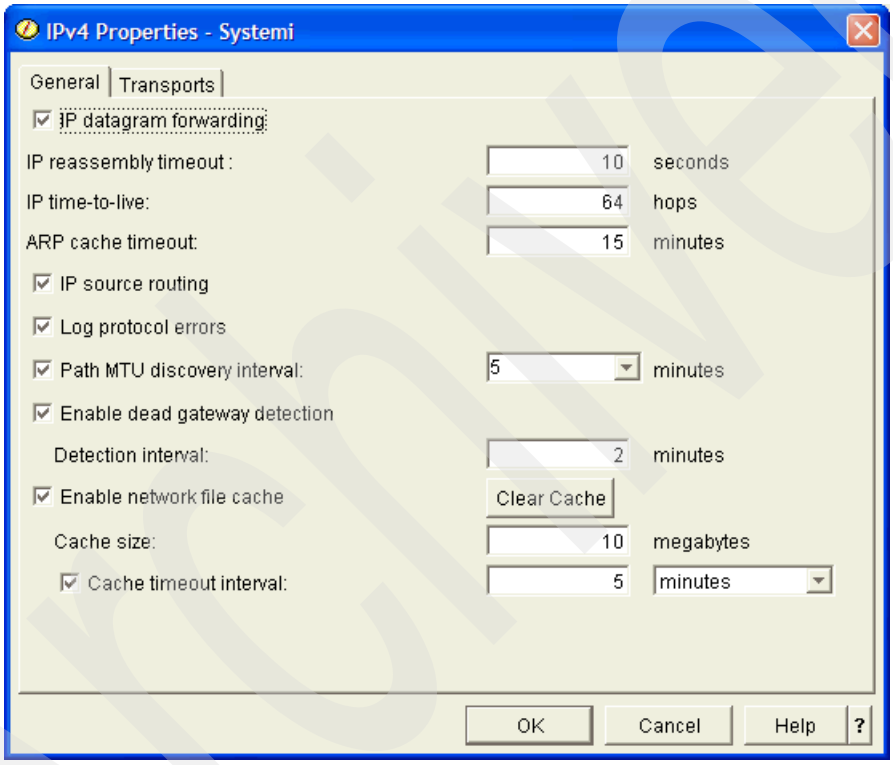


Figure 2-22 IPV4 Properties General tab

3. Click the **Transport** Tab (Figure 2-23) to change the parameters for TCP keep-alive timer, TCP urgent pointer, TCP receive and send buffers, and more:
   - TCP keep-alive specifies the amount of time, in minutes, that TCP waits before sending a probe to the other side of a connection. TCP sends the probe when the connection is otherwise idle, even when there is no data to be sent.
   - The TCP urgent pointer defines the convention to follow when TCP interprets which byte the urgent pointer in the TCP header points to. This value is set on a system basis and must be consistent between the local and remote ends of a TCP connection. All applications using this system use this value. The default is BSD.
   - The UDP checksum parameter specifies whether User Datagram Protocol (UDP) processing generates and validates checksum. If you select UDP checksum, the system performs UDP checksum calculations that verify the integrity of received data. Checksums are useful in unreliable networks, but their calculations reduce performance. We recommend that you check the UDP checksum check box.
   - TCP R1 and R2 retransmission values specifies the number of times to retry the TCP transmission before TCP assumes that a connection is lost.

   Note: The TCP R1 retransmission value must be less than or equal to the TCP R2 retransmission value.

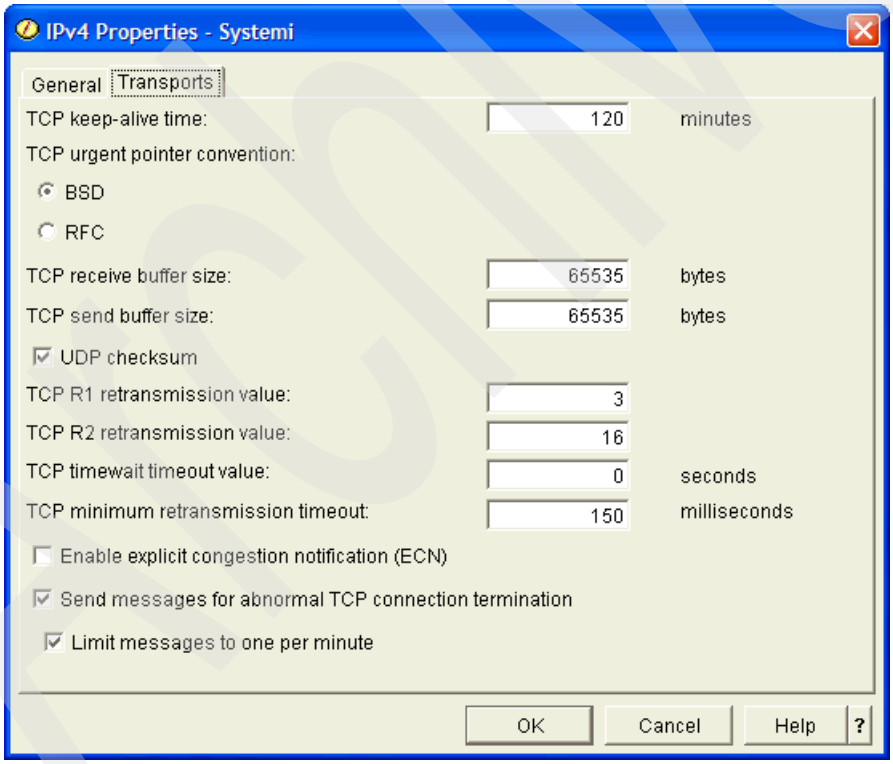


Figure 2-23 IPV4 Properties: Transports tab

4. After you have entered all of the desired parameters, click **OK**.

Verifying a TCP/IP connection

The TCP/IP connection to the other TCP/IP hosts can be verified using the utilities provided in the iSeries Navigator. They are described in detail.

PING

Verifying a network connection (PING) function is one of the best problem-determination tools available for quick diagnosis of a problem in your TCP/IP network. PING tests the TCP/IP connection between a system and the remote system specified on the remote system parameter. It tells you whether you can see the host to which you are trying to connect.

When you PING a machine, you send an Internet Control Message Protocol (ICMP) echo request to that machine. A successful reply means that the network's primary transport and communication systems are functioning properly.

To PING a machine using iSeries Navigator, complete the following steps:

1. Expand **My Connection** → <server name> → **Network**. Right-click **TCP/IP configuration** and select **Utilities** → **Ping** (Figure 2-24).

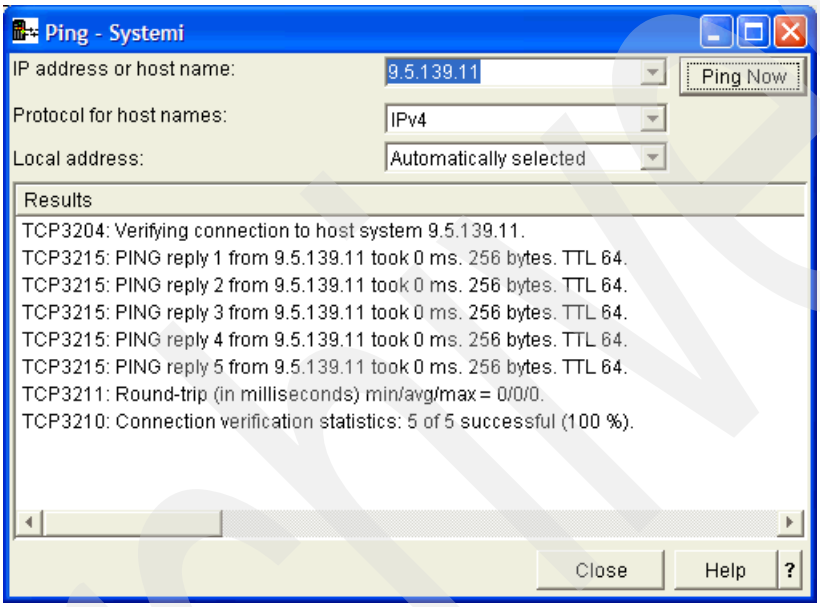


Figure 2-24 Ping window

2. Enter the IP address of the system you want to ping.
3. Click **Ping Now**. The results are shown in the window.

Trace Route

This utility can be used trace a route and show a list of routers between your site and the host that you specify. You can specify parameters to consider when performing the trace; otherwise, the default values are used.

To trace the route to a host from the i5/OS server and discover all routers:

1. In iSeries Navigator expand **My Connection** → <server name> → **Network**. Right-click **TCP/IP configuration** and select **Utilities** → **Trace Route**. This opens the window shown in Figure 2-25.

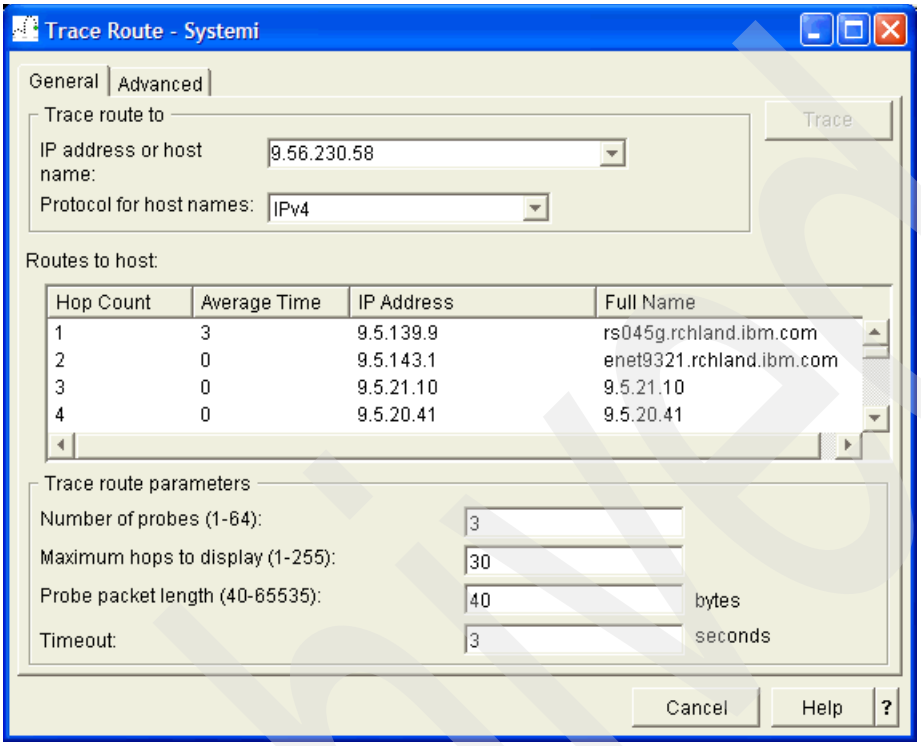


Figure 2-25 Trace Route: General window

2. Enter the IP address of the host and parameters if you want to change any; otherwise, use the defaults.
3. The utility shows all of the routers passed by the TCP/IP packet to reach the host. It is very helpful in debugging the network problems.

ARP cache

The ARP cache provides a one-to-one mapping of IP addresses to MAC addresses. This cache is used when communicating with hosts via a LAN interface, which resides on the same subnet as the i5/OS server.

The i5/OS server automatically clears the ARP cache based on the time specified in the TCP/IP attribute ARP cache time-out. (The default is 15 minutes.) This is part of the normal operation of the LAN interface ARP cache on the i5/OS server.

Note: The System i only clears entries that have not been used for the past 15 minutes (default). It will not remove active entries in an attempt to improve performance.

This attribute can be changed through iSeries Navigator: Select **Network → TCP/IP Configuration.** Right-click **IPv4** and select **Properties**. The IPv4 window is presented as shown in Figure 2-26.

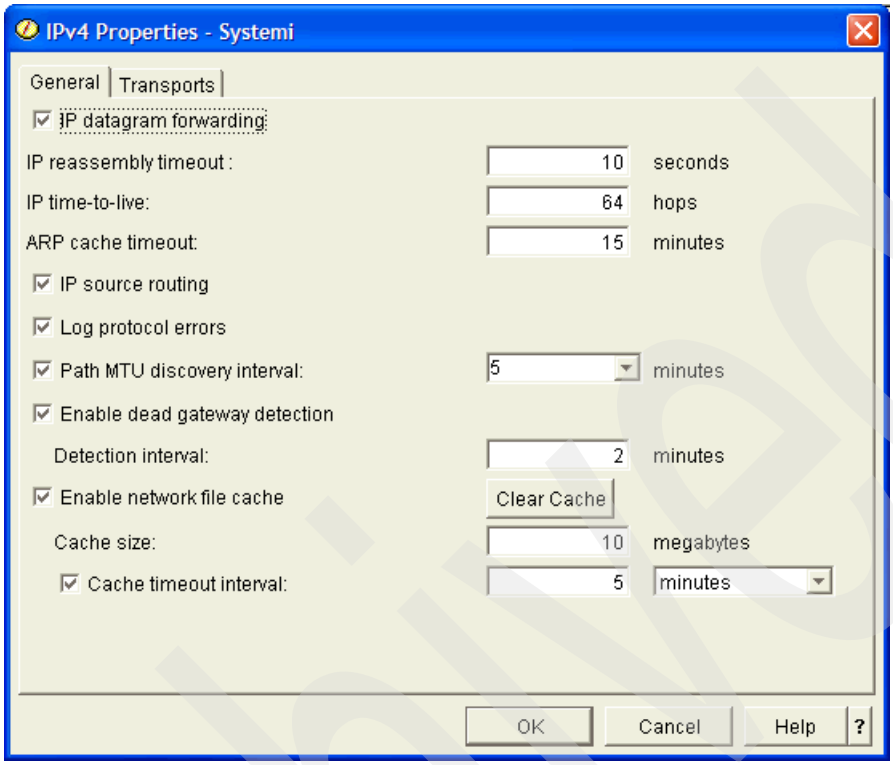


Figure 2-26 TCP/IP properties

In fact, the ARP cache is kept at the physical interface level even though you use the logical IP address to drive to the ARP cache.

In V5R1 of OS/400 the ability to view and manage the ARP cache became available. The ARP cache is viewed via iSeries Navigator (only). One ARP cache is kept for each physical LAN interface on the i5/OS server.

To view the ARP cache for a specific line, expand **Network → TCP/IP Configuration → IPv4** and then select **Interfaces**. Right-click the **IP address** associated with the physical interface and select **ARP** to view the ARP cache table as displayed in Figure 2-27.

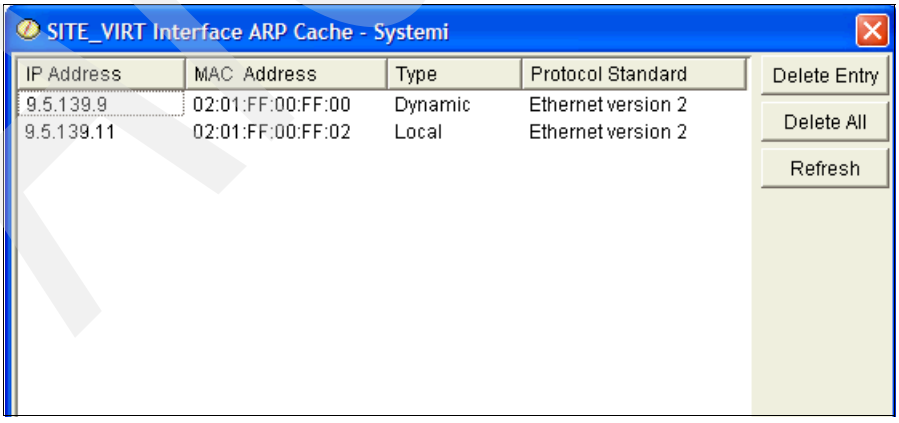


Figure 2-27 ARP cache table

You can select and delete a specific dynamic entry or all dynamic entries. Only dynamic entries may be deleted.

The sequence to delete a single ARP cache entry is to first select the entry, then click **Delete Entry**. Only when you click **Apply** will the entry be deleted.

Tip: The easiest way to clear a single ARP entry is to simply click **Delete All**. The effort for the i5/OS server to refill the ARP cache with new entries is minimal.

2.1.2 Wide area network (WAN) interface wizard for frame relay

The WAN interface wizard is used to create an interface that is used for frame relay. The wizard enables the creation of either a non-broadcast multi-access connection or a point-to-point connection. The WAN interface wizard is not commonly used.

Other types of WAN interfaces do exist on the i5/OS server (for example, Point-to-Point Protocol (PPP) and X.25). PPP interfaces are created via the Remote Access Services folder in iSeries Navigator. For more information about PPP, see Chapter 5, “Point-to-Point Protocol (PPP)” on page 87. For X.25 configuration, visit the System i and i5/OS Information Center at:

http://publib.boulder.ibm.com/infocenter/iseries/v5r4/index.jsp

2.1.3 Virtual IP address

Another popular term for the virtual IP address is *circuitless*.

Virtual IP address (VIPA) support was first introduced in V4R3 of OS/400.

Virtual IP is a powerful new feature with many different applications. Because VIPAs are not bound to a single physical interface, they provide a simple way to define system-wide IP addresses. They enable the i5/OS server to be known by a single IP address, even when it is attached to many different networks.

This is an *extremely* important building block to your i5/OS server TCP/IP network, as Virtual IP allows for the implementation of load balancing, fault tolerance, and anchors for unnumbered interfaces.

VIPA pre-V5R2: not directly routable

Pre-V5R2, VIPAs were not directly routable. This means that prior to V5R2, the i5/OS server would never respond to an ARP request for a virtual IP address. In order for other hosts to reach the i5/OS server VIPA, they had to have an indirect route defined that specified the IP address of the System i physical adapter as the next hop gateway to be used to reach the VIPA. Remote hosts would connect through a gateway router. Therefore, only the local gateway router had to have indirect routes defined for the i5/OS server virtual IP address. Local hosts each had to have a host route defined for the VIPA or point to the local router that has the indirect routes defined for the VIPA.

Virtual IP can be used to provide continuous availability even through an interface adapter failure. Introduced in V4R3, Virtual IP provides a way to define an IP address for the system that is not bound to any one physical adapter. Virtual IP was originally introduced for load balancing, but it can also be used to provide fault tolerance across a local adapter failure. If remote clients or the DNS interrogated by the remote clients only know the i5/OS server by its virtual IP address, and the local gateways know the paths to reach the VIPA, then the system will stay accessible as long as at least one physical interface is active. This provides enhanced system availability, a feature being requested more and more.

In Figure 2-28, the i5/OS server has three adapters connecting it to the 172.23.10.0 network. Remote clients know the i5/OS server by the virtual IP address of 172.23.10.88. Routers R1 and R2 each have three routes configured to the virtual IP address, with next hop gateway addresses of 172.23.10.1, 172.23.10.2, and 172.23.10.3. As long as at least one of the three System i adapters is active, the System i remains accessible to remote clients.

Note: The clients connected to the local LAN shown in Figure 2-28 do not have *easy* access to AS20 via the VIPA of 172.23.10.88. Both the routers and all of the local LAN clients must have indirect routes defined for the VIPA.

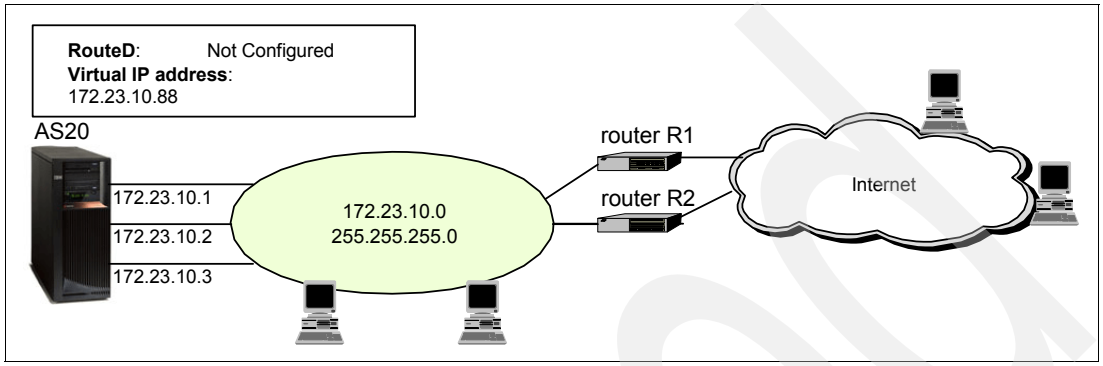


Figure 2-28 Fault tolerance with Virtual IP

As stated earlier, pre-V5R2 VIPA will not respond to ARP requests. That is why in Figure 2-28, both of the routers have explicit routes configured in order to forward packets to the Virtual IP interface. On the plus side, this enables the same virtual IP address to be configured on multiple machines. This is exactly what is commonly done for high-availability Web serving with multiple CPUs. (See “IP address takeover” on page 49.)

In order for locally attached hosts to participate at pre-V5R2, they must have an indirect route defined for the virtual IP address or point to a local router that has indirect routes defined. The downside for locally attached hosts is that configuring routes on each locally attached client is cumbersome at best and impractical at worst. Additionally, local attached hosts sending data through the router just add unnecessary traffic to the router. An alternative is for local clients to access the System i server using only one of the physical IP addresses, but if the adapter on which that IP address is defined goes down, the System i appears down to the client. The net result is that, for locally attached clients, the fault tolerant advantages of Virtual IP are not available (without additional configuration on each local client).

The previously mentioned problems are resolved at V5R2 through the introduction of proxy ARP for virtual IP addresses. Proxy ARP for Virtual IP enables the System i to answer ARP requests for the virtual IP address.

The Route Daemon (RouteD) server (RIPv2) on the System i can be used to advertise routes for the VIPA. The RouteD server advertises one of the physical interfaces as being the next hop for accessing the VIPA.

VIPA V5R2 and beyond: directly routable with proxy ARP support

The whole point of the proxy ARP for VIPA is so that local LAN hosts no longer see the VIPA any differently.

V5R2 of OS/400 brought even more power to the VIPA with the ability to do proxy ARP for a virtual IP address. Proxy ARP enables the System i to respond to ARP requests for the VIPA. This simplifies fault tolerance for locally attached LAN devices because the configuration of indirect routes in the local LAN attached hosts is no longer necessary. Proxy ARP is also supported for virtual Ethernet addresses.

Figure 2-29 shows a System i named AS20 with two physical interfaces. Each interface has a unique MAC address. We also have configured a VIPA with proxy ARP enabled. When the interfaces are started, the VIPA is associated with any active IP address within the same subnet. For example, in Figure 2-29 the VIPA is associated with IP address 172.23.10.1.

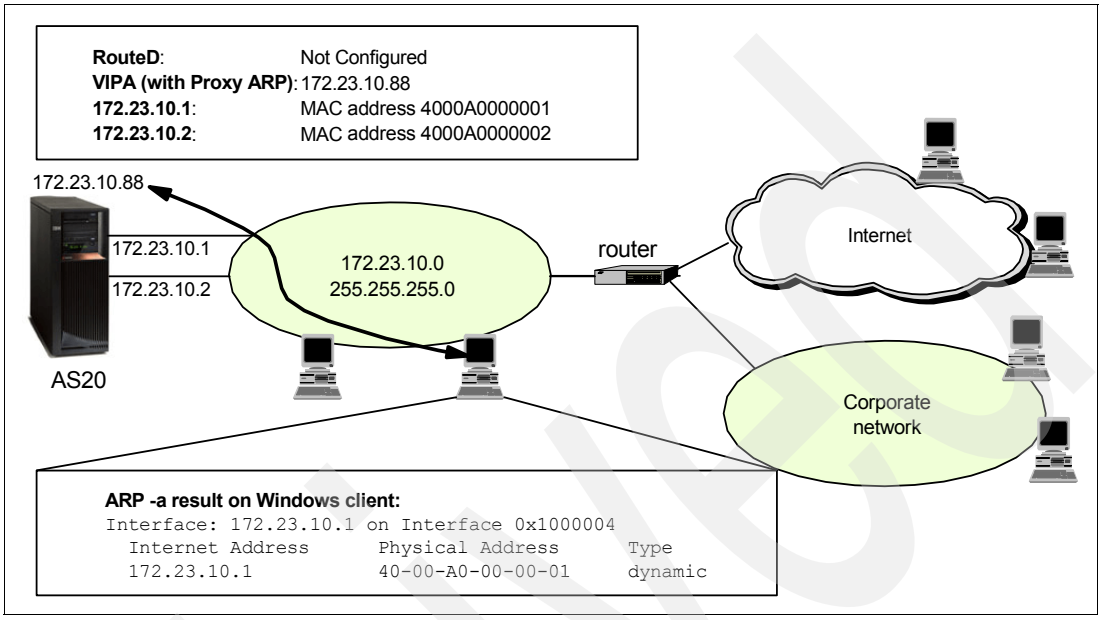


Figure 2-29 Virtual IP: proxy ARP enables local clients to map the virtual IP address to MAC address

Tip: If you create a VIPA and select proxy ARP, there *must* be at least one local and active LAN interface connected to a subnet that includes the VIPA. Otherwise, the proxy ARP function cannot work.

When a host (shown as a PC client) on the same subnet does an ARP (maybe as the result of a PING to 172.23.10.88), the System i responds to the ARP with the MAC address of 4000A0000001. At that point the host updates its own ARP cache with the MAC address of 4000A0000001.

If there is a failure of the 172.23.10.1 interface, the System i automatically switches the association of the VIPA to another active interface. (In the case of Figure 2-29, this would be 172.23.10.2.) The System i sends a broadcast ARP that causes all clients in the local LAN segment to update their ARP cache with the new MAC address of 4000A0000002.

Communications between the client and server will continue as normal. No interruption to the application will be seen.

The iSeries Navigator interface wizard does not allow for the enablement of proxy ARP during the creation of an interface. Proxy ARP can be enabled by displaying the properties of an existing virtual IP or virtual Ethernet interface and selecting the Advanced tab. Proxy ARP for a physical interface will default off when the interface is first created. Figure 2-30 shows the contents of the interface's Advanced tab. To enable proxy ARP, select **Enable proxy ARP**.

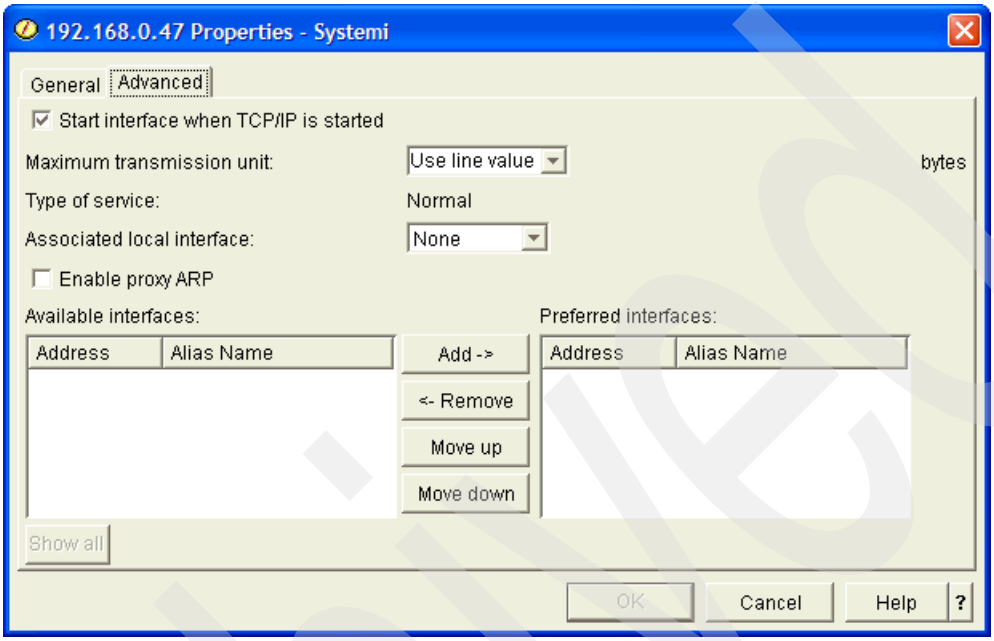


Figure 2-30 Proxy ARP enablement

For a detailed example of how to configure this support in V5R2, see 12.2, "Fault tolerance: proxy ARP for the virtual IP address" on page 189.

How a MAC address is selected for a VIPA

The process for selecting a MAC address or a proxy ARP agent for a VIPA has undergone several changes since proxy ARP was first introduced in V5R2. At that time, the proxy agent selected was the *first acceptable proxy agent* that was found (in other words, the first LAN adapter found that has an active IP address configured on it, and whose network address includes the VIPA would be selected). This selection method provided the user with little control over which agent got selected. The way a user could control agent selection was to start the VIPA and agent interfaces in a specific order so that the preferred agent interface was always started after the VIPA and before any other potential agents were started.

In V5R3, proxy agent selection was changed so that it was also based on the speed of the available interfaces. The highest available interface would be selected first. In addition, if multiple VIPAs were being proxied, the selected proxy agent selection would be spread across the available interfaces. Again, the user had little control over agent selection.

In V5R4, the user has the ability to control the specific proxy agents and the order in which the agents are selected for a specific virtual IP or virtual Ethernet interface. This is accomplished through the configuration of a preferred interface list. The preferred interface list can be configured through iSeries Navigator or through the Change IPv4 Interface API (QTOCC4IF). For information about configuring the preferred interface list programmatically, see Chapter 22, "Using alias names and setting proxy ARP and preferred interface lists programmatically" on page 669.

The iSeries Navigator interface wizard does not allow for configuration of the preferred interface list during the creation of an interface. By default, a preferred interface list will not be set, so agent selection will be done in the same manner as the automatic agent selection in V5R3. The default is off when the interface is first created. The configuration of the preferred interface list must be done by displaying the properties of an existing virtual IP or virtual Ethernet interface and selecting the Advanced tab. The preferred interface list is an ordered list of up to 10 interfaces. Figure 2-31 shows an example of the contents of the Advanced tab for a virtual IP address with a preferred interface list.

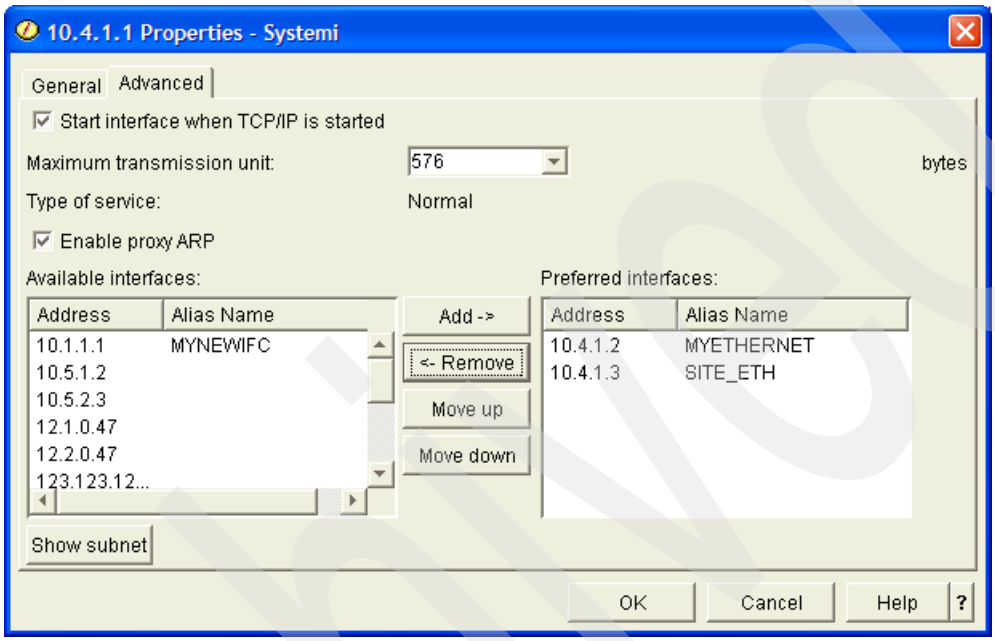


Figure 2-31 Configuration of a preferred interface list

In this example, the virtual IP address 10.4.1.1 has a preferred interface list that consists of the interfaces 10.4.1.2 and 10.4.1.3. Interface 10.4.1.2 is the first interface in the list, so it is preferred over interface 10.4.1.3. When all of the above interfaces are active, 10.4.1.2 will respond to any ARP requests for virtual IP address 10.4.1.1. If interface 10.4.1.2 fails or becomes inactive, the next interface in the list 10.4.1.3 will respond to any ARP requests for interface 10.4.1.1. Once 10.4.1.2 recovers from the failed or inactive state and becomes active, it will once again respond to ARP requests for 10.4.1.1.

2.2 Routing

Routes on the System i provide a path to a network or a host. The route destination and subnet mask define the host or range of hosts that are reachable via a route. Each route is bound to an interface. In turn, the interface points to a line description and the line description points to a hardware resource.

2.2.1 Types of routes

It is very important to know the types of routes on the System i in order to understand how routing works on the System i. Routes are added manually or dynamically.

Direct routes

Direct routing occurs when the destination host is attached to the same physical network as the source host, which means that IP datagrams can be directly exchanged. Direct routes are automatically added by the System i when an interface is added. A direct route defines the range of external hosts that are locally connected (for example, locally reachable).

Indirect routes

Indirect routing occurs when the destination host is not connected to a network directly attached to the source host. The only way to reach the destination is by way of one or more IP gateways.

An indirect route can be a type of host, network, or default. Indirect routes can be configured manually or added dynamically via RIP, ICMP, and so on.

Manually creating routes with iSeries Navigator

Host, network, and default routes may be created using iSeries Navigator route wizard. To create a route, expand **Network** → **TCP/IP configuration** → **IPv4** and then right-click **Routes** and select **New Route**. Routes may be added to active (and inactive) interfaces. Figure 2-32 shows the first window presented by the route wizard.

Tip: In V5R1 and earlier versions of iSeries Navigator (the V5R1 Operations Navigator was renamed iSeries Navigator in V5R2), you could only add a route to an inactive interface. In addition, manually added routes are initiated by expanding **Network** → **TCP/IP configuration** and clicking **Interfaces**. Right-click the IP address of the *inactive* interface to which you want to add a new route, and select **Associated Routes** from the context menu. Here you can add new routes of type: host, network, and default.

For 5250 command entry, Add TCP/IP Route (ADDTCPRTE) enables you to add a new indirect route against an active (or inactive) interface.

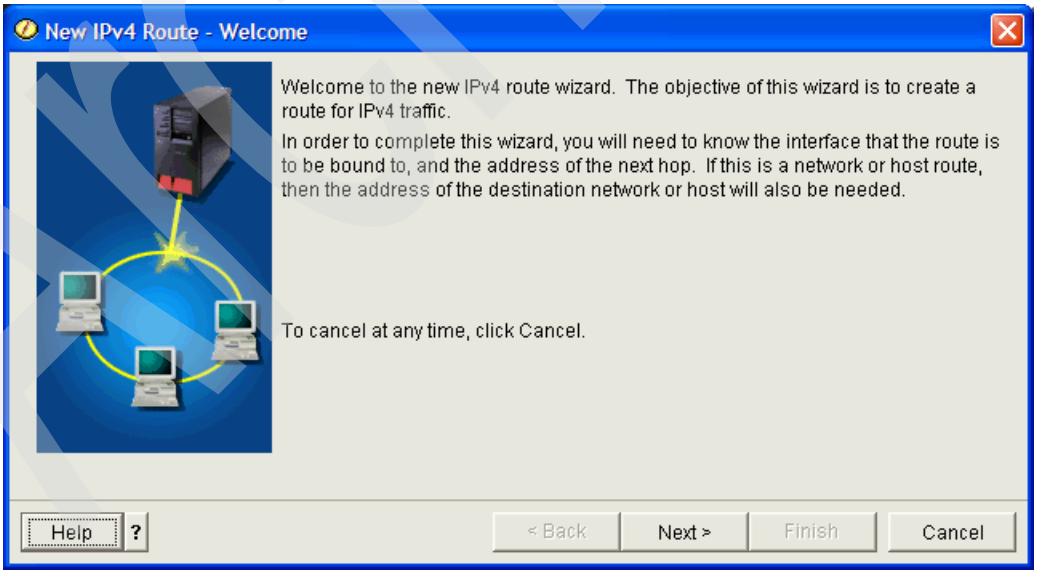


Figure 2-32 Route wizard

The two parameters to consider when creating routes are route precedence and preferred binding interface. Both of these parameters are very important in the discussion of load balancing and fault tolerance. Read more in 2.3.2, “Load balancing” on page 50.

Route precedence cannot be set via the wizard. Routes created with the wizard are created with a default route precedence of 5 (five). The route priority may be changed through iSeries Navigator by right-clicking on the inactive route and selecting **Properties**. (Changes to the route cannot be made unless the interface to which the route is bound is ended.) If your iSeries Navigator is using this interface for its connection to the System i, it would then be necessary to make the changes from the console.

The preferred binding interface parameter can be configured via the route wizard. The preferred binding interface enables the user to explicitly bind a route to a specific interface by IP address, rather than have it bound to the first one the system sees.

Pre-V4R2, there was no user control of which interface a route would bind to. As a result, the indirect routes would bind to the first acceptable interface found. It was possible that all of the routes would be bound to a single interface.

These problems were solved through the introduction of preferred binding interface in V4R2. The preferred binding interface enables the user to explicitly bind a route to a specific interface by IP address rather than have it bound to the first one the system sees. The preferred binding interface is key to duplicate route round-robin load balancing. Figure 2-33 shows the iSeries Navigator route wizard with the option to specify a preferred binding interface.

Tip: The definition of *round-robin* from Merriam-Webster's online dictionary at `http://www.m-w.com/` is:

> **a:** A written petition, memorial, or protest to which the signatures are affixed in a circle, so as not to indicate who signed first **b:** A statement signed by several persons **c:** Something (as a letter) sent in turn to the members of a group, each of whom signs and forwards it sometimes after adding comment.

We use round-robin to indicate that the interfaces will be selected one after the other. For example, with three interfaces, the first, second, then third will be selected. For the next connection the System i will start back at the first.

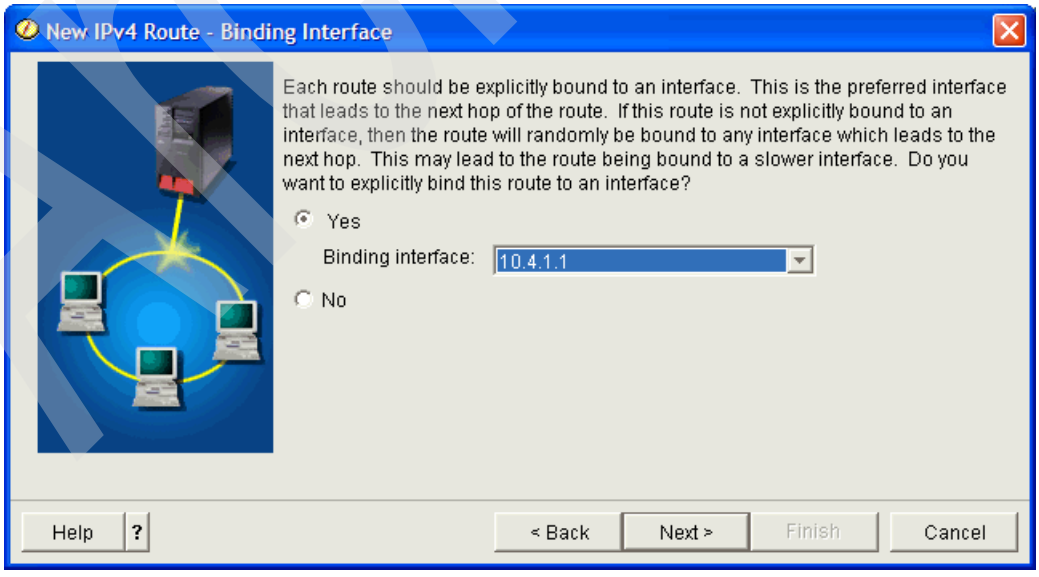


Figure 2-33 iSeries Navigator route wizard with the option to specify a preferred binding interface

Figure 2-34 shows three interfaces that are connected to the same network. We want to guarantee that no matter which interface receives the inbound request, it will be possible to send the reply back out the same interface. To do this, we must add equivalent, duplicate routes to each interface. In this example, we add three default routes, and each one is explicitly bound to a different interface. This binding will not change regardless of the order in which interfaces are started or ended.

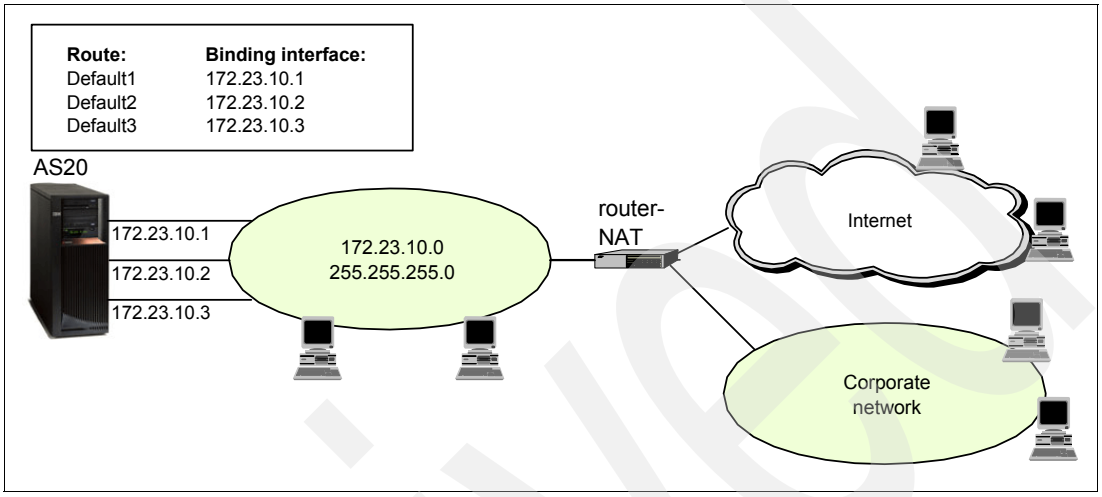


Figure 2-34 Route binding

Routing Information Protocol (RIP)

In V4R2, RIPv2, a widely used routing protocol, was added to the System i. It is an Interior Gateway Protocol (IGP) used to assist TCP/IP in the routing of IP data packets within an autonomous system (a group of networks and gateways controlled by a single administrative authority). Dynamic routing protocols enable you to handle larger networks where automatic switching to redundant routes or use of multiple routers on a network is desirable.

Figure 2-35 shows an example of a RIP implementation.

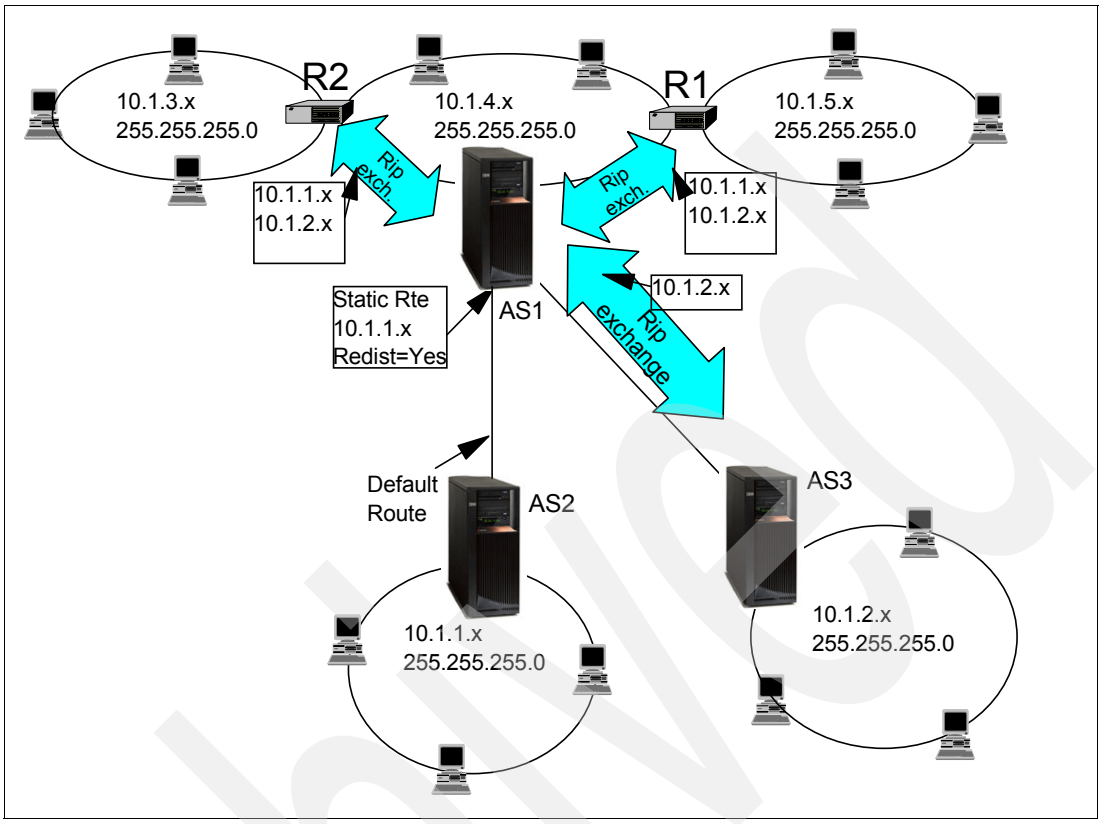


Figure 2-35 RIP to exchange routing information between System i and RIP capable routers

This series of steps represents an example of how RIP is used to exchange routing information in a network of System i and RIP-capable routers:

1. A static route is added to the central system (AS1) that describes the connection to the network 10.1.1.x via AS2. This is a static route (added by the network administrator) with Route redistribution set to *YES. This causes this route to be shared with other routers and systems, so that when they have traffic for 10.1.1.x they will route the traffic to the central System i (AS1).
2. Or, AS1 can distribute the route information that it receives from a remote RouteD server. In this example, AS3 has the RouteD server started so that it sends and receives RIP information. It sends the message that AS3 has a direct connection to 10.1.2.x.
3. AS1 receives this RIP packet and processes it. If it does not have a route to 10.1.2.x, it will store this route. If it does have a path to 10.1.2.x that is the same number of hops or fewer, it will discard this new route information. In this example it keeps the route data.
4. AS1 receives information from R1 with route information to 10.1.5.x. AS1 keeps this route information.
5. AS1 receives information from R2 with route information to 10.1.3.x. AS1 keeps this route information.
6. The next time AS1 sends RIP messages, it will send information to R1 and R2 that describes all of the connections AS1 knows about that R1 or R2 may not know about. AS1 sends route information about 10.1.1.x and 10.1.2.x, but does not send information about 10.1.4.x to R1 and R2, because AS1 knows that R1 and R2 are connected to 10.1.4.x and do not need a route. AS1 does not send information about 10.1.3.x and 10.1.5.x to R1 and R2, because AS1 learned those routes from routers R1 and R2.

7. Route information for 10.1.1.x, 10.1.3.x, 10.1.4.x, and 10.1.5.x is similarly sent to AS3.

Internet Control Message Protocol (ICMP)
An ICMP route on the System i is created when an ICMP Redirect is received from an intermediate router. The intermediate router is telling the System i that it should send future datagrams for a host to a router whose IP address is specified in the ICMP message. This preferred router will always be on the same subnet as the System i and the router that returned the ICMP redirect.

2.2.2 System i rules for route selection

Some basic routing rules apply to TCP/IP in general and to TCP/IP on the System i. You should consider these rules as you implement routing functions on your System i. These rules will help you to determine which route will be selected for a connection. The destination IP address in the packet and the route destination of the configured routes are used in the selection process. The route selected is based on:

- Route group search order:
  a. Direct routes
  b. (Sub)network routes
  c. Default routes

  Note: This is read as: the first choice for an IP packet would be any direct routes, then network or subnetwork routes, and finally any default routes.

- Within a group (a, b, or c from above), the route with the most specific subnet mask is chosen.
- Among equally specific routes, the route bound to the preferred source IP address is chosen.
- Equally specific routes are subject to list order or load balancing options.

2.3 Network administrator's tricks of advanced IP networks

We now discuss ways in which fault tolerance and load balancing can be implemented for the System i. Fault tolerance and load balancing improve performance and availability.

2.3.1 Fault tolerance

Fault tolerance means keeping your systems available 24 hours a day, seven days a week. This can be accomplished by many different techniques. One way is obviously to increase the reliability and availability of each individual component.

But often, a more cost-effective approach is to provide redundancy in your network so that a single point of failure does not take the entire system (or systems) down.

Various System i routing capabilities can play a role in the design of such redundant networks. Among them are:

- "Dead gateway detection" on page 48
- "IP address takeover" on page 49

Dead gateway detection

Per RFC 1122, the System i employs a Dead gateway (DG) detection mechanism to detect failures in locally connected gateways. DG is triggered by either of two events:

- Failure to receive a response to an ARP request sent to a gateway
- Excessive TCP re-transmits on a TCP connection that is using an indirect route

A series of enhancements to the System i DG support have been integrated into V4R5 (and later releases of OS/400 and i5/OS). DG now primarily depends on ARP to determine whether the gateway is dead. That is, if the gateway does not respond to ARP it *must* be broken. Moreover, ARP cache entries are purged and ARP re-resolves are forced when a problem is suspected.

The ARP-initiated process works as follows: If a packet is to be sent and no ARP entry exists, an ARP request is sent. If no ARP reply is received to successive requests, the gateway is considered down, affected routes are marked inactive, and DG slow polling starts. Slow polling is no longer just a PING request. Prior to sending the PING, any existing ARP cache entry for the suspect gateway is purged, forcing an ARP re-resolve. If ARP replies are received, even if no PING reply comes in, the routes are re-marked as active and the gateway is considered alive. So although PING is still involved, it is used mostly as a way to force the ARP cycle rather than the way to decide whether a gateway is alive.

Similar to above, when TCP hits its re-transmit threshold and tells IP that there may be a problem, IP now purges the ARP cache entry and sends the PING. As long as an ARP reply is received, the gateway is considered alive and no routes are inactivated. Only if no ARP reply is received are the routes marked inactive and does DG slow polling start and proceed as above. With TCP connections, gateway failures may be recognized in as little as 10 to 20 seconds.

When routes are marked inactive, the System i TCP/IP attempts to reroute connections over an alternate route. In Figure 2-36, when router R1 fails, if a route exists that goes through router R2, connections will be rerouted.

When R1 comes back, active connections will stay routed through R2. However, new connections will be routed over R1, just as prior to the failure.

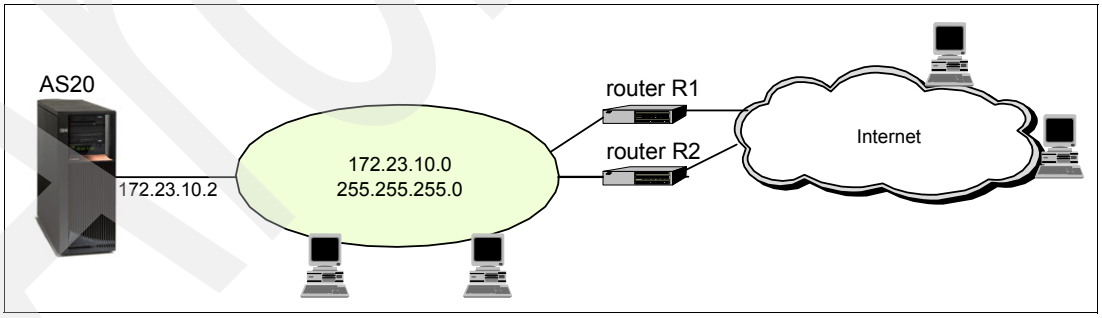


Figure 2-36 Dead gateway processing

At V5R1 and V5R2, dead gateway processing can be disabled. Figure 2-37 shows the iSeries Navigator TCP/IP attributes window and the option to enable dead gateway processing.

Tip: One reason you might want to disable dead gateway processing is to avoid the slow poll of the gateway that might be on the other side of an ISDN connection. Doing so might increase your connection fees from your network provider.

Another would be if there is only one gateway available through which traffic can be routed. If no backup path exists, you might as well leave the primary path active.

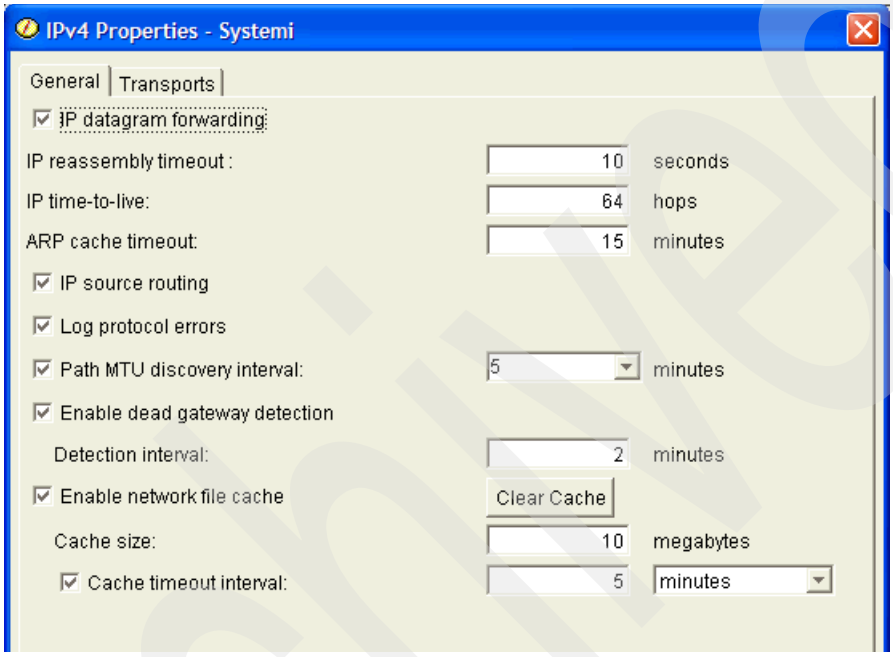


Figure 2-37 IPv4 properties: dead gateway processing can be disabled or enabled

IP address takeover

A virtual IP addresses (VIPA) can improve system availability when used in conjunction with the clustering product. The clustering application controls which system the VIPA is active on at any point in time. When that system is taken down, the same VIPA is activated on a backup system.

Tip: Go to the System i Information Center at:

`http://publib.boulder.ibm.com/infocenter/iseries/v5r4/index.jsp`

Select **Systems management** → **Clusters** for more information about clustering and the IP address.

In addition, the Redbooks publication *IBM HTTP Server (powered by Apache): An Integrated Solution for IBM eServer iSeries Servers*, SG24-6716, has an example of two HTTP servers running on two different System i's in a Highly Available HTTP server configuration. A failure of the primary System i causes the backup server to automatically take over the IP address of the primary.

If the backup system is connected to the same network as the primary system, no special routing procedures are required. Consider AS1 as the primary system and AS2 the backup (Figure 2-38). When the takeover IP address comes active on AS2, it broadcasts an ARP packet to the rest of the local network, informing all other hosts that the IP address has moved to a new system.

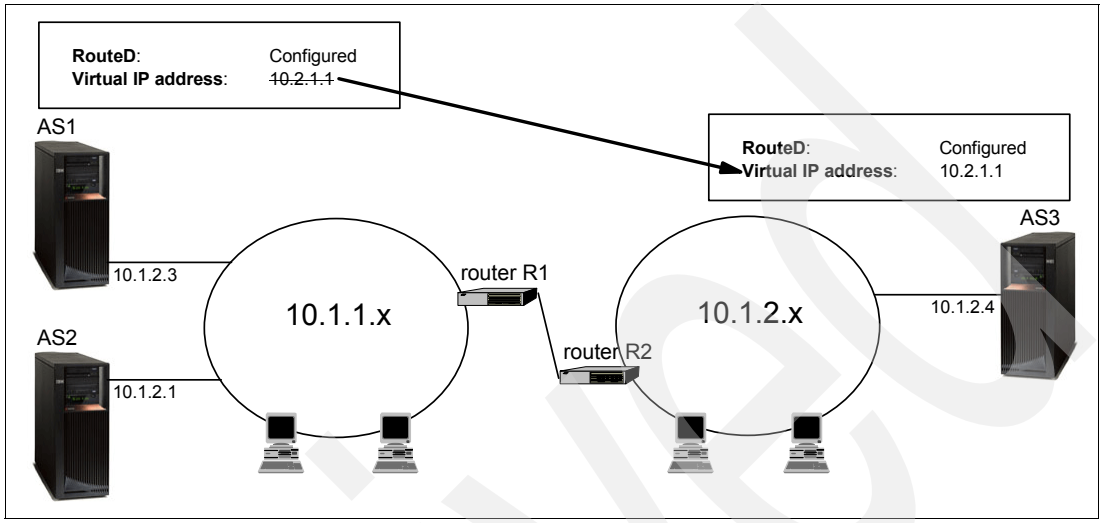


Figure 2-38 IP address takeover using virtual IP addresses

IP address takeover is not limited to both machines being on the same network. All we need is to define the takeover address as an address that is not directly accessible from either of the local networks (that is, a virtual IP address).

For example, consider the backup system being AS3, rather than AS2. In this case, we need to define the takeover address as a VIPA that is not part of either of the local networks to which the System i's are attached. That is why the virtual IP address is defined as 10.2.1.1. This address is not part of either the 10.1.1.x or the 10.1.2.x networks.

When the 10.2.1.1 takeover address is moved from AS1 to AS3, RIPv2 will advertise to the rest of the network that 10.2.1.1 is now reachable by AS3. Assuming that intermediate routers are also running RIPv2, the route tables throughout the rest of the network will be updated within a few minutes.

2.3.2 Load balancing

As the amount of IP traffic handled by your System i continues to increase, you should look for ways in which to balance the workload. Load balancing provides a way to accomplish this. TCP/IP data flow can be balanced across network adapters or between System i's.

To study load balancing on the System i it is best to divide the problem in two. We define outbound load balancing for IP datagrams sent from your System i. We define inbound load balancing for IP datagrams received to your System i.

Outbound

The System i has the ability to balance outbound IP traffic across multiple TCP/IP interfaces. This was made possible by the introduction of the Duplicate Route Round-Robin (DRRR) method of load balancing in V4R2. This method is oriented toward remotely connected clients. This method of load balancing is based on two indirect route parameters:

Duplicate Route Priority Specifies a priority of a route in comparison to matching routes

Preferred Binding Interface Gives the ability to bind a route to a specific IP interface rather than binding to the first matching IP interface

Figure 2-39 shows a DRRR configuration. The example has three adapters on the system, all connected to the same LAN segment. Adapter 172.23.10.1 has been set up for inbound traffic only, and the other two adapters (172.23.10.2 and 172.23.10.3) are configured as outbound.

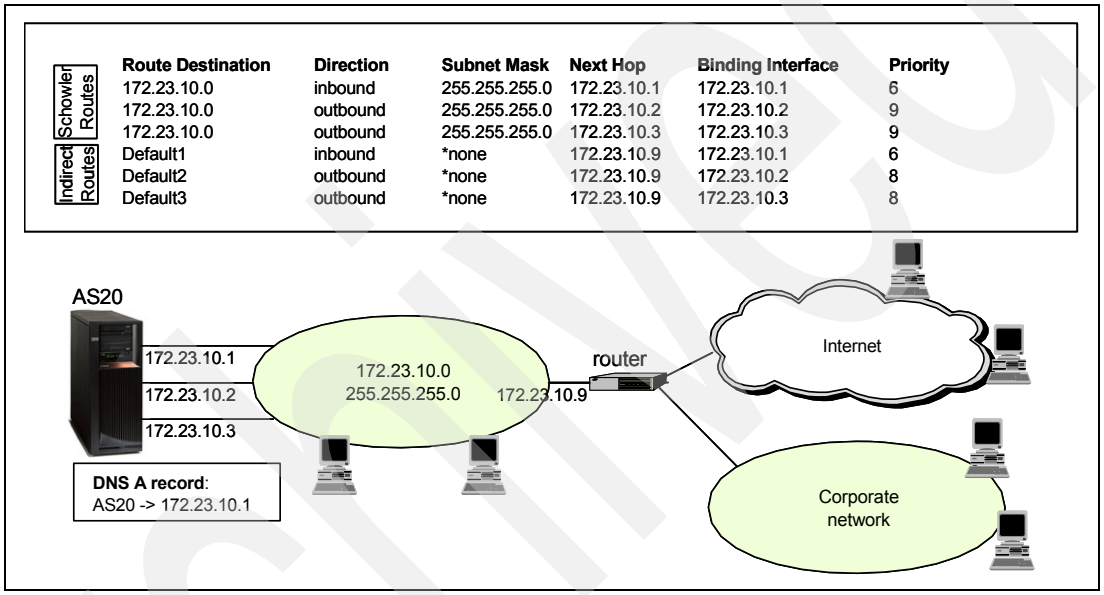


Figure 2-39 Duplicate Route Round-Robin (DRRR) for indirect routes Default1, Default2, and Default3

How are the two outbound adapters configured for DRRR? The Duplicate route priority (DUPRTEPTY) and the Preferred binding interface (BINDIFC) parameters were added to the configuration for indirect routes in V4R2. The DUPRTEPTY has to be set to a number higher than five, and the BINDIFC has to specify an physical interface. Routed connections will round-robin between the routes that have the same priority.

In Figure 2-39 we want to use the 172.23.10.1 interface as our primary inbound adapter. This is accomplished by configuring only this IP address in the DNS for our host and by configuring a lower DUPRTEPTY of six in the default route that is bound to the 172.23.10.1 interface. We configured the other two adapters with a DUPRTEPTY of eight. The DUPRTEPTY of six will cause this interface not be selected for outbound connections unless all of the DUPRTEPTY of eight interfaces are down.

Notice that the DNS is pointing to the 172.23.10.1 interface, making it the inbound interface.

This works well for indirect routes. But what can be done for the direct routes and those local hosts connected directly to the 172.23.10.0 subnet? Schowler routes, which were originally introduced via PTFs in V4R3 and V4R4, enable load balancing to be applied to local hosts. See “Schowler routes” on page 52.

Tip: The load balancing example in Figure 2-39 on page 51 can be combined with Virtual IP to give the added benefit of fault tolerance.

Schowler routes

Schowler routes extend the load balancing capability to locally connected hosts. A Schowler route is functionally equivalent to the *DIRECT route that it replaces, but it is added just like any other indirect route so the two load-balancing parameters (DUPRTEPTY and BINDIFC) can now be configured by the user. Schowler routes have three special characteristics:

- They are network routes.
- These network routes have the same route destination, subnet mask, and Type of Service (TOS) setting as the equivalent *DIRECT route.
- The Next Hop and Preferred Binding Interface IP addresses are both set to the IP address of the associated physical interface.

When the Duplicate Route Priority is set to greater than the default of five, the equivalent Schowler routes are selected in a round-robin fashion, identical to what can be done with other indirect routes.

Another use of Schowler routes is to reverse the default System i TCP/IP routing logic that always prioritizes direct routes over indirect routes, even host routes. (See 2.2.2, “System i rules for route selection” on page 47 for details.) By replacing the direct routes with Schowler routes, no highest priority direct routes will be found during route lookup. All candidate routes are now indirect and prioritized by subnet mask. Thus, a host route with a subnet mask of 255.255.255.255 will be considered the highest priority route.

Tip: How can you tell whether you have configured a Schowler route correctly? A quick and easy way is to look at your IPv4 Routes using iSeries Navigator (or from 5250 command entry with the Work with TCP/IP Network Status (NETSTAT) option 2=Display TCP/IP Route Information). The direct routes that are automatically created for the interface will have been replaced by the active Schowler routes. When the Schowler routes are removed, the direct routes will dynamically come back.

Looking back to Figure 2-39 on page 51, three Schowler routes are configured that replace the implicit direct routes created when you started the 172.23.10.1, 172.23.10.2, and 172.23.10.3 interfaces. All outbound traffic will be directed through the two interfaces, 172.23.10.2 and 172.23.10.3. You accomplish this by configuring a higher route priority on the Schowler routes for 172.23.10.2 and 172.23.10.3. When routes are selected for local outbound traffic only, these two interfaces will be selected.

Inbound

There are numerous ways in which to load balance on the inbound. Some methods used to accomplish inbound load balancing are DNS, router, and the IBM WebSphere Edge Server.

DNS-based inbound load balancing

The DNS method works by passing out multiple addresses for the same system name. The DNS serves a different IP address each time a request is made for the address (A) record for the system name. The addresses given out can reside on the same System i or multiple System i. In Figure 2-40, each address corresponds to a different system. This enables users to provide load balancing across two separate systems. This type of load balancing provides balancing by the connect request. In most cases, after a client has resolved the address, the client caches the address and does not ask again. This type of load balancing does not consider the amount of traffic going to each system.

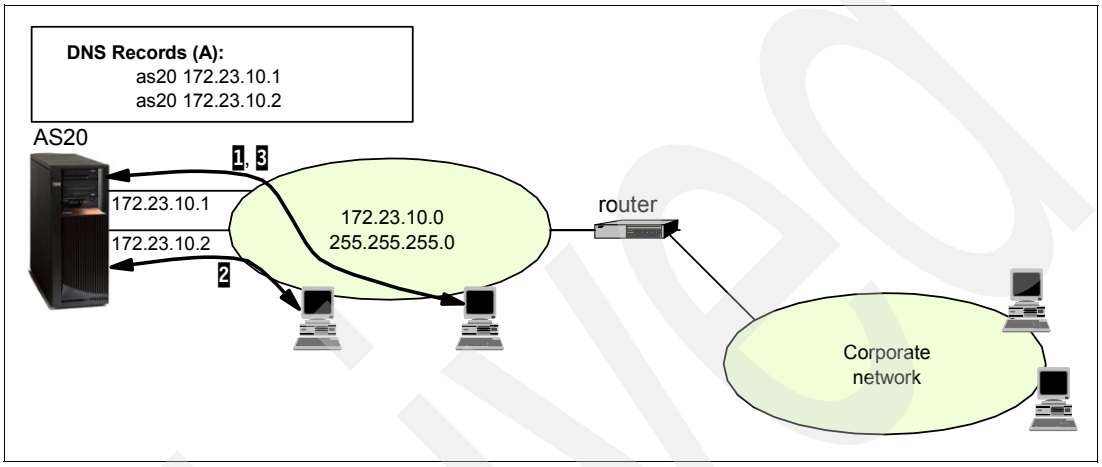


Figure 2-40 DNS-based load balancing

Router-based inbound load balancing

The use of a router for load balancing is done by configuring the router with multiple indirect routes pointing to the System i. The router then round-robins in its selection of routes when forwarding packets to the System i.

IBM WebSphere Edge Server–based inbound load balancing

The IBM WebSphere Edge Server differs from the DNS and router-based methods in that it does load balancing based on usage, not just connections. IBM WebSphere Edge Server provides extensive and powerful user controls on how network connections should be distributed across multiple interfaces or servers. For additional details, see:

`http://www.ibm.com/software/webservers/edgeserver/`

2.3.3 Classless InterDomain Routing (CIDR)

CIDR (or sometimes supernetting) is a way to combine several Class C network address ranges into a single network or route. This was implemented on the System i in V4R3. In the past you were required to enter a subnet mask that was equal to or greater than the mask required for the network class. For Class C addresses, this meant that a subnet of 255.255.255.0 was the largest (253 host) that could be specified. When a company needed more than 253 hosts, they would obtain additional Class C network addresses. This made the configuration of routes and other things difficult.

CIDR allows contiguous Class C addresses to be combined into a single network address range by using the subnet mask. For example, if you are given the Class C network addresses 208.222.148.0, 208.222.149.0, 208.222.150.0, and 208.222.151.0 with a subnet mask of 255.255.255.0, you could ask your ISP to make them a supernet by using the subnet mask 255.255.252.0. This mask would combine the four network addresses into one for routing purposes.

In Figure 2-41, the router is set up to send one RIP message with the network address 210.1.0.0 subnet mask 255.255.240.0. This tells the systems that receive the RIP message that networks 210.1.0.0 through 210.1.15.0 can be reached using this router. This sends one message rather than the 16 that it would take to convey the same information if CIDR was not available.

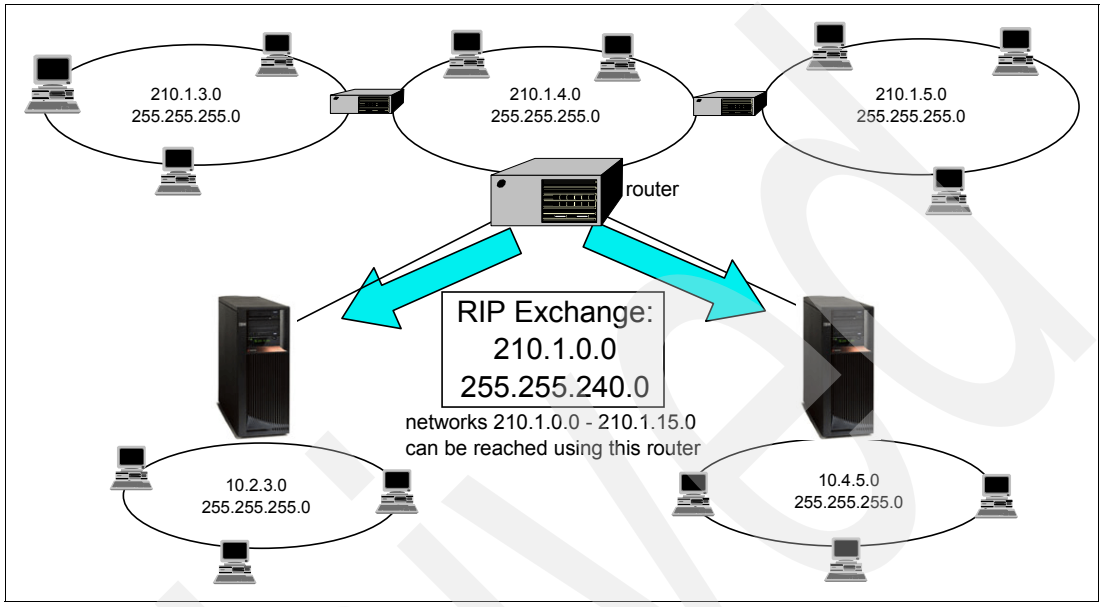


Figure 2-41 CIDR allows contiguous Class C addresses to be combined into a single network address

2.3.4 Transparent subnetting

Transparent subnetting was introduced on the System i in V4R2 to provide twinax attached network stations and PCs with twinax card access beyond the workstation controller. It is based on RFC1027, *Using ARP to Implement Transparent Subnet Gateways*.

Its use has grown to include providing proxy ARP support for real LANs.

Transparent subnetting for twinax connections

The twinax LANs are defined in address ranges that are within the real LAN address range. Prior to V4R2, the edits on the Add TCP/IP Route (ADDTCPRTE) and Add TCP/IP Interface (ADDTCPIFC) would not allow this to happen. In V4R2 the edits were relaxed, enabling two interfaces in different segments to have addresses that appear to be in the same segment. When the System i sees this happen, it will automatically proxy ARP for any systems that are attached behind the twinax controller.

Transparent subnetting is nothing more than extending the proxy ARP concept from proxy for a single host to proxy for an entire subnet or range of hosts. Figure 2-42 shows an example of classic transparent subnetting. It enables all of the systems on the 10.1.x.x network to communicate with all of the subnetted systems with no changes to the systems on the 10.1.x.x network.

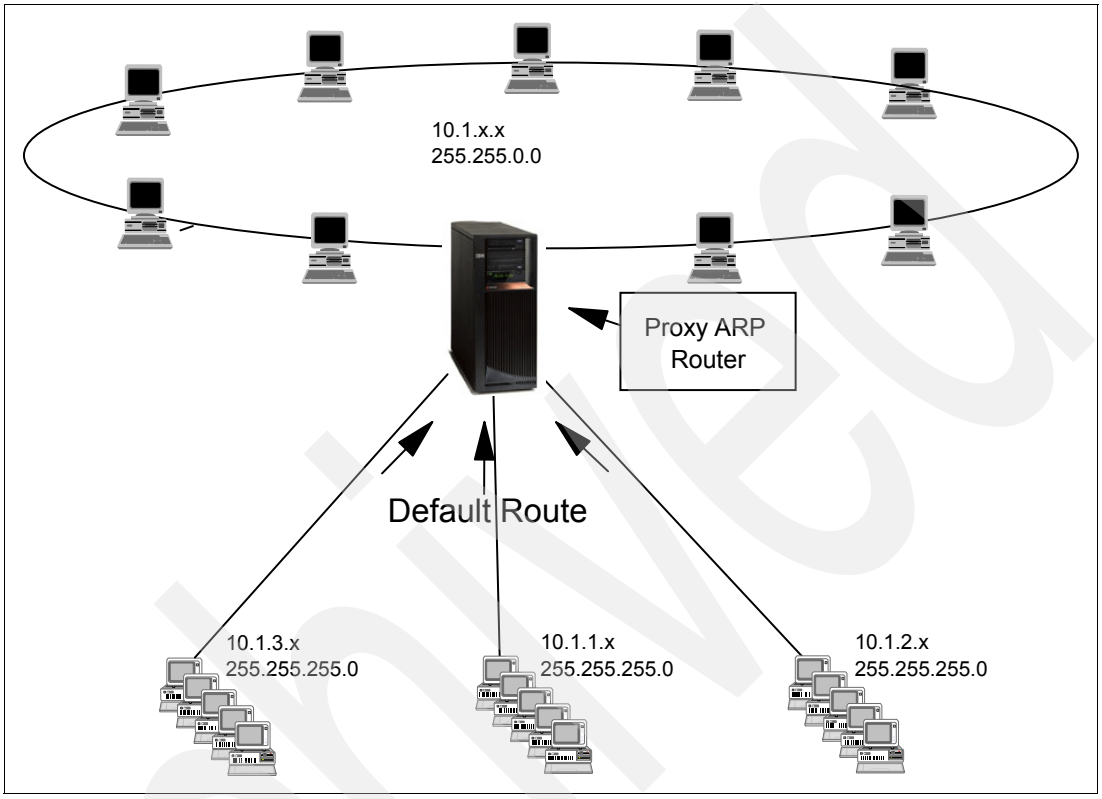


Figure 2-42 Classic transparent subnetting

Transparent subnetting for WAN attached LANs

The transparent subnet feature can be further expanded to handle *real* LANs that are remotely located over a WAN connection.

We have three networks that are attached to the home 10.1.x.x network via the System i. These networks are all defined using a subnet mask that makes them a transparent subnet to the home network. Once again, proxy ARP will respond to any ARP request on the home network for systems in the 10.1.1.x, 10.1.2.x, and 10.1.3.x subnets. This will cause the traffic for the home network to be routed automatically to the System i in the home network. This System i will in turn route the data to the correct remote System i. The remote System i will either process the data, or forward it to the correct system within the remote LAN.

The workstations in the remote LAN must have a default route that points to the remote System i in their network as the first hop gateway.

The workstations in the home LAN do not need any additional route entries. No new logical networks are created.

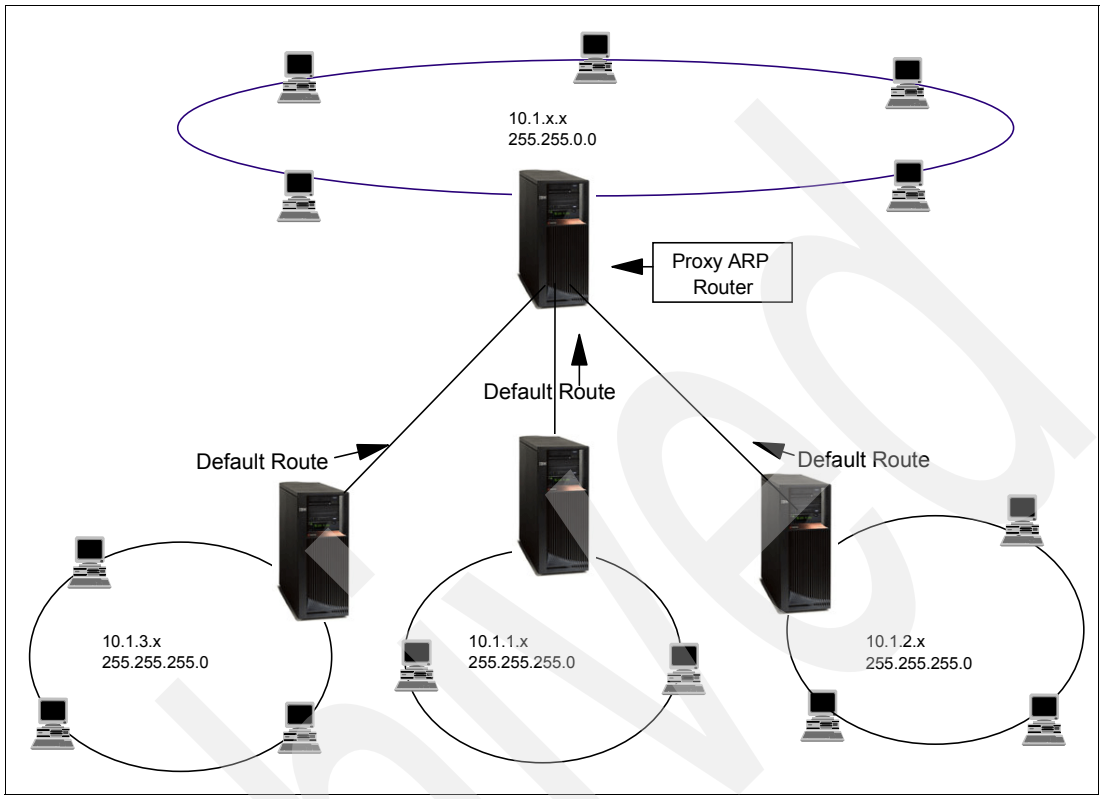


Figure 2-43 Transparent WAN subnetting: remote networks appear connected to the home network

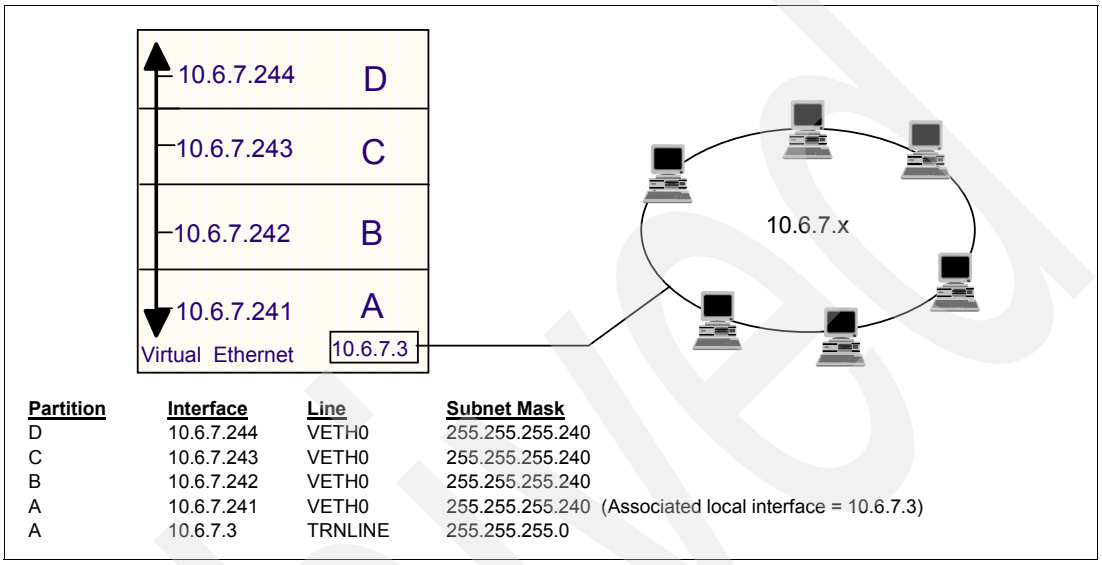
Transparent subnetting for external LAN and I/O-less partitions

The advent of Logical Partitioning (LPAR) provides yet another environment to apply the same transparent subnetting routing concepts.

With LPAR, a single System i is logically partitioned in multiple virtual machines. Each partition has its own address space, its own instance of TCP/IP, and may have its own dedicated I/O adapters. To TCP/IP, each partition appears as a distinct and separate System i.

Tip: i5/OS logical partitions also benefit from the new Virtual Ethernet LAN capability, which emulates a high-speed Ethernet. It is used to establish multiple high-speed TCP/IP connections between logical partitions without additional communication hardware and software. For more information about this other way to use transparent subnetting, see *LPAR Configuration and Management Working with IBM eServer iSeries Logical Partitions*, SG24-6251.

In Figure 2-44, only one LAN adapter is installed in the system. It is allocated to partition A. To enable the clients in the LAN to communicate with the other partitions defined on the system, we define a virtual Ethernet and configure it following the proxy ARP concept. A physical system enables you to configure up to 16 different Virtual Ethernet networks.The LAN has an network address of 10.6.7.x. We want to plan for additional partitions, so we need 12 IP addresses. To get 12 addresses we must use a subnet mask of 255.255.255.240. This gives us 10.6.7.241 to 10.6.7.254, for a total of 14 usable addresses.



Partition	Interface	Line	Subnet Mask
D	10.6.7.244	VETH0	255.255.255.240
C	10.6.7.243	VETH0	255.255.255.240
B	10.6.7.242	VETH0	255.255.255.240
A	10.6.7.241	VETH0	255.255.255.240 (Associated local interface = 10.6.7.3)
A	10.6.7.3	TRNLINE	255.255.255.0

Figure 2-44 Example transparent subnetting for external LAN and I/O-less partitions

We must ensure that these addresses are not already in use in the real LAN.

First, we create an Ethernet line description for each partition by using the Create Line Desc (Ethernet) (CRTLINETH) command. We then define a TCP/IP address from the range of usable addresses referred to above, on top of each Ethernet line description by using the Add TCP/IP Interface (ADDTCPIFC) command.

As long as the Virtual Ethernet subnet is a subnet of the LAN network address, then transparent subnetting will be enabled automatically and the interface 10.6.7.3 will proxy ARP for all of the interfaces defined in the partitions. This enables clients on the LAN to connect to the partitions.

If the LAN adapter 10.6.7.3 fails, connection from the external LAN to all partitions is lost. However, the interpartition communications across the Virtual Ethernet will continue to work.

You should also consider the amount of traffic expected to the other partitions when choosing this configuration because all traffic related to the secondary partitions will be directed to and routed through the external LAN adapter 10.6.7.3 interface.

For complete configuration details, refer to Chapter 10, “Interpartition communications,” in Redbooks publication *LPAR Configuration and Management Working with IBM eServer iSeries Logical Partitions*, SG24-6251.

We discuss the complete setup with four different partitions in Chapter 13, “Virtual Ethernet within an LPAR environment” on page 211.

Tip: The configuration demonstrated in Figure 2-44 is an example of how to provide maximum connectivity between the LAN hosts and the partitions. Alternatively, if maximum security or isolation is needed between the LAN and one or more partitions, different Virtual Ethernet line descriptions with different IP subnets should be considered.

2.3.5 Virtual Ethernet within LPAR environment

This capability was introduced in V5R1M0 of OS/400 and was known as Virtual LAN (local area network). Each Virtual LAN enables you to establish communication via TCP/IP between logical partitions. Each partition can define up to 16 virtual LAN ports. Partitions defined to use the same port can communicate through that link.

In OS/400 V5R2M0, this capability was renamed to Virtual Ethernet. Virtual Ethernet enables you to establish communication via TCP/IP between logical partitions. Each partition can define up to 16 virtual LANs. Partitions defined to use the same port can communicate through that link.

Virtual Ethernet can be used without any additional hardware or software and Linux interpartition connectivity. This emulates a high-speed Ethernet environment and is used to establish multiple high-speed TCP/IP connections between logical partitions on the same System i without any additional communications hardware and software. For more information about the Virtual Ethernet setup, see these Redbooks publications:

- *LPAR Configuration and Management Working with IBM eServer iSeries Logical Partitions*, SG24-6251
- *Implementing POWER Linux on IBM System i Platform*, SG24-6388

In this book we provide three examples of configuring a Virtual Ethernet on the System i:

- "Virtual Ethernet and proxy ARP configuration" on page 212: This scenario demonstrates the steps required to configure proxy ARP for the System i virtual Ethernet. The goal is to allow connections from the external LAN to secondary partitions on the System i that do not have a physical adapter on the LAN.
- "Virtual Ethernet and NAT scenario" on page 224: Here we use Network Address Translation (NAT) to route traffic between secondary partitions across the Virtual Ethernet and the external LAN to which the System i is connected.
- "Virtual Ethernet and routing scenario" on page 236: Here we use basic TCP/IP routing on the System i in order to route traffic from the external LAN toward secondary partitions across the Virtual Ethernet.

2.3.6 Connect to a TCP/IP application while in restricted state

Starting in V5R2M0 of OS/400, the TCP/IP protocol stack and interfaces can be started while the operating system is in restricted state. One major restriction is that only your own user-written sockets application can be running on the system in this state. That is, none of the applications in the TCP/IP protocol suite (such as Telnet, FTP, SMTP, DNS, and so on) can be started.

As an example, a network administrator can obtain status reports while you are running backup procedures. The operating system must be in restricted state to prevent users from changing any configuration. You can now remotely access status reports using a PDA device (or any TCP/IP networking device). The PDA uses a sockets-enabled application that requires an active TCP/IP interface available to communicate with the server.

To allow this communication, you must first start TCP/IP using special parameters. After you start TCP/IP, you must start the required TCP/IP interface to allow access to the system.

See 12.5, “Connect to a TCP/IP application while in restricted state” on page 209, for more information.

3

IPv6: the next generation of the Internet

Internet Protocol Version 6 (IPv6) is a revision of the IPv4 addressing scheme currently in use by the majority of the computing world today. IPv6 will eventually become the standard for IP addressing. There are many reasons for this:

- Increased address space
- Autoconfiguration
- Mobility
- Security
- Scalability

This chapter addresses the improvements IPv6 offers, the basics of the protocol, and how you can take advantage of the IPv6 enhancements.

3.1 Benefits of IPv6

Careful attention is being paid to IPv6 design. The protocol enhancements are intended to make IPv6 easier to use and solve known problems and limitations of IPv4. This section highlights the features that make IPv6 superior to IPv4. Keep in mind that not all of these features are available with the V5R4 release of i5/OS. For specific information about what you can do at V5R4, refer to 3.3, "IPv6 support in the V5R4 release of i5/OS" on page 67.

3.1.1 Increased address space

The IPv6 address space equals approximately 665,570,793,348,866,943,898,599 IPv6 addresses for every square meter of the Earth. And that includes oceans, mountains, and polar ice caps.

To overcome the IPv4 address shortage, IPv6 expands the IP address space from 32 bits to 128. The address is in the format:

```
xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx
```

Each x is a hexadecimal digit representing 4 bits. The increase allows virtually unlimited unique IP addresses (literally: 340,282,366,920,938,463,374,607,431,768,211,456 addresses).

In addition to this addressing capability, IPv6 provides new functions that simplify the tasks of configuring and managing the addresses in the network. IPv6 automates several of the network administrator's tasks. For example, the IPv6 autoconfiguration feature automatically configures interface and router addresses for you. In stateless autoconfiguration, IPv6 can create a new, unique IPv6 address for a device. This feature eliminates the need for a DHCP server to manage addresses.

For more detailed information about IPv6, including descriptions of the differences between IPv4 attributes and IPv6 attributes and configuration scenarios, visit the V5R4 Information Center at:

http://publib.boulder.ibm.com/infocenter/iseries/v5r4

The topic is located under **Networking** → **TCP/IP setup** → **Internet Protocol version 6**.

3.1.2 Autoconfiguration

IPv6 offers a large address space. But the consequences of a large address space are longer IP addresses and exponential growth in the number of devices that can be accommodated on the network. To alleviate the administration needs of this address space, automated configuration has been enabled using concepts that are still fresh in terms of Dynamic IPv4.

The IPv6 stack is self-learning: It is capable of discovering routes and information about routes automatically. Nodes can also discover other nodes on the network in this process, called *neighbor discovery*. Nodes that are added to the network can be configured in one of two ways: using DHCPv6, which is known as *stateful* autoconfiguration, or by *stateless* autoconfiguration, which is a new feature of IPv6 and relies on ICMPv6 Internet Control Message Protocol). DHCPv6 is currently not available on i5/OS.

Neighbor discovery

Neighbor discovery functions are used by IPv6 nodes (hosts or routers) to discover the presence of other IPv6 nodes, to determine the link-layer addresses of nodes, to find routers that are capable of forwarding IPv6 packets, and to maintain a cache of active IPv6

neighbors. IPv6 nodes use these five Internet Control Message Protocol V6 (ICMPv6) messages to communicate with other nodes:

Router solicitation	Hosts send these messages to request routers to generate router advertisements. When a host first becomes available on the network, it sends an initial router solicitation.
Router advertisement	Routers send these messages either periodically or in response to a router solicitation. The information provided by router advertisements is used by hosts to automatically create global interfaces and associated routes. Router advertisements also contain other configuration information used by a host, such as maximum transmission unit and hop limit.
Neighbor solicitation	Nodes send these messages to determine the link-layer address of a neighbor or to verify that a neighbor is still reachable.
Neighbor advertisement	Nodes send these messages in response to a neighbor solicitation or as an unsolicited message to announce an address change.
Redirect	Routers use these messages to inform hosts of a better first hop for a destination.

Tip: Routes are automatically created when i5/OS discovers a router on the LAN, when a interface starts, when a local router tells i5/OS about new routes, and when a local router tells i5/OS about new on-link prefixes. Multicast routes are completely automatic and are created as needed by the TCP/IP stack.

See RFC 2461 for more information about neighbor discovery and router discovery. The RFC is available through RFC Editor:

`http://www.rfc-editor.org/rfcsearch.html`

Stateless autoconfiguration

IPv6 was designed with the capability to automatically assign an address to an interface at initialization time, with the intention that a network can become operational with minimal to no action on the part of the TCP/IP administrator. IPv6 nodes generally use autoconfiguration to obtain their IPv6 address.

Stateless address autoconfiguration is the process that IPv6 nodes (hosts or routers) use to configure IPv6 addresses for interfaces automatically. It consists of the following steps:

1. During system startup, the node builds IPv6 addresses by combining an address prefix with either the MAC address of the node or a user-specified interface identifier.
2. The node creates a tentative link-local unicast address. This is done by combining the well-known link-local prefix (fe80::/10) with the interface token.
3. The node performs *duplicate address detection* to verify the uniqueness of the address before assigning it to an interface. The node sends out a neighbor solicitation query to the new address and waits for a response. If the node receives a response in the form of a neighbor advertisement, the address is already in use. If a node determines that its tentative IPv6 address is not unique, then autoconfiguration stops and manual configuration of the interface is required.
4. If the node does not receive a response, then the address is assumed to be unique, and the node assigns the link-level address to its interface.

3.1.3 Mobility

As mobile devices become more popular, they not only consume IP addresses, they also place demands on routing and other record changes because the current methods rely on IP addresses to stay static and remain in the same routing path. There are methods that work around this problem, but IPv6 has been defined to provide better mobility support.

Certain enhancements in the IPv6 protocol lend themselves particularly to the mobile environment. For example, unlike Mobile IPv4, there is no requirement for routers to act as *foreign agents* on behalf of the mobile node. Autoconfiguration enables the node to operate away from home without any special support from a local router. Also, most packets sent to a mobile node while it is away from its home location can be tunneled by using IPv6 routing (extension) headers, rather than a complete encapsulation (as used in Mobile IPv4), which reduces the overhead of delivering packets to mobile nodes.

3.1.4 Security

The IPv6 protocol includes the same features as IPv4 in regard to IPSec requirements. It offers built-in authentication and encryption. The IPv6 support in the V5R4 release of i5/OS does not support IPSec. SSL encrypted connections over IPv6 are supported.

RFC 2411 provides an overview of the IPSec documentation, and RFC 4301 documents the security architecture.

3.1.5 Scalability

Globally unique and hierarchical addressing, based on prefixes rather than address classes, keeps routing tables small and backbone routing efficient.

CISCO offers IPv6-capable routers. For information about Cisco support of IPv6, visit:

`http://www.cisco.com/en/US/products/ps6553/products_ios_technology_home.html`

3.1.6 Quality-of-Service capabilities

The header structure for IPv6 packets allows applications to specify a certain priority for the traffic they generate. This enables packets to be labeled as belonging to traffic flows to allow Differentiated Services. QoS support for IPv6 is not enabled in V5R4.

3.2 IPv6 addressing

IPv6 is an evolutionary step from IPv4. Most of the Internet currently uses IPv4, and this protocol has been reliable and resilient for more than 20 years. However, IPv4 has limitations that are causing problems as the Internet expands.

IPv6 has 4 billion times 4 billion times 4 billion (2^^96) more addresses than IPv4.

The basic reason for IPv6 enhancement is a growing demand for IPv4 addresses as more devices connect to the Internet, especially mobile devices. Other potential markets could conceivably require many addresses, and the IP Next Generation developers kept this potential in mind when defining the new protocol.

For a detailed comparison of IPv4 and IPv6 concepts and services, including specific information about the IPv6 support in the V5R4 release of i5/OS, refer to Appendix B, “IPv6 reference information” on page 715.

3.2.1 IPv6 address format

The IPv6 address size is 128 bits. The preferred IPv6 address representation is `xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx` where each `x` is a hexadecimal digit representing four bits.

IPv6 addresses range from `0000:0000:0000:0000:0000:0000:0000:0000` to `ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff`.

In addition to this preferred format, IPv6 addresses may be specified in two other shortened formats:

Omit leading zeros Specify IPv6 addresses by omitting leading zeros. For example, IPv6 address 1050:0000:0000:0000:0005:0600:300c:326b may be written as 1050:0:0:0:5:600:300c:326b.

Double colon Specify IPv6 addresses by using double colons (`::`) in place of a series of zeros. For example, IPv6 address ff06:0:0:0:0:0:c3 may be written as ff06::c3. Double colons may be used only once in an IP address.

An alternative format for IPv6 addresses combines the colon and dotted decimal notation, so the IPv4 address may be embedded in the IPv6 address. Hexadecimal values are specified for the left-most 96 bits, and decimal values are specified for the right-most 32 bits indicating the embedded IPv4 address. This format allows compatibility between IPv6 nodes and IPv4 nodes when you are working in a mixed network environment.

```
IPv4-mapped IPv6 address (::ffff:<IPv4_address>)
```

This type of address is used to represent IPv4 nodes as IPv6 addresses. It enables IPv6 applications to communicate directly with IPv4 applications. For example, for an IPv4 address of 192.1.56.10, the IPv4-mapped IPv6 address would be 0:0:0:0:0:ffff:192.1.56.10 (or ::ffff:192.1.56.10 in the shortened format). See Table 24-5 on page 716 for more information.

3.2.2 IPv6 address types

IPv6 addresses are categorized into three basic types: Unicast, Multicast, and Anycast.

Unicast address

The unicast address specifies a single interface. A packet sent to a unicast address destination travels from one host to the destination host. Two types of unicast addresses include:

Link-local address Link-local addresses are designed for use on a single local link (local network). Link-local addresses are automatically configured. The prefix used for a link-local address is **fe80::/10**. Routers do not forward packets with a destination or source address containing a link-local address.

Global address Global addresses are designed for use on any network.

Note: When a link-local address is made using the MAC address, it is likely to be globally unique (due to the uniqueness of MACs), but that does not matter because no router will allow the link-local traffic to leave the link. Architecturally, link-local addresses are only *guaranteed* unique with their respective scopes.

As far as renumbering, an ISP will typically assign global (unicast) prefixes, which will be propagated around intranets via internal routers. In theory, renumbering will occur when you change ISPs and it can be largely automatic.

Two special types of unicast addresses include:

Unspecified address (::) The unspecified address is 0:0:0:0:0:0:0:0 or may be abbreviated with two colons (::). The unspecified address indicates the absence of an address, and it may never be assigned to a host. It may be used by an IPv6 host that does not yet have an address assigned to it. For example, when the host sends a packet to discover an address from another node, the host uses the unspecified address as its source address.

Loopback address (::1) The Loopback address is 0:0:0:0:0:0:0:1 or may be abbreviated as ::1. This address is used by a node to send a packet to itself.

Multicast address

The multicast address specifies a set of interfaces, possibly at multiple locations. The prefix used for a multicast address is 0xFF. If a packet is sent to a multicast address, one copy of the packet is delivered to all interfaces corresponding to that address. Multicast addresses supersede the use of IPv4 broadcast addresses.

Certain special-purpose multicast addresses are predefined. The following list describes some of the common reserved multicast addresses. A more complete listing may be found in RFC 2375, *IPv6 Multicast Address Assignments*:

FF01::1 All interfaces node-local. Defines all interfaces on the host itself.

FF01::2 All routers node-local. Defines all routers local to the host itself.

FF02::1 All nodes link-local. Defines all systems on the local network.

FF02::2 All routers link-local. Defines all routers on the same link as the host.

FF02::B Mobile agents link-local.

FF02::1:2 All DHCP agents link-local.

FF02::1:FFxx:xxxx Solicited node address in which xx:xxxx is taken from the last 24 bits of a node's unicast address. For example, the node with the IPv6 address of 4025::01:800:100F:7B5B belongs to the multicast group FF02::1:FF 0F:7B5B. The solicited node address is used by ICMPv6 for neighbor discovery and to detect duplicate addresses.

Anycast address

The anycast address specifies a set of interfaces, possibly at different locations, that all share a single address. A packet sent to an anycast address goes only to the nearest member of the group. The V5R4 release of i5/OS does not support anycast addressing.

3.3 IPv6 support in the V5R4 release of i5/OS

Because the IPv4 standard is so widely used, the transition to IPv6 will take many years. During this transition phase, both the IPv4 and IPv6 protocols will coexist. The IPv6 protocol has been designed so that it can inter-operate with existing IPv4 applications and hosts. This is accomplished using the IPv4-mapped IPv6 addresses that were mentioned earlier and using other transition techniques.

In the V5R4 release, the IPv6 support that is provided in i5/OS is considered stable and is *production ready*. Unfortunately, many of the i5/OS base applications have not been updated yet to use IPv6. These application updates will be occurring in future releases of i5/OS. Tools for working with IPv6 have been supplied in i5/OS and developers should start updating their applications to support IPv6 today. For additional information about the relationship between IPv4 and IPv6 concepts, refer to “Comparison: IPv4 to IPv6” on page 716.

3.3.1 IPv6 in iSeries Navigator

You must use iSeries Navigator to configure IPv6. To view the configuration, expand **Networking** → **TCP/IP configuration** → **IPv6**, as shown in Figure 3-1.

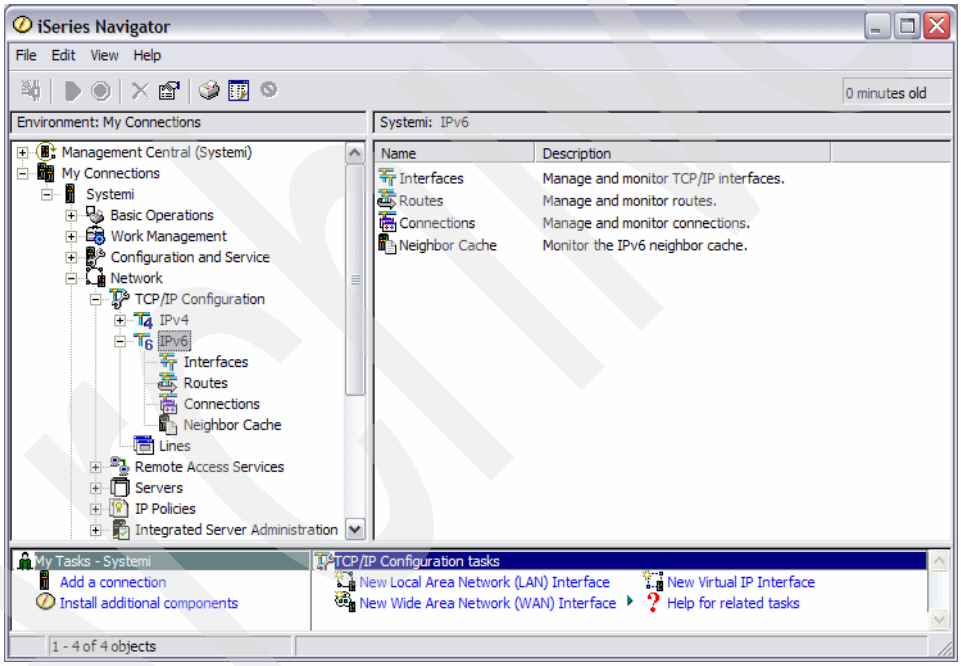


Figure 3-1 IPv6 in iSeries Navigator

3.3.2 Sockets enhancements

All new application development should be designed to support both IPv4 and IPv6 addresses.

You can write applications using sockets APIs to concurrently and transparently handle IPv4 and IPv6 addresses.

The AF_INET6 address family was added in the V5R2 release. This address family enables you to write TCP/IP applications that use IPv6 addresses. The AF_INET6 address family also allows applications to use IPv4-mapped IPv6 addresses to communicate with IPv4 applications. Existing socket APIs were updated, new APIs were added, and new structures were defined to support the AF_INET6 address family and IPv6.

New functions were added in support of the AF_INET6 family. The getaddrinfo() function translates the a host name and returns a set of socket addresses and associated information to be used in creating a socket with which to address the specified service. This API can be used to obtain (at application choice) IPv6 only, IPv4 only, or IPv6 and IPv4 addresses. This replaces the gethostbyname() function. The getnameinfo() function translates a socket address to a node name and service location, and replaces gethostbyaddr().

We recommend that you use the enhanced sockets support for all new application development. These changes enable you to support both address formats, whether or not you are currently using IPv6. The IPv6 support in the V5R4 release of i5/OS provides all of the capabilities that you need to change and test your applications using the enhanced sockets.

For more information about the sockets enhancements for IPv6, refer to the V5R4 Information Center:

http://publib.boulder.ibm.com/infocenter/iseries/v5r4

The topic is located under **Programming** → **Communications** → **Socket Programming** → **Socket characteristics** → **Socket address family** → **AF_INET6 address family**.

You can also see an example in the Socket Programming topic by clicking **Socket scenario: Create an application to accept IPv4 and IPv6 clients**.

3.3.3 i5/OS DNS support for IPv6

IPv6 addressing may solve the problem of address space, but it makes remembering addresses even more difficult. DNS will be more important than ever to help people work without needing to remember long strings of numbers. BIND (Berkeley Internet Name Domain) Version 8.2.5 is available in i5/OS. It provides AAAA (pronounced *quad A*) records for IPv6 addresses, in the format as:

```
owner class ttl AAAA IP_v6_address
```

Example:

```
host1.itsoroch.ibm.com. IN AAAA 1234::206:29ff:feec:c4b
```

The reverse lookup is the hexadecimal digits written backward, separated by dots. The IPv6 address reverse domain is ip6.arpa, such as:

```
b.4.c.0.c.e.e.f.f.f.9.2.6.0.2.0.0.0.0.0.0.0.4.3.2.1.ip6.arpa. IN PTR host1.itsoroch.ibm.com
```

Although BIND 8.2.5 supports AAAA records, it does not support IPv6 communication. Queries must be sent to DNS servers using IPv4.

In addition to DNS support of AAAA records, the Sockets Network Functions API set has been updated to enable resolution of IPv6 address and associated names using the resolver APIs or the new protocol-independent APIs (getaddrinfo and getnameinfo). See 3.3.2, “Sockets enhancements” on page 67. Support has also been added to allow for dynamically updating the DNS IPv6 associated records using the appropriate resolver APIs.

3.3.4 Troubleshooting and test tools

PING, Trace Route, Communications Trace, and Netstat have all been updated to support IPv6 to enable you to test your configuration and applications.

PING

PING supports both IPv4 and IPv6 addresses. If you have AAAA records defined, you can PING an IPv6 host by entering the name and selecting IPv6 as the protocol.

To access PING in iSeries Navigator expand **Networking** → **TCP/IP Configuration**, right-click **TCP/IP Configuration**, and choose **Utilities** → **Ping**, as shown in Figure 3-2.

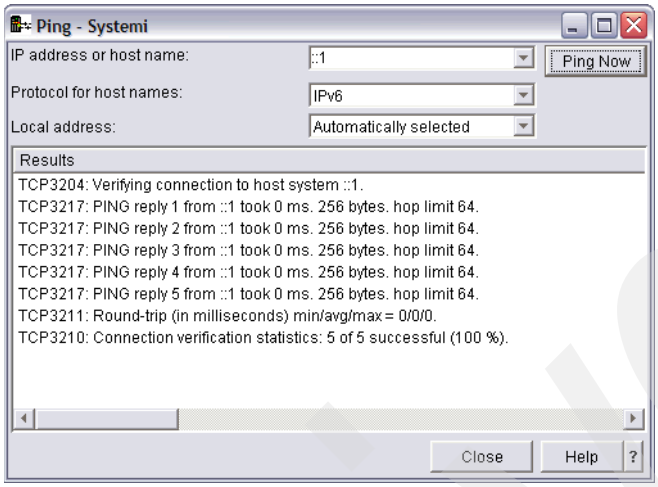


Figure 3-2 IPv6 PING of the loopback address ::1

Trace Route

Trace Route supports both IPv4 and IPv6 addresses. If you have AAAA records defined, you can trace the route to an IPv6 host by entering the name and selecting IPv6 as the protocol.

To access Trace Route in iSeries Navigator expand **Networking** → **TCP/IP Configuration**, right-click **TCP/IP Configuration**, and choose **Utilities** → **Trace route**, as shown in Figure 3-3.

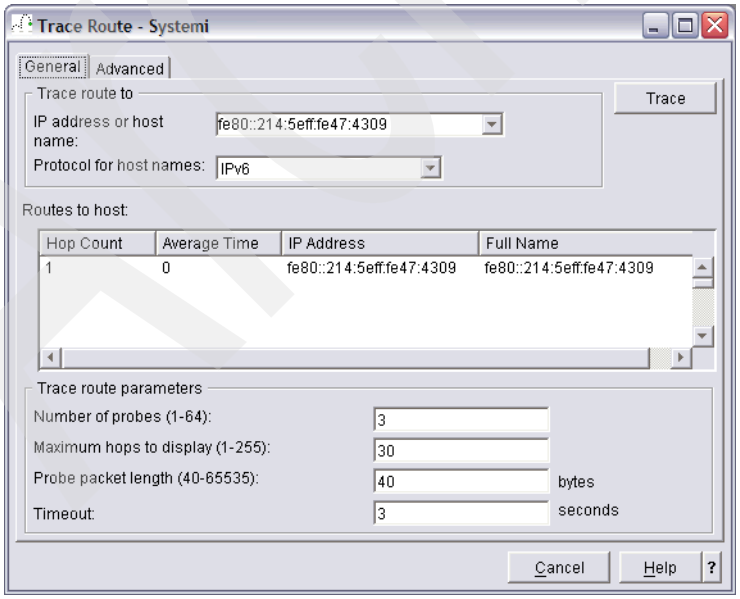


Figure 3-3 IPv6 Trace Route

Netstat

In addition to the views and actions available in iSeries Navigator, you can use Netstat on the green screen to view IPv6 information. Type NETSTAT at the command line. The IPv6 options (items 4-6 in Figure 3-4) will not be displayed if the IPv6 protocol was not activated when TCP/IP was started on your system. By default in the V5R4 release of i5/OS, the IPv6 protocol and the IPv6 loopback address, ::1, is automatically configured and enabled when TCP/IP is started.

```
Work with TCP/IP Network Status
                                                          System: SYSTEMI
Select one of the following:

     1. Work with TCP/IP interface status
     2. Display TCP/IP route information
     3. Work with TCP/IP connection status
     4. Work with IPv6 interface status
     5. Display IPv6 route information
     6. Work with IPv6 connection status


Selection or command
===>

F3=Exit   F4=Prompt   F9=Retrieve   F12=Cancel
```

Figure 3-4 IPv6: Work with TCP/IP Network Status (NETSTAT)

Communications Trace

The Communications Trace function has been updated to log IPv6 activity. You will need to use Communications Trace CL commands — not Start Service Tools (SST) — for traces. Step-by-step instructions are available in “Using IPv6 Communications Trace” on page 724.

3.4 Configuring IPv6

In order to configure IPv6 support in i5/OS, you must first have IPv4 configured, have TCP/IP running, and be able to access your system using the iSeries Navigator. IPv6 cannot be configured from the command line. You must use iSeries Navigator.

Three different types of IPv6 interfaces can be configured in the V5R4 release:

- IPv6 Loopback Interface, ::1
- Manually configured IPv6 interface
- IPv6 stateless interfaces

3.4.1 IPv6 Loopback Interface, ::1

The IPv6 loopback interface is just like the IPv4 loopback interface. It does not require an Ethernet adapter or even another node. The loopback interface allows you to PING your own system and it is useful for application development and testing over IPv6. Starting with the V5R4 release, the IPv6 loopback interface, ::1, is automatically configured and enabled when TCP/IP is started.

The IPv6 loopback interface can be viewed by expanding **IPv6** and selecting **Interfaces**, as shown in Figure 3-5.

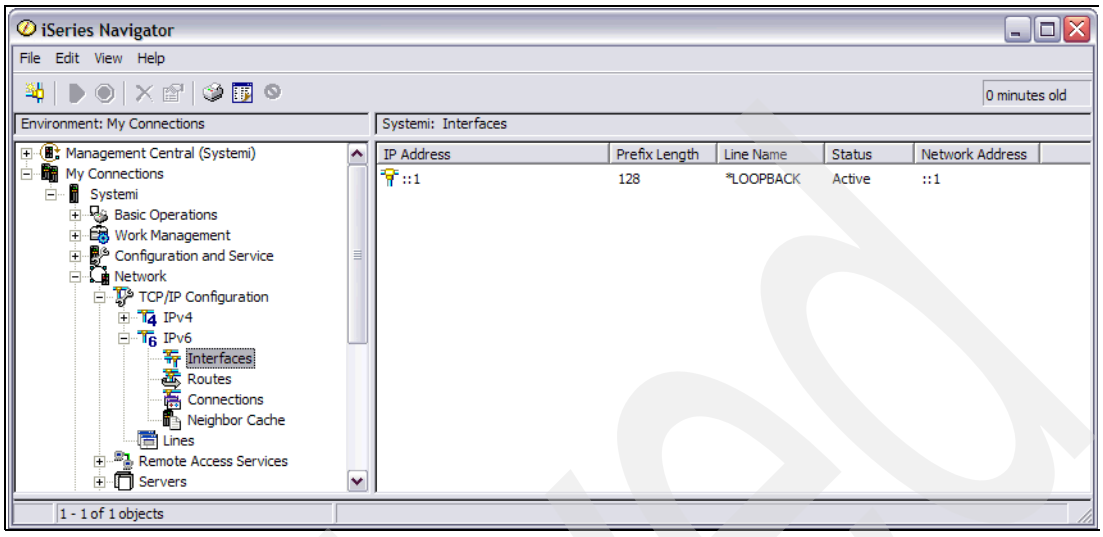


Figure 3-5 IPv6 Configuration: Interfaces - loopback

3.4.2 Manually configured IPv6 interface

With the V5R4 release of i5/OS, IPv6 can be configured on any Ethernet or Virtual Ethernet line. The adapter type restrictions for IPv6 in the V5R2 and V5R3 releases no longer apply. In addition, IPv6 can be configured on multiple Ethernet adapters, and those same adapters can be shared with IPv4 and PPPoE traffic. IPv6 support in V5R4 no longer requires a dedicated Ethernet line.

1. Right-click **IPv6** → **Interface** and choose **New Interface**, as shown in Figure 3-6.

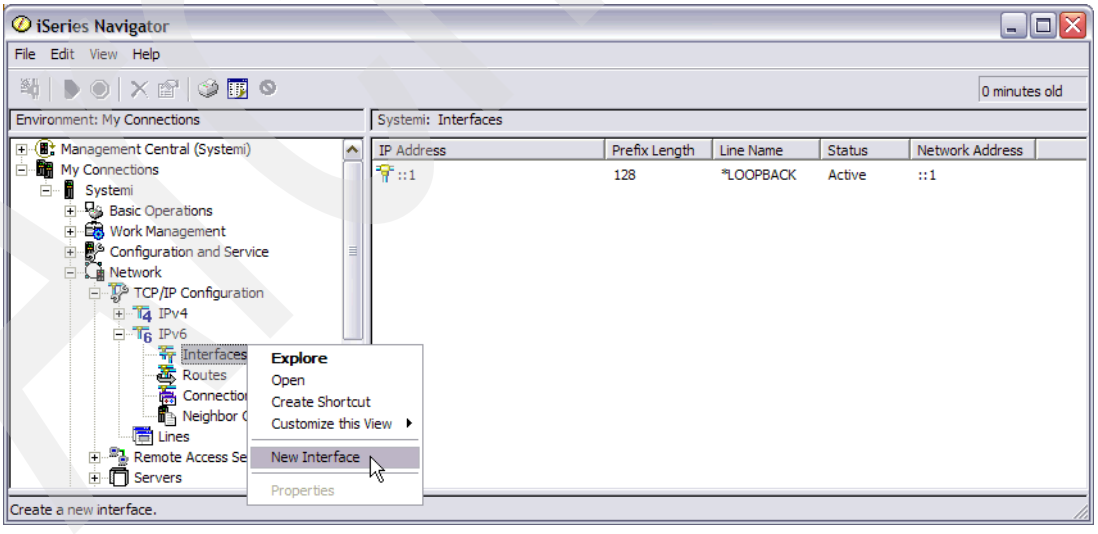


Figure 3-6 IPv6 Configuration: New Interface

2. The new IPv6 Interface wizard is displayed, as shown in Figure 3-7. Click **Next** to continue.

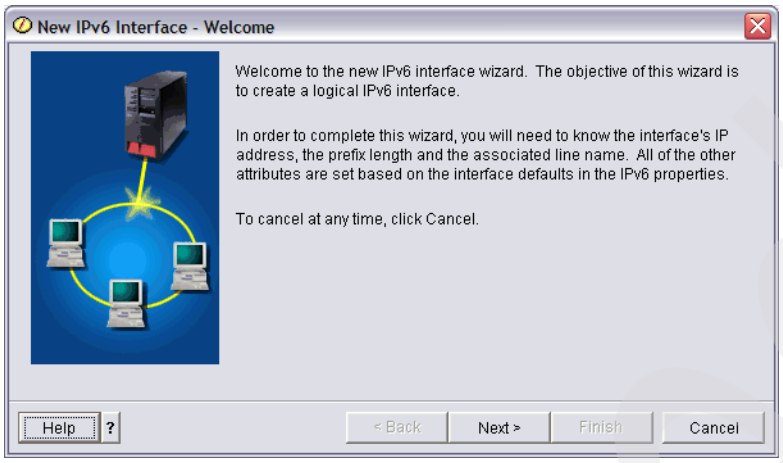


Figure 3-7 IPv6 Configuration: New IPv6 Interface wizard

3. Fill in the attributes for your new IPv6 interface. Fill in the IPv6 address that you would like created. Specify the prefix length. A prefix length of 64 is commonly used for IPv6 addresses. Specify an alias name and description for this new interface if desired. Use the drop-down menu to select which line should be used for the new interface. Only existing Ethernet lines will be displayed. We also recommend that you enabled “Use duplicate address detection” and that you specify the maximum transmits to be 1. Finally, you may also want to enable “Start with TCP/IP is started” so that this interface will be started the next time that you start TCP/IP. Figure 3-8 shows the manual configuration of a new IPv6 interface.

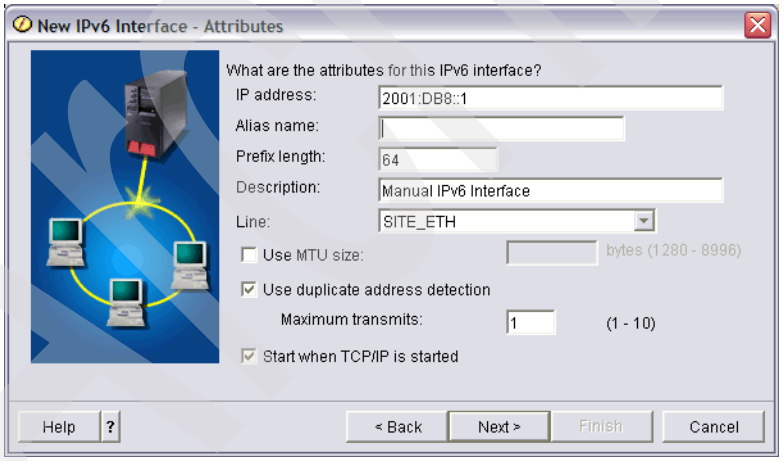


Figure 3-8 IPv6 configuration: New IPv6 Interface attributes

4. The summary page verifies your selections. Click **Finish** to create the new IPv6 interface (Figure 3-9).

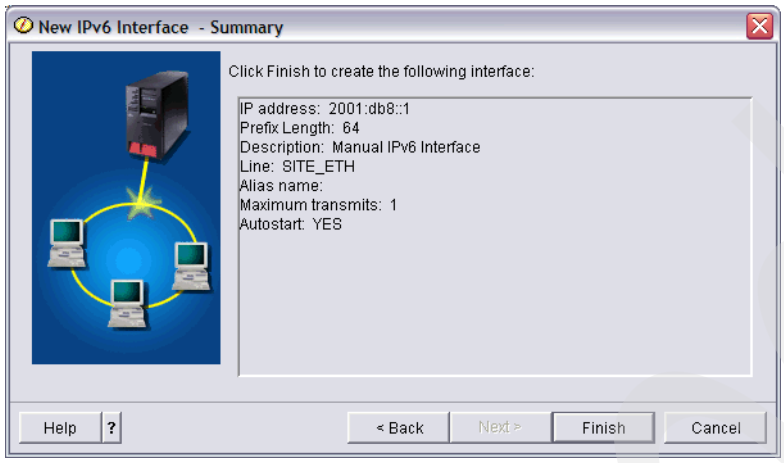


Figure 3-9 IPv6 configuration: New IPv6 Interface Summary

5. Unlike the IPv4 new interface wizard, there is no way to configure the IPv6 interface so that it is automatically started after it has been created. In order to start your new IPv6 interface, you need to go to the **IPv6** → **Interface** panel, right-click the new interface, and choose **Start**.

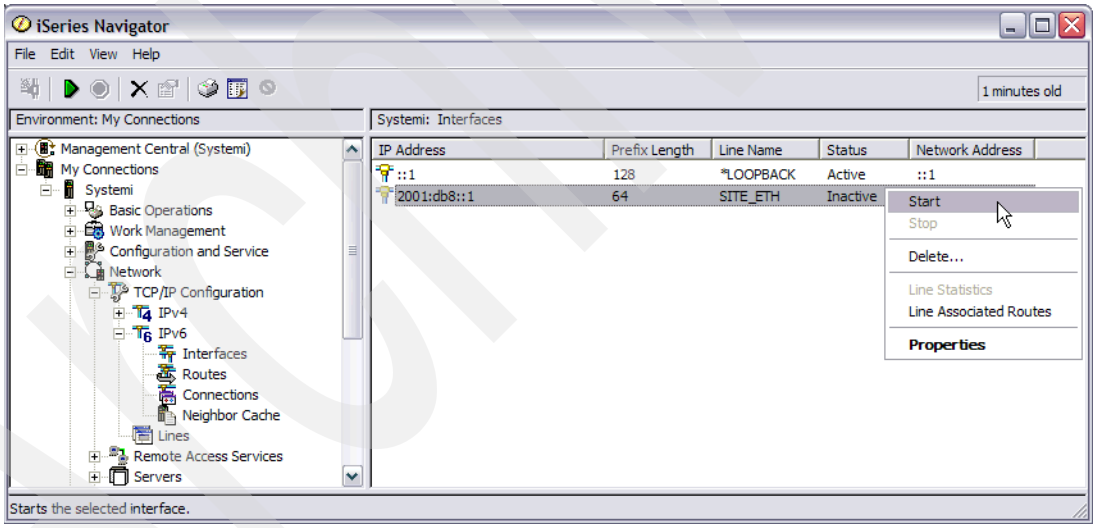


Figure 3-10 IPv6 configuration: Start new IPv6 interface

6. Figure 3-11 shows the new IPv6 interface that has been created.

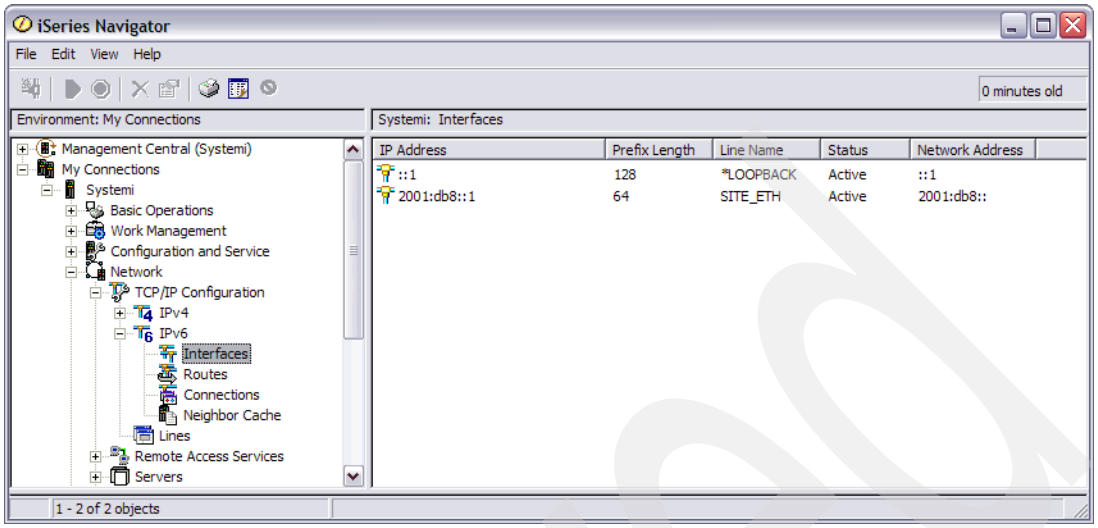


Figure 3-11 IPv6 Configuration: Interfaces

The following interfaces are shown:

::1 IPv6 loopback address.

2001:db8::1 Manual IPv6 interface that was created.

3.4.3 IPv6 stateless interfaces

The stateless address autoconfiguration functionality allows IPv6 interfaces to be automatically generated without requiring the use of a DHCP server. The IPv6 address is calculated using the network prefix from IPv6 routers on the network and the MAC address of the Ethernet adapter or a user-specified interface identifier.

1. The first step to configure a stateless IPv6 interface is to display the list of lines that are configured on your system. Select **Lines** under the **Networking** → **TCP/IP Configuration** (Figure 3-12).

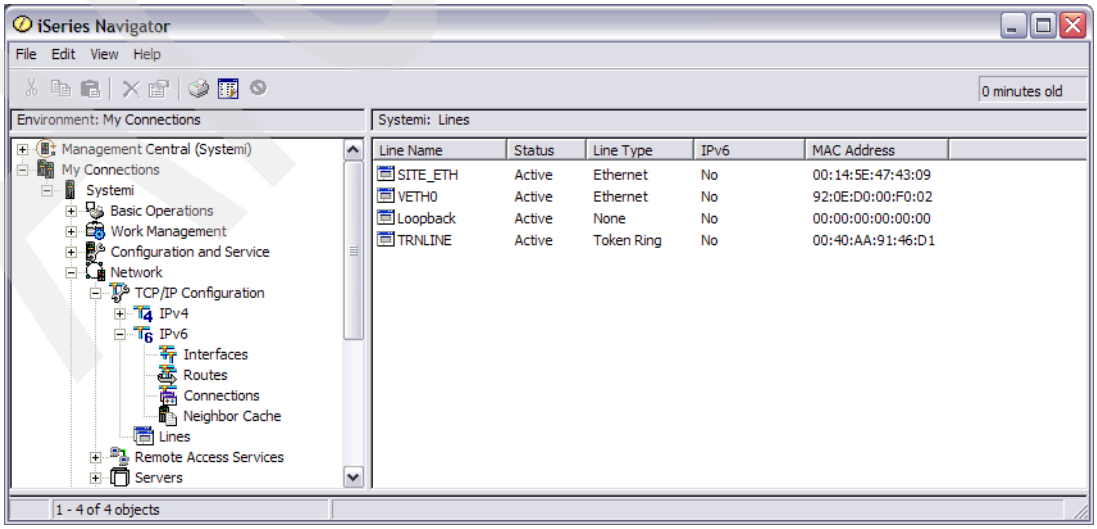


Figure 3-12 IPv6 Configuration: Lines

2. Right-click an Ethernet line that you would like to configure and select **IPv6 Stateless Address Autoconfig** → **Configure**, as shown in Figure 3-13.

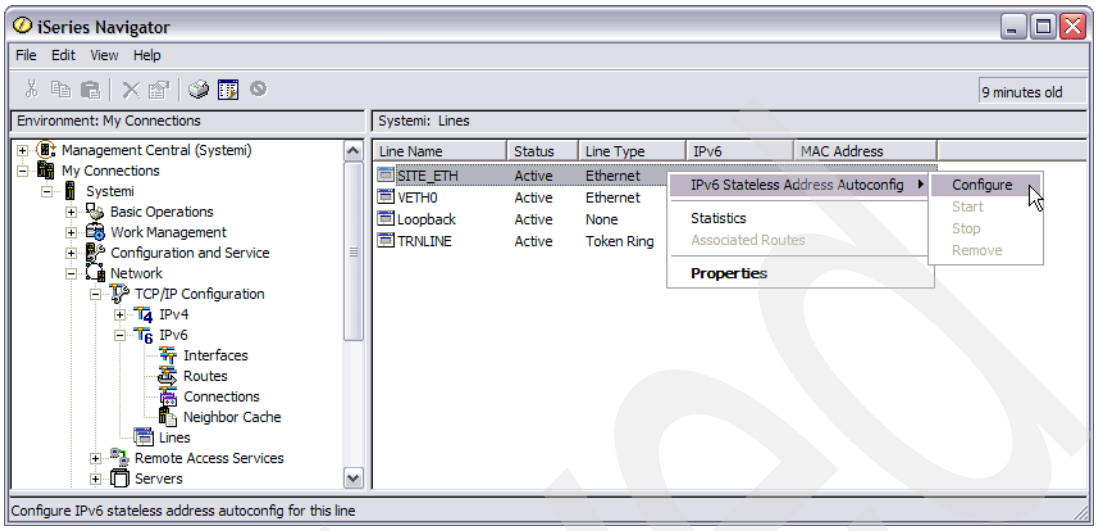


Figure 3-13 IPv6 Configuration: Configure IPv6 Stateless

3. Fill in the attributes for your new IPv6 stateless interface. Use the drop-down menu to select which line should be used. By default the line that you had highlighted in the previous step will be displayed, but you can select a different Ethernet line if desired. We recommend that you enable “Use duplicate address detection” and that you specify the maximum transmits to be 1. By default the IPv6 address will be constructed using the MAC address, but you can specify your own interface identifier (last 64 bits of the IPv6 address) if desired. You may also want to enable “Start with TCP/IP is started” so that this interface will be started the next time that you start TCP/IP.

   Figure 3-14 shows the configuration of an IPv6 stateless interface. This configuration panel can also be invoked by right-clicking on **Lines** and choosing **Configure IPv6 Stateless Address Autoconfig**.

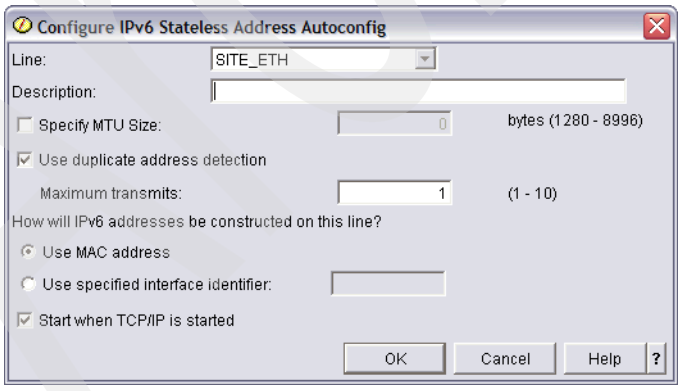


Figure 3-14 IPv6 Configuration: Configure IPv6 stateless interface

4. The IPv6 column has been updated to indicate that IPv6 stateless has been configured. The status has changed from No to Yes (Inactive). In order to activate the IPv6 stateless interface that was just configured, you need to right-click the line and select **IPv6 Stateless Address Autoconfig** → **Start**, as shown in Figure 3-15.

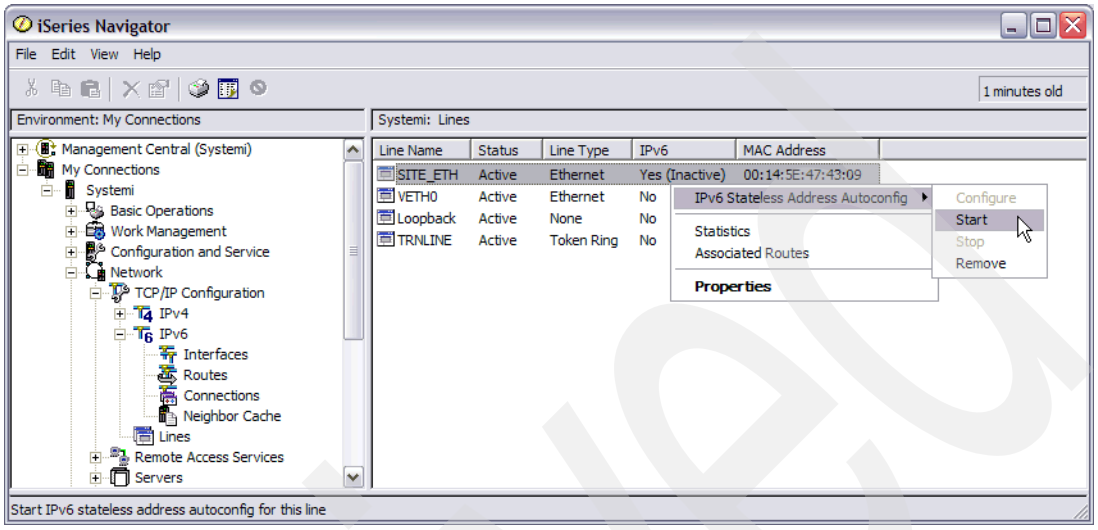


Figure 3-15 IPv6 Configuration: Start IPv6 stateless interface

5. As shown in Figure 3-16, the IPv6 status for the line is now Yes (Active).

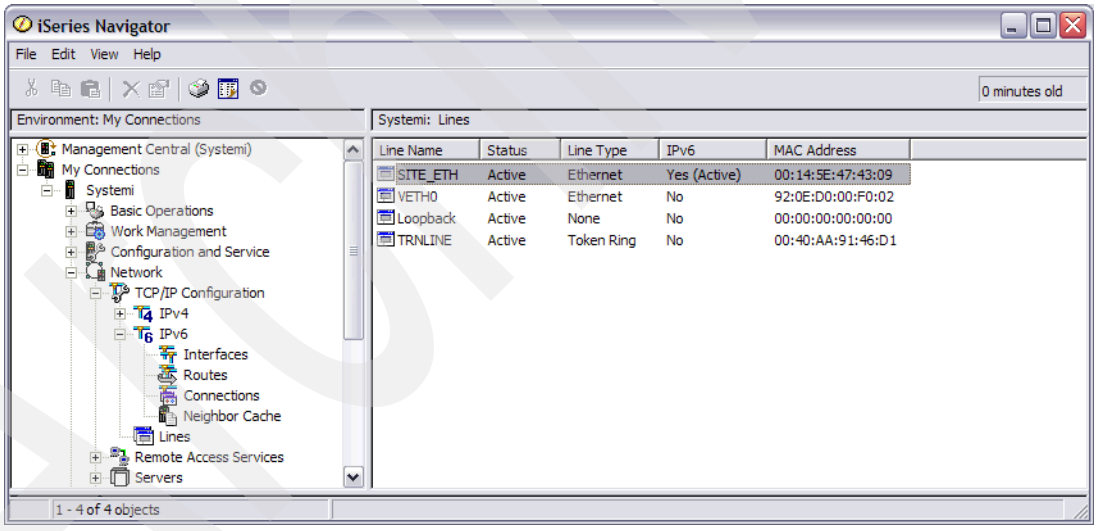


Figure 3-16 IPv6 Configuration: Active IPv6 stateless interface

6. The IPv6 stateless interface has been configured. Figure 3-17 shows the IPv6 interfaces that were automatically created.

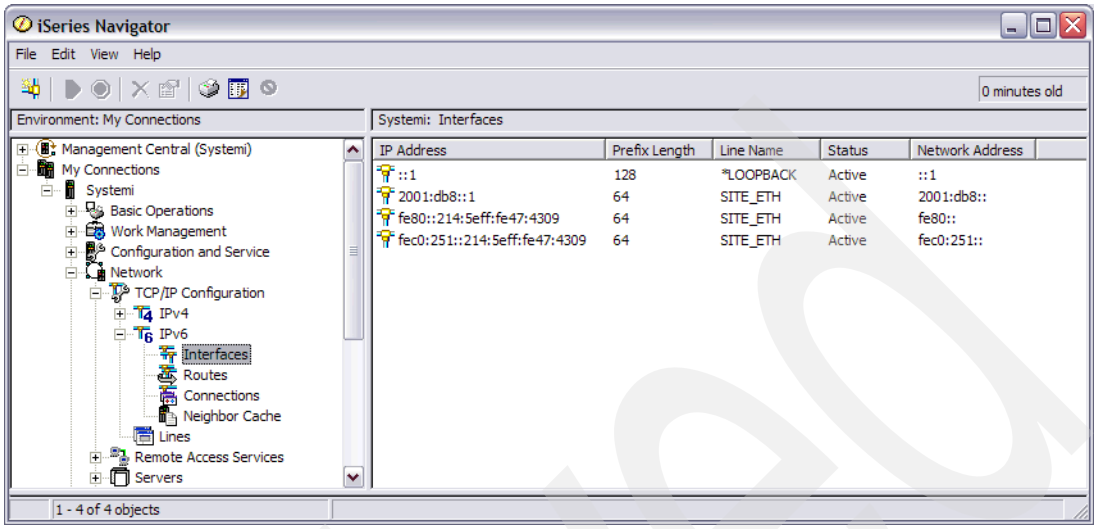


Figure 3-17 IPv6 Configuration: IPv6 stateless interfaces

The first two interfaces are from earlier:

::1 IPv6 loopback address

2001:db8::1 Manual IPv6 interface that was created earlier

The next two interfaces were created as a result of enabling IPv6 stateless address autoconfiguration on the SITE_ETH line:

- fe80::214:5efff:fe47:4309

  A link-local IPv6 stateless interface. fe80::/10 is the prefix for all link-local addresses. The MAC address was used for the interface identifier (last 64 bits of the IPv6 address).

- fec0:251::214:5efff:fe47:4309

  A globally routable IPv6 stateless interface. The fec0:251::/64 is the network prefix that was obtained from an IPv6 router on the Ethernet network that line SITE_ETH is attached to. The MAC address was used for the interface identifier (last 64 bits of the IPv6 address).

If there were additional IPv6 routers on the Ethernet network that SITE_ETH is attached to and those routers were configured with a different network prefix (other than fec0:251::/64), additional global IPv6 stateless addresses using those network prefixes would also have been appeared in this list.

3.5 Additional IPv6 information

The information presented here has given an overview of IPv6 and the function provided in V5R4. For more information about the IPv6 support on i5/OS, you can refer to the Information Center at:

`http://publib.boulder.ibm.com/infocenter/iseries/v5r4`

Full documentation on the IPv6 support in V5R4 is located under **Networking** → **TCP/IP setup** → **Internet Protocol version 6**.

Documentation of the AF_INET6 family is located under **Programming** → **Communications** → **Socket Programming** → **Socket characteristics** → **Socket address family** → **AF_INET6 address family**.

You can also see an example in the Socket Programming topic by clicking **Socket scenario: Create an application to accept IPv4 and IPv6 clients**.

If every person on earth had their own IPv6 network, each would have 18,000,000, 000,000,000, 000 addresses.

The *TCP/IP Tutorial and Technical Overview*, GG24-3376, contains a chapter on IPv6 that provides a more detailed explanation of the protocol, including header formats and services that are not yet available on i5/OS, such as DHCPv6.

The following RFCs (available by searching at http://www.rfc-editor.org/rfcsearch.html) contain detailed information about IPv6. The following list, while lengthy, is only a subset of the IPv6 related RFCs.

RFC 1752 The Recommendation for the IP Next Generation Protocol

RFC 1886 DNS Extensions to Support IP Version 6

RFC 1981 Path MTU Discovery for IP version 6

RFC 2185 Routing Aspects of IPv6 Transition

RFC 2375 IPv6 Multicast Address Assignments

RFC 2403 The Use of HMAC-MD5-96 within ESP and AH

RFC 2404 The Use of HMAC-SHA-1-96 within ESP and AH

RFC 2410 The NULL Encryption Algorithm and Its Use With IPsec

RFC 2411 IP Security Document Roadmap

RFC 2451 The ESP CBC-Mode Cipher Algorithms

RFC 2460 Internet Protocol, Version 6 (IPv6) Specification

RFC 2461 Neighbor Discovery for IP Version 6 (IPv6)

RFC 2462 IPv6 Stateless Address Autoconfiguration

RFC 2464 Transmission of IPv6 Packets over Ethernet Networks

RFC 2710 Multicast Listener Discovery (MLD) for IPv6

RFC 2711 IPv6 Router Alert Option

RFC 3041 Privacy Extensions for Stateless Address Autoconfiguration in IPv6

RFC 3152 Delegation of IP6.ARPA

RFC 3363 Representing IPv6 Addresses in the Domain Name System (DNS)

RFC 3484 Default Address Selection for Internet Protocol version 6 (IPv6)

RFC 3493 Basic Socket Interface Extensions for IPv6

RFC 3542 Advanced Socket Application Program Interface (API) for IPv6

RFC 3587 IPv6 Global Unicast Address Format

RFC 3590 Source Address Selection for the Multicast Listener Discovery (MLD) Protocol

RFC 3596 DNS Extensions to Support IP Version 6

RFC 3602 The AES-CBC Cipher Algorithm and Its Use with IPsec

RFC 3775 Mobility Support in IPv6

RFC 3810 Multicast Listener Discovery Version 2 (MLDv2) for IPv6

RFC 4007 IPv6 Scoped Address Architecture

RFC 4191 Default Router Preferences and More-Specific Routes

RFC 4193 Unique Local IPv6 Unicast Addresses

RFC 4213 Basic Transition Mechanisms for IPv6 Hosts and Routers

RFC 4291 IP Version 6 Addressing Architecture

RFC 4294 IPv6 Node Requirements

RFC 4301 Security Architecture for the Internet Protocol

RFC 4302 IP Authentication Header (AH)

RFC 4303 IP Encapsulating Security Payload (ESP)

RFC 4305 Cryptographic Algorithm Implementation Requirements for ESP and AH

RFC 4306 Internet Key Exchange (IKEv2) Protocol

RFC 4311 IPv6 Host-to-Router Load Sharing

RFC 4429 Optimistic Duplicate Address Detection (DAD) for IPv6

RFC 4443 Internet Control Message Protocol (ICMPv6) for the IPv6 Specification

4

Multilink Protocol

Multilink Protocol (MP) enables multiple PPP links to be grouped together to form a single virtual link or bundle. The benefits of MP include:

- Reducing the latency of data sent between systems by increasing the total effective bandwidth.
- Increased reliability through the use of multiple lines. (If a line fails, the link is maintained as long as one line in the MP bundle remains operational.)
- The ability to dynamically add and remove lines from a bundle, enabling bandwidth to be supplied as needed and making more efficient use of the bandwidth available.

We also have two scenarios that demonstrate the two most important features of MP:

- 14.1, “Multilink: dynamic bandwidth allocation” on page 242
- 14.2, “Multilink: Fault tolerance” on page 255

4.1 An introduction to Multilink Protocol (MP)

The Point-to-Point (PPP) Multilink Protocol (MP) follows RFC 1990, *The PPP Multilink Protocol*. MP enables multiple PPP links to be grouped together to form a single virtual link or bundle. The links that make up the bundle must be the same type (for example, all L2TP lines, all PPP analog leased lines, or all PPP ISDN switched lines). MP requires that MP support is implemented on both ends of a PPP or L2TP link.

The multilink protocol is transparent to applications. It sits logically between the data link layer and the network layer as shown in Figure 4-1. MP appears to the network layer protocols as a single link, regardless of how many physical links are in the multilink bundle.

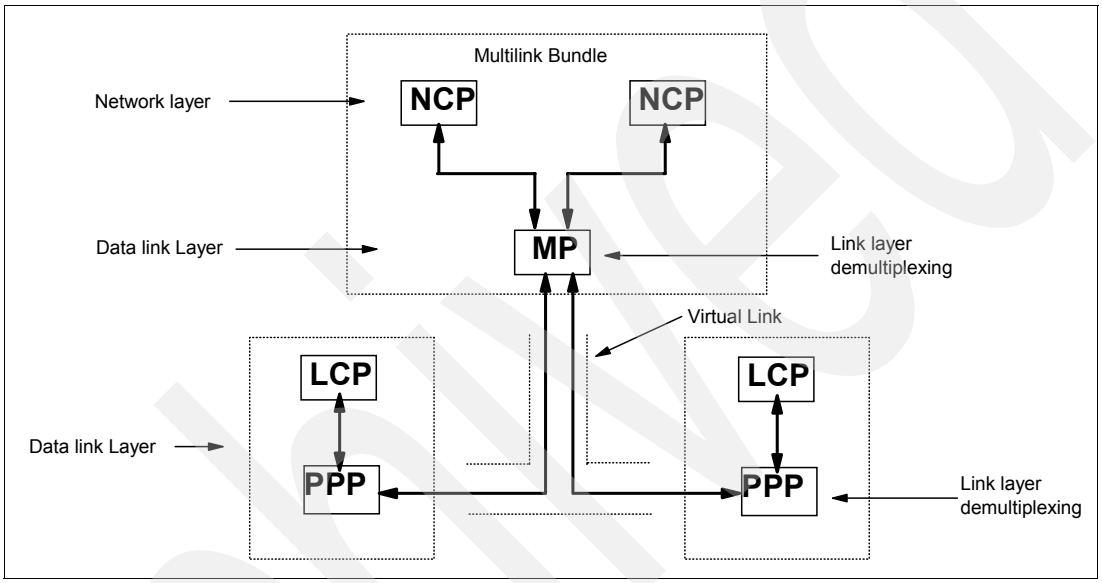


Figure 4-1 Multilink Protocol

To establish an MP operation, a system sends the Maximum Reconstructed Receive Unit (MRRU) option to its peer during LCP option negotiation on a PPP link. The inclusion of the MRRU LCP option indicates that a peer supports MP. If both peers successfully negotiate the MRRU LCP option, MP negotiation completes successfully. At this time, the sending system is free to receive protocol data units (PDU) from the network layers. The correct order is guaranteed by using the MP sequence number in the MP headers.

4.2 Multilink implementation on the System i

MP on the System i is implemented for both the originator and receiver PPP profiles and for the receiver L2TP profiles. MP enhances total effective bandwidth and reliability. MP was first made available on the System i in V5R1 of OS/400. The System i PPP supports Bandwidth Allocation Protocol (BAP) and Bandwidth Allocation Control Protocol (BACP).

4.2.1 BAP and BACP

These two protocols are defined by RFC 2125, *The PPP Bandwidth Allocation Protocol and The PPP Bandwidth Allocation Control Protocol*. BAP and BACP enable links to be added and removed from an MP bundle in response to dynamically changing bandwidth utilization. BAP and BACP work in conjunction with bandwidth utilization monitoring. BAP and BACP for

an Originator profile is enabled by checking the box next to **Enable multilink protocol** as shown in Figure 4-2.

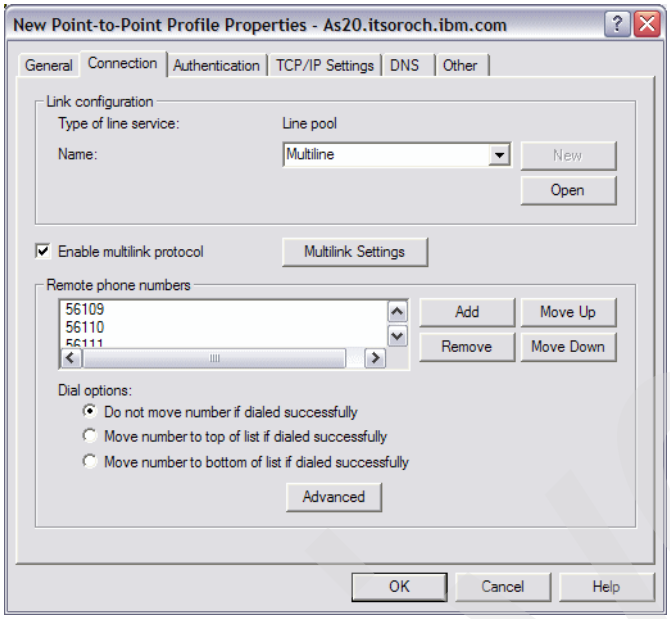


Figure 4-2 Originator profile: enabling BAP and BACP

Figure 4-3 shows BAP and BACP being enabled for a Receiver profile by the **Enable multilink protocol** box being checked. If **Require bandwidth allocation protocol** is enabled, as shown in Figure 4-3, all incoming connections will be required to negotiate BAP and BACP. The connection will fail if BAP and BACP negotiation does not occur.

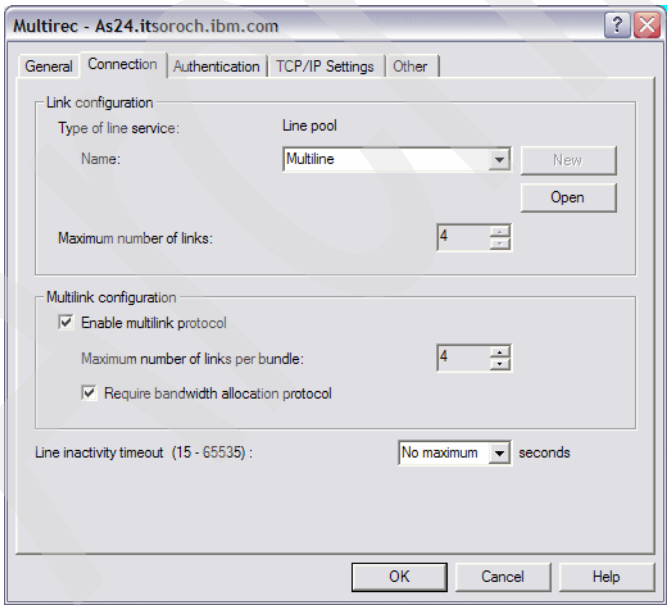


Figure 4-3 Receiver profile: enabling BAP and BACP

4.2.2 Bandwidth utilization monitoring

In order to realize the benefits of dynamically changing the bandwidth based on demand, at least one peer must be capable of monitoring the utilization of total bandwidth in an MP

bundle. The ability to do so is called bandwidth utilization monitoring. Links may be added or removed from a bundle as needed. The System i supports bandwidth utilization monitoring for Originator profiles but not for Receiver profiles.

Note: To be clear, via the BAP and BACP protocol, the System i will add or remove links from a bundle based on a remote Bandwidth Utilization Monitor if the System i is the receiver.

Bandwidth utilization monitoring determines the capacity of a link using three methods:

- The bandwidth of a switched asynchronous link is always assumed to be 28.8 Kbps for a single line. The reason that the System i must make this assumption is that it is very difficult to determine the actual modem-to-modem speed of an asynchronous link due to data compression (or expansion) and dynamically negotiated rates.
- The bandwidth of a switched synchronous link is the value specified in the link's line description.
- For L2TP, the bandwidth utilization monitoring is not done.

To enable bandwidth utilization monitoring for an Originator profile, click **Multilink Settings** (Figure 4-2 on page 83). The Define window (Figure 4-4 on page 85) appears.

This window shows how links can be added and removed depending on utilization as a percentage of the current active links in the bundle.

There is also a parameter (`Allow remote system to initiate a call from this system`) that enables the remote system to request the System i to initiate another link to the remote system. This would occur when the Receiver profile and Originator are doing bandwidth utilization monitoring and the receiver (via BAP and BACP) tells us to start another link connection. You may want to leave this unchecked so that the System i retains control of line cost being incurred and so an additional link may be added.

How do you determine what percentage utilization to use for your asynch link?

How do you determine what percentage utilization to use for your asynchronous link? As stated earlier, the System i monitors asynchronous line utilization based on an assumed value of 28.8 Kbps. Because the majority of modems put into use today will be faster than 28.8 Kbps, some math must be done to determine your desired utilization. This formula helps to determine the utilization percentage to use in the bandwidth utilization monitoring settings:

```
(Average_modem_speed Kbps / 28.8 Kbps) * Desired_Utilization% = Configured_Utilization%
```

For example, suppose that your modems communicate at an average rate of 50 Kbps. This is just an estimate, as a modem's data compression varies. You wish to start an additional line when utilization for the connection equals 30% (or greater). To put the equation into practice, use:

```
(50 Kbps / 28.8 Kbps) * 30% = 52%
```

The closest percentage to 52% that can be selected is 50%. (See Figure 4-4.)

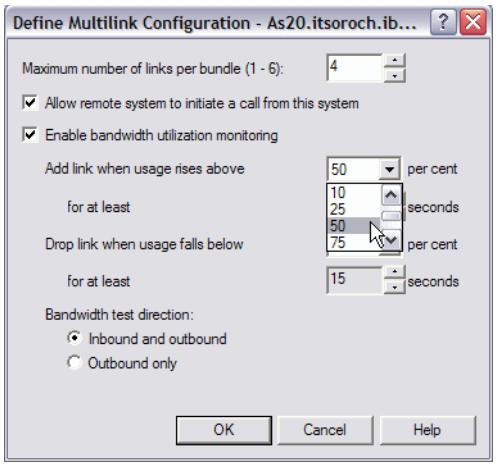


Figure 4-4 Originator profile: bandwidth utilization monitoring settings

Table 4-1 summarizes multilink support on the System i. The chart indicates the use of MP with PPP or ISDN. ISDN can also be utilized in conjunction with an asynchronous PPP profile through the use of a Terminal Adapter (TA). The TA would handle the multilink function. The System i currently has three TAs defined: Courier, Adtran, and Zyxel. Zyxel comes the closest to the System i 2750/2751 ISDN adapter.

Note: Support for the 2750 and 2751 ISDN adapters was removed in V5R3. Customers needing ISDN support in V5R3 and later releases should use an ISDN Terminal Adapter.

Table 4-1 Multilink support on the System i

Connection type	Operating mode	Allow MP/BAP?	Allow bandwidth utilization monitoring?
Switched line	Dial Answer Dial-on-demand (dial only) Dial-on-demand (answer enabled dedicated peer)	Yes Yes Yes No	Yes No Yes No
Leased line	Initiator Terminator	No No	No No
Virtual line (L2TP)	Initiator Terminator (network server)	No Yes	No No

5

Point-to-Point Protocol (PPP)

One goal of PPP is to foster interoperability among the remote access software of different manufacturers. Another goal is to enable the same physical communication line to be used by multiple network communication protocols.

The PPP protocol is described in multiple RFC standards:

RFC1661 Point-to-Point Protocol
RFC1662 PPP on HDLC-like framing
RFC1994 PPP Challenge Handshaking Protocol (CHAP)

More information about the RFCs can be found at:

http://www.rfc-editor.org

In addition, the IBM Redbooks publication *V4 TCP/IP for AS/400: More Cool Things Than Ever*, SG24-5190, has a very good introduction to WAN connectivity using PPP and SLIP.

This chapter assumes that you know the history of PPP on the System i and simply updates you on all of the enhancements since V5R1 of OS/400 and i5/OS.

To take advantage of these V5 enhancements we have included the following scenarios in this book:

- “Multilink: dynamic bandwidth allocation” on page 242
- “Multilink: Fault tolerance” on page 255
- “PPPoE branch office with secured connection” on page 522
- “Dynamic resource sharing scenario” on page 567
- “Dial-on-demand with unnumbered PPP connection” on page 574
- “System i RADIUS NAS” on page 586
- “Assigning an IP address to PPP client from DHCP server” on page 610

5.1 A brief introduction to WAN connectivity on the System i

PPP and SLIP enable WAN (point-to-point) connectivity. Examples are System i to System i, PC to System i, and System i to ISP (Internet).

When two systems are physically connected, typically it is referred to as a point-to-point connection or link. Several different protocols, such as Point-to-Point Protocol (PPP), SLIP, X.25, and frame relay, are viewed as Point-to-Point Protocols. However, in many cases, PPP and SLIP offer a lower cost, more efficient connection alternative to X.25 and frame relay. Support for PPP and SLIP is included on your System i as part of WAN connectivity.

Table 5-1 shows some of the different WAN alternatives available.

Table 5-1 WAN connectivity alternatives

Service	Line speed	Required equipment	Interface
Analog (leased and switched)	56 Kbps or less	Modem	RS232 Asynchronous
Digital Data service (DDS)	56 Kbps or less	CSU/DSU	X.21/V.35/RS-449 Synchronous
Switched -56	56 Kbps	CSU/DSU with V.25bis dial	V.35/RS-449 Synchronous
ISDN switched	56, 64, 112 or 128 Kbps	ISDN terminal adapter	RS232 Asynchronous
Fractional T1	56 Kbps to 1544 Kbps	CSU/DSU or T1 mux	X.21/V.35/RS-449 synchronous
T1/E1	56 Kbps to 1544/2048 Kbps	CSU/DSU or T1 mux	X.21/V.35/RS-449 synchronous

These connection methods include:

Analog phone lines Use standard V.42bis modems, or the latest technology: V.92 with V.44 compression technology, which extends the speed to 56 Kbs download and 48 Kbs upload.

DDS or digital service lines The most basic form of digital services, it enables speeds up to 56 Kbps. Requires a special box called Channel Service Unit/Data Service Unit (CSU/DSU), which replaces the modems used in an analog scenario.

Switched-56 Another digital service. Connects via CSU/DSUs and includes a dialing pad from which you enter the phone number of the remote host. It is available only in North America.

ISDN Switched end-to-end digital connectivity. ISDN can carry both voice and data over the same connection. There are different types of ISDN services, with Basic Rate Interface (BRI) being the most common. BRI consists of two 64-Kbps B channels to carry customer data and a D channel to carry signaling data. The two B channels can be linked together to give a combined rate of 128 Kbps. The ISDN modems are called Terminal Adapters (TA).

T1/E1 A T1 connection bundles together twenty-four 64-Kbps channels over a 4-wire copper circuit, giving a total bandwidth of 1544 Kbps. An E1 circuit in Europe bundles thirty-two

	64 Kbps lines for a total of 2048 Kbps. They typically connect using a V.35 interface to a CSU/DSU and a synchronous protocol.
Fractional T1	A customer can lease any 64 Kbps sub-multiple of a T1 line.

5.2 What you need to know to use the PPP scenarios

The sections that follow are an introduction to many of the features and enhancements to System i PPP support since V5R1 of OS/400 and i5/OS. Table 1-1 on page 4 gives an overview of the changes to PPP (and the wider WAN point-to-point connectivity) on the System i.

5.2.1 Dial-on-demand with unnumbered PPP connection scenario

Dial-on-demand is a typical dynamic TCP network that was first introduced on the System i with OS/400 at V4R2. Figure 5-1 shows the network configuration of dial-on-demand with unnumbered PPP connection. Two System i servers are connected with a dial-on-demand PPP connection. If there is IP traffic on AS20 that has to get to AS24, AS20 starts an on-demand PPP connection to AS24 using switched line and an unnumbered PPP connection. An unnumbered PPP connection is easy to configure because you simply specify the existing IP interface's IP address as a local or remote side IP address of the PPP connection. This on-demand PPP connection is disconnected after the specified idle time.

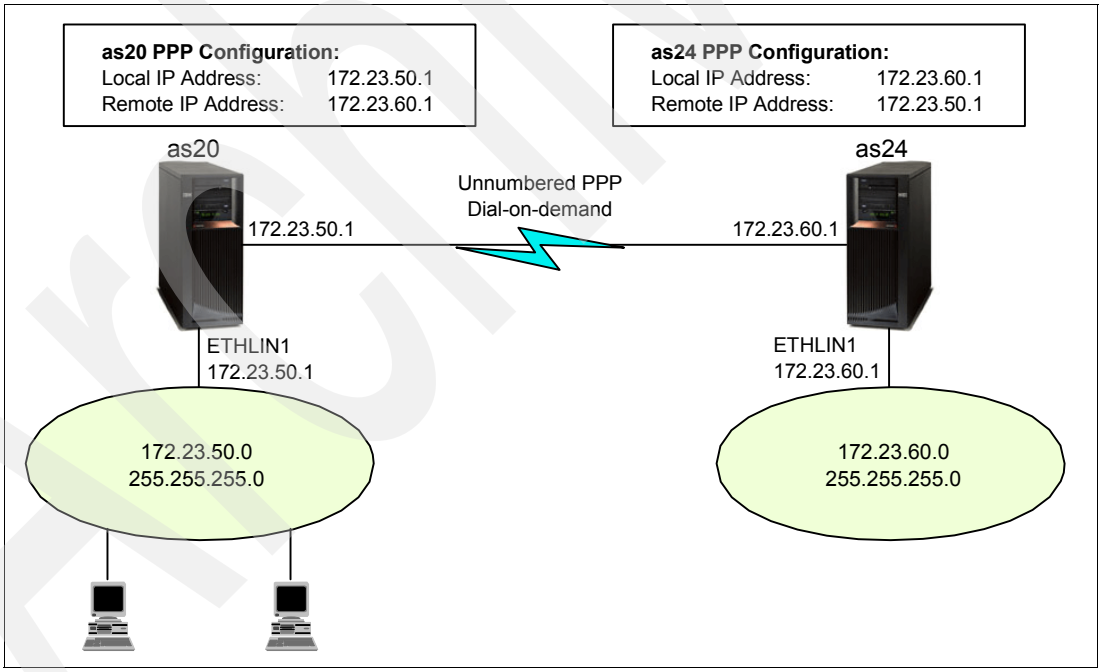


Figure 5-1 Dial on demand with unnumbered PPP connection scenario

We provide you with a how-to scenario in 17.3, "Dial-on-demand with unnumbered PPP connection" on page 574.

This is dynamic IP.

Tip: An unnumbered network

The best method of defining a point-to-point connection is to define it as an unnumbered connection.

The route selection process in the System i depends on having an IP address for an interface. In an unnumbered connection, the point-to-point interface that you configure will not have unique addresses either for the local or remote IP address. The remote IP address of the System i interface for an unnumbered connection is actually the IP address of the remote system.

In Figure 5-1 on page 89, AS20 is connected to the LAN network 172.23.50.0 with an address of 172.23.50.1. This enables AS20 to communicate directly with any system on the 172.23.50.0 network. The example also shows AS24, which is connected to the LAN network 172.23.60.0 with an address of 172.23.60.1. This enables AS24 to communicate directly with any system on the 172.23.60.0 network.

Now we have a need to connect AS20 to the 172.23.60.0 network and to connect AS24 to the 172.23.50.0 network. If these two systems were in the same room, we would simply add a LAN adapter to each system and plug the new interface into the correct LAN. If we did this, AS20 and AS24 would not have to have any routing entries added. In our case, however, the systems are in different cities, so we must use a point-to-point connection, but we would like to avoid creating a new network and manually adding route entries across that connection.

By defining the point-to-point connection as an unnumbered connection, we achieve the same results as though we had used LAN adapters and do not have to add any route entries to the System i. To do this, each System i borrows the IP address of the remote system for use with route resolution.

Each System i (AS20 and AS24) automatically adds the remote IP address to its route table as a local interface. The address is treated specially, so packets destined for that address will not be processed locally. The packets for the remote address will be placed on the interface and transported to the other end of the connection. When the packet arrives at the other end of the connection, normal packet processing is used.

AS20 looks as though it has an interface in the 172.23.60.0 network. AS24 appears to have an interface in the 172.23.50.0 network. Additional route table entries for AS20 and AS24 are added automatically when the unnumbered interfaces are started. For example, if these were PPP lines, the additional route is added when the PPP profile is started.

Again, no new networks or manually added route table entries must be created. This is dynamic IP at its best.

5.2.2 What is new in V5R1 PPP

The existing Point-to-Point folder has been renamed Remote Access Services, as seen in Figure 5-2. It contains three folders: Originator Connection Profiles, Receiver Connection Profiles, and Modems (identical to pre-V5R1). The new Originator Connection Profiles folder provides functions for defining and administering point-to-point profiles originating from the System i. The new Receiver Connection Profiles folder provides functions for defining and administering point-to-point profiles that receive connections originating from a remote system.

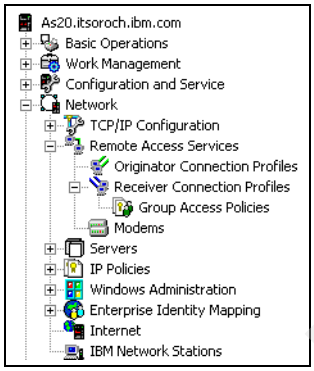


Figure 5-2 iSeries Navigator showing new and expanded Remote Access Services folder

Group Access Policies is a new option available under Receiver Connection Profiles. A group policy overrides the settings (multilink and TCP/IP) associated with an individual Receiver Connection Profile. The administrator can use the validation list under the PPP profile to define which group policy each user name uses. For example, you could have different user names use different filter rules.

Two new APIs have been provided for use in monitoring PPP profiles

Two new APIs have been provided for use in monitoring PPP profiles:

- List PPP connection profiles (QtocLstPPPCnnPrf): This API returns a list of PPP connection profiles with some basic information about each profile.
- Retrieve PPP connection profiles (QtocRtvPPPCnnPrf): This API retrieves the details of a specific PPP connection job profile. If the connection profile describes multiple connections, details of each connection are also retrieved.

SLIP over Async line profiles are no longer supported

As of V5R1, SLIP over Async line profiles are no longer supported. SLIP is still supported, but only though iSeries Navigator and over a *PPP line description. The profiles will not be removed. They can be viewed via Print Point-to-Point Profile (PRTTCPPTP). iSeries Navigator has an option available to migrate them to a PPP line or profile.

Trace TCP/IP Application (TRCTCPAPP) enhancements

Trace TCP/IP Application (TRCTCPAPP) has new choices:

***PPP**	Point-to-point protocol (PPP)
***QOS**	Quality of Service (QoS)
***NTP**	Simple Network Time Protocol (SNTP) client
***DIRSRV**	Directory Services server

These are new in addition to those already available in V4R5: *FTP, *CERTSRV, *SMTPSVR, *L2TP,*SMTPCLT, *TELNET, *VTAPI, *CENTRAL, *DTAQ, *RMTCMD, *SIGNON, *NETPRT, *SVRMAP, *DDM, and *VPN.

Connection data

Previous connection data is kept by right-clicking the connection profile and selecting **Connections**. The System i keeps data associated with previous connections, such as the user who connected and the IP address they used.

Select **Connections** for details such as:

- Group policy in effect
- IP forwarding, VJ compression, masquerading
- Line inactivity time-out, negotiated authentication protocol, and MTU size
- Filter rule name in effect
- Whether multilink is enabled and other multilink-related data

CL commands for PPP

At V4R5 and earlier, iSeries Navigator was required to configure PPP profiles. This works fine for those who can make a connection with iSeries Navigator, but if they cannot make such a connection, there is no way to create a PPP connection.

V5R1 introduces three CL commands that enable PPP to be administered via a green screen. They can be executed by selecting options from the Work with Point-to-Point TCP/IP (WRKTCPPTP) panel. The commands can also be executed from a command line. The commands are:

- Add Point-to-Point Profile (ADDTCPPTP): creates originator or receiver PPP profiles.
- Remove Point-to-Point Profile (RMVTCPPTP): removes any TCP/IP point-to-point profile.
- Print Point-to-Point Profile (PRTTCPPTP): prints any TCP/IP point-to-point profile configuration to a spooled file.

Support for SLIP via GUI (only)

In V5R1, you cannot create SLIP profiles via a green screen, but only via iSeries Navigator. Additionally, the support of SLIP over *ASYNC lines was removed in this version. Now, any SLIP profile is supported via *PPP line descriptions. The profiles will not be removed (you can still see them via PRTTCPPTP). An iSeries Navigator option is available to migrate them to SLIP over a *PPP line. All data that can be copied is used in the new profile. Authentication information is lost (because there is no way to extract the encrypted passwords).

DHCP enablement for PPP

The System i server acts as a DHCP WAN client for remote access dial-in and L2TP tunnel users. This enables WAN remote access users to obtain the same IP address services as LAN attached network DHCP clients do.

The DHCP WAN client must be enabled before you can use it. (In iSeries Navigator, expand **Network**. Right-click **Remote Access Services** and select **Services** from the context menu.)

The DHCP WAN client sends its request to either a DHCP server or DHCP relay agent. The Remote Access Services folder provides a property sheet with a selection for enabling a DHCP WAN client. When the DHCP WAN client is enabled, a dialog box is launched that requests that the user specify the IP addresses of the local interfaces to be used to connect to the DHCP server or relay agent. If it is detected that neither the DHCP server nor the relay agent are running, the GUI code launches a dialog box to configure the relay agent. The PPP Receiver Connection Profile can now be set to use DHCP. Do this by selecting the TCP/IP

Settings tab. The DHCP option for the Remote IP address can then be selected from a pull-down box.

See 6.4.7, “DHCP wide area network (WAN) client support” on page 114, for a more detailed description of this feature and configuration information. Also see 17.5, “Assigning an IP address to PPP client from DHCP server” on page 610, for a detailed scenario that leads you through the steps to configure a System i PPP Receiver profile to act as a DHCP WAN client to the System i DHCP server.

Multilink

The PPP Multilink Protocol (MP) follows RFC 1990. It enables multiple PPP links to be grouped together to form a single virtual link or bundle. The links that make up the bundle must be the same type (for example, all L2TP lines, PPP analog leased lines, or PPP ISDN switched lines).

MP on the System i, first made available in V5R1 of OS/400, is implemented for both the Originator and Receiver PPP profiles and for the Receiver L2TP profiles. Along with enhancing total effective bandwidth and reliability, the benefits of MP include:

- Reducing the latency of data sent between systems by increasing the total effective bandwidth.
- Increased reliability through the use of multiple lines. (If a line fails, the link is maintained as long as one line in the MP bundle remains operational.)
- The ability to dynamically add and remove lines from a bundle enables bandwidth to be supplied as needed, making more efficient use of the available bandwidth.

For more information see Chapter 4, “Multilink Protocol” on page 81. In addition, we provide you with two how-to scenarios in Chapter 14, “Multilink in action” on page 241.

Remote Authentication Dial-In Service (RADIUS)

Remote Authentication Dial-In Service (RADIUS) is a distributed security system developed by Lucent Technologies InterNetworking Systems. RADIUS was designed based on a previous recommendation from the IETF’s Network Access Server Working Requirements Group. RADIUS is the de facto industry standard for user authentication, authorization, and accounting. It has three main functions:

Authentication RADIUS server authenticates users for dial-in remote access based on user ID and password.

Authorization The RADIUS server can be configured to control access to specific services in the network for an authenticated user. Such services are routes, time-outs, and port limits. This is a very powerful feature in large distributed networks of i5/OS servers and employees.

Accounting RADIUS server accounting permits system administrators to track dial-in use. This is often used for billing purposes.

The RADIUS server is installed on a central computer at the customer’s site. The RADIUS Network Access Server (NAS) can be installed on the i5/OS server. The NAS is responsible for passing user information designated for the RADIUS servers and then acting on the response that is returned.

System i RADIUS NAS server working with RADIUS server scenario

Figure 5-3 shows a network configuration scenario of a System i NAS server working with a RADIUS server. In this scenario, the System i NAS server generates an Access request packet with the incoming client's user ID, password, and other information. The System i NAS server sends the Access request packet to the RADIUS server to get the authentication. This network configuration is most likely used in companies that have many mobile users.

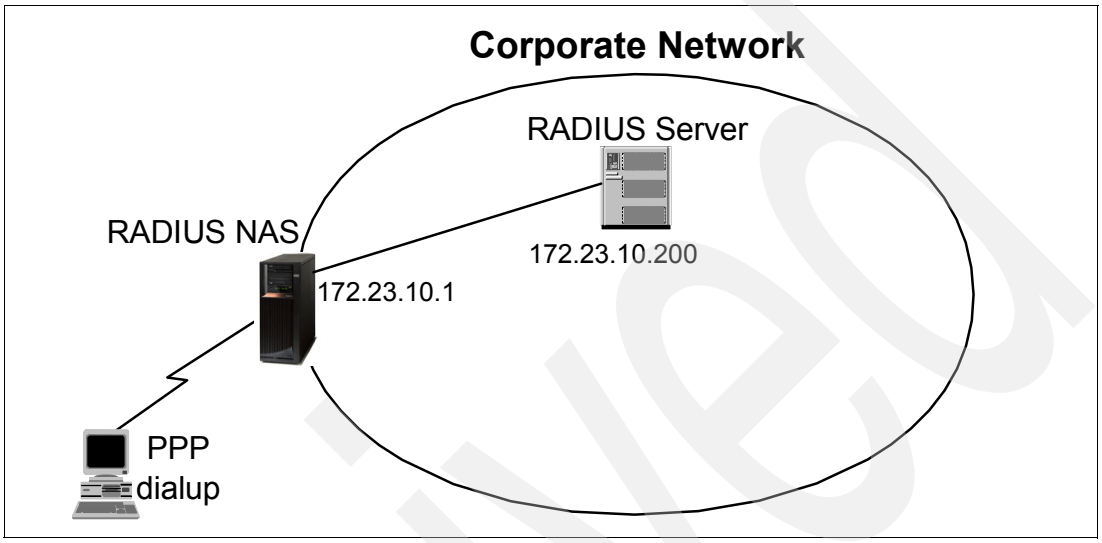


Figure 5-3 i5/OS NAS server working with RADIUS server scenario

For more information see Chapter 9, "Remote Authentication Dial-In User Service (RADIUS)" on page 137. In addition, we provide you with a how-to scenario in 17.4, "System i RADIUS NAS" on page 586.

RADIUS enablement for PPP and L2TP

Remote Access Services must be enabled for RADIUS in order for PPP and L2TP to use it. In iSeries Navigator, expand **Network**. Right-click **Remote Access Services** and select **Services** from the context menu.

You can enable some or all of the RADIUS functions. If the RADIUS NAS server has not been configured, a configuration wizard is initiated when Enable RADIUS Network Access Server connection is selected. Separate RADIUS servers can be used for authentication and accounting. The following items are required when setting up the configuration:

- Local IP address
- Server IP address (RADIUS server)
- Password (shared secret)
- Port number: This is the port under which the RADIUS server listens to authentication or accounting requests initiated from an NAS server. When RADIUS was first introduced in the market, servers listened on UDP port 1645 for authentication and 1646 for accounting requests. Newer implementations default to UDP port 1812 for authentication and 1813 for accounting requests. Always refer to the RADIUS server user's guide or help text to determine on which ports this server listens for RADIUS NAS requests.

5.2.3 What is new in V5R2 PPP

In this section we explain what is new in System i V5R2 PPP support.

PPPoE support

PPP over Ethernet (PPPoE) provides the ability to connect a network of hosts over a simple bridging access device to a remote access concentrator. With this model, each host utilizes its own PPP stack and the user is presented with a familiar PPP user interface. Access control, billing, and type of service can be done on a per-user basis, identical to conventional PPP links.

RFC 2516 is the specification that defines the encapsulation of PPP, the Link Control Protocol, Network-layer Control Protocols, authentication, and IP datagrams. PPPoE is a point-to-point relationship between the peers and is not designed for the multi-point relationships that are available in Ethernet and other multi-access environments.

This specification can be used by multiple hosts on a shared Ethernet to open PPP sessions to multiple destinations via one or more bridging modems. It is intended to be used with broadband remote access technologies that provide a bridged Ethernet topology, when access providers wish to maintain the session abstraction associated with PPP. PPPoE is primarily used in xDSL environments and in many cases is the only protocol supported by xDSL implementations.

PPPoE defines two new Ethernet frame types, x'8863' and x'8864', which must be supported by bridges and the System i communication adapter IOPs. For V5R2, PPPoE is limited to the F/C 2838 and F/C 2849 10/100 Mbps adapter. Also, the 2838 and 2849 cannot be shared as both the PPPoE and standard TCP/IP communications adapter. For V5R2, the 2838 or 2849 must be dedicated to either PPPoE or TCP/IP, but not both at the same time.

Note: The list of Ethernet adapters supporting PPPoE has expanded in V5R4. Check the Information Center for the adapters supported on your release. In addition, the restriction on sharing with standard TCP/IP communications has been removed in V5R4.

PPPoE scenarios

In V5R2, System i supports PPPoE as in PPPoE host (client) role. Figure 5-4 shows the connection diagram of a DSL connection with the PPPoE protocol. The System i has a user ID and password to authenticate with the PPPoE DSL Concentrator on the ISP side.

Since broadband Internet connection service was announced, the number of broadband Internet users has been increasing. Network connectivity in branch offices was significantly changed by high-speed and reasonably priced broadband connection service. Previously, dedicated line, frame relay, T1, or ISDN line were used to connect two branch office networks. Now the trend is to install xDSL or a cable modem in branch offices for a faster connection at a reasonable price.

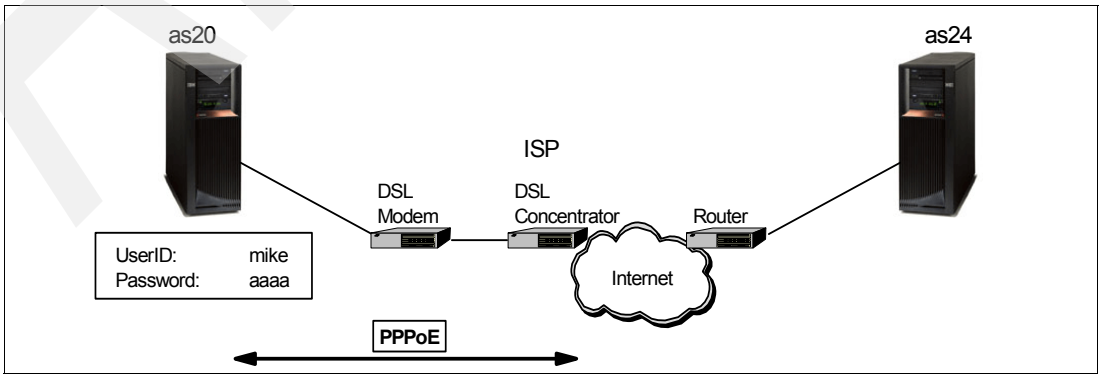


Figure 5-4 Branch office DSL PPPoE connection

To connect two branch offices over the Internet, we must configure a secured connection. In 17.1, “PPPoE branch office with secured connection” on page 522, we implement a typical branch office network configuration, which includes PPPoE for DSL connection and VPN for encrypted communication, as shown in Figure 5-5.

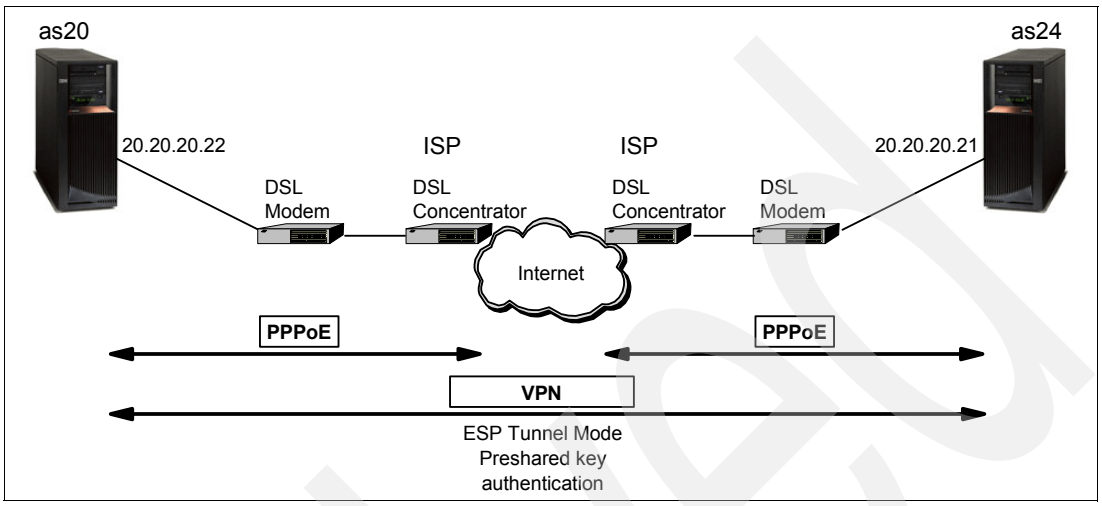


Figure 5-5 PPPoE branch office with secured connection

We provide a how-to scenario in 17.1, “PPPoE branch office with secured connection” on page 522.

PPP Dynamic Resource Sharing for analog connections

In V5R1, analog modem ports may not be dynamically shared, requiring manual intervention to change which function has ownership of the resource.

The PPP Dynamic Resource Sharing function introduced in V5R2 provides the capability to designate an (analog) line resource as *shared*, enabling PPP dial profiles to *borrow* a line being used to listen for incoming calls in order to place an outgoing call. This permits shared use of any PPP-supported adapters on the System i. See this topic in the Information Center. Go to:

`http://publib.boulder.ibm.com/iseries/`

Select your language and release. Then select **Networking → TCP/IP Applications, protocols, and services → Remote Access Services: PPP connections → Plan PPP → Software and hardware requirements** for a list of the supported adapters.

Dynamic resource sharing scenario

Using this function, you can minimize the required hardware resources if your communication application needs some Receiver connection profiles for incoming calls and Originator profiles for outgoing calls. Figure 5-6 shows the dynamic resource sharing scenario.

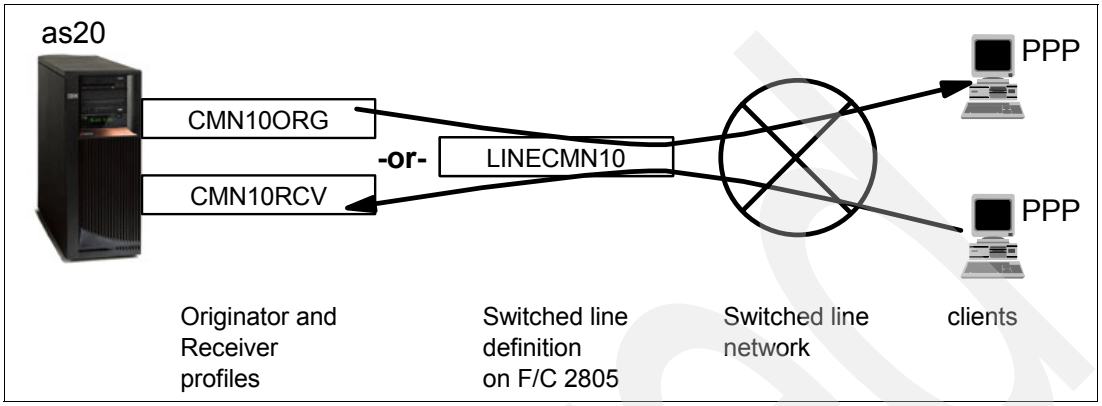


Figure 5-6 Dynamic resource-sharing scenario

In this scenario, Originator profile CMN10ORG shares a line resource LINECMN10 with Receiver profile CMN10RCV. While CMN10ORG is initiating an outgoing call using line resource LINECMN10, the status of CMN10RCV is Resource sharing, or waiting for the line to become active, as shown in Figure 5-7.

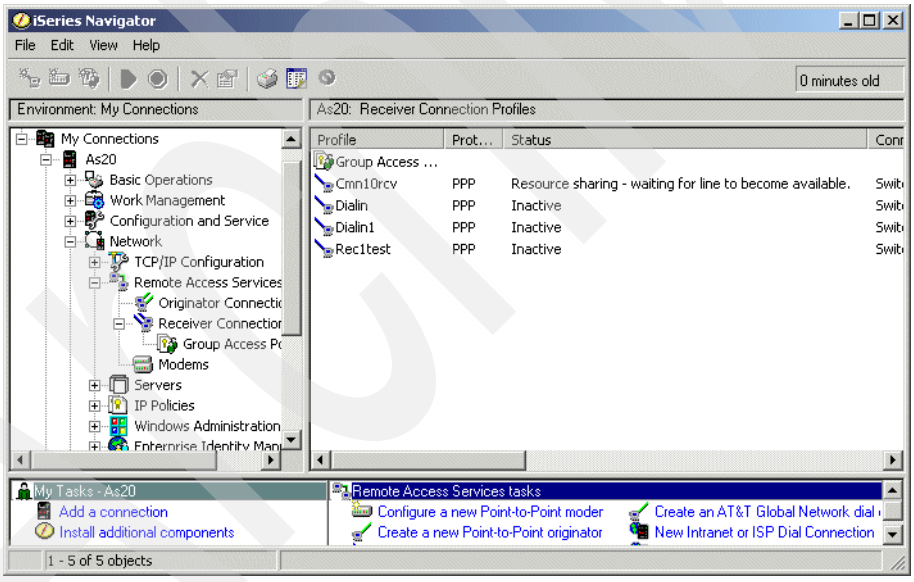


Figure 5-7 Dynamic resource sharing: waiting for line to become available

We provide a how-to scenario in 17.2, “Dynamic resource sharing scenario” on page 567.

ADDDFTRTE parameter change in ADDTCPPTP command

In V5R2, a parameter ADDDFTRTE in Add Point-to-Point Profile (ADDTCPPTP) command was modified to enabled a default route to be added for *DIAL profiles.

ADDDFTRTE specifies whether you want a default route added when this Point-to-Point Connection profile is started. This parameter is only in effect if OPRMODE(*DIAL) is specified.

***NO** A default route will NOT be added automatically.

***YES** A default route will be added automatically when this Point-to-Point profile is started. The next hop address will be the IP address of the remote system.

5.2.4 What is new in V5R3 PPP

In this section we explain what is new in System i V5R3 PPP support.

L2TP outgoing call support

L2TP outgoing call support allows multiple systems or partitions to share a single modem or a pool of modems — another excellent example of Dynamic IP. See an example scenario of how to use this support in the Information Center. Go to:

http://publib.boulder.ibm.com/iseries/

Select your language and release V5R3 or later. Then select **Networking** → **TCP/IP Applications, protocols, and services** → **Remote Access Services: PPP connections** → **PPP Scenarios** → **Scenario: Share a modem between logical partitions using PPP and L2TP**.

Automatic starting of profiles with TCP/IP

PPP, PPPoE, and L2TP profiles can now be started automatically with TCP/IP if desired. The New Profile GUI Interface in iSeries Navigator allows you to specify whether a profile should automatically start with TCP.

5.2.5 What is new in V5R4 PPP

In this section we explain what is new in System i V5R4 PPP support.

PPPoE Ethernet Adapter sharing

Starting in V5R4, you can share the same Ethernet Adapter for PPPoE and TCP/IP Communications using IPv4 and IPv6. In addition, the list of adapters supporting PPPoE was greatly expanded. See this topic in the Information Center. Go to:

http://publib.boulder.ibm.com/iseries/

Select your language and release. Then select **Networking** → **TCP/IP Applications, protocols, and services** → **Remote Access Services: PPP connections** → **Plan PPP** → **Software and hardware requirements** for a list of the supported adapters.

6

Dynamic Host Configuration Protocol (DHCP)

DHCP is the king of System i dynamic IP. The DHCP server is central to a wide range of techniques to automate your network configuration. In this IBM Redbooks publication we have 11 how-to scenarios that directly involve the System i

The Dynamic Host Configuration Protocol (DHCP) provides a framework for passing configuration information to hosts on a TCP/IP network. This configuration information, called options, can be related directly to IP such as IP address, subnet mask, or the default gateway router for a particular host. The System i DHCP server can also tell the host the domain it finds itself in and the IP address of the DNS server. In addition, many more options have been defined to support Network Station® and many other TCP/IP applications.

DHCP is based on the Bootstrap Protocol (BOOTP), adding the capability of automatic allocation of reusable network addresses and additional configuration options. DHCP is described by RFC 2131, *Dynamic Host Configuration Protocol* and RFC 2132, *DHCP Options and BOOTP Vendor Extensions*.

Tip: The best place to find and read RFCs is on the Internet: The Web site `http://dir.yahoo.com/Computers_and_Internet/Standards/RFCs/` has a list of sites that enable you to search for and then read specific RFCs.

In this chapter, we introduce you to BOOTP (6.1, “BOOTP: The predecessor to DHCP” on page 101) and then present an overview of DHCP concepts and components and the implementation of DHCP on System i.

We also have many how-to scenarios to demonstrate that DHCP is at the very core of your System i dynamic IP networks.

Chapter 15, “DHCP: Dynamic allocation of IP addresses” on page 269, contains a group of scenarios that has a focus on just DHCP:

- “DHCP: One physical network, one logical network, one DHCP server” on page 270
- “DHCP: One physical network, multiple logical networks, one DHCP server” on page 292
- “DHCP: One physical subnet, one logical subnet, multiple DHCP servers” on page 307
- “DHCP: multiple physical networks, logical networks, and DHCP servers” on page 322

- “DHCP: multiple physical, logical networks, and DHCP servers using Relay Agents” on page 343

The next group demonstrates the power of the System i DHCP dynamically updating your domain’s DNS server:

- “Single DDNS and DHCP server on the same server” on page 368
- “Single DDNS and DHCP servers without secured updates” on page 402
- “Single DDNS and DHCP servers with secured updates” on page 438
- “Primary DDNS and DHCP servers on one server, secondary server as backup” on page 447
- “Primary DDNS and DHCP servers, secondary DNS server Red Hat Linux 7.2” on page 460

This last scenario demonstrates the ability of the System i DHCP server to dynamically assign IP addresses to remote PPP clients: 17.5, “Assigning an IP address to PPP client from DHCP server” on page 610.

6.1 BOOTP: The predecessor to DHCP

The Bootstrap Protocol (BOOTP) was developed originally as a mechanism to enable diskless hosts to be booted remotely over a network. It allows a minimum IP protocol stack with no configuration information to obtain enough information to begin the process of downloading the necessary boot code.

BOOTP is described by RFC 951, *Bootstrap Protocol (BOOTP)*.

BOOTP describes the first phase of the bootstrap operation, which is address determination and bootfile selection. After the client determines this information, control passes to the second phase of the bootstrap where a file transfer occurs. Usually, the file transfer used is the Trivial File Transfer Protocol (TFTP).

The BOOTP protocol defines three network components:

- BOOTP client
- BOOTP server
- BOOTP forwarding agent

The BOOTP server contains a database with the clients' MAC addresses and the associated IP addresses and boot files. The server is listening on well-known UDP port 67 for requests from BOOTP clients.

The BOOTP client requests configuration parameters from the BOOTP server. It receives messages on port 68.

The forwarding agent is used to forward BOOTP messages between a client and a server, which are located in different physical networks.

The BOOTP flow between a client and a server is shown in Figure 6-1 and consists of the following steps:

1. The BOOTP client uses the special broadcast address of all ones (255.255.255.255) to send a request to the server.
2. When the server receives a BOOTP request from a client, the server looks for a defined IP address based on the client MAC address. It then replies with the client IP address and the name of the load file.
3. The client initiates a TFTP request to the server for the load file.

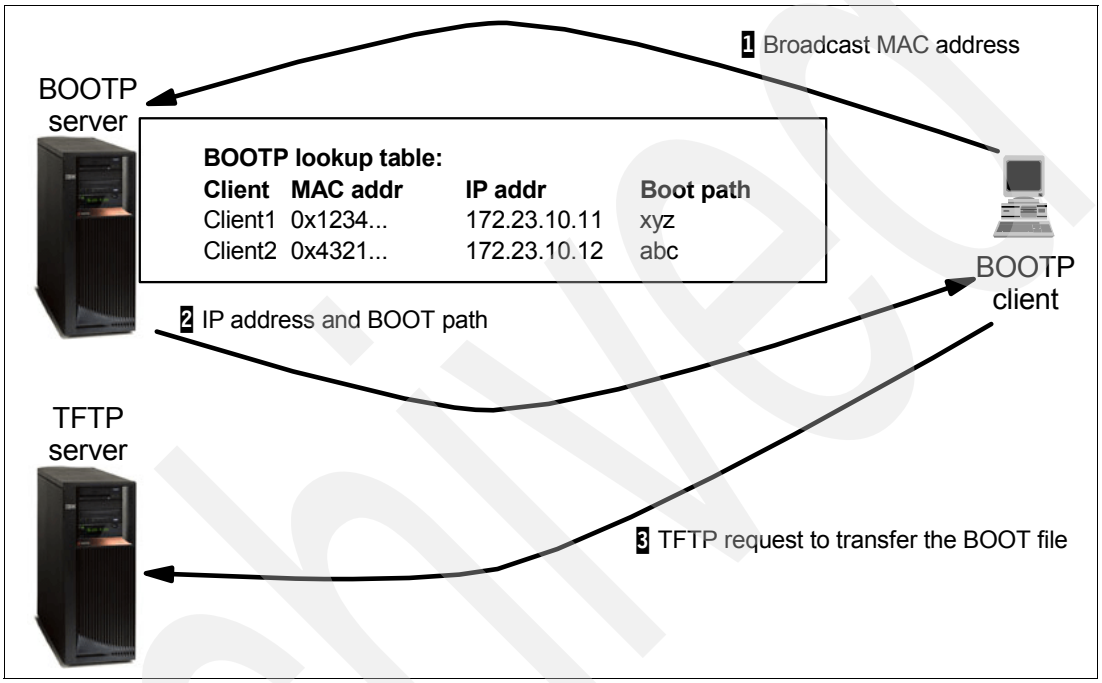


Figure 6-1 BOOTP flow between the client and the server

BOOTP clients can interact with DHCP servers, and DHCP servers and BOOTP servers can coexist, if configured properly. DHCP clients cannot interoperate with BOOTP servers.

The System i DHCP server support accommodates the already existing BOOTP server that was available in earlier releases of OS/400. Also, the DHCP server accommodates BOOTP clients. Additionally, it performs all of the functions specific to BOOTP, as well as the added function that a DHCP server is assumed to carry. BOOTP and DHCP servers cannot run at the same time on the system because both use the well-known port 67.

Tip: Unless you have any specific reason to use BOOTP, we recommend that you use DHCP protocol to assign IP addresses to clients. A migration program is also available to migrate your old BOOTP configuration to DHCP. See Figure 6.4.12 on page 122, for more information about this topic.

6.2 Overview of DHCP

Compared with BOOTP protocol, DHCP adds the capability of automatically allocating reusable network addresses and distributing additional host configuration options.

The DHCP server has three components that are represented in Figure 6-2. These components are:

DHCP host clients These run the DHCP client programs. Examples of DHCP host clients: Linux workstations, IBM Network Stations, Windows systems. The System i cannot be a DHCP client. The DHCP client listens for messages from the DHCP server on UDP port 68.

DHCP servers These provide addresses and configuration information to BOOTP and DHCP clients within the TCP/IP network. The DHCP server listens for requests from DHCP clients on UDP port 67.

Since OS/400 V4R2 and i5/OS, the System i can be a DHCP server.

BOOTP/DHCP Relay Agents The Relay Agent is used to relay the conversation between a DHCP server and a DHCP client, when the server and the client are in different physical subnets. The Relay Agent eliminates the need of having a DHCP server on each subnet to serve the existing DHCP clients.

The System i can act as a BOOTP/DHCP Relay Agent since V4R2. You cannot run the BOOTP/DHCP Relay Agent and the DHCP server at the same time on the same System i.

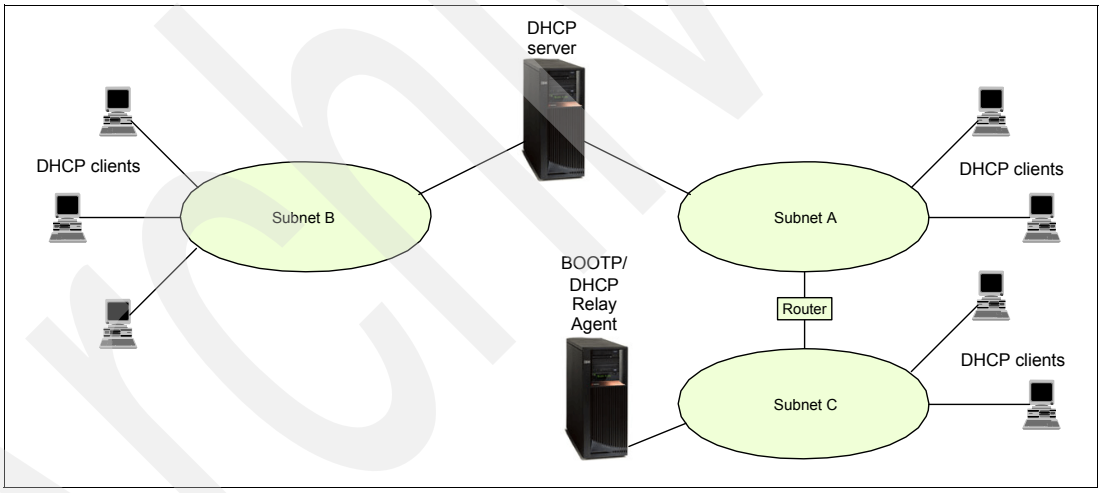


Figure 6-2 DHCP network components

6.3 How DHCP works

The DHCP protocol enables clients to obtain their IP addresses and additional IP network configurations from a DHCP server.

In addition, the DHCP server can be configured to return many options to the DHCP clients. For information about DHCP options, the DHCP clients can obtained from the DHCP server. See RFC 2132, *DHCP Options and BOOTP Vendor Extensions*. Figure 6-3 shows just a few of the many options that can be sent from the DHCP server to the DHCP clients during the execution of the DHCP cycle. (See Figure 6-4.)

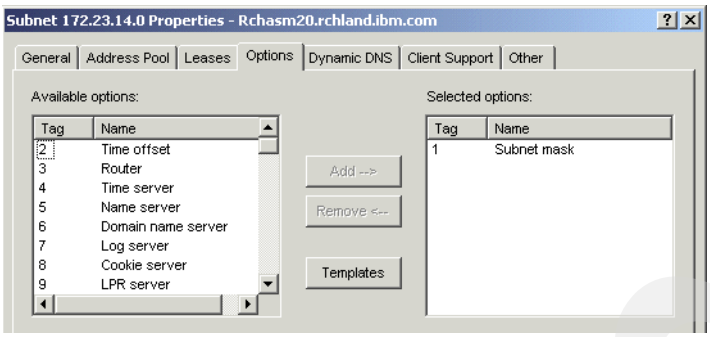


Figure 6-3 A few of the many DHCP options available to be sent to DHCP clients

The IP addresses can be allocated permanently or leased for a specific time period. If the server leases the IP address for a period of time, the client who received the IP address has to check with the server to revalidate the address and renew the lease on a periodic basis.

In the sections that follow, we detail these interactions between the DHCP clients, DHCP servers, and BOOTP/DHCP Relay Agents.

6.3.1 Acquiring configuration information

DHCP enables DHCP clients to obtain an IP address and other configuration information through a request process to a DHCP server. The dialog between DHCP client and DHCP server is based on RFC-designed messages.

Figure 6-4 shows a high-level overview of the DHCP protocol cycle.

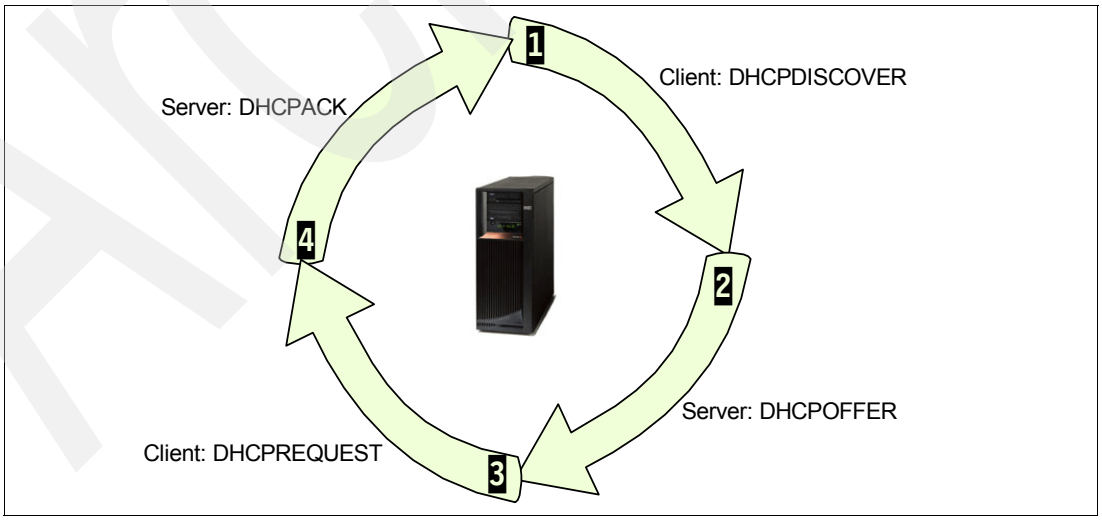


Figure 6-4 DHCP cycle overview

Here are the details of the four steps in the DHCP cycle:

1. The client broadcasts a DHCPDISCOVER message that contains its client ID on its local physical subnet. (See Figure 6-5.)

   Tip: The client ID for a Windows system is the MAC address of the physical interface. For an Apple MAC, the client ID can be a name or the MAC address of the physical interface.

   By issuing the DHCPDISCOVER message, the client requests an IP address from any DHCP server. The DHCPDISCOVER message may include other options requested by the client, such as a subnet mask, domain name server, domain name, and so on. The client can suggest specific values for some options.

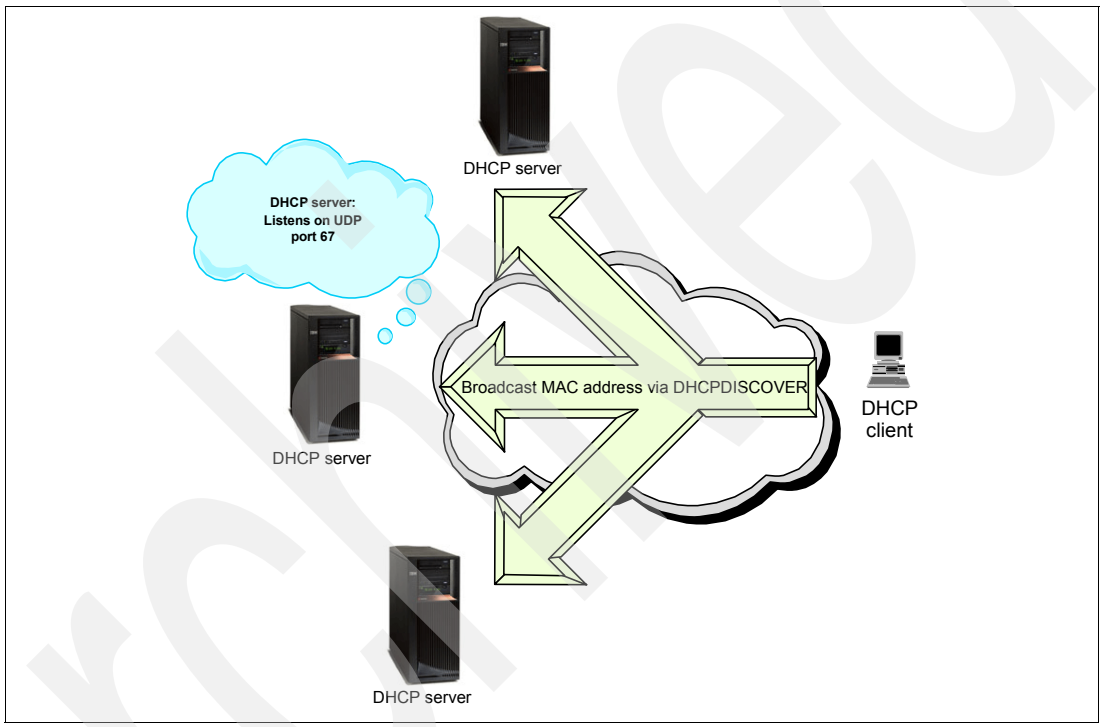


Figure 6-5 DHCP client broadcasts DHCPDISCOVER on its subnet

2. When a DHCP server receives a DHCPDISCOVER message from the client, the server chooses a network address for the requesting client.

   If the DHCP server has multiple physical interfaces, the DHCP server listens for requests on all of its interfaces.

Note: If the System i has multiple logical IP interfaces configured on one physical interface, it is difficult to predetermine which logical IP interface the DHCP server will use to receive the DHCPDISCOVER messages. This has to do with the multihomed capabilities of the System i (see 2.1, “Interfaces” on page 16).

As it turns out, the TCP/IP stack on the System i must pick one of the logical IP interfaces as the destination IP address. In most situations, the TCP/IP stack will assign the last logical IP interface started as the destination IP address for the DHCPDISCOVER message. But our V5R2 testing showed that this is not always the case.

This choice will eventually influence the DHCP server as it must assign an IP address and subnet for this new DHCP client that will best fit based on this logical IP interface assignment by the System i TCP/IP stack.

For more about the Relay Agent, see 6.3.4, “BOOTP/DHCP Relay Agent” on page 110.

When the System i DHCP server receives a DHCPDISCOVER packet that has the RELAY AGENT field equal to zero (that is, a Relay Agent was not used to forward the DHCPDISCOVER packet to this System i), the DHCP server uses the IP address of the interface the DHCPDISCOVER packet was received on as a clue to determine the network address of the IP address that will be assigned to the client. The DHCP server will try to assign to the DHCP client an IP address from its configuration that has the same network address as the IP interface that received the DHCP message. If the DHCP server could not find such an IP address, the server will be unable to send an offer to the client.

When the DHCP server receives a DHCPDISCOVER packet that has the RELAY AGENT field not equal to zero, the DHCP server will use the value specified in this field as a clue to determine the network address of the IP address that will be assigned to the client.

If the IP address that the server wants to allocate to the DHCP client has not been assigned previously, the DHCP server checks that the address is not already in use in the network before issuing an offer.

The server checks the configuration file to see whether it needs to assign a static or a dynamic address to the client:

- If a dynamic address must be assigned, the DHCP server selects the least recently used IP address from the address pool. The address pool is a range of IP addresses that are potentially leased to DHCP clients.
- If a static IP address must be assigned, the DHCP server uses a client statement from the server configuration file to assign an IP address to that specific DHCP client.

After the server chooses the IP address and the other configuration options for the client, it builds a DHCPOFFER and sends it to the client (Figure 6-6).

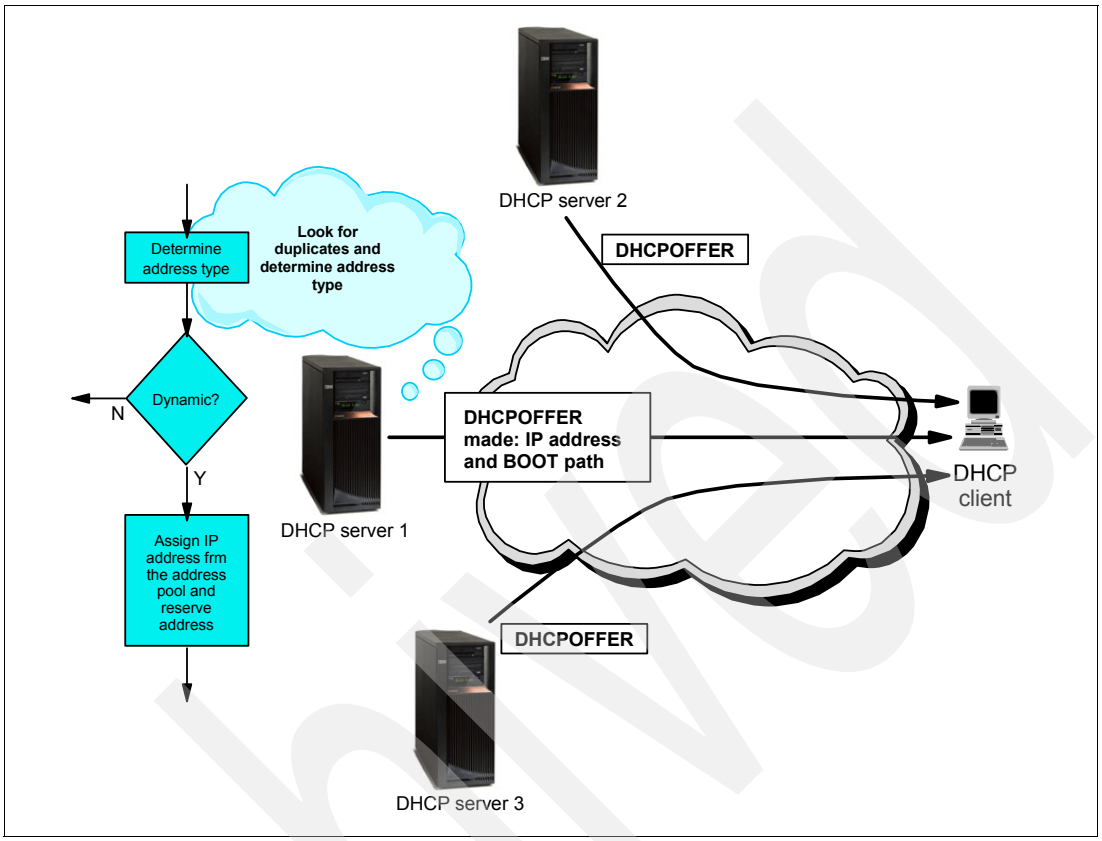


Figure 6-6 All DHCP servers in the subnet send a DHCPOFFER message to client

3. The DHCP client receives DHCPOFFER messages from several DHCP servers. The client compares all received offers and selects the one that meets its criteria.

> **Tip:** Not all DHCP clients can wait for several offers and evaluate them afterwards. Many DHCP clients available today accept the first offer they get from a DHCP server.
>
> As an example, we found it difficult to control the Windows 2000 DHCP clients' choice of DHCP server offers based on the options offered.

After the client decides which offer to accept, the client broadcasts a DHCPREQUEST message, which contains the IP address of the server that was selected (Figure 6-7). By sending the DHCPREQUEST message, the client acquires the IP address that was served by that specific DHCP server to use it.

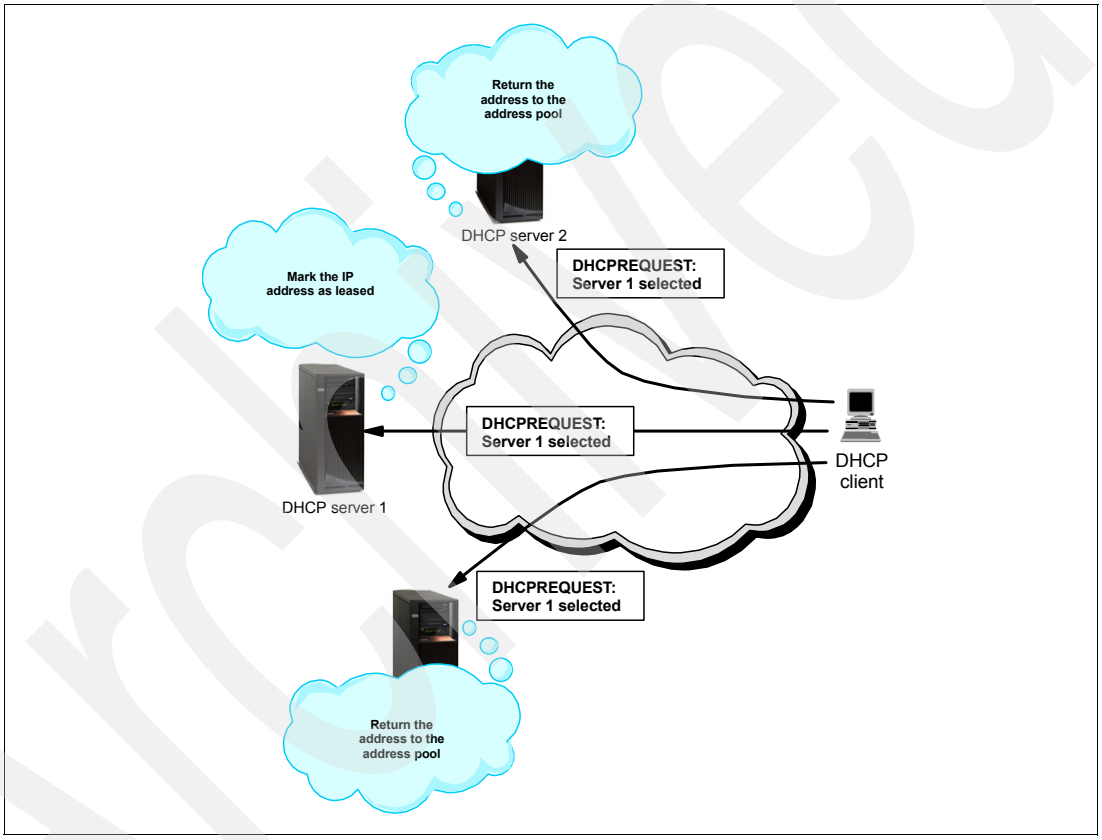


Figure 6-7 The client broadcasts the DHCPREQUEST message

4. When a DHCP server receives the DHCPREQUEST from the client, first it checks to see which server was chosen by the client and performs one of the following:
   - If the client accepted its offer, the server marks the corresponding IP address as leased and responds with a DHCPACK (DHCP Acknowledge) message containing the configuration parameters for the requesting client (Figure 6-8).
   - If the client accepted the offer of another DHCP server, the server returns the address to the available pool. The server will perform in the same way if does not receive any message from the client.

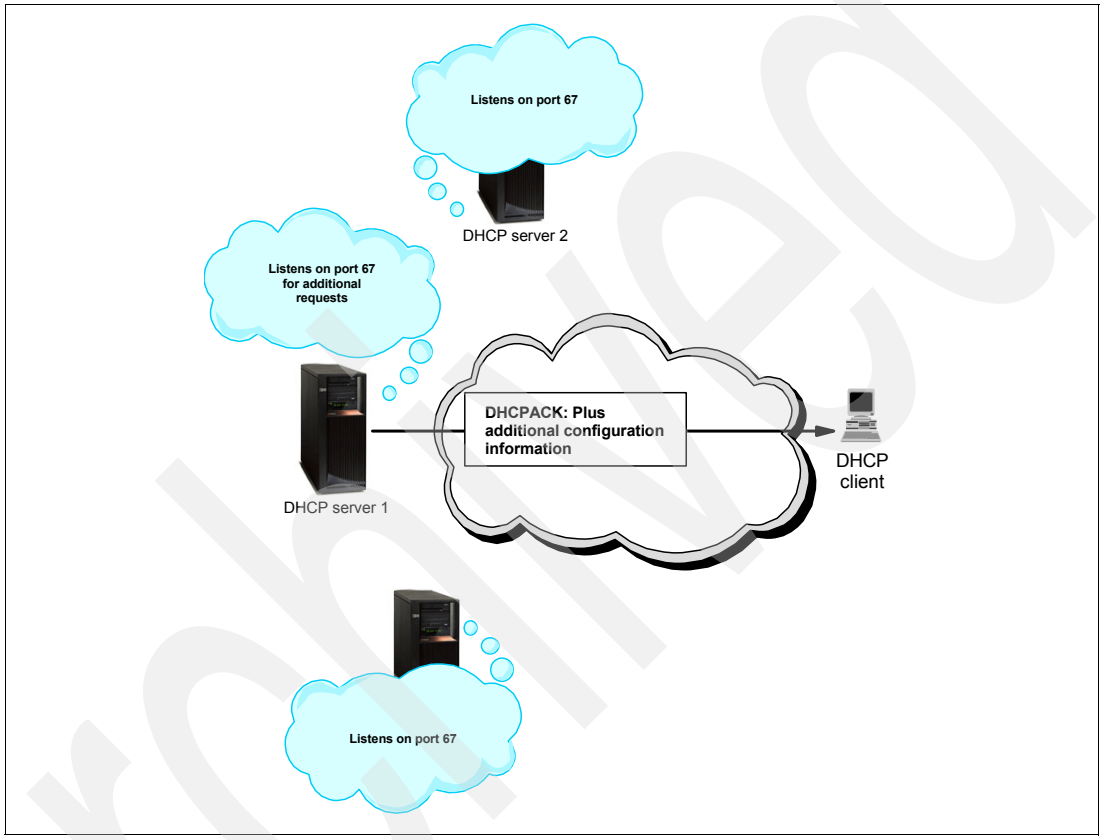


Figure 6-8 The chosen DHCP server responds with DHCPACK to the client

The DHCP client then determines whether the configuration information is valid. If the lease is valid, the client specifies a BINDING state with the DHCP server and proceeds to use the IP address as well as the specified options.

6.3.2 Lease renewal

The lease represents the time interval that the server allows the DHCP client to use an IP address.

The DHCP client keeps track of how much time is remaining on the lease. At a specified time prior to the expiration of the lease (usually when half of the lease time has passed), the client sends a renewal request to the leasing server. This request contains its current address and configuration information.

If the server responds with a DHCPACK, the client lease is renewed.

If the DHCP server explicitly refuses the request, the DHCP client continues to use the IP address until lease time expires. At this time, the client initiates the address request process, including broadcasting the address request. If the server is unreachable, the client continues to use the assigned address until the lease expires (Figure 6-9).

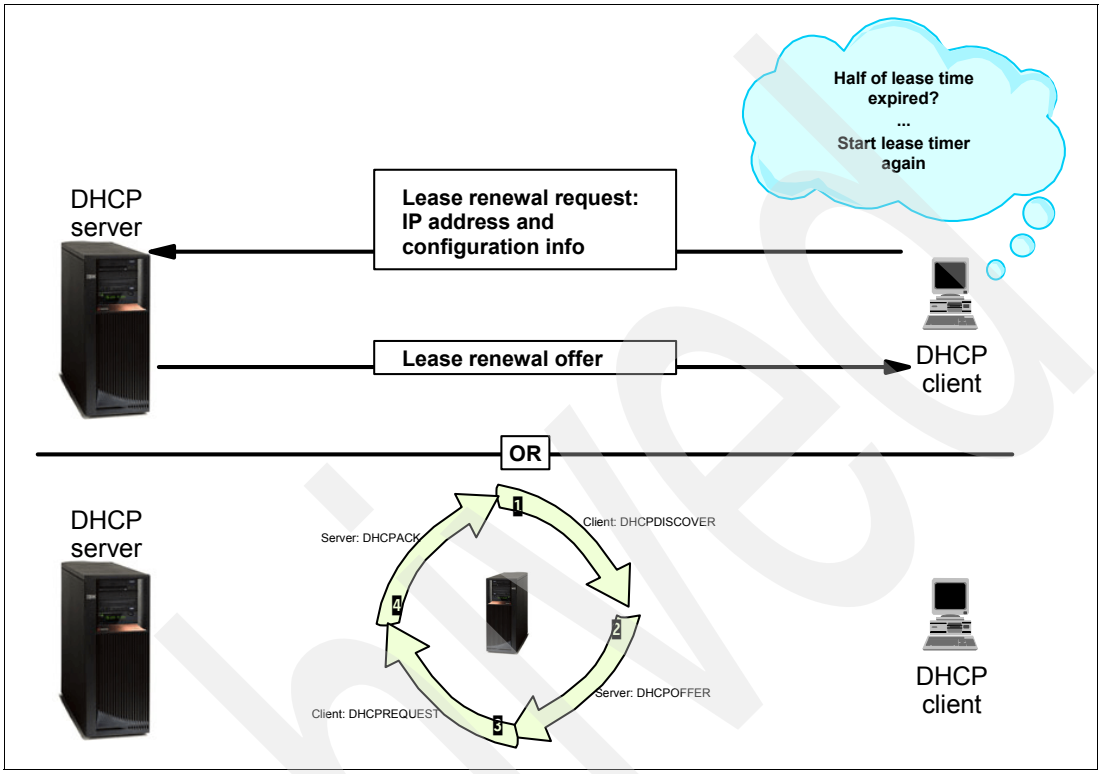


Figure 6-9 How leases are renewed

6.3.3 DHCP server configuration changes

A DHCP client retains option values being assigned by the DHCP server for the duration of the lease. If the configuration of the DHCP server changes while a client is already up and running, those changes are not processed by the DHCP client until either it attempts to renew its lease or it is restarted.

6.3.4 BOOTP/DHCP Relay Agent

A relay agent intermediates the dialog between the DHCP client and the DHCP servers when these two are located in different physical subnets. The BOOTP/DHCP Relay Agent forwards any BOOTP/DHCP requests that it receives on its subnet, or from other subnets, in the direction of the DHCP server. The Relay Agent is configured with the IP address where the DHCP messages received from DHCP clients should be sent (next hop). The next hop can be either a DHCP server or another Relay Agent.

Usually, the Relay Agent is configured on routers, but it also can be configured on a server. Starting with V4R2, the System i can act as a BOOTP/DHCP Relay Agent.

When the DHCP client creates a DHCPDISCOVER packet, the special field in the message, called RELAY AGENT, is set to zero. When the Relay Agent intercepts a DHCPDISCOVER message, first it looks at the RELAY AGENT field.

If the RELAY AGENT field is zero (that is, the DHCP message comes directly from the DHCP client), the Relay Agent will write its own IP address in this field and then forward the packet to the next hop and increase the hop count.

If the RELAY AGENT field is not zero (that is, the DHCP message comes from another Relay Agent), the Relay Agent just forwards the packet to the next hop and increments the hop count without modifying the RELAY AGENT field.

This process is repeated until the packet reaches the DHCP server.

When the DHCP server sends the DHCPOFFER, it sends the message back to the first Relay Agent, and the Relay Agent forwards it to the originator client. After the client receives an IP address, the communication is direct between the DHCP server and the DHCP client.

6.4 DHCP implementation on the System i

The following sections describe the System i DHCP server, BOOTP/DHCP Relay Agent, migration from BOOTP to DHCP, the backup of DHCP configuration files, and other useful topics associated with DHCP.

6.4.1 DHCP software prerequisites

The native DHCP support on System i running V5R2 or later requires the following software products:

- 5722-SS1 option 3: Extended Base Directory Support.
- 5722-XE1: iSeries Access for Windows. Specifically, you will use iSeries Navigator to configure your DHCP server on System i.

6.4.2 DHCP installation

To install the DHCP support on your System i server running OS/400 or i5/OS, you must install 5722-SS1 option 3.

The configuration of the DHCP server running on System i is performed through iSeries Navigator. Therefore, iSeries Navigator must be installed on your administrator workstation.

The installation program performs the following actions:

- It creates the IFS subdirectory /QIBM/UserData/OS400/DHCP/ .
- It sets up the IFS files required for DHCP in directory /QIBM/UserData/OS400/DHCP. If any file already exists, it remains *as is*.

Tip: Perform the following steps to reset an existing configuration to start over:

1. Delete the IFS file dhcpsd.cfg in /QIBM/UserData/OS400/DHCP/ .
2. From an i5/OS command line, enter the command:

```
CALL QSYSDIR/QTODDINS
```

You can perform the steps previously mentioned when you suspect that some DHCP files are corrupted. As earlier stated, reinstalling 5722-SS1 option 3 does not replace the existing files.

After the installation, you can proceed with the DHCP configuration using iSeries Navigator.

Figure 6-10 illustrates these steps.

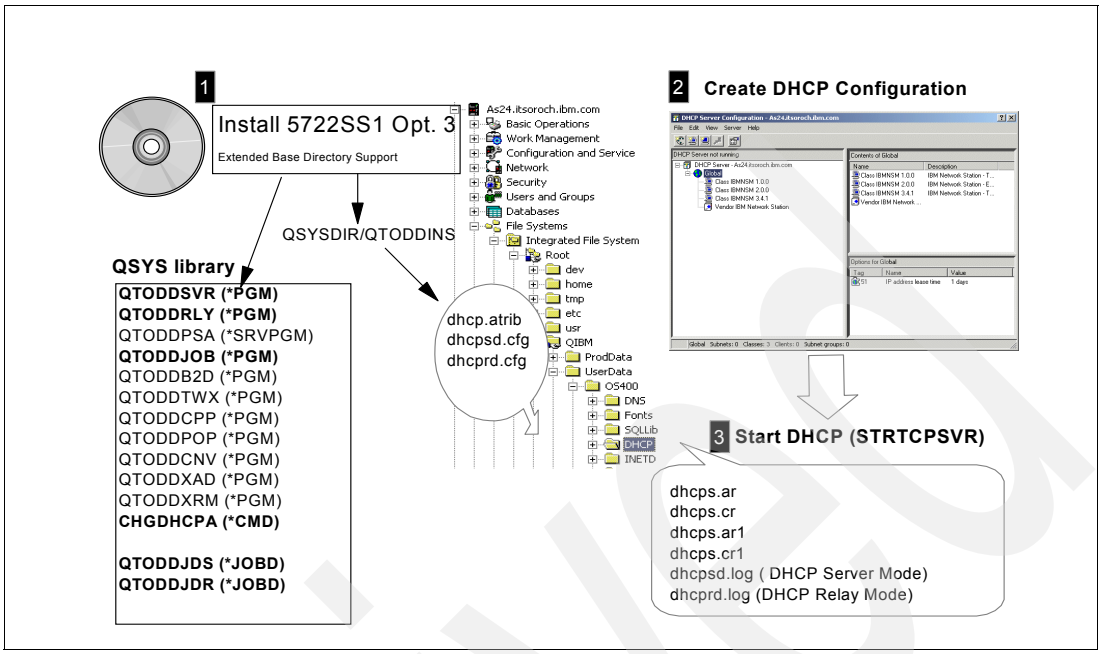


Figure 6-10 System i DHCP server support installation and configuration overview

6.4.3 DHCP server jobs

The DHCP server jobs are:

xxxxxx/QTCP/QTODDHCPS This is the job that starts when you run the regular DHCP transaction processing server (that is, when the DHCP Server Mode is *SERVER). The DHCP uses the well-known port 67. The program QSYS/QTODDSVR runs within this job. This job starts with job description QSYS/QTODDJDS.

xxxxxx/QTCP/QTODDHCPR This is the job that starts when you run the BOOTP/DHCP Relay Agent (that is, when the DHCP Server Mode is *RELAY). The BOOTP/DHCP Relay Agent uses the well-known port 67. The program QSYS/QTODDRLY runs within this job. This job starts with job description QSYS/QTODDJDR.

6.4.4 DHCP configuration files

The files used by the DHCP server, which are located in /QIBM/UserData/OS400/DHCP, are:

dhcpsd.cfg DHCP server configuration file. When the DHCP server starts, it loads the configuration from this file.

dhcprd.cfg BOOTP/DHCP Relay Agent configuration file. When the BOOTP/DHCP Relay Agent starts, it loads the configuration from this file.

dhcps.ar DHCP server non-volatile address record. This file contains up-to-the minute, actual address allocation from the address pools that the DHCP server administers when running in regular DHCP server mode.

dhcps.cr DHCP server non-volatile client records. This file contains up-to-the minute data on the actual clients that this DHCP server is servicing when running in regular DHCP server mode.

dhcps.ar1 DHCP server backup of non-volatile address records. The DHCP server takes an hourly backup of dhcps.ar, the non-volatile address record file.

dhcps.cr1 DHCP backup of server non-volatile client records. The DHCP server takes an hourly backup of dhcps.cr, the non-volatile client records file.

dhcp.attrib DHCP attributes file. The DHCP server stores the current value of the CHGDHCPA command parameters, with the exception of the AUTOSTART parameter.

6.4.5 DHCP server log file

The DHCP server logs all of the actions it performs in a file in the IFS directory /QIBM/UserData/OS400/DHCP/. By default, the log file is named dhcpsd.log for the DHCP server. You can enable logging through a configuration option in iSeries Navigator and you can configure this file to roll into multiple files based on the maximum size. Also, you can select the types of actions that you want to be logged.

To enable DHCP server logging, perform the following steps:

1. From iSeries Navigator, expand **Network** → **Servers** then click **TCP/IP** to see a list of all TCP/IP servers that are available on your System i in the right-side pane.
2. Right-click the **DHCP** server and select **Configuration** from the context menu.
3. Right-click your DHCP server and select **Properties** from the context menu.
4. The Server Properties window is displayed (Figure 6-11). Select the **Logging** tab.

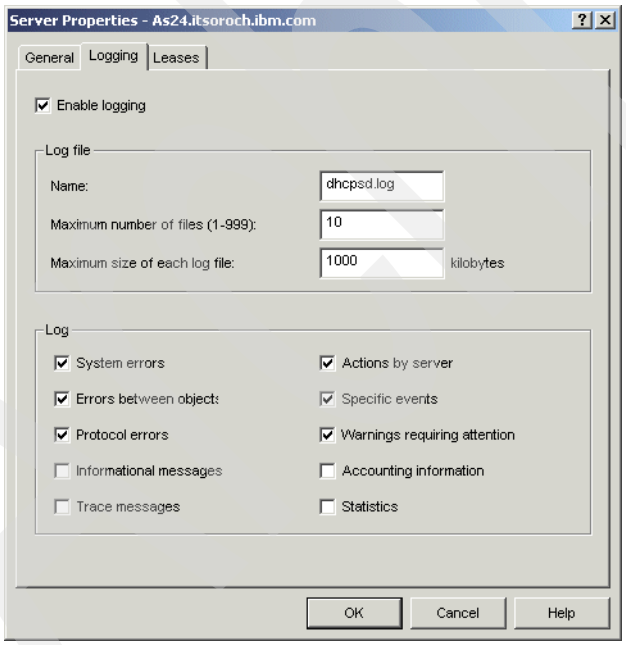


Figure 6-11 DHCP Server Properties: Logging tab

5. In the Logging tab, select the **Enable logging** check box. You can specify the log file name, the maximum size of the log file, and the number of log files the system should keep. Also, you can specify the actions that will be logged. Click **OK** to save the configuration.

6.4.6 BOOTP/DHCP Relay Agent log file

The BOOTP/DHCP Relay Agent logs all of the actions it performed in a file in the IFS directory /QIBM/UserData/OS400/DHCP/. By default, the log file is named dhcprd.log for the DHCP server. You can enable logging through a configuration option in iSeries Navigator, and you can configure this file to roll into multiple files based on the maximum size. Also, you can select the types of actions that you want to have logged.

To enable BOOTP/DHCP Relay Agent logging, perform the following steps:

1. From iSeries Navigator expand **Network** → **Servers** then click **TCP/IP** to see a list of all available TCP/IP servers on your System i in the pane on the right.
2. Right-click the BOOTP/DHCP Relay Agent server, and select **Configuration** from the context menu.
3. Select the **Logging** tab as shown in Figure 6-12.

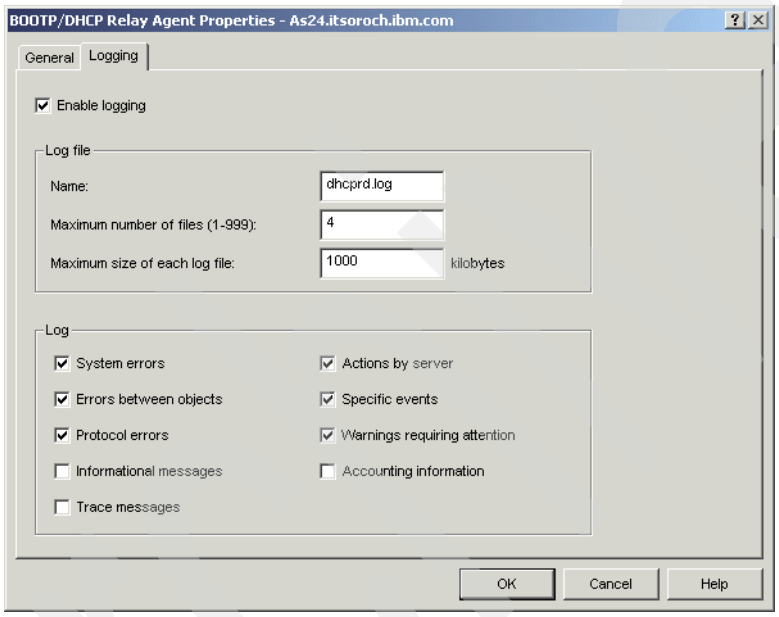


Figure 6-12 BOOTP/DHCP Relay Agent Properties: Logging tab

4. Select the **Enable logging** check box. You can specify the log file name, the maximum size of the log file, and the number of log files the system should keep. Also, you can specify the actions that will be logged. Click **OK** to save the configuration.

6.4.7 DHCP wide area network (WAN) client support

A new feature of the DHCP server on System i introduced in V5R1 is the DHCP WAN Client support. The DHCP WAN Client acts as a DHCP client, requesting IP addresses from the DHCP server address pool on behalf of an application that called it.

An example of an application that can use the DHCP WAN Client is a Point-to-Point Protocol (PPP) Receiver profile.

PPP provides a standard method for transporting multiprotocol datagrams over point-to-point links. PPP hosts can obtain an IP address by connecting to a System i for which a PPP Receiver profile has been defined.

Starting with OS/400 V5R1, you can define a PPP Receiver connection profile that will assign an IP address to clients using the DHCP server address pool. This gives the administrator greater flexibility in managing a precious resource: the IP addresses.

Note: Before OS/400 V5R1 and i5/OS, the system administrator defined within the PPP profile a single IP address or a range of IP addresses to be assigned to PPP hosts. Similarly, LAN-connected hosts could obtain an IP address from a System i server via DHCP.

Thus, an administrator had to separately define address pools onto serve both PPP and LAN connected hosts. There was no mechanism to share free addresses from either pool.

For a how-to scenario, see 17.5, “Assigning an IP address to PPP client from DHCP server” on page 610.

6.4.8 DHCP support of a Dynamic DNS

A new feature of the System i DHCP server that was introduced in OS/400 V5R1 and i5/OS is to support the automatic update of the DNS server as new DHCP clients are added and removed from the TCP/IP network.

DHCP enables dynamic allocation of IP addresses and TCP/IP configuration information to IP hosts. A DNS server maps IP addresses to host names and host names to IP addresses. Dynamic DNS enables authorized clients (DHCP servers are one example of authorized clients) to dynamically update host-name-to-IP-address (and IP-address-to-host-name) mappings.

Starting with OS/400 V5R1 and i5/OS, Dynamic Domain Name System (DDNS) on the System i provides an API (QTOBUPDT - Update DNS) that is used by the DHCP server to update the DNS records.

When the DHCP server issues an IP address to a client, it can perform the following operations using the DDNS support:

- Add a PTR resource record (mapping from IP address to a Fully Qualified Domain Name or FQDN) to DNS configuration
- Add an A resource record (mapping from FQDN to IP address) to DNS configuration

When the DHCP receives a DHCPRELEASE from a DHCP client, or when a lease expires, the DHCP server can perform the following operations using the DDNS support:

- Remove the corresponding PTR record from the DNS configuration.
- Remove the corresponding A record from the DNS configuration.

Informing the DHCP server how to perform these updates is done through a series of new keywords in the DHCP configuration file:

- updateDNSA
- updateDNSP
- releaseDNSP
- releaseDNSA
- proxyARec
- appendDomainName

To avoid delays in serving clients, the System i DHCP server starts a separate thread to handle dynamic updates.

To activate the Dynamic DNS support on the System i DHCP server:

1. From the DHCP Server Configuration window, right-click the element for which you want to activate the Dynamic DNS support and select **Properties** from the context menu. The Properties window is displayed.

   Tip: You can activate the Dynamic DNS support at a global level (for all elements in the DHCP configuration) or for each element in the DHCP configuration (subnet, class, client).

2. Select the **Dynamic DNS** tab (Figure 6-13 on page 117). If you want the DHCP server to update the DNS with PTR and A records, select **Update A** and **PTR records** in Update client records. If you want the DHCP server to update only the PTR records, select only **Update PTR** records in Update client records. Select **Inherited** if you want the Update client record value to be inherited from the global level.

   Tip: The automatic update of the PTR (or PTR and A) records by the DHCP server is worth some explanation.

   By convention, in a TCP/IP network the host client owns its name and the DHCP server owns the IP address that is currently leased to the host client.

   So, by convention, the DHCP server needs only to update the PTR record in the DNS as this is the reverse mapping of IP-address-to-name. The host client, by convention, owns the responsibility of updating the A record in the DNS, as this is the mapping of name-to-IP-address.

   But, it is very common for the DHCP server to take on both the update of the A and PTR records. The configuration here in the System i DHCP server should match the configuration and behavior of your host clients.

   The *Verify client identifier before updating A records* check box specifies that the server verifies the client ID before performing the A record update or delete.

   Tip: Two situations that require an administrative decision arise out of a dynamic DNS environment:

   1. Name collisions (host name conflicts are possible).
   2. Multiple DHCP servers can update the same DNS zone (that is, the client switches subnets but remains within the zone).

   Ultimately, you have two options:

   1. An existing entry is protected and is not overwritten (first update wins). The same client is allowed to update its own record.
   2. Even if an entry exists, you can update DNS records with the latest information (last update wins).

   This select box is used to configure this policy. The System i accomplishes this by writing a corresponding TXT (documenting) record with each A and PTR record. The TXT record contains the MAC address of the client.

   If you want the DHCP server to append the domain name to the host name, select **Yes** in Append domain name to host name. Select **No** in Append domain name to host name if you do not want the DHCP server to append the domain name to host name. Select

Inherited if you want the Append domain name to host name value to be inherited from the global level.

Tip: If Yes is selected in Append domain name to host name, option 15 (domain name) must be configured.

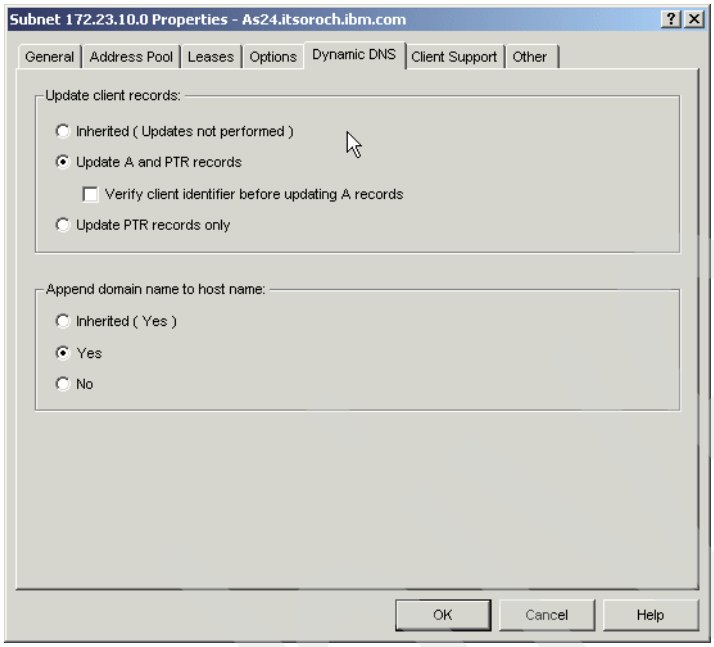


Figure 6-13 Activating the DDNS support on DHCP server

3. Click **OK** to save the configuration.

For more information about Dynamic DNS, refer to Chapter 8, “Dynamic Domain Name System (Dynamic DNS)” on page 133.

For some scenarios in which the DHCP server dynamically updates the DNS, see:

- 16.1, “Single DDNS and DHCP server on the same server” on page 368
- 16.2, “Single DDNS and DHCP servers without secured updates” on page 402
- 16.3, “Single DDNS and DHCP servers with secured updates” on page 438

6.4.9 Configuring the DHCP server through iSeries Navigator

The configuration of DHCP server on System i can be performed only through iSeries Navigator.

1. Start the DHCP server configuration from iSeries Navigator by selecting your System i. You may be asked to enter your user ID and password.

2. Expand **Network** → **Servers** and then click **TCP/IP**. A list of all of your TCP/IP servers will be displayed in the pane on the right as shown in Figure 6-14.

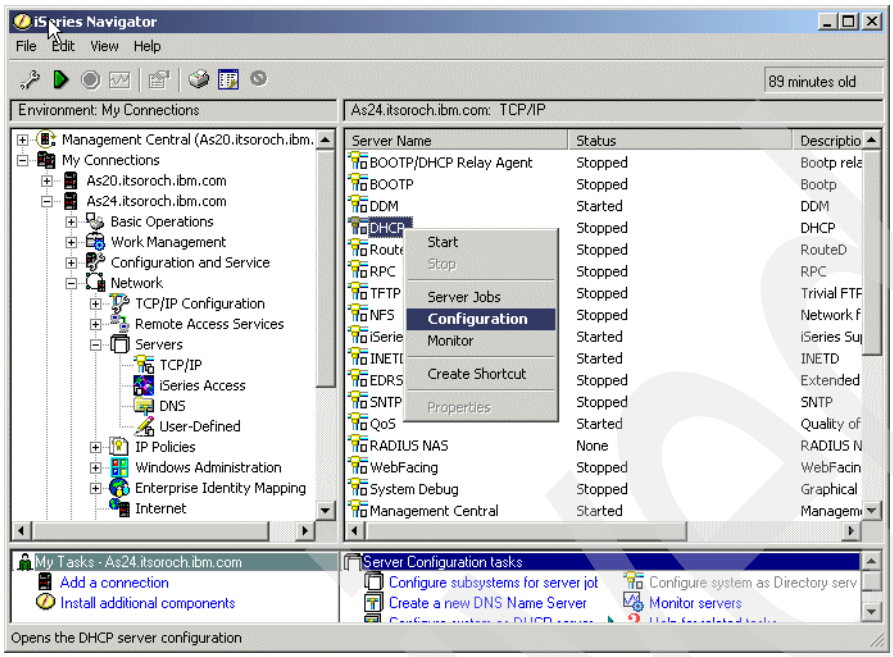


Figure 6-14 iSeries Navigator: starting the DHCP server configuration interface

3. Right-click **DHCP** and select **Configuration** from the context menu. The DHCP Server Configuration pane is displayed as shown in Figure 6-15.

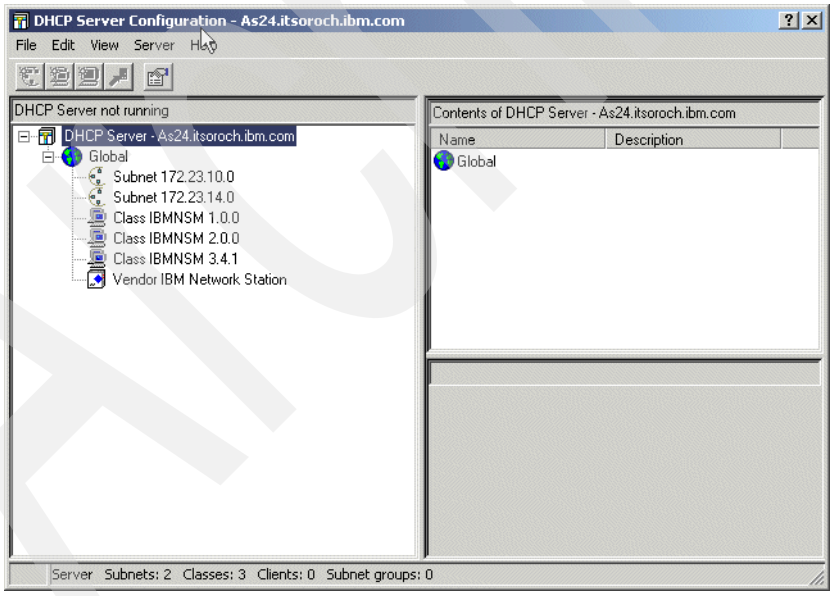


Figure 6-15 DHCP Server Configuration interface

You can configure your DHCP server from this interface.

Tip: One of the strengths of the GUI configuration for the System i DHCP server is the configuration parameter hierarchy.

Depending on your network needs, you can specify a hierarchy of default configuration parameters (options) that the DHCP server provides to a client in response to a client request for an IP address. Possible options are:

- Globally
- Subnet
- Class
- Vendor
- Client

Each level inherits the configuration parameters of the higher levels. You can define configuration parameters for the various levels based on the network location of the client, type of equipment vendor used for the client, or the user characteristics of the client. You can exclude IP addresses at either the global or the subnet levels. You may find it helpful to refer to a network diagram showing the relationship between your subnets, vendors, classes, and individual clients and the DHCP configuration parameters (options) you need.

For a very good description of this configuration feature, click **Help** → **Help Topics** from the DHCP Server Configuration window (Figure 6-15). In the Help window, select **Help Topics** → **Find,** and type `configuration parameter hierarchy`.

6.4.10 Change DHCP Attributes Command (CHGDHCPA)

Use the Change DHCP Attributes (CHGDHCPA) command to set the AUTOSTART and MODE attributes (Figure 6-16).

```
                      Change DHCP Attributes (CHGDHCPA)

Type choices, press Enter.

Autostart server . . . . . .   *YES          *YES, *NO, *SAME
Mode . . . . . .               *SERVER       *SERVER, *RELAY, *SAME




                                                                    Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

Figure 6-16 Change DHCP Attributes panel

The AUTOSTART attribute determines whether the DHCP server starts automatically when TCP/IP is started. This attribute is ignored by the Start TCP/IP Server (STRTCPSVR) command.

The MODE attribute determines the DHCP server behavior.

Set the MODE attribute to *SERVER if you want the DHCP server to automatically assign reusable IP addresses to DHCP clients in response to DHCP requests.

Set the MODE attribute to *RELAY if you want the DHCP server to function only as a BOOTP/DHCP Relay Agent. A BOOTP/DHCP Relay Agent forwards BOOTP or DHCP

packets from hosts to BOOTP or DHCP servers and from the servers back to the hosts. It performs no BOOTP or DHCP server functions.

The MODE attribute is saved in file /QIBM/UserData/OS400/DHCP/dhcp.attrib.

You can also set the AUTOSTART attribute from iSeries Navigator. Open the DHCP Server Configuration window, click **DHCP Server**, and select **Properties** from the context menu (Figure 6-17).

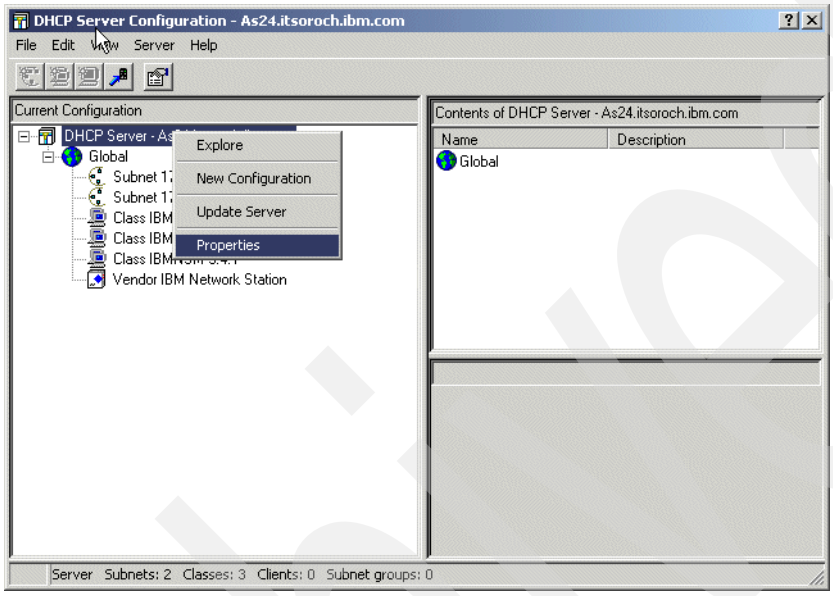


Figure 6-17 DHCP Server Configuration: server properties

The window shown in Figure 6-18 is displayed. In the **General** tab, check the **Start when TCP/IP is started** check box to set the AUTOSTART attribute to *YES.

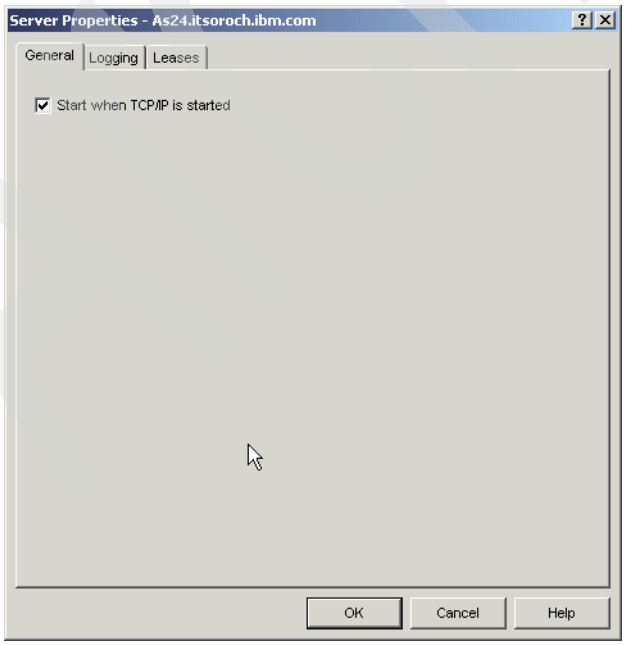


Figure 6-18 DHCP Server Properties window: General tab

6.4.11 Starting and stopping the DHCP server

You can start the DHCP server from a command line or using iSeries Navigator:

- To start the DHCP server using iSeries Navigator, perform the following:
  a. From iSeries Navigator select your System i. You may be asked to enter your user ID and password.
  b. Expand **Network** → **Servers** and then click **TCP/IP.** A list of all of your TCP/IP servers will be displayed in the pane on the right.
  c. Right-click **DHCP** and select **Start** from the context menu (Figure 6-19).

  Tip: If the server is already started, you can stop the server using the same context menu.

- To start the DHCP server from a command line, run the Start TCP/IP Server (STRTCPSVR) command:

```
STRTCPSVR *DHCP
```

Tip: If the server is already started, you can stop the server using the End TCP/IP Server (ENDTCPSVR) command:

```
ENDTCPSVR *DHCP
```

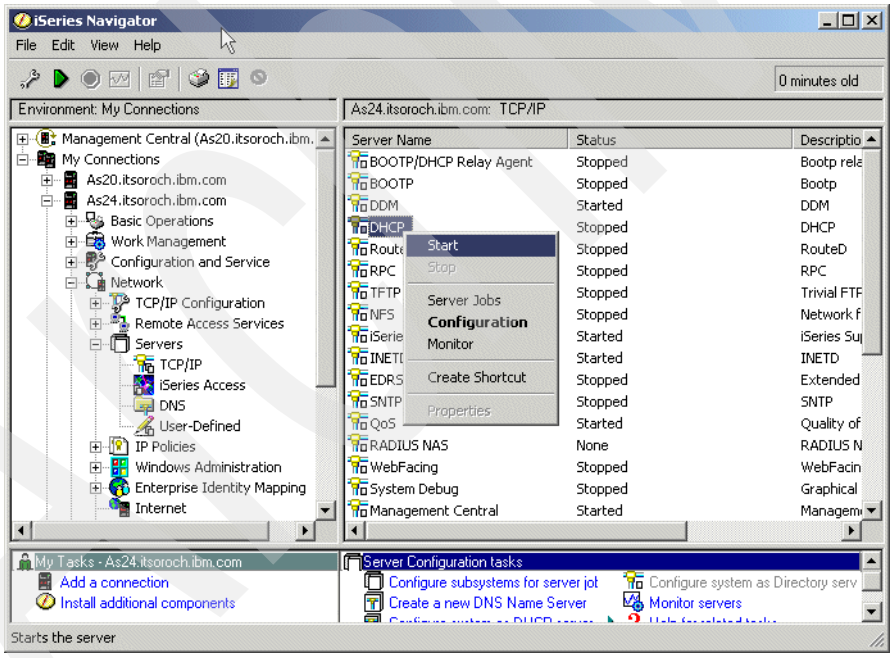


Figure 6-19 Starting the DHCP server through iSeries Navigator

6.4.12 BOOTP to DHCP migration program

iSeries Navigator uses the migration program QSYS/QTODDB2D to migrate a BOOTP configuration to one that can be used by the DHCP server. If the program detects that there is a BOOTP table in the system, it gives you options through the iSeries Navigator to migrate the BOOTP table to a DHCP server configuration (Figure 6-20). If you choose Yes, all of the clients listed in the BOOTP table will be included in the DHCP configuration file.

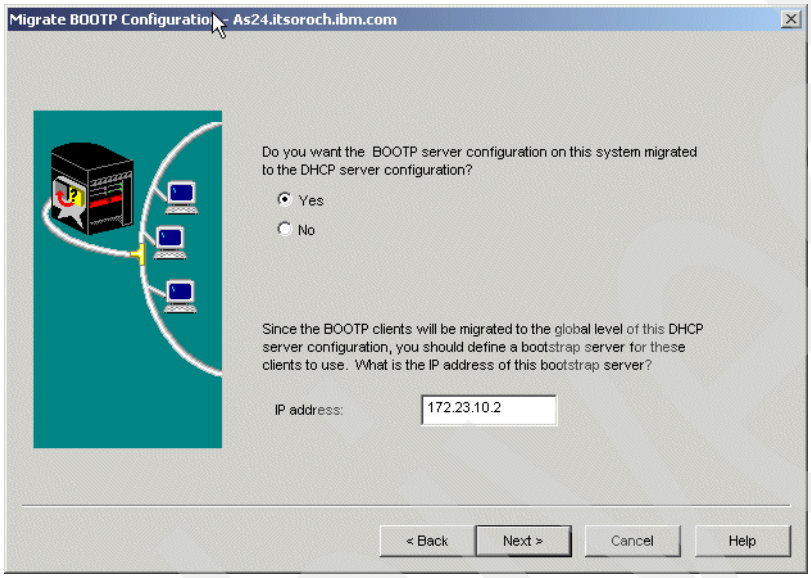


Figure 6-20 Migrate BOOTP Configuration to DHCP Configuration window

6.4.13 DHCP Monitor

The iSeries DHCP Monitor enables you to view and administer IP addresses that are managed by the System i DHCP server.

The DHCP Monitor is part of iSeries Navigator and was introduced in V5R1. This graphical interface enables you to view which IP addresses are leased, how long they have been leased, and when they will be available to lease again.

You can also use the DHCP Server Administration tool to reclaim IP addresses that are no longer being used. If the DHCP address pool has been exhausted, you can look through the active lease information to determine whether there are any leases that you may want to delete, making the IP address available to other clients. For example, you may have a client that is no longer on the network, but still has an active IP address lease. You can delete the active IP address lease for this client.

To start the DHCP Monitor, perform the following steps:

1. To start the DHCP Monitor from iSeries Navigator, select your System i. You may be asked to enter your user ID and password.

2. Expand **Network** → **Servers** and then click **TCP/IP**. A list of all of your TCP/IP servers will be displayed in the pane on the right as shown in Figure 6-21.

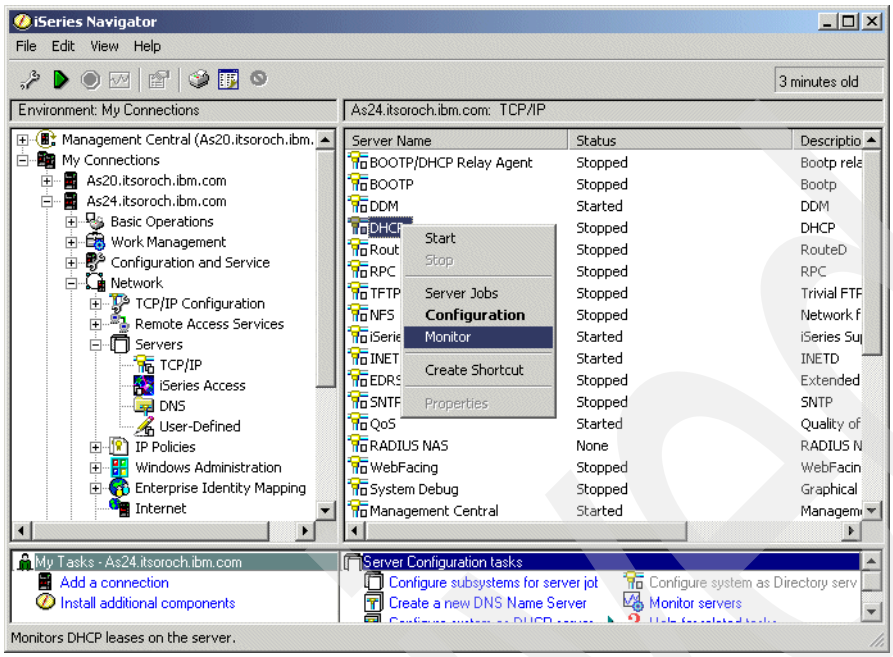


Figure 6-21 iSeries Navigator: starting the DHCP Monitor

3. Right-click **DHCP** and select **Monitor** from the context menu. The DHCP Monitor window is displayed (Figure 6-22). We expanded the tree on the left and selected our subnet.

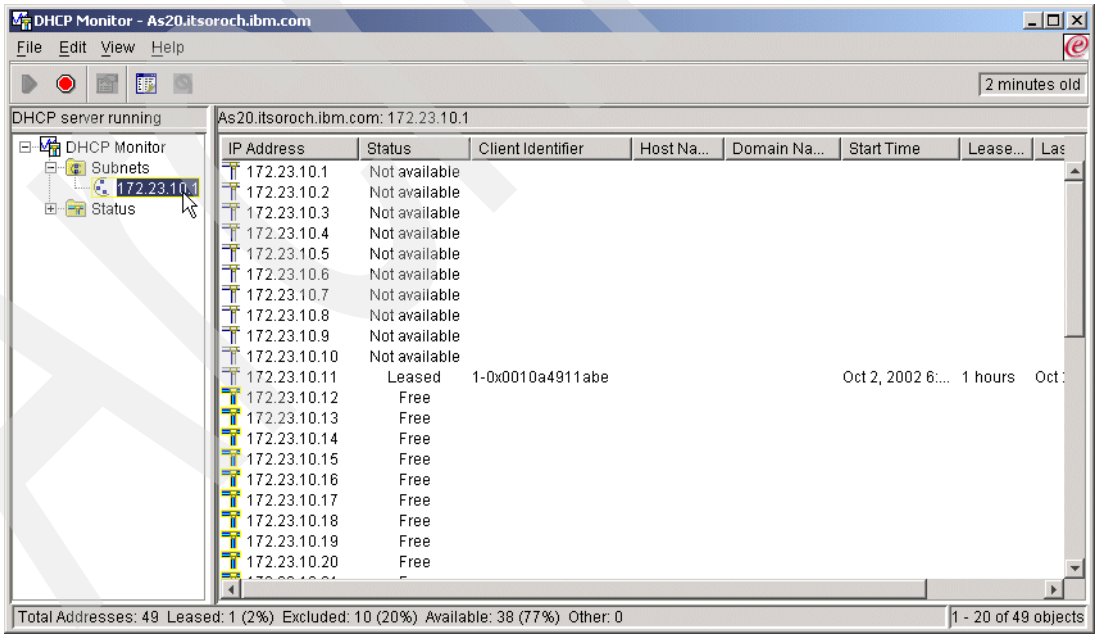


Figure 6-22 DHCP Monitor showing excluded IP addresses, one active lease, and many free

From this interface you can monitor the activity of the System i DHCP server.

6.4.14 DHCP server exit programs

The System i DHCP server assigns and releases TCP/IP addresses to and from client hosts in a network. Exit points have been provided so that user-written programs are called from the running DHCP server when specific events take place. They allow for customer-supplied security validation of incoming client requests, as well as for notification when an IP address is assigned or released.

The exist programs are run as part of the DHCP server job.

The DHCP-supplied exit points are:

QIBM_QTOD_DHCP_ABND The DHCP Address Binding Notify exit point allows for notification each time the DHCP server assigns an IP address to a specific host.

QIBM_QTOD_DHCP_ARLS The DHCP Address Release Notify exit point allows for notification each time the DHCP server releases an IP address from its specific client host assignment binding.

QIBM_QTOD_DHCP_REQ The DHCP Request Packet Validation exit point provides additional control for restricting which incoming DHCP and BOOTP message request packets from client hosts are processed and which are rejected by their DHCP server.

For more information about these and other exit points, refer to documentation in Information Center found at:

http://publib.boulder.ibm.com/iseries/

6.4.15 DHCP server backup and recovery considerations

You should back up the following files on a regular basis and as part of your normal backup procedures:

- If you run the DHCP Server, back up the following files located in directory /QIBM/UserData/OS400/DHCP:
  - dhcpsd.cfg
  - dhcps.ar
  - dhcps.cr
  - dhcps.ar1
  - dhcps.cr1
- If you run the BOOTP/DHCP Relay Agent, back up the following file located in /QIBM/UserData/OS400/DHCP:
  - dhcprd.cfg
- Back up /QIBM/UserData/OS400/DHCP/dhcp.attrib to back up the general DHCP attributes.

Optionally, you can save all of the files in the IFS directory /QIBM/UserData/OS400/DHCP. In this case, the backup includes other files that exist in this directory, such as log files. The other files are not necessary for recovery, but this might be an easier approach.

Tip: Shut down the servers before you perform these backups. This prevents saving the files while one or more files are in the middle of an update.

To back up the files, use the Save Object (SAV) command. In this way, the ownership, CCSID, and the authorizations are restored when you restore the files.

To recover the DHCP files saved with the SAV command, use following guidelines:

- If you saved the files using the SAV command, use the Restore Object (RST) command to restore the files.
- You must shut down the server prior to restoring any file.

The following backups take place automatically during the normal operation of the DHCP server:

- After every transaction processed, the server stores its current state in the following non-volatile files:

  ```
  /QIBM/UserData/OS400/DHCP/dhcps.ar
  /QIBM/UserData/OS400/DHCP/dhcps.cr
  ```

  Hourly backups of the non-volatile file previous mentioned are taken in:

  ```
  /QIBM/UserData/OS400/DHCP/dhcps.ar1
  /QIBM/UserData/OS400/DHCP/dhcps.cr1
  ```

The following run-time recoveries take place automatically:

- If the DHCP server is shut down intentionally or terminates abnormally, you must restart the server. You must also have it re-initialize itself to its state just after it processed the last successful transaction. It does this by reading the /QIBM/UserData/OS400/DHCP/dhcps.ar and /QIBM/UserData/OS400/DHCP/dhcps.cr files.
- If the previous re-initialization fails due to the corruption of one or both of the primary non-volatile files, the DHCP server automatically deletes them. It then renames the hourly backup versions to the primary version file names and tries again. It sends messages to the log to signal the event.
- If both of your re-initialization attempts fail, you must recover using your own backups.

7

Routing Information Protocol Version 2 (RIPv2)

The IETF recognizes two versions of RIP:

RIP Version 1 (RIPv1) This protocol is described in RFC 1058.

RIP Version 2 (RIPv2) RIPv2 is also a distance vector protocol designed for use within a System i platform. It was developed to address the limitations observed in RIPv1. RIPv2 is described in RFC 1723. The standard was published in late 1994.

In practice, the term RIP refers to RIPv1. Whenever the reader encounters the term RIP in TCP/IP literature, it is safe to assume that the reference is to RIP Version 1, unless otherwise stated. This same convention is used in this document. However, when the two versions are being compared, the term RIPv1 is used to avoid confusion.

7.1 Routing Information Protocol Version 1 (RIPv1)

RIP is an example of an interior gateway protocol designed for use within small autonomous systems. RIP is based on the Xerox XNS routing protocol. Early implementations of RIP were readily accepted because the code was incorporated into the Berkeley Software Distribution (BSD) UNIX-based operating system. RIP is a distance vector protocol. In mid-1988, the IETF issued RFC 1058, which describes the standard operations of a RIP system. However, the RFC was issued after many RIP implementations had been completed. For this reason, some RIP systems do not support the entire set of enhancements to the basic distance vector algorithm (for example, poison reverse and triggered updates).

7.1.1 RIPv1 packet types

The RIP protocol specifies two packet types. These packets may be sent by any device running the RIP protocol:

- Request packets: A request packet queries neighboring RIP devices to obtain their distance vector table. The request indicates whether the neighbor should return either a specific subset or the entire contents of the table.
- Response packets: A response packet is sent by a device to advertise the information maintained in its local distance vector table. The table is sent during the following situations:
  - The table is automatically sent every 30 seconds.
  - The table is sent as a response to a request packet generated by another RIP node.
  - If triggered updates are supported, the table is sent when there is a change to the local distance vector table.

When a response packet is received by a device, the information contained in the update is compared against the local distance vector table. If the update contains a lower-cost route to a destination, the table is updated to reflect the new path.

7.1.2 RIPv1 packet format

RIP uses a specific packet format to share information about the distances to known network destinations. RIP packets are transmitted using UDP datagrams. RIP sends and receives datagrams using UDP port 520. RIP datagrams have a maximum size of 512 octets. Updates larger than this size must be advertised in multiple datagrams. In LAN environments, RIP datagrams are sent using the MAC all-stations broadcast address and an IP network broadcast address. In point-to-point or non-broadcast environments, datagrams are specifically addressed to the destination device. The RIP packet format is shown in Figure 7-1.

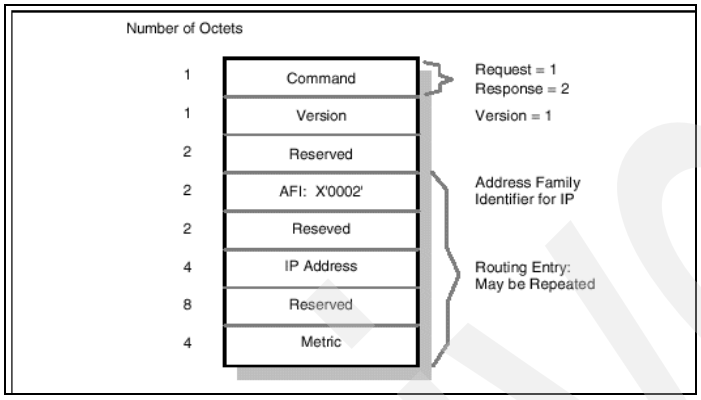


Figure 7-1 RIPv1 packet format

A 512-byte packet size allows a maximum of 25 routing entries to be included in a single RIP advertisement.

7.1.3 RIPv1 limitations

There are a number of limitations observed in RIP environments:

- Path cost limits: The resolution to the counting to infinity problem enforces a maximum cost for a network path. This places an upper limit on the maximum network diameter. Networks requiring paths greater than 15 hops must use an alternate routing protocol.
- Network-intensive table updates: Periodic broadcasting of the distance vector table can result in increased utilization of network resources. This can be a concern in reduced-capacity segments.
- Relatively slow convergence: RIP, like other distance vector protocols, is relatively slow to converge. The algorithms rely on timers to initiate routing table advertisements.
- No support for variable-length subnet masking: Route advertisements in a RIP environment do not include subnet masking information. This makes it impossible for RIP networks to deploy variable length subnet masks.

7.2 Routing Information Protocol version 2 (RIPv2)

RIPv2 is similar to RIPv1. It was developed to extend RIPv1 functionality in small networks. RIPv2 provides these additional benefits not available in RIPv1:

- Support for CIDR and VLSM: RIPv2 supports supernetting (that is, CIDR) and variable-length subnet masking. This support was the major reason that the new standard was developed. This enhancement positions the standard to accommodate a degree of addressing complexity not supported in RIPv1.

- Support for multicasting: RIPv2 supports the use of multicasting rather than the simple broadcasting of routing announcements. This reduces the processing load on hosts not listening for RIPv2 messages. To ensure interoperability with RIPv1 environments, this option is configured on each network interface.
- Support for authentication: RIPv2 supports authentication of any node transmitting route advertisements. This prevents fraudulent sources from corrupting the routing table.
- Support for RIPv1: RIPv2 is fully interoperable with RIPv1. This provides backward compatibility between the two standards.

As noted in the RIPv1 section, one notable shortcoming in the RIPv1 standard is the implementation of the metric field. RIPv1 specifies the metric as a value between 0 and 16. To ensure compatibility with RIPv1 networks, RIPv2 preserves this definition. In both standards, network paths with a hop-count greater than 15 are interpreted as unreachable.

7.2.1 RIPv2 packet format

The original RIPv1 specification was designed to support future enhancements. The RIPv2 standard was able to capitalize on this feature. RIPv2 developers noted that a RIPv1 packet already contains a version field and that 50% of the octets are unused. Figure 7-2 illustrates the contents of a RIPv2 packet. The packet is shown with authentication information. The first entry in the update contains either a routing entry or an authentication entry. If the first entry is an authentication entry, 24 additional routing entries can be included in the message. If there is no authentication information, 25 routing entries can be provided.

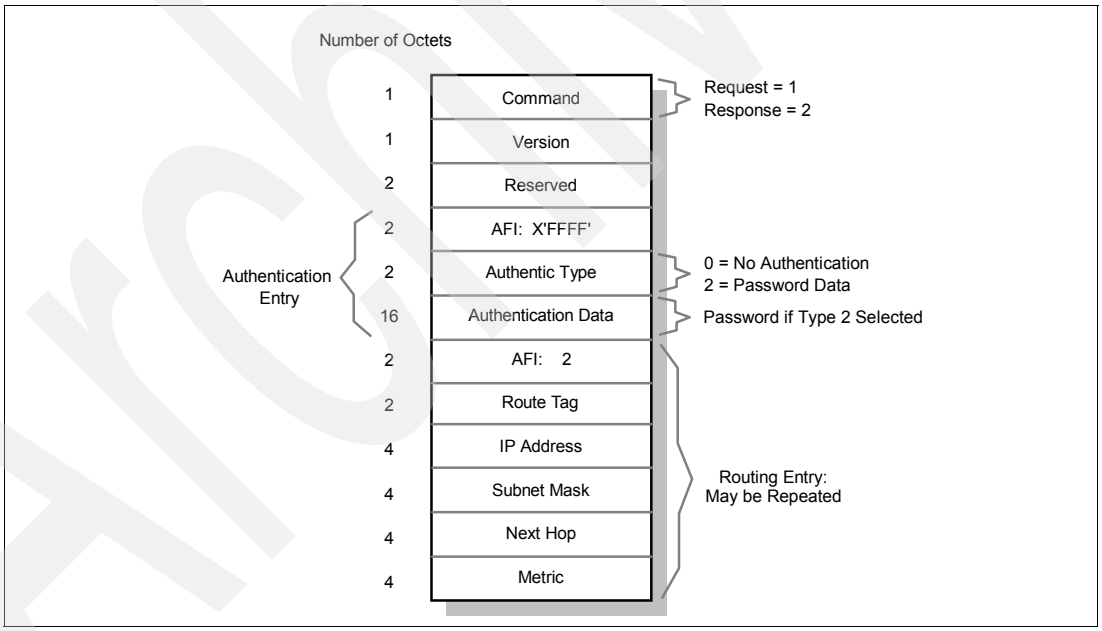


Figure 7-2 RIPv2 packet format

The use of the command field, IP address field, and metric field in a RIPv2 message is identical to the use in a RIPv1 message. Otherwise, the changes implemented in RIPv2 packets include:

- Version: The value contained in this field must be 2. This instructs RIPv1 routers to ignore any information contained in the previously unused fields.
- AFI (Address Family): A value of x'0002' indicates that the address contained in the network address field is an IP address. A value of x'FFFF' indicates an authentication entry.

- Authentication Type: This field defines the remaining 16 bytes of the authentication entry. A value of 0 indicates no authentication. A value of 2 indicates that the authentication data field contains password data.
- Authentication Data: This field contains a 16-byte password.
- Route Tag: This field is intended to differentiate between internal and external routes. Internal routes are learned via RIPv2 within the same network or AS.
- Subnet Mask: This field contains the subnet mask of the referenced network.
- Next Hop: This field contains a recommendation about the next hop the router should use when sending datagrams to the referenced network.

7.2.2 RIPv2 limitations

RIPv2 was developed to address many of the limitations observed in RIPv1. However, the path cost limits and slow convergence inherent in RIPv1 networks are also concerns in RIPv2 environments. In addition to these concerns, there are limitations to the RIPv2 authentication process. The RIPv2 (and RIPv1) standard does not encrypt the authentication password. It is transmitted in clear text. This makes the network vulnerable to attack by anyone with direct physical access to the environment.

8

Dynamic Domain Name System (Dynamic DNS)

If you are a network administrator in your company, you may know the difficulty of keeping your DNS server maintained. A static DNS needs a high maintenance after every IP address or host name change; a network administrator has to update the DNS records manually. Is there a solution so that the DNS record can be updated automatically?

Thinking of a mobile networking environment, each person carries a mobile computer and works at different locations. Each time a person hooks a laptop up to the branch office, a new IP address is issued by the DHCP server. If there is a reason for the laptop's host name to remain the same even as the laptop's IP address varies, is there a solution so that the DNS records in the DNS server will be kept updated automatically?

Dynamic DNS is the solution. Dynamic DNS enables A and PTR records to be updated dynamically from the DHCP server. Every time a DHCP server issues an IP address to the client, DHCP server updates an A record, a PTR record, or both with a Dynamic DNS update API. If the DHCP server detects that the leased IP address is expired and is not being used by the client any more, the DHCP server deletes the A records and the PTR record from DDNS server dynamically.

Several how-to scenarios are related to your DDNS-capable System i:

- 16.1, "Single DDNS and DHCP server on the same server" on page 368
- 16.2, "Single DDNS and DHCP servers without secured updates" on page 402
- 16.3, "Single DDNS and DHCP servers with secured updates" on page 438
- 16.4, "Primary DDNS and DHCP servers on one server, secondary server as backup" on page 447
- 16.5, "Primary DDNS and DHCP servers, secondary DNS server Red Hat Linux 7.2" on page 460
- 16.6, "Split DNS: Private and Public DNS with masquerade NAT" on page 466

8.1 i5/OS V5 Dynamic DNS

Prior to the V5 releases, the System i DNS server was based on Berkeley Internet Name Domain (BIND) 4.x. Beginning with V5R1, you have the option to use BIND 8.

This section provides information about the new features available with BIND 8.2.5, the automatic migration of BIND 4.x configurations to BIND 8.2.5, and the system requirements at V5R1 and V5R2 to run a BIND 8 Dynamic DNS server on your System i.

8.1.1 New features

BIND 8.2.3 implements the following new functions in addition to those available in BIND 4.x:

Multiple DNS servers on a single System i

In past releases, only one DNS server could be configured on your System i. You can now configure and run multiple DNS servers (or instances). This enables you to set up a logical division between servers. When you create multiple instances, you must explicitly define the listen-on interface IP addresses for each one. Two DNS instances cannot listen on the same interface.

One practical application of multiple servers is split DNS, where one server is authoritative for an internal network, and a second server is used for external queries. In past releases, two System i servers were required to configure split DNS. Since V5R1, only one System i server is required to configure split DNS. The how-to scenario in 16.6, "Split DNS: Private and Public DNS with masquerade NAT" on page 466, provides a good example of configuring a split DNS on your System i.

Conditional forwarding to fine-tune your forwarding preferences

Conditional forwarding enables you to configure your DNS server to fine-tune your forwarding preferences. You can set a server to forward all queries for which it does not know the answer. You could set forwarding at a global level while adding exceptions to domains for which you want to force normal iterative resolution. Or, you could set normal iterative resolution at the global level and then force forwarding within certain domains.

Secured dynamic updates

DHCP and other authorized sources can send dynamic resource record updates using Transaction Signatures (TSIG) and source IP address authorization. This reduces the need for manual updates of zone data while ensuring that only authorized sources are used for updates.

Dynamic updates are performed by:

- Network client: A host, such as a dynamic client, which typically updates its A record with current IP address information, and may also update its text (TXT) record, such as with the IBM Dynamic IP client.
- DHCP server: A host that updates PTR records with current host name information for the addresses it allocates, and which under certain circumstances updates A records for clients that either cannot or do not update the A records themselves.
- Any application that has access to BIND 8 dynamic updates API. For example, nsupdate for UNIX and QTOBUPDT for OS/400 and i5/OS.

Because security is always important in our scenarios in Chapter 16, “Dynamic DNS scenarios” on page 367, use the secured form of the update.

NOTIFY command

When NOTIFY is turned on, the DNS NOTIFY function is activated whenever zone data is updated on the primary server. The primary server sends out a message indicating that data has changed to all known secondary servers. Secondary servers may then respond with a zone transfer request for updated zone data. This helps improve secondary server support by keeping backup zone data current.

Zone transfers (IXFR and AXFR)

In the past, whenever secondary servers needed to reload zone data, they had to load the entire data set in an all-zone transfer (AXFR). BIND 8 supports a new zone transfer method called incremental zone transfer (IXFR). IXFR is a method other servers use to transfer only changed data rather than the entire zone.

When enabled on the primary server, data changes are assigned a flag indicating that a change has occurred. When a secondary server requests a zone update in an IXFR, the primary server sends just the new data. IXFR is especially useful when a zone is dynamically updated, and it reduces the traffic load by sending smaller amounts of data.

Note: Both the primary server and secondary server must be IXFR-enabled to use this feature.

The Update DNS API (QTOBUPDT)

This enables the caller to send one or more update instructions to a System i Dynamic DNS. The instructions allow for adding or deleting DNS Resource Records (RRs). Optionally, the instructions can include any number of prerequisite conditions that must be true for the actual updates to take place. This API is based on the Berkeley Internet Name Domain (BIND) Version 8.2.x implementation of dynamic DNS updates. Therefore, it also can be used to send update requests to DNS servers running on other operating system platforms that conform to BIND Version 8 update protocols.

5722-SS1 Option 31 (Domain Name System) must be installed to use this API.

8.1.2 i5/OS and System i requirements

If you want to run your DNS server at BIND 8 (available on all currently supported releases of OS/400 and i5/OS), you must install the following 5722-SS1 options on your System i:

- Option 12 - Host Servers.
- Option 31 - Domain Name System.
- Option 33 - Portable Application Solutions Environment (PASE). PASE may require a small license fee (depending upon your release).

Tip: Why does 5722-SS1 Option 31 - Domain Name System require PASE? The answer is that the version of BIND 8.2.5 is a direct port of the same DNS BIND level from AIX. That is, the pSeries® and System i share similar binary code.

8.2 Automatic (yet optional) migration and conversion

The new BIND 8.x requires a migration and conversion when moving from a BIND 4.x base to a BIND 8.x base.

Note: There are no BIND versions between BIND 4.x and 8.x. That means that BIND 5, 6, and 7 never really existed.

The basic BIND 4.x configuration file, called the boot file, must be converted to a completely new format. All BIND 4.x-based OS platforms including OS/400 and i5/OS require this same conversion.

You are not required to migrate to the newer BIND 8.x DNS.

You are not required to migrate to the newer BIND 8.x DNS. That is, the BIND 4.x DNS is still shipped with i5/OS. One reason that you would not migrate to BIND 8.x may be that you cannot run PASE on your AS/400 system. (AS/400e™, iSeries, or System i5 servers are required. See 8.1.2, “i5/OS and System i requirements” on page 136.) Another reason may be that you did not purchase or install 5722-SS1 Option 33 - Portable Application Solutions Environment (PASE) (may require a small license fee). Or, you may be satisfied with your current BIND 4.x support.

You can still use the Change DNS Server Attributes (CHGDNSA) command to change the attributes for a particular BIND 4.x or BIND 8.x instance. The DNS Server instance parameter (DNSSVR) determines which specific DNS server should have its attributes changed. If your system is configured to use the older BIND version 4 server, which does not support multiple instances, only the default server instance value of *ALL is allowed.

The possible values are:

***ALL** Specify this if you want all of the DNS server instances that are currently configured on the system to have their attributes values changed to those specified in the other parameters of this command. If you are still using the BIND 4 DNS server, use *ALL to CHGDNSA for that single server.

Server instance name The specified DNS server instance has the attributes values changed to those in the other parameters of this command. If you provide a server instance name when prompting on this command, the remaining attributes parameters display the actual current values for the specified instance. The other parameters are autostart server (AUTOSTART) and debug level (DBGLVL), which have not changed.

9

Remote Authentication Dial-In User Service (RADIUS)

Remote dial-in to the corporate intranet and to the Internet has made the remote access server a very vital part of today's internetworking services. More and more mobile users require access not only to central-site resources, but to information sources on the Internet. The widespread use of the Internet and the corporate intranet has fueled the growth of remote access services and devices. There is an increasing demand for simplified connection to corporate network resources from mobile computing devices, such as notebook computers or personal digital assistants for e-mail access.

To take a peek at a typical RADIUS network configuration, see Figure 9-2 on page 140.

The emergence of remote access has caused significant development work in the area of security. The AAA (triple A) security model has evolved in the industry to address the issues of remote access security. Authentication, authorization, and accounting answers the questions: who, what, and when respectively. A brief description of each of the three elements in the AAA security model is listed below:

Authentication This is the action of determining who a user (or entity) is. Authentication can take many forms. Traditional authentication utilizes a name and a fixed password. Most computers work this way; however, fixed passwords have limitations, mainly in the area of security. Many modern authentication mechanisms utilize one-time passwords or a challenge-response query. Authentication generally takes place when the user first logs in to a machine or requests a service of it.

Authorization This is the action of determining what a user is allowed to do. Generally, authentication precedes authorization, but again, this is not required. An authorization request may indicate that the user is not authenticated (we do not know who they are). In this case, it is up to the authorization agent to determine whether an unauthenticated user is allowed to access the services in question. In current remote authentication protocols, authorization does not merely provide yes or no answers, but it may also customize the service for the particular user.

Accounting This is typically the third action after authentication and authorization. But again, neither authentication nor authorization are required. Accounting is the action of recording what a user is doing or has done.

In the distributed client/server security database model, several communications servers (or clients) authenticate a dial-in user's identity through a single central database or authentication server. The authentication server stores all information about users, their passwords, and access privileges. Distributed security provides a central location for authentication data that is more secure than scattering the user information about different devices throughout a network. A single authentication server can support hundreds of communications servers, serving up to tens of thousand of users. Communications servers can access an authentication server locally or remotely over WAN connections.

Several remote access vendors and the Internet Engineering Task Force (IETF) have been in the forefront of this remote access security effort and the means whereby such security measures are standardized. Remote Authentication Dial-In User Service (RADIUS) as shown in Figure 9-1 and Terminal Access Controlled Access Control System (TACACS) are two such cooperative ventures that have evolved out of the Internet standardizing body and remote access vendors.

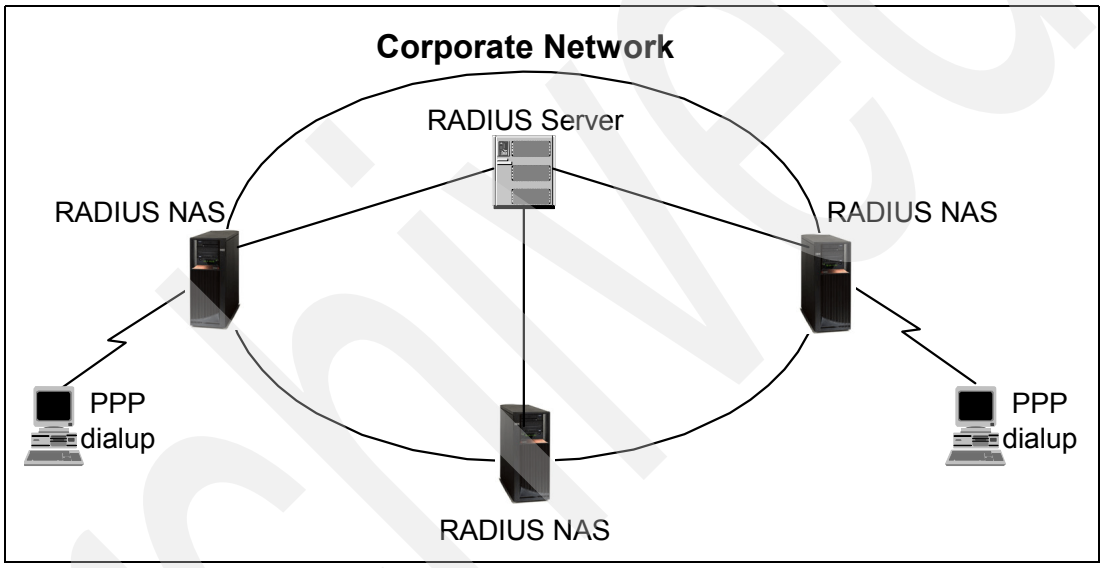


Figure 9-1 Typical RADIUS NAS network scheme

Remote Authentication Dial-In User Service (RADIUS) is a distributed security system developed by Livingston Enterprises. RADIUS was designed based on a previous recommendation from the IETF Network Access Server Working Requirements Group. An IETF Working Group for RADIUS was formed in January 1996 to address the standardization of RADIUS protocol. RADIUS is now an IETF-recognized dial-in security solution: RFC 2058 and RFC 2138.

Although RADIUS and other authentication servers can be set up in a variety of ways, depending on the security scheme of the network they are serving, the basic process for authenticating a user is essentially the same. Using a modem, a remote dial-in user connects to a remote access server (also called the Network Access Server or NAS) with a built-in analog or digital modem. When the modem connection is made, the NAS prompts the user for a name and password. The NAS then creates the so-called authentication request from the supplied data packet, which consists of information identifying the specific NAS device that is sending the authentication request, the port that is being used for the modem connection, and the user name and password.

For protection against eavesdropping by hackers, the NAS, acting as the RADIUS client, encrypts the password before it sends it to the authentication server. If the primary security server cannot be reached, the security client or NAS device can route the request to an

alternate server. When an authentication request is received, the authentication server validates the request and then decrypts the data packet to access the user name and password information. If the user name and password are correct, the server sends an authentication acknowledgment packet. This packet may include additional filters, such as information about the user's network resource requirements and authorization levels. The security server may, for instance, inform the NAS that a user needs TCP/IP or IPX™ using PPP, or that the user needs SLIP to connect to the network. It may include information about the specific network resource that the user is allowed to access.

To circumvent snooping in the network, the security server sends an authentication key, or signature, identifying itself to the security client. When the NAS receives this information, it enables the necessary configuration to allow the user the necessary access rights to network services and resources. If at any point in this log-in process all necessary authentication conditions are not met, the security database server sends an authentication reject message to the NAS device and the user is denied access to the network.

9.1 RADIUS support and implementation on i5/OS

RADIUS on System i (OS/400 V5R1 or later and i5/OS) is implemented as a Network Access Server (NAS) for PPP and L2TP only. To use RADIUS in your network, you will have to implement a RADIUS server somewhere in your corporate network, as shown in Figure 9-2.

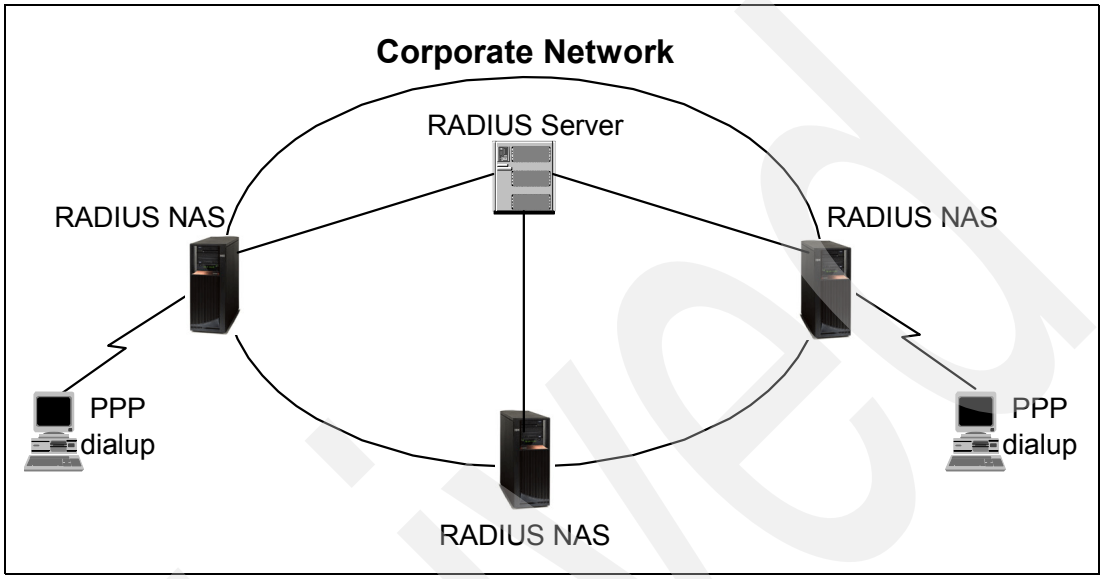


Figure 9-2 RADIUS in a network

RADIUS servers are available for several different operating systems (such as Linux, Windows NT®, Windows 2000, or Windows XP) from a variety of vendors. A very basic setup and install for one such server based on Windows XP is discussed in “Step 1: Setting up a RADIUS server on your network” on page 588.

10

Quality of Service (QoS)

With the increased use of IP-based networks, including the Internet, there has been a large focus on providing necessary network resources to certain applications. That is, it has become better understood that some applications are more *important* than others, thereby demanding preferential treatment throughout an inter-network. Additionally, applications have different demands, such as real-time requirements of low latency, and high bandwidth.

This chapter discusses the topic of traffic prioritization, or Quality of Service (QoS). It explains why QoS may be desirable on an intranet as well as on the Internet, and presents the three main approaches to implementing QoS in TCP/IP networks:

- Integrated Services (IntServ)
- Differentiated Services (DiffServ)
- Inbound admission

10.1 An introduction to QoS

In the Internet and intranets of today, bandwidth is an important subject. More and more people are using the Internet for private and business purposes. The amount of data that is being transmitted through the Internet is increasing exponentially. Multimedia applications, such as IP telephony and videoconferencing systems, need a lot more bandwidth than the applications that were used in the early years of the Internet. Whereas traditional Internet applications such as HTTP, FTP, or Telnet cannot tolerate packet loss but are less sensitive to variable delays, most real-time applications show just the opposite behavior, meaning that they can compensate for a reasonable amount of packet loss, but are usually very critical toward high-variable delays.

This means that without any bandwidth control, the quality of these real-time streams depends on the bandwidth that is currently available. For example, low or unstable bandwidth causes bad quality in real-time transmissions by leading to dropouts and hangs. Even the quality of a transmission using the real-time protocol (RTP) depends on the utilization of the underlying IP delivery service.

Therefore, certain concepts are necessary to guarantee a specific Quality of Service (QoS) for real-time applications in the Internet. A QoS can be described as a set of parameters that define the quality (for example: bandwidth, buffer usage, priority, CPU usage, and so on) of a specific stream of data. The basic IP protocol stack provides only one QoS, which is called *best-effort*. Packets are transmitted from point to point without any guarantee for a special bandwidth or minimum time delay. With the best-effort traffic model, Internet requests are handled with the *first-come, first-served* strategy. This means that all requests have the same priority and are handled one after the other. There is no possibility of making bandwidth reservations for specific connections or raising the priority for special requests. Therefore, new strategies were developed to provide predictable services for the Internet.

To carry out QoS, you configure policies. A policy is a set of rules that designate an action. The policy basically states which client, application, and schedule (which you designate) must receive a particular service. You can ultimately configure three policy types:

- Differentiated service
- Integrated service
- Inbound admission

Differentiated service and integrated service are considered outbound bandwidth policies. Outbound policies limit data leaving your network and help control server load. The rates you set within an outbound policy control how and what data is or is not limited within the server. Both outbound policy types might require a service-level agreement (SLA) with your Internet service provider (ISP).

Inbound admission policies control connection requests coming into your network from some outside source. Inbound policies are not dependent on a service-level from your ISP.

10.1.1 Differentiated Services

Differentiated Services (DiffServ) mechanisms do not use per-flow signaling and, as a result, do not consume per-flow state within the routing infrastructure. Different service levels can be allocated to different groups of users, which means that all traffic is distributed into groups or classes with different QoS parameters. This reduces the maintenance overhead in comparison to Integrated Services.

In general the characteristics of DiffServ are:

- Traffic is classified, and each class can be given different treatment.
- Each class is best effort.
- Replaces the current Type of Service (TOS).
- Transparent to applications (routers not signalled before data transfer). No modifications to your sockets application is necessary to take advantage of DiffServ.

10.1.2 Integrated Services

Integrated Services (IntServ) bring enhancements to the IP network model to support real-time transmissions and guaranteed bandwidth for specific flows. In this case, we define a *flow* as a distinguishable stream of related datagrams from a unique sender to a unique receiver that results from a single user activity and requires the same QoS.

For example, a flow might consist of one video stream between a given host pair. To establish the video connection in both directions, two flows are necessary. Each application that initiates data flows can specify which QoS are required for this flow. If the video conferencing tool requires a minimum bandwidth of 128 Kbps and a minimum packet delay of 100 ms to assure a continuous video display, such a QoS can be reserved for this connection.

In general, the characteristics of IntServ are:

- The client/server application is negotiated (end-to-end) and is dedicated for the duration of the request.
- Uses Resource Reservation Protocol (RSVP) and X/Open RSVP API. This requires your sockets application to make some modifications to make use of the new APIs.
- Can dynamically change bandwidth. This requires RSVP-aware routers for end-to-end quality of service.
- Good for applications requiring dedicated QoS.

10.1.3 Inbound admission policy

The inbound admission policy is used to control connection requests coming into your network. The inbound policy is used to restrict traffic attempting to connect to your server. You can restrict access by client, Uniform Resource Identifier (URI), application, or local interface on your System i server. In addition, you can enhance server performance by applying a class of service to inbound traffic.

10.2 QoS implementation on the System i

QoS is well defined in the RFCs and these terms can be found in multiple sources, such as:

`http://www.ietf.org`

The System i can only act as a client or a server, not a QoS-aware router.

We will only cover the basics as they specifically apply to and are implemented on your System i.

One of the most important parts of implementing quality of service is your server itself. You need to understand the concepts in 10.2, “QoS implementation on the System i” on page 143, and also to be aware of the role your server plays in implementing these concepts. The System i can only act as a client or a server, not a QoS-aware router. Take this into

consideration as you learn more about the concepts below and begin planning for Quality of Service.

To initiate QoS on the System i, you must first enable it in the TCP/IP configuration properties. From iSeries Navigator, expand **Network**. Right-click **TCP/IP Configuration** and select **Properties** from the context menu. As seen in Figure 10-1, click the **Quality of Service** tab and select **Quality of Service (QoS)**. Optionally, you can change the Datagram batching and Timer resolution, which affect the behavior of QoS (see Help). Click **OK** to save your changes.

Tip: Notice that in Figure 10-1 on page 144, you can select either QoS, the old IP Type of Service (TOS), or Do not use. That is, you cannot choose to run with both QoS and TOS. The reason is that QoS uses the same bits in the IP header as TOS to set IP datagram priority and drop (or discard) precedence. See 10.2.1, “Differentiated Services (DiffServ)” on page 146 for more information.

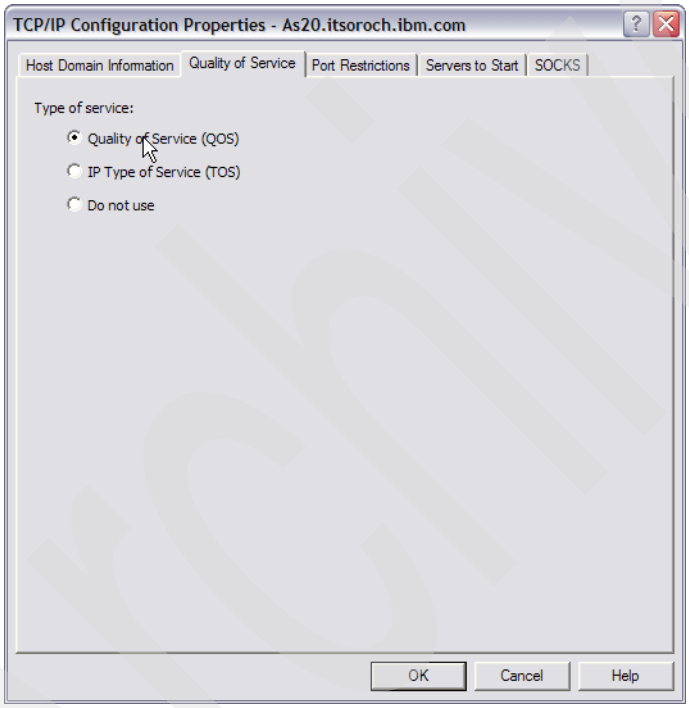


Figure 10-1 Enabling QoS on the System i

To implement QoS, create policies for your traffic. A policy is a set of rules that designate an action and state what client, application, and schedule (of your designation) should receive a particular service. Policies are created and managed in iSeries Navigator. To manage your QoS, click **Network** → **IP Policies** → **Quality of Service** and then right-click **Quality of Service** and select **Configuration**, as shown in Figure 10-2.

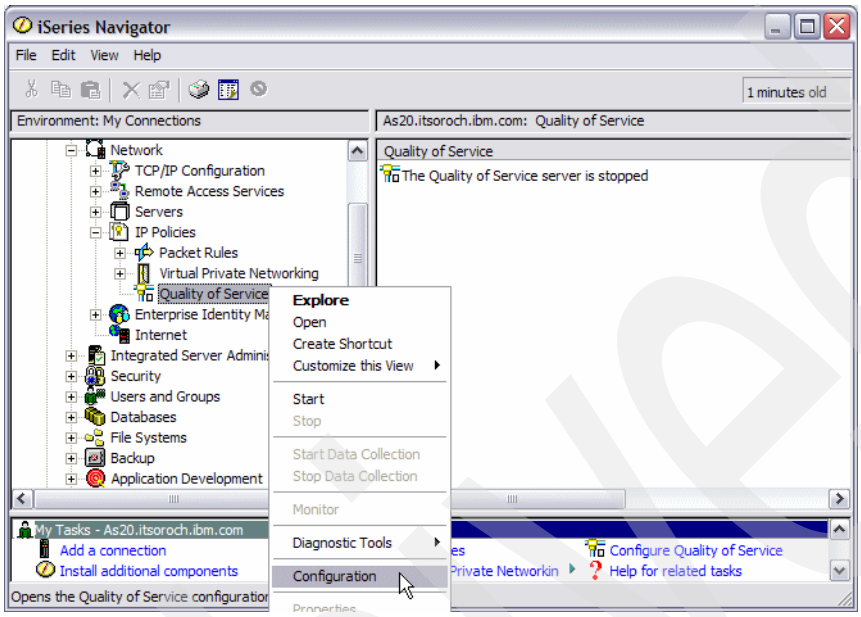


Figure 10-2 Finding QoS in iSeries Navigator

You can ultimately implement three policy types. The policies are first broken into two categories: outbound bandwidth and inbound admission. Within outbound bandwidth policies you can create two service types: Integrated Service policy or Differentiated Service policy.

Inbound refers to policies that control connection requests coming into your network from some outside source. Outbound refers to policies that place limits or help benefit traffic trying to leave your network. To decide which policy to use, evaluate the reason why you want to use QoS. Read the overview in Table 10-1 to obtain what situations apply to each policy type.

Table 10-1 Policy type overview

Policy	Description
Differentiated Services (DiffServ)	This is the first type of outbound bandwidth policy that you can create on your server. Differentiated services is the portion of QoS that divides your traffic into classes. To implement QoS in your network, determine how you want to classify your network traffic and how to handle the different classes. Then you can create the classes of service to use with your Differentiated Services policy.

Policy	Description
Integrated Services (IntServ)	The second type of outbound bandwidth policy that you can create is an Integrated Service policy. Integrated services provides the capability for IP applications to request and reserve bandwidth using the RSVP protocol. Integrated service policies use the RSVP protocol to guarantee an end-to-end connection. This is the highest level of service you can designate; however, it is also the most complex. When you create an Integrated Service policy, you will designate one of two service classes: guaranteed service or controlled load service.
Integrated Services using Differentiated Services markings	Generally, this type of policy is used when an Integrated Services policy may cross a mixed network environment. A mixed network environment contains some network nodes that are RSVP-enabled and some that are not RSVP-enabled.
Inbound admission policy	The inbound admission policy is used to control connection requests coming into your network. The inbound policy is used to restrict traffic attempting to connect to your server. You can restrict access by client, Uniform Resource Identifier (URI), application, or local interface on your System i. In addition, you can enhance server performance by applying a class of service to inbound traffic.

10.2.1 Differentiated Services (DiffServ)

Differentiated Services divides your traffic into classes. To implement quality of service in your network, you must determine how you want to classify your network traffic and how to handle the different classes.

The server uses 5 bits in the IP header to identify an IP packet's level of service. Routers and switches allocate their resources based on the per-hop behavior (PHB) information in the IP header's Type of Service (TOS) field. The TOS field was redefined in RFC 1349 and OS/400 V5R1. A PHB is the forwarding behavior a packet receives at a network node. It is represented by a hexadecimal value known as a codepoint. Packets can be marked at either the server or other parts of the network, such as a router. For a packet to retain the service requested, every network node must be Differentiated Services–enabled. That is, the equipment must be able to enforce per-hop behaviors. To enforce PHB treatment, the network node must be able to use queue scheduling and outbound priority management.

Tip: QoS on i5/OS does not implement queue scheduling. This is the reason why i5/OS can be used as a QoS edge device in your network but not as a QoS-aware router.

If your packet passes through a router or switch that is not Differentiated Services-enabled, it will lose its level of service. Note that the packet is still handled, but it may experience unexpected delivery. On your System i you can use the standard PHB codepoints, or you may define your own class. It is not recommended that you create your own codepoints for use outside your private network as these will not be supported on external routers without defining them on those routers.

Unlike Integrated Services, DiffServ traffic does not require a reservation or per-flow treatment. All traffic placed in the same class is treated equally. DiffServ also does not require your application to be QoS enabled, in contrast to Integrated Services.

DiffServ is also used for traffic control into or out of a server. This means that your System i uses DiffServ to limit performance. Limiting a less-critical application allows a mission-critical application to exit your private network first. When you create a policy, you are asked to set various limits on your server. The performance limits include token bucket size, peak rate limit, and average rate limit. The help topics within the QoS function of iSeries Navigator gives you more specific information about these limits. To configure a DiffServ policy, right-click **Quality of Service** (Figure 10-2 on page 145) and click **Configuration** from the context menu. If you have never configured QoS on your System i, the New Quality of Service Configuration wizard will be launched. Use this to configure the QoS server attributes (such as whether QoS should start automatically with TCP/IP). Once complete, this returns you to the QoS Server Configuration window shown in Figure 10-3.

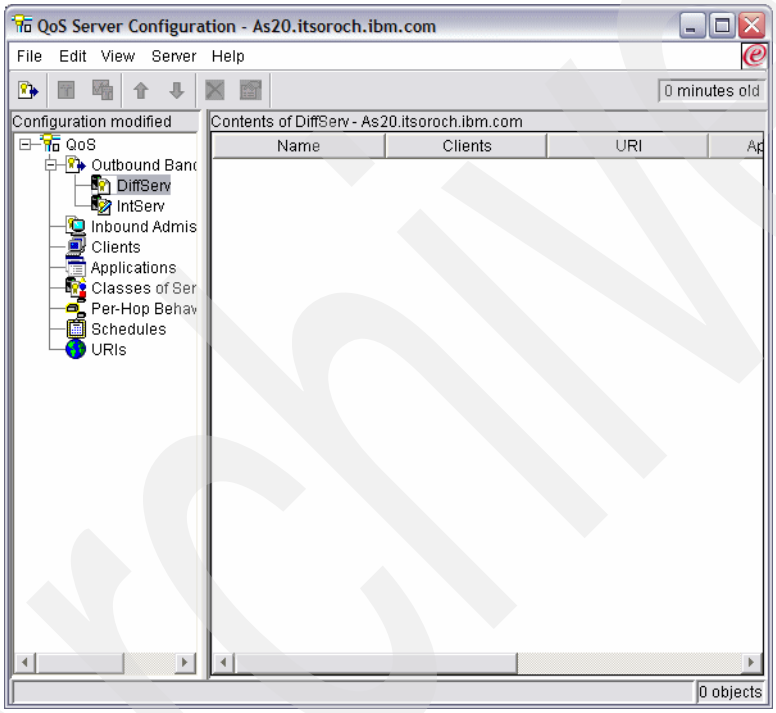


Figure 10-3 DiffServ configuration

Expand **Outbound Bandwidth Policies** and right-click **DiffServ** to add a new policy.

Differentiated Class of Service

The previous section discusses how the Differentiated Services function groups your traffic into classes. Even though most of this happens through equipment, you control how you group traffic and what priority the traffic should receive.

As you implement QoS, first you will define policies. The policies determine the who, what, where, and when. Then you must assign a Class of Service to your policy. The Class of Service is defined separately and may be reused by policies. A Class of Service is comprised of a per-hop behavior, traffic limits, and out-of-profile handling in the class of service.

To configure a Class of Service, open the QoS server configuration in iSeries Navigator and right-click **Classes of Service** to launch the New Class of Service wizard, as shown in Figure 10-4.

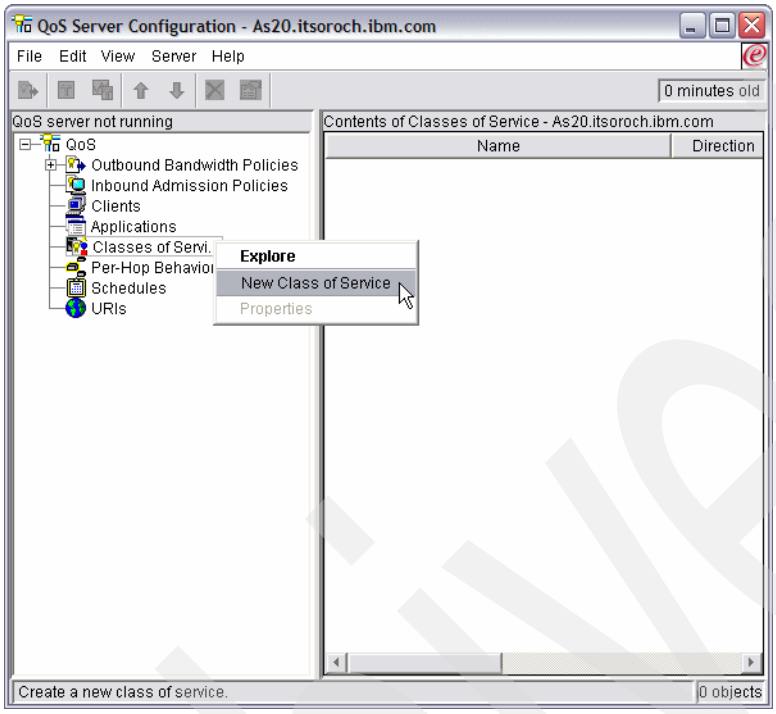


Figure 10-4 Class of Service configuration

Per-hop behaviors

QoS uses the recommended codepoints to assign per-hop behaviors to traffic. Routers and switches use these codepoints to give traffic priority levels. You should determine which codepoints to use based on your individual network needs. Consider what applications are most important to you and what policies should be assigned higher priority. The most important thing is to be consistent with your markings, so that you get the results you expect. These codepoints will be a key part of differentiating different classes of traffic.

Tip: Your System i cannot use these codepoints because it does not act as a QoS-aware router. As an application end-point (client or server) you can configure the System i to properly set these codepoints. But if the System i is routing IP datagrams it will not use these codepoints to give priority to one datagram over another.

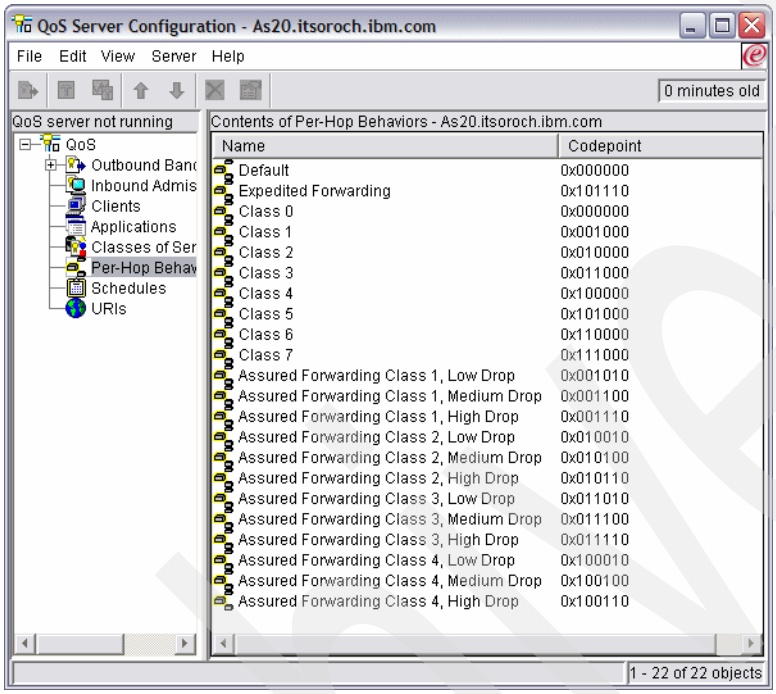


Figure 10-5 Per-hop behavior (PHB) configuration

Performance limits

QoS uses performance limits to restrict traffic through your network. These limits are placed by setting the token bucket size, peak rate limit, and average rate limit.

Out-of-profile handling

The final component of a Class of Service is out-of-profile handling. When you assign the performance limits above, you set values to restrict traffic. When traffic exceeds these restrictions, the packets are considered out-of-profile. This information in a Class of Service tells the server whether to drop, shape, or retransmit these out-of-profile packets. If you decide to drop out-of-profile packets, they are retransmitted after a specified amount of time. If you delay the out-of-profile packets, they are shaped to conform to your defined handling characteristics. If you remark out-of-profile packets with a Differentiated Service Code Point (DSCP), they are reassigned a new codepoint. When you assign these handling instructions in the wizard, click Help for more specific information.

10.2.2 Integrated Services (IntServ)

IntServ deals with traffic delivery time and assigning special handling instructions to particular traffic. It is important to be conservative with your IntServ policies, because it is still relatively expensive to guarantee data transfer. However, overprovisioning your resources can be even more expensive.

IntServ reserves resources for a particular policy before the data is sent. The routers are signaled before data transfer and the network actually agrees to and manages (end-to-end)

data transfer based on a policy. A policy is a set of rules that designates an action. It is basically an admission control list. The bandwidth request comes in a reservation from the client. If all of the routers in the path agree to the requirements coming from the requesting client, the request gets to the server and IntServ policy. If the request falls within the limits defined by the policy, the QoS server grants permission for the Resource Reservation Protocol (RSVP) connection and sets aside the bandwidth for the application. The reservation is performed using the RSVP API (RAPI), qtoq QoS sockets APIs, or both.

Every node that your traffic travels through must have the ability to use the RSVP protocol. The QoS-aware routers provide quality of service through the following traffic control functions: packet scheduler, packet classifier, and admission control.

The ability to carry out this traffic control is often referred to as RSVP-enabled. As a result, the most important part of implementing Integrated Services policies is being able to control and predict the resources in your network. To get predictable results, every node in the network should be RSVP-enabled. For example, your traffic is routed based on resources, not on which paths have RSVP-aware routers. Crossing routers that are not RSVP-aware may cause unpredictable performance problems. The connection is still made, but the performance that the application requests is not guaranteed by that router.

Figure 10-6 shows how the IntServ Resource Reservation Protocol (RSVP) flow logically works:

1. Sender transmits PATH test to receiver.
2. Receiver transmits RESV message to sender.
3. Resources reserved in routers along the path.
4. Data follows same route as PATH and RESV messages.

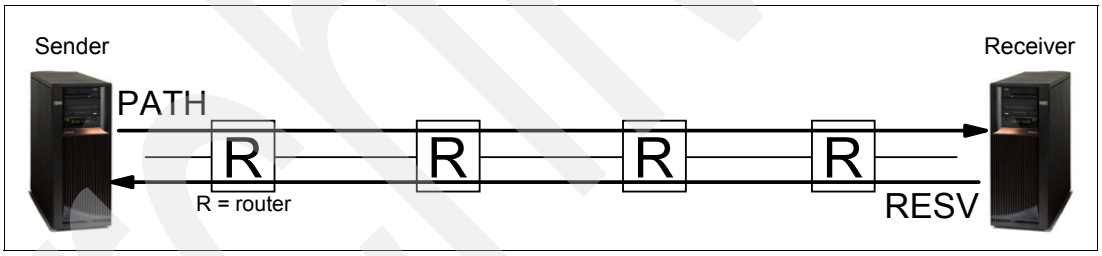


Figure 10-6 RSVP reservation flow

The RSVP-enabled application on the server detects a connection request from the client. In response, the server's application issues a PATH command to the client. This command is issued using the RAPI APIs or qtoq QoS sockets APIs and contains router IP address information. A PATH command contains information about the available resources on the server and the routers along the path as well as route information between the server and the client. The RSVP-enabled application on the client then sends a RESV command back along the network path to signal the server that the network resources have been allocated. This command makes the reservation based on the router information from the PATH command. The server and all routers along the path reserve the resources for the RSVP connection. When the server receives the RESV command, the application starts transmitting data to the client. The data is transmitted along the same route as the reservation. Again, this shows how important the routers' abilities to carry out this reservation are to the success of your policies.

IntServ is not meant for short-term RSVP connections such as HTTP. Consider what areas and applications are having performance problems and need QoS. Applications used in an Integrated Services policy must be able to use the RSVP protocol. Currently, your i5/OS does not have any RSVP-enabled servers, so you will need to write or buy an application to use RSVP.

As packets arrive and attempt to leave your network, your server determines whether it has the resources to send the packet. This acceptance is determined by the amount of space in the token bucket. You manually set the number of bits to allow into your token bucket any bandwidth limits, token rate limits, and the maximum number of connections your server should allow. These values are referred to as *performance limits*. If the incoming packets will cause the bucket to exceed its limit, the packets are considered non-conformant. Your server can handle non-conformant traffic in a few different ways. It can either delay, shape, retransmit, or drop the packets. If the packets will remain within the server's limits, the packets conform and are sent out. In IntServ, each connection is granted its own token bucket. In Differentiated Services, the whole subnet or range of clients shares a token bucket.

RSVP protocol and QoS APIs

The Resource Reservation Protocol (RSVP), along with the RAPI or qtoq QoS sockets APIs, perform your Integrated Service reservation. Every node that your traffic travels through must have the ability to use the RSVP protocol. The ability to carry out IntServ policies is often referred to as RSVP-enabled.

The RSVP protocol is used to create an RSVP reservation in all of the network nodes along your traffic's pathway. It maintains this reservation long enough to provide your policies' requested services. The reservation defines the handling and bandwidth that the data in this conversation require. The network nodes each agree to provide the data handling defined in the reservation.

RSVP is a simple protocol in that reservations are only made in one direction (from the receiver). For more complex connections, such as audio and video conferences, each sender is also a receiver. In this case, you must set up two RSVP sessions for each side.

In addition to RSVP-enabled routers, you need to have RSVP-enabled applications to use Integrated Services. Because the System i does not have any RSVP-enabled applications at this time, you will need to write the applications using the RAPI API or the qtoq QoS Sockets APIs. This will enable the applications to use the RSVP protocol. If you want an in-depth explanation, there are many sources that explain these models, their operation, and messaging. You need a thorough understanding of the RSVP protocol and the contents of Internet RFC 2205.

qtoq Sockets APIs

With OS/400 V5R2 and i5/OS, you can use the qtoq QoS sockets APIs to simplify the work required to use the RSVP protocol on the System i. The qtoq sockets APIs call the RAPI APIs and perform some of the more complex tasks. The qtoq sockets APIs are not as flexible as the RAPI APIs, but provide the same function with less effort. The *No Signal* versions of the APIs enable you to write the following:

- An application that will load an RSVP rule on the server
- An application that only requires the server side application (of the TCP/IP conversation) to be RSVP-enabled

The RSVP signalling is done automatically on behalf of the client side.

See the QoS API connection-oriented functional flow section below for typical QoS API flow for an application/protocol using connection-oriented or connectionless qtoq QoS sockets.

QoS API connection-oriented functional flow

Figure 10-7 illustrates the client/server relationship of the QoS enabled API qtoq sockets functions for a connection-oriented protocol, such as Transmission Control Protocol (TCP).

When the QoS-enabled API functions are called for a connection-oriented flow requesting that RSVP be initiated, additional functions are initiated. These functions cause the QoS agents on the client and server to set up the RSVP protocol for the data flow between the client and the server.

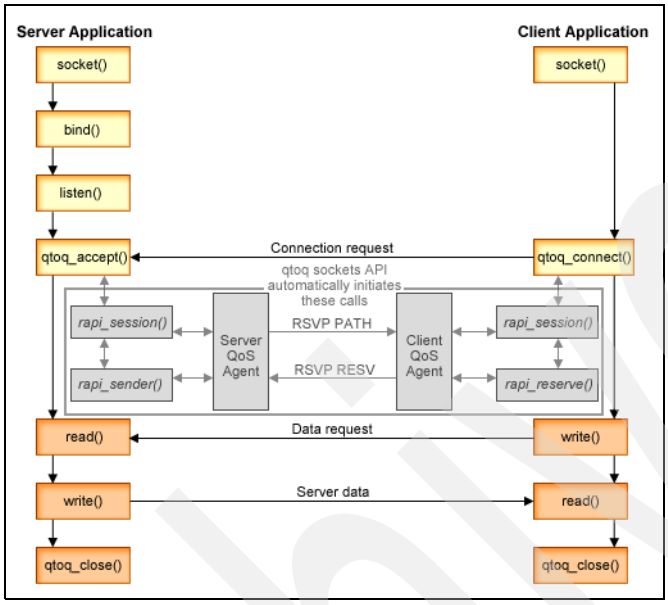


Figure 10-7 IntServ API negotiation using qtoq QoS sockets APIs

10.2.3 Connection request rate and URI request rate

Inbound policies are used to control traffic that attempts to connect to your server. Two types of policies enable you to define and configure inbound controls: URI policies and connection rate policies. The two policy types are described below.

URI request rate policies

URI request rate policies are part of a solution to help protect servers against overload. This type of policy applies admission controls, based on application level information, to limit the URI requests accepted by the server. In industry this is also referred to as header-based connection request control, which uses URIs to set priorities.

Unlike connection rate policies, URI policies have more control because they examine content, not just packet headers. The content they examine could include URI name or other application-specific information. For System i, the relative URI name is used to define the policy (for example, /products/clothing). The examples below describe the relative URI.

Relative URI

The relative URI is actually a subset of an absolute URI (similar to the old absolute URL). Consider this example: http://www.ibm.com/software. The http://www.ibm.com/software segment is considered the absolute URI. The /software segment is the relative URI. All relative URI values must begin with one forward slash (/). The following are valid relative URI examples:

- /market/grocery#D5
- /software
- /market/grocery?q=green

Tip: The default protocol, host name, and port are all inherited from the HTTP server. Also, there is an implicit wildcard when you specify a URI. For example, /software will include anything within the software directory.

URI policies are considered inbound policies because they control the traffic requests entering your network. As part of this inbound control, you can specify the priority in which URI requests are handled after they have been accepted by the policies. By prioritizing policies, you actually prioritize the connection requests in the queue based on the configured priority for each connection.

Figure 10-8 shows the initial step in configuring a New URI DiffServ policy.

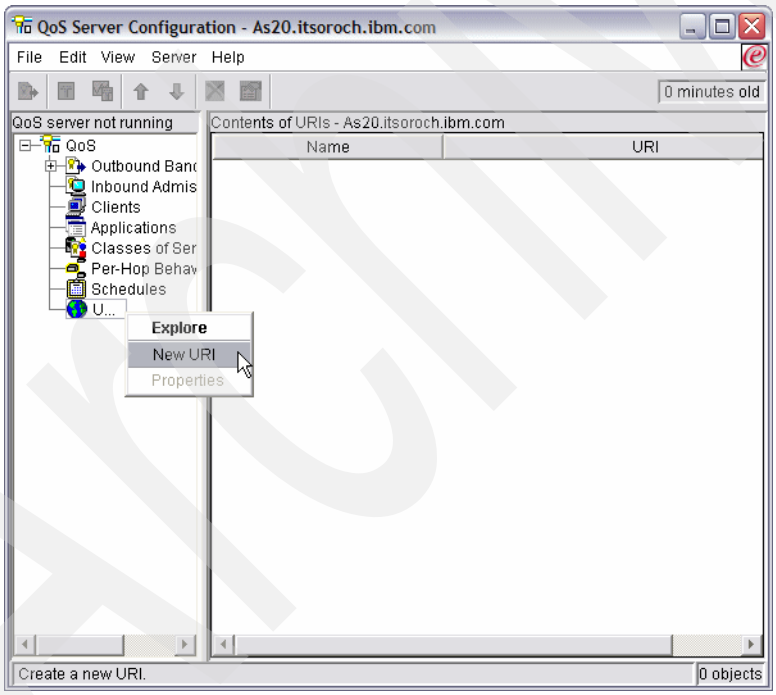


Figure 10-8 URI policy configuration

10.2.4 Connection rate policies

Connection rate policies are also part of a solution to help protect servers against overload. This type of policy applies admission controls, based on connection-level information, to limit the connections accepted by the server. In industry this is also referred to as TCP SYN policing.

Connection rate policing accepts or denies new incoming connections based on the average number of connections established per second and the maximum number of established connections (in any given instant) defined in the policy you create. These connection limits consist of average rate and burst limits, which the wizards in iSeries Navigator prompt you to enter. When incoming connection requests reach the server, the server analyzes the packet header information to determine whether this traffic is defined in a policy. The system verifies this information against the connection limits profile. If the policy is within the policy limits, it is placed into the queue. Packets that do not comply with a policy are discarded.

Similar to URI policies, connection rate policies are considered inbound policies because they control the connection rate of traffic entering your network. As part of this inbound control, you can specify the priority in which connections are handled after they have been accepted by the policies. By prioritizing policies, you actually prioritize the connection requests in the queue based on the configured priority for each connection.

Both the URI policy and connection rate policy require you to set connection rates and burst limits for the traffic defined in each policy. These rate limits help restrict inbound connections trying to enter your server. The average connection rate specifies the limit of new, established connections or the rate of accepted URI requests allowed into a server.

10.2.5 Storing your configuration

You can export your policies to a directory server. QoS policy configuration can be exported to a directory server, using LDAP Version 3.

Exporting QoS policies to a directory server makes your policies easier to manage. There are three ways to use the directory server:

- The configuration data can be stored on one local directory server for many systems to share.
- The configuration data can be configured, stored, and only used by one system (not shared).
- The configuration data can reside on a directory server that holds data for other systems, but is not shared between those other systems. This allows you to use a single location to back up and save data for several systems.

Saving QoS policies on your local server is not as complex. There are a number of advantages to using policies locally:

- Eliminate the complexity of LDAP configuration for users who do not need it.
- Improve performance, because writing to the local server IFS is faster.
- Easier to duplicate a configuration between different System i's. You can copy the file from one system to another. Because there is no primary or secondary machine, you can tailor each policy directly on the individual servers.

If you decide to export your policies to an LDAP server, you must be familiar with LDAP concepts and directory structures before you continue. Review LDAP basics within the Directory Services (LDAP) topic of the System i Information Center.

QoS tree structure

When you want to manage part of your directory, you reference the Distinguished Name (DN) or (if you choose) a keyword. Specify the DN when you configure the directory server. DNs usually consist of the name for the entry itself, as well as the objects (top to bottom) above the entry in the directory. The server can access all objects on the directory that are below the

DN. For example, assume that the LDAP server is contained the directory structure as shown in Figure 10-9.

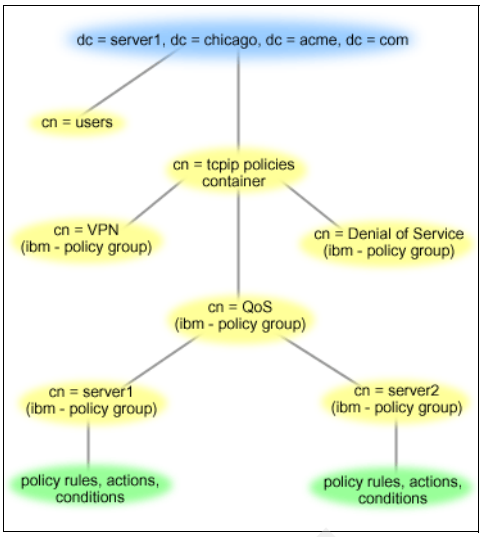


Figure 10-9 QoS in an LDAP structure

Server1 at the top (dc=server1,dc=chicago,dc=acme,dc=com) is the server on which the directory server resides. The other servers, such as cn=QoS or cn=tcpip policies are where the QoS servers reside.

So on cn=server1 the default DN would read cn=server1,cn=QoS,cn=tcpip policies,dc=server1,dc=chicago,dc=acme,dc=com. On the cn=server2 the default DN would read cn=server2,cn=QoS,cn=tcpip policies,dc=server1,dc=chicago,dc=acme,dc=com.

When managing your directory, it is important to change the proper server in the DN, such as cn or dc. Be careful when editing the DN, especially because the string is usually too long to be displayed without scrolling.

11

Intrusion Detection System (IDS)

The Intrusion Detection System (IDS) introduced in IBM i5/OS V5R4 is a notification system of attempts to hack into, disrupt, or deny service to the system. Prior to IDS, the i5/OS took some protective measures against the types of intrusions described here. However, with the new IDS support, the i5/OS system can now tell you about the intrusions.

On the i5/OS system, the following types of intrusions are caught, audited, and, in many cases, discarded before they become a threat:

- "Attacks" on page 159
  - "IP fragments" on page 159
  - "Malformed packets" on page 159
  - "SYN floods" on page 159
  - "ICMP redirect messages" on page 160
  - "Perpetual echo" on page 160
  - "Restricted IP options" on page 160
  - "Restricted IP protocols" on page 161
- "Scans" on page 161
- "Traffic regulation anomalies for TCP and UDP" on page 161

Throughout this book the term *hacker* is used to describe computer attackers. Occasionally, the term *intruder* or *perpetrator* is also used. In short, whatever the terminology, those described by such terms are, perhaps, amateurs who are just playing around; or they might be intellectual-types who may *hack* just for the challenge or, worse, because they are disgruntled employees; or they might be professionals.

The Internet is a vast playground. There are unsophisticated hackers who try to disrupt or deny service to both targeted and random IP addresses. Viruses, floods, ping-of-death attempts, *Smurf* attacks, User Datagram Protocol (UDP) *fraggle* attacks, and so on are designed just for the purpose of disruption and Denial of Service (DoS). These types of attacks are commonplace by now and on the decline.

There are two types of sophisticated hackers, the ethical variety and the malicious criminal. The ethical hacker looks for vulnerabilities in a company's security defenses and suggests how to plug the holes. The malicious hacker is out to exploit those vulnerabilities and steal information in such a way that the victim may never be aware of the infiltration. The malicious

hacker's intent is to *own* your system, gaining access using stealthy scanning techniques and then using trojan horses or root kits to wreak havoc or steal information. It is interesting to note here where these two types of hackers have crossed paths in the past. Sometimes the ethical hacker publicly exposes a vulnerability before the provider of the software has had a chance to come up with a fix. In some cases, this exposure has been intentional either for publicity or to precipitate a quick solution to the problem. Whatever the reason, this practice opens a window of opportunity for the malicious hacker. Typically, during these *windows* of opportunity, a virus might be launched, or, worse yet, a company's secrets might be compromised.

The i5/OS system has long detected attempts to disrupt and deny service. With IDS, there is now notification that these potential attacks have taken place. Additionally, other types of intrusions are now monitored and handled. Some events may be just legitimate attempts to connect to the system. It is up to a system administrator or the person monitoring the security audit journal to decide.

This is just the beginning of intrusion detection on the i5/OS system. This chapter describes the Intrusion Detection System currently offered on the i5/OS system.

11.1 Intrusion types

This section describes the most common types of intrusion, which are attacks, scans, and traffic regulation anomalies.

11.1.1 Attacks

Attacks may or may not be malicious. As mentioned in the introduction, IDS is notified of various events and some of these notifications may be false alarms. A description of these attacks follows.

IP fragments

Datagrams that are too big to be transmitted over a network are broken down into fragments. The fragmentation process involves tacking on an IP header to each piece of the fragmented datagram, setting the *More Fragments* (MF) flag, and providing the offset of where this fragment lies within the original datagram. This information along with a fragment identification number and the length of data in the fragment itself is used by the target in reassembling the original datagram.

On the i5/OS system, fragments that IDS is notified about fall into three categories:

- Fragments that when reassembled would be greater than 64 K in size and, therefore, too large (see “Malformed packets” on page 159)
- Fragments that are less than 576 bytes in length
- Fragments with an offset of less than 256 bytes (This does not mean that the fragment itself is less than 576 bytes. This may be an attempt to overlay data in the first fragment.)

In the case of a fragment that is too large, the intent may be to crash or hang a system. In the other two cases, where the fragments are smaller, the intent may be a malicious attempt to slip through a firewall. Then again, it could just be a normal case of packet retransmission. At any rate, a packet is not forwarded to the next layer until it is completely reassembled.

Malformed packets

Malformed packets may be designed to cause a system to crash or hang. They are detected by the TCP/IP stack in the following instances:

- When a checksum is wrong
- For a destination port of 0
- When a packet size, including fragments, is greater than 64 K
- When both SYN and FIN are set (indicating that a client is attempting to establish a connection, but has no more data to send)

The TCP/IP stack notifies IDS of these malformed packets and then, in most cases, discards them.

SYN floods

SYN floods are an attempt to tie up system resources and deny service. They occur when the TCP/IP three-part handshake does not complete. An attacker will initiate a connection attempt to a host (first part of the handshake: SYN) and provide a spoofed source address for the host to acknowledge (second part of the handshake: SYN/ACK), and then leave the host waiting on an acknowledgment (third part of the handshake: ACK) that will never come from the spoofed address. This ties up system resources.

On the i5/OS system, these incomplete connections are queued. Once the queue limit is exceeded, the oldest incomplete connection attempts are dropped from the backlog one at a time. Each time one such connection is dropped, the TCP/IP stack notifies IDS of the possible flood situation.

ICMP redirect messages

The Internet Control Message Protocol (ICMP) is used for sending out-of-band messages concerning network operations. There are many types of ICMP messages (see Request for Comments (RFC) 792). An ICMP type 5 message is a *redirect* message. This message may be sent by a router to a host on the same subnet to indicate a more optimal route for packets sent by that host to some other target host in another network. The message will indicate that there is another gateway (that is, router) on the same subnet to which the packets may be sent and forwarded more efficiently. The original packet from the host that precipitated the ICMP redirect message from the router is forwarded anyway and the host does not have to honor the ICMP redirect message from the router. That is, it can choose to ignore ICMP redirect messages.

ICMP redirect messages can be used maliciously by a hacker for Man-in-the-Middle (MITM) attacks. In this type of an attack, the hacker, posing as a router, sends an ICMP redirect message to a host indicating that all future traffic must be directed his way as the more optimal route to the intended destination.

On the i5/OS system, the TCP/IP stack will notify IDS in the event of any ICMP redirect message whether the intent is valid or not, and whether or not the stack has been configured to ignore such messages.

Perpetual echo

The UDP *fraggle* attack is an annoying DoS attack involving UDP echo port 7. An attacker sends a UDP echo request to an IP broadcast address and provides a spoofed source address for all of the targets to echo back responses. The spoofed source address, which is not the hacker's address, becomes the victim of a potentially large amount of network traffic. If the source port is also port 7, then a *perpetual echo* results.

On the i5/OS system, any UDP request to destination port 7 is signaled by the TCP/IP stack to IDS as a possible perpetual echo attack. IDS then checks the source port to determine whether the packet truly is a perpetual echo attempt.

Restricted IP options

An IP header may contain the Loose Source and Record Route (LSRR) option traditionally used by *traceroute* to map out a network's topology. This option has been used by network administrators to obtain why two hosts on a network are not communicating, or to specify alternate routes to relieve network congestion. A hacker may try to use LSRR to get through firewalls. By specifying LSRR and a hop that is reachable both by the hacker and private IP addresses, the hacker may reach what was previously thought to be a protected IP address.

On the i5/OS system, the TCP attribute IPSRCRTG (IP source routing) may be set to either on or off through the CHGTCPA command. If IPSRCRTG is on, the packet is forwarded if it can be. If off and the system is not the destination of the packet, the packet is discarded. At any rate, any datagrams with IP options are signaled by the TCP/IP stack to IDS as possible suspicious events.

Restricted IP protocols

The IP protocols most often used are ICMP, TCP, UDP, and IGMP. Other protocols listed as part of the Internet Assigned Numbers Authority (IANA) may be used in an attempt to gain back door entry into a system.

On the i5/OS system, unrecognized IP protocols are signaled by the TCP/IP stack to IDS and handed off to raw support. If there is no application listening on the raw port, the packet is discarded. If, however, there is an application listening on the port, this could be the back door that the perpetrator is trying to access. The restricted IP protocol event logged in the security audit journal must alert a system administrator to the possibility of such a rogue application.

11.1.2 Scans

Scanning entails sending a datagram to a system in order to determine what the listening ports are. Once the open ports are discovered, the hacker tries to discover the weaknesses and gain access to the system.

On the i5/OS system, the TCP/IP stack signals IDS when connection attempts to non-listening ports are made (that is, *undemuxable SYNs*) or when a connection attempt is made in which the source address is the same as the target address (which could be a *spoofing* attempt).

Scans may be innocent attempts at connections to a server that may be down, making the resulting attention event a false alarm. However, they may be of interest if they come in at a very high rate or a very slow rate. The high rate variety may be quick attempts at gathering information or attempts to deny service. They are more readily identifiable in the system logs than slow scans. The slow, *stealthy* variety are of more interest. A perpetrator may be seeking information about what ports to probe, what operating system is running, and so forth. Hackers may scan from an Internet cafe, a library, and so on (in short, a *disposable* source). If tracked down in a log, the source IP address will no longer be valid. Also, by the time a suspicious IP address is noticed in a log, the hacker may have already gained access to the system, having sneaked in under the radar, and stolen valuable information.

11.1.3 Traffic regulation anomalies for TCP and UDP

Traffic regulation anomalies are events that cover TCP established connections or UDP transmissions. Their purpose is to single out an inordinate number of connections to a certain range of addresses/ports/applications. The UDP variety, being connectionless, is tougher to monitor than the TCP variety. These anomalies may indicate a DoS attack or be used to monitor certain connections and usage of certain applications on a system.

11.2 Setup for IDS notification on i5/OS

To enable IDS on the i5/OS system, here are a few steps that generally have to be performed only once:

1. Enable Quality-of-Service (QoS). The i5/OS system currently uses the QoS server to push its intrusion detection policies down to the network level. Figure 11-1 shows QoS being enabled through the CHGTCPA command.

```
                          Change TCP/IP Attributes (CHGTCPA)

Type choices, press Enter.

IP time to live (hop limit)  . .   64            1-255, *SAME, *DFT
IP QoS enablement  . . . .   *YES          *SAME, *TOS, *YES, *NO
IP dead gateway detection:
  Enablement . . . . . . . . . .   *YES          *SAME, *DFT, *NO, *YES
  Interval . . . . . . . . . . .   2             1-60
ARP cache timeout  . . . . . . .   15            1-1440, *SAME, *DFT
Enable ECN . . . . . . . . . . .   *NO           *SAME, *YES, *NO
Network file cache:
  Enablement . . . . . . . . . *YES          *DFT, *CLEAR, *SAME, *YES, *NO
  Cached file timeout  . . . . .   300           *NOMAX,30-604800 sec (1week)
  Cache size . . . . . . . . . .   10            10-100000 megabytes
Log protocol errors  . . . . . .   *NO           *SAME, *YES, *NO




                                                                        Bottom
F3=Exit   F4=Prompt   F5=Refresh   F10=Additional parameters   F12=Cancel
F13=How to use this display        F24=More keys
```

Figure 11-1 Enable QoS by changing TCP attributes

2. Enable i5/OS system security auditing in order to have intrusions show up in the system security audit journal. This is done by changing the system value of QAUDCTL to allow for auditing (*AUDLVL), using the WRKSYSVAL QAUDCTL command and taking option 2, as shown in Figure 11-2.

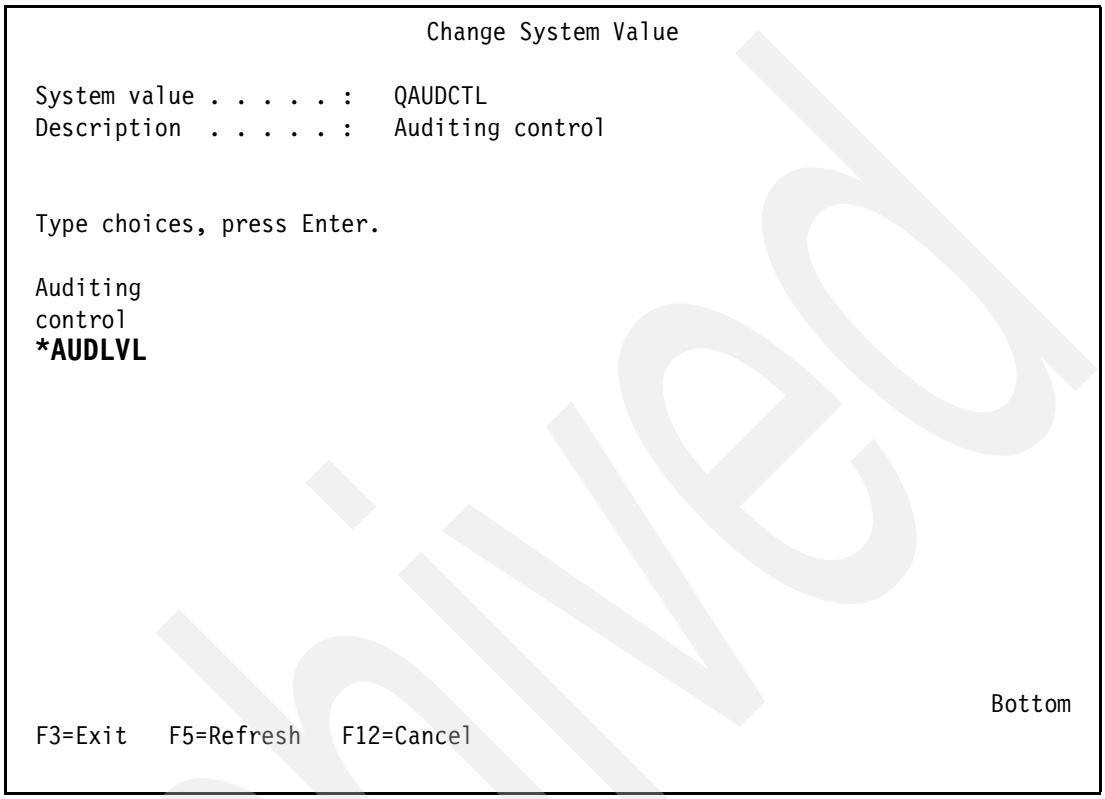


*Figure 11-2 Turn auditing on (*AUDLVL)*

3. Also, either one of the two system values — QAUDLVL or QAUDLVL2 — should allow for auditing *attention events* (*ATNEVT), which can be accomplished using the WRKSYSVAL command on either QAUDLVL or QAUDLVL2 and taking option 2 to change or add the option, as shown in Figure 11-3.

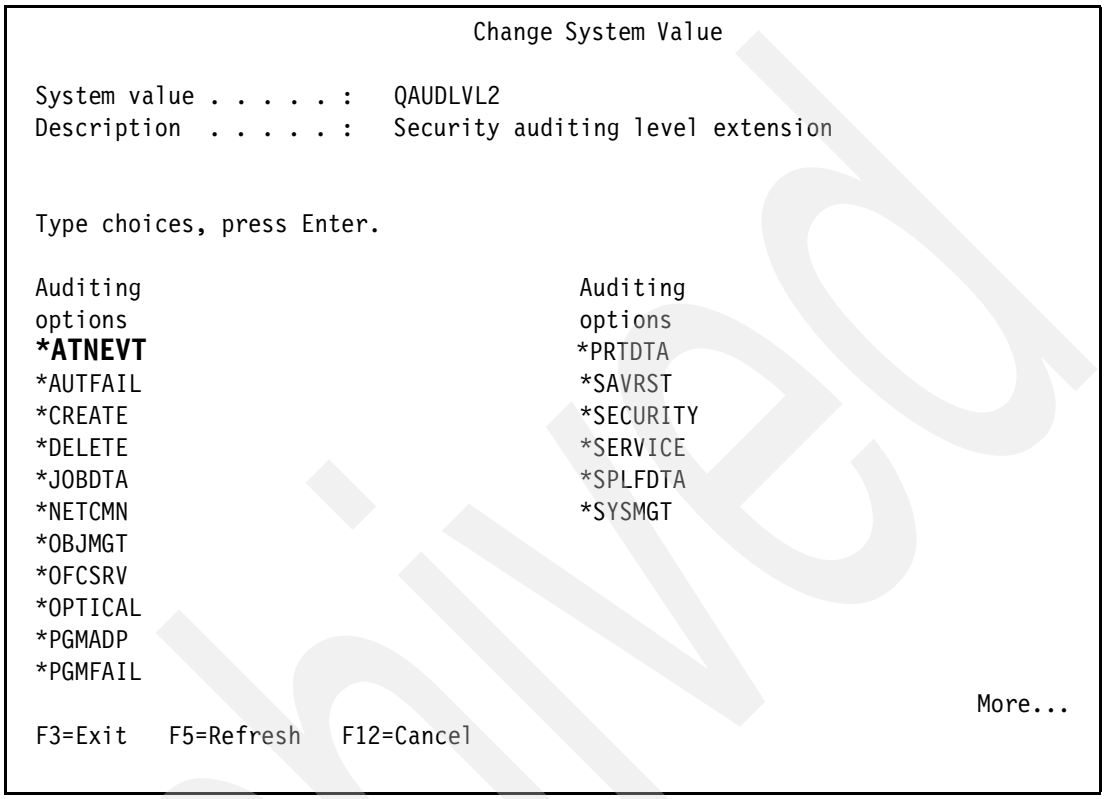


*Figure 11-3 Allow auditing of attention events (*ATNEVT)*

Note: If *ATNEVT is not set as an option in QAUDLVL, then *AUDLVL2 must be set in QAUDLVL in order to direct the system to interrogate QAUDLVL2 for any auditing options not covered in QAUDLVL.

4. A viable policy file, IDSPOLICY.CONF, must exist at /QIBM/USERDATA/OS400/QOS/ETC or there will be no IDS policies to load. The conditions and actions in the policy file must be created by the user. To facilitate the creation of conditions and associated actions, a commented out sample policy is provided in the initial version of IDSPOLICY.CONF. A sample of a viable IDS policy file is shown in Figure 11-4.

Note: A copy of the IDS policy file is shipped in /QIBM/PRODDATA/OS400/QOS and loaded into /QIBM/USERDATA/OS400/QOS/ETC at installation time. The system administrator should edit the file in /QIBM/USERDATA/OS400/QOS/ETC to create conditions and actions that will be loaded to the network layer and enable IDS.

```
 Edit File: /qibm/userdata/os400/qos/etc/idspolicy.conf
 Record :       1   of      13 by   8           Column :    1     59 by  74
 Control :

CMD ....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+
     ************Beginning of data**************
    ibm-idsConditionAuxClass        idscond1
    {
    ibm-idsConditionType            ATTACK
    ibm-idsAttackType               FLOOD
    ibm-idsLocalPortRange           1-65535
    ibm-idsLocalHostIPAddress       2-9.000.000.000-8
    ibm-policyIdsActionName         idsact1
    }
    ibm-idsActionAuxClass           idsact1
    {
    ibm-idsActionType               ATTACK
    ibm-idsMaxEventMessage          25
    }
     ************End of Data********************

 F2=Save   F3=Save/Exit   F12=Exit   F15=Services   F16=Repeat find
 F17=Repeat change  F19=Left  F20=Right
```

Figure 11-4 A viable IDS policy file (idspolicy.conf) example

The sample in Figure 11-4 consists of one condition and one associated action. Normally, there will be several conditions covering all types of intrusions. An action may be referenced by more than one condition.

Note: In its most primal form, an IDS policy consists of at least one condition and an associated action. There may be multiple conditions and multiple associated actions. There may also be multiple conditions associated with one action.

5. Finally, in order to load the IDS policies, the QoS server should be started, as shown in Figure 11-5. This step will need to be done more than once if policy changes are made.

```
                                Command Entry
                                                          Request level:   1
Previous commands and messages:

  (No previous commands or messages)









                                                                       Bottom
Type command, press Enter.
===> STRTCPSVR *QOS



F3=Exit   F4=Prompt   F9=Retrieve   F10=Include detailed messages
F11=Display full      F12=Cancel    F13=Information Assistant   F24=More keys
```

Figure 11-5 Start the QoS server

To have the QoS server start automatically with TCP, select the **QoS Server Properties** by right-clicking **QoS** when using the iSeries Navigator and check the box next to **Start QoS server when TCP/IP is started**.

Note: To reload the IDS policy file after changes have been made, the QoS server needs to be ended and restarted. To end the QoS server, type the following command and click **Enter**:

```
ENDTCPSVR *QOS
```

11.3 IDS policy file

The IDS policy file is used to define the intrusions that will be audited. It is set up with *directives* or keywords for both conditions and actions. The various directives with their range of values are given in “IDS policy directives defined in terms of their possible values” on

page 167. The example shows the prolog of the shipped IDS policy file (/QIBM/PRODDATA/OS400/QOS/IDSPOLICY.CONF). Following the definition of the directives is Table 11-2 on page 168, which shows both the condition and action directives.

Example 11-1 IDS policy directives defined in terms of their possible values

```
# IDS Policy File Keywords and Values:
#
# ibm-idsConditionAuxClass          <condition name> (max of 31 characters)
#   {
#   ibm-idsConditionType:           ATTACK | TR | SCAN_GLOBAL | SCAN_EVENT
#   ibm-idsAttackType:              MALFORMED_PACKET | FLOOD | OUTBOUND_RAW |
#                                   ICMP_REDIRECT | PERPETUAL_ECHO | IP_FRAGMENT |
#                                   RESTRICTED_IP_OPTIONS | RESTRICTED_IP_PROTOCOL
#   ibm-idsLocalPortRange:          <from-port>[:<to-port>] (ports range from 1 to 65535)
#   ibm-idsRemotePortRange:         <from-port>[:<to-port>] (ports range from 1 to 65535)
#   ibm-idsProtocolRange:           <from-protocol>[:<to-protocol>] (protocols range from 1 to 255)
#                                   (See www.iana.org/assignments/protocol-numbers)
#   ibm-idsIPOptionRange:           <from-option>[:<to-option>] (options range from 1 to 255)
#                                   (See www.iana.org/assignments/ip-parameters)
#   ibm-idsLocalHostIPAddress:      1-All local addresses
#                                   2-<IPv4Address>-<PrefixMaskLength>
#                                   3-<IPV4Address1><-IPV4Address2>
#   ibm-idsRemoteHostIPAddress:     1-All remote addresses
#                                   2-<IPv4Address>-<PrefixMaskLength>
#                                   3-<IPV4Address1><-IPV4Address2>
#   ibm-policyIdsActionName         <action name> (max of 31 characters)
#   }
# ibm-idsActionAuxClass             <action name> (max of 31 characters)
#   {
#   ibm-idsActionType:              ATTACK | TR | SCAN_GLOBAL | SCAN_EVENT
#   ibm-idsStatInterval:            n       (Default is 60, max is 4294967295)
#                                   where n is the interval length in minutes to collect
#                                   IDS statistics
#   ibm-idsMaxEventMessage:         n       (Default is 5)
#                                   where n is the maximum number of attack event
#                                   messages to be audited per interval specified
#                                   with ibm-idsStatInterval.
#   ibm-idsTRtcpTotalConnections:   n
#                                   where n is the total number of connections
#                                   allowed for a listening server application.
#   ibm-idsTRtcpPercentage:         n       (Default is 100)
#                                   where n represents anything in the range
#                                   of 0 - 100%
#   ibm-idsTRtcpLimitScope:         PORT | PORT_INSTANCE  (Default is PORT_INSTANCE)
#                                   PORT specifies that traffic regulation parameters
#                                   apply to the aggregate of all sockets bound to
#                                   the target port (i.e., regardless of IP address)
#                                   PORT_INSTANCE specifies that traffic regulation
#                                   parameters apply to each socket bound to the
#                                   target port individually (i.e., IP address range taken into account)
#   ibm-idsFSInterval:              n       (Default is 1 minute)
#                                   where n is the interval in minutes for monitoring fast
#                                   scanning attacks (maximum value is 1440)
#   ibm-idsFSThreshold:             n       (Default is 5)
#                                   where n is the fast scanning threshold (maximum value is 64)
#   ibm-idsSSInterval:              n       (Default is 120 minutes)
#                                   where n is the interval in minutes for monitoring slow
#                                   scanning attacks (maximum value is 1440)
#                                   NOTE: This interval must be greater than the fast scan
```

```
#                                       interval.  However, a value of 0 can be specified
#                                       to indicate that no slow scan interval exists (to
#                                       "turn off" slow scan processing).
#   ibm-idsSSThreshold:             n       (Default is 10)
#                                   where n is the slow scanning threshold (maximum value is 64)
#                                   NOTE: This threshold must be greater than the fast scan
#                                       threshold.  However, a value of 0 can be specified
#                                       to indicate that no slow scan threshold exists (to
#                                       "turn off" slow scan processing).
#   }
```

Note: Here it may be pointed out that the directive ibm-idsMaxEventMessage can gate the number of attention events that are written to the security audit journal. If the value is reasonable (for example, 25) in the action associated with, for example, a *malformed packet* attack condition, only that small number of Intrusion Monitor (IM) records are cut in the audit journal. If, however, the value of this action directive is large, the security audit journal could be *flooded* by IM records during a packet storm. A value of 0 for ibm-idsMaxEventMessage implies no limit to the number of audit journal entries that can be generated as a result of executing the corresponding action.

Given the IDS directives in Figure 11-1 on page 162, it is now worth discussing how to construct meaningful conditions and actions. For example, it should be obvious that ibm-idsFSInterval does not go with an ibm-idsActionType of ATTACK since it applies to either an action type of SCAN_EVENT (preferred) or SCAN_GLOBAL (which is interpreted as a SCAN_EVENT internally). For this purpose, Table 11-2 is presented as a cheat sheet when constructing conditions and their associated actions. Table 11-1 provides the key information to interpreting the IDS policy file directives in Table 11-2.

Table 11-1 KEY to IDS policy file directive Table 11-2

o - optional **r** - required **x** - not supported **i** - ignored **d** - depends on type of attack	**TR** - (Traffic Regulation) **SE** - SCAN_EVENT **SG** - SCAN_GLOBAL **AT** - ATTACK	**MP** - MALFORMED_PACKET **FL** - FLOOD **OR** - OUTBOUND_RAW **IR** - ICMP_REDIRECT **PE** - PERPETUAL_ECHO **IF** - IP_FRAGMENT **RO** - RESTRICTED_IP_OPTIONS **RP** - RESTRICTED_IP_PROTOCOL

Table 11-2 IDS policy file directives

	ibm-idsConditionType			**ibm-idsActionType**								
	TR	**SE**	**SG**[a]	**AT**	**MP**	**FL**	**OR**	**IR**	**PE**	**IF**	**RO**	**RP**
Condition Directives:												
ibm-idsLocalPortRange[b]	o	r	i	o	o	o	x	o	o	o	o	o
ibm-idsRemotePortRange	o	o	i	o	o	o	x	o	o	o	o	o
ibm-idsProtocolRange	r	x	i	d	x	x	x	x	x	x	x	r
ibm-idsIPOptionRange	x	x	i	d	x	x	x	x	x	x	r	x
ibm-idsLocalHostIPAddress	r	r	i	o	o	o	x	o	o	o	o	o
ibm-idsRemoteHostIPAddress	o	o	i	o	o	o	x	o	o	o	o	o
ibm-policyIdsActionName	r	r	r	r	r	r	x	r	r	r	r	r

Action Directives:												
ibm-idsActionType	r	r	r	r								
ibm-idsStatInterval	o	i	i	o								
ibm-idsMaxEventMessage	o	o	o	o								
ibm-idsTRtcpTotalConnections	r	x	x	x								
ibm-idsTRtcpPercentage	r	x	x	x								
ibm-idsTRtcpLimitScope	o	x	x	x								
ibm-idsTRudpQueueSize	o	x	x	x								
ibm-idsFSInterval[c]	x	o	o	x								
ibm-idsFSThreshold	x	o	o	x								
ibm-idsSSInterval	x	o	o	x								
ibm-idsSSThreshold	x	o	o	x								

a. The TCP/IP stack can only detect single scan events. IDS keeps a tally of scan events and is better able to determine when a global scan has occurred.

b. If no local port (range) is given, the condition applies to all local ports.

c. The scan action directives (ibm-idsFSInterval, ibm-idsFSThreshold, ibm-idsSSInterval, ibm-idsSSThreshold) are assigned the default values if not specifically assigned values in the policy file.

Note: Directives that do not appear in the above table (ibm-idsMessageDest, ibm-ICMPRedirect, ibm-idsNotification, ibm-idsLoggingLevel, ibm-idsTypeActions, ibm-idsSensitivity, ibm-idsScanExclusion) are ignored.

11.3.1 Examples of IDS policy conditions and actions

This section provides a few examples of IDS policy conditions and actions depending on attack types.

Sample flood policy

Consider the condition and action for a flood attack in Figure 11-6.

```
 Edit File: /qibm/userdata/os400/qos/etc/idspolicy.conf
 Record :       1   of      13 by   8           Column :    1     59 by  74
 Control :

CMD ....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+
     ************Beginning of data**************
    ibm-idsConditionAuxClass        idscond1
    {
    ibm-idsConditionType            ATTACK
    ibm-idsAttackType               FLOOD
    ibm-idsLocalPortRange           1-65535
    ibm-idsLocalHostIPAddress       2-9.000.000.000-8
    ibm-policyIdsActionName         idsact1
    }
    ibm-idsActionAuxClass           idsact1
    {
    ibm-idsActionType               ATTACK
    ibm-idsMaxEventMessage          25
    }
     ************End of Data*******************

 F2=Save   F3=Save/Exit   F12=Exit   F15=Services   F16=Repeat find
 F17=Repeat change  F19=Left  F20=Right
```

Figure 11-6 IDSPOLICY.CONF file example: flood attack detection

This condition has a name idscond1. It describes a flood attack on any one of the local ports 1 through 65535 to the range of addresses 9.0.0.0 to 9.255.255.255. If any intrusions fit this description, then the action named idsact1 is taken. The action simply states that an event should be signaled, provided that 25 events that call out this action have not already been signaled. (And 25 IM records should then be seen in the audit journal.)

In the case of a flood attack, once IDS is notified by the TCP/IP stack, the system most likely is under attack. Floods are only signaled after an aging connection attempt (SYN) is dropped from the queue of all incomplete connection attempts. The attack is signaled immediately.

Sample Traffic Regulation policy

Consider a Traffic Regulation (TR) condition/action combination in Example 11-2.

Example 11-2 IDSPOLICY.CONF file example: Traffic regulation

```
  ibm-idsConditionAuxClass     idscond2
  {
  ibm-idsConditionType         TR
  ibm-idsLocalPortRange        80
  ibm-idsProtocolRange         6
  ibm-idsRemoteHostIPAddress   2-9.124.1.0-24
  ibm-policyIdsActionName      idsact2
  }
  ibm-idsActionAuxClass        idsact2
  {
```

```
ibm-idsActionType            TR
ibm-idsStatInterval          10
ibm-idsTRtcpTotalConnections 1000
ibm-idsTRtcpPercentage       10
ibm-idsMaxEventMessage       50
}
```

The condition and associated action shown in Example 11-2 on page 170 indicate that an attention event is generated when the number of established TCP connections to port 80 (HTTP server) from the range of remote IP addresses 9.124.1.0 to 9.124.1.255, over a certain period of time (10 minutes), has either exceeded a preset limit (1000) of connections, or exceeded a preset percentage (10%) of the total number of established connections to the system. At the end of the interval, internal counts are reset and a new statistical interval begins. Note that notification only occurs when the conditions are met. Nothing is done dynamically to remedy the situation.

Sample Scan Policy

A sample condition/action combination is given for scan events in Example 11-3.

Example 11-3 IDSPOLICY.CONF file example: scan monitoring

```
ibm-idsConditionAuxClass     idscond3
{
ibm-idsConditionType         SCAN_EVENT
ibm-idsLocalPortRange        26-136
ibm-idsRemoteHostIPAddress   2-9.0.0.0-8
ibm-policyIdsActionName      idsact3
}
ibm-idsActionAuxClass        idsact3
{
ibm-idsActionType            SCAN_EVENT
ibm-idsFSInterval            1
ibm-idsFSThreshold           5
ibm-idsSSInterval            120
ibm-idsSSThreshold           10
ibm-idsMaxEventMessage       50
}
```

As shown in Example 11-3, a scan event will be signaled to the audit journal if the following conditions are met: a connection attempt is made to non-listening ports (that is, undemuxable SYNs) 26 to 136 from remote IP addresses in the range of 9.0.0.0 to 9.255.255.255, and such attempts number 5 or more for a 1-minute interval (fast scan), or such attempts take place at the rate of 10 every 120 minutes (slow scan). To distinguish between a fast scan and a slow scan, internal counts for both fast scans and slow scans are reset when a threshold is met. Also, the internal count for fast scans is reset when the fast scan interval expires. Similarly, the internal count for slow scans is reset when the slow scan interval expires. For non-listening ports, any connection attempt may be of interest to a system administrator, especially the slow scan case where a perpetrator may be trying to bypass defenses and sneak in under the radar.

11.4 Intrusion Monitor entries

When intrusions are signaled from the IDS task, an entry is made in the security audit journal. An example of an Intrusion Monitor (IM) entry is given in Figure 11-7.

```
                              Display Journal Entry

Object . . . . . :                  Library  . . . . . :
Member . . . . . :
Incomplete data  . . :   No          Minimized entry data :   *NONE
Sequence . . . . . :   46500
Code . . . . . :   T  - Audit trail entry
Type . . . . . :   IM - Intrusion monitor

            Entry specific data
Column      *...+....1....+....2....+....3....+....4....+....5
00001      'P2006-06-25-18.11.41.0911761105 169909.5.175.141  '
00051      '                                   027549.10.229.77'
00101      '                                    TR-TCPO233     '
00151      '                          á    ß    Ja  V(  ýâ; B '
00201      'M OnýÐ Ø Ø zÅ       K ©vÓ  '



                                                                Bottom
Press Enter to continue.

F3=Exit   F6=Display only entry specific data
F10=Display only entry details   F12=Cancel   F24=More keys
```

Figure 11-7 TCP TR event denoted by IM entry in the security audit journal (QSYS/QAUDJRN)

Referring to the Intrusion Monitor audit record layout in Table 11-3, the audit journal entry in Figure 11-7 on page 172 describes a Traffic Regulation event:

- With a *P* for *potential* intrusion (They are all potential intrusions.)
- occurring on 6/25/2006 at 18:11:41.091176
- detected by internal code point 1105
- and, though we cannot see it, within the address family of IPv4 (0x02)
- to local port 16990 and local IP address 9.5.175.141
- from remote port 2754 and remote IP address 9.10.229.77
- of the Traffic Regulation variety depicting an established TCP connection
- with an event correlator of 233 (used for debugging)
- and a suspected packet consisting of a 2-byte non-displayable length and up to 1000 bytes, which can best be viewed in hex format by pressing F11.

If the above intrusion appears over and over again from the same remote IP address, a system administrator could create an IP filter rule denying any more input from that address. IP filtering is not discussed in this chapter, but is one preventive measure that a system administrator could take to deny access to suspicious clients. (See the i5/OS System Information Center: Networking → Networking security → IP filtering and network address translation.)

Table 11-3 Intrusion Monitor audit record

Offset J5	Field	Format	Description
1			Heading fields common to all entry types.
610	Entry Type	Char(1)	The type of entry. **P** Potential intrusion event detected.
611	Time of Event	TIMESTAMP	Timestamp of when the event was detected in SAA® time stamp format.
637	Detection Point Identifier	Char(4)	This is a unique identifier for the processing location that detected the intrusion event. This field is intended for use by service personnel.
641	Local Address Family	Char(1)	Local IP address family associated with the detected event.
642	Local Port Number	Zoned(5,0)	Local port number associated with the detected event.
647	Local IP Address	Char(46)	Local IP address associated with the detected event.
693	Remote Address Family	Char(1)	Remote address family associated with the detected event.
694	Remote Port Number	Zoned(5,0)	Remote port number associated with the detected event.
699	Remote IP Address	Char(46)	Remote IP address associated with the detected event.

745	Probe Type Identifier	Char(6)	Identifies the type of probe used to detect the potential intrusion. Possible values include: ► **ATTACK** – attack action detected event ► **TR–TCP, TR-UDP**– Traffic Regulation event ► **SCANG** – scan global action detected event ► **SCANE**– scan event action detected event
751	Event Correlator	Char(4)	Unique identifier for this specific intrusion event. This identifier can be used to correlate this audit record with other intrusion detection information.
755	Event type	Char(8)	Identifies the type of potential intrusion that was detected. The possible values are: ► **MALFPKT** – malformed packet ► **FLOOD** – flood event ► **ICMPRED** – ICMP (Internet Control Message Protocol) redirect ► **PERPECH** – perpetual echo ► **IPFRAG** – IP fragment ► **RESTOPT** - restricted IP options ► **RESTPROT** – restricted IP protocol
763	Reserved	Char(20)	Reserved for future use
783	Suspected Packet	Char(1002)[1]	This is a variable length field which may contain up to the first 1000 bytes of the IP packet associated with the detected event. This field contains binary data and should be treated as though it has a CCSID of 65535.

Intrusion Monitor entries in the audit journal may be viewed by entering the following command:

```
DSPJRN QAUDJRN ENTTYP(IM)
```

This command can be specified with multiple parameters, which can help limit the number of IM entries returned. For example, the most recent entries could be viewed by specifying a value for the FROMTIME parameter that was after the time that the last DSPJRN command was run. A CL program could be written to exploit these capabilities and return only the latest entries that have been added to the audit journal. This is an exercise left to the reader.

11.5 Verifying IDS policy implementation

To verify whether your IDS policy is active, complete the following steps:

1. Verify whether jobs QTOQRAGENT and QTOQSRVR are active in subsystem QSYSWRK.
2. If QTOQRAGENT is not active, review its job log to help identify policy directive errors.

To verify whether your IDS directives are correct, do the following:

1. Send datagrams that match the IDS policy directives that you are monitoring.
2. Review the audit journal for IM type entries.

Using nmap 4.11 you can generate datagrams to send to your i5/OS host for IDS policy verification.

Note: For more information regarding nmap, go to:

http://www.insecure.org/nmap/

The following is a scan example that generates both attack and TCP traffic regulation notifications:

```
nmap -sO i5oshostname
```

The next example uses a SYN-scan directly to an i5/OS host and causes IDS policy notifications to be generated:

```
nmap -sS i5oshostname
```

The following example uses a zombie with stealth idle scan to generate scan event notifications. Note that notification logs will only show the source as the zombie host:

```
nmap -P0 -sI zombiehost i5oshostname
```

To assist in mining the audit journal for intrusion monitor records, use the following i5/OS commands:

- `CPYAUDJRNE IM`

  This will copy IM entries, if any, to qtemp/qauditim.

- `RUNQRY *NONE QAUDITIM`

  This will query the IM entries from qtemp/qauditim.

11.6 Tips and techniques

The following are a few tips and techniques for running i5/OS IDS:

- It is important to verify that IDS policies are continually maintained.
- Remember, inbound datagrams may be stopped by active packet filtering rules, preventing them from reaching IDS for processing.
- Ensure that your policy action directive criteria are met. Thresholds and intervals can be overlooked when attempting to identify why datagram traffic did not trigger a policy directive notification.
- Only TCP/IP scans to non-listening ports and spoofing attempts are signaled for IDS *scan* processing. UDP scans are not detected.

11.7 i5/OS intrusion detection and prevention — a summary

The Intrusion Detection System on the i5/OS system is an integrated, host-based, highly secure and yet flexible notification system. Potential intrusions are presented as events in the system security audit journal. These potential intrusions may signify that a firewall is not doing its job and may need reprogramming by a network administrator. On the other hand, the intrusions may signify that the firewall is doing its job, within its limitations, and that the host-based IDS is catching intrusions that have sneaked through the firewall. At any rate, the i5/OS system IDS can stand alone as an intrusion detection system or be used in conjunction with a firewall for even greater security and peace of mind.

This chapter has primarily dealt with intrusion detection. With i5/OS system IDS, IBM has extended its capabilities in network security to place it at the vanguard of the IT industry.

Part 2

Scenarios

In Part 1, “Dynamic IP,” we detailed a wide range of functions built into i5/OS that can help you automate your own TCP/IP network. We did this function-by-function to detail the characteristics of that particular function.

In this part of the book, we define problems and their solutions and demonstrate step-by-step how to apply these TCP/IP protocols to solve your networking problems. In some cases our scenarios are simple and encompass just a single function. But, as the scenarios get more and more complex, the solutions will start to include a mixture of functions that interoperate and rely on each other.

Each scenario in this part follows a similar style:

Problem definition We define the problem from the customer’s point of view. A network diagram enables you to quickly match our scenario with your existing network (or subnet). Scenarios can be built on other scenarios, so that the problem solved in one scenario can be used as the basis for the next scenario.

Solution definition We define the solution. At first the solution might be very simple. As you progress through this book, the solutions become more complex and require a mixture of the different functions defined in Part 1.

How to Step-by-step instructions show how to use the iSeries Navigator to implement the solution.

Before you enter any of the scenarios in this part ensure that the System i and clients meet these criteria:

- Ensure that the System i has the LAN or WAN adapters necessary for the particular scenario. Some of the scenarios require multiple adapters:
  - Ensure that for each network adapter on the System i there is a line description. Verify basic functionality by varying on the corresponding line description.

  - Ensure that for each line description that describes a network adapter on the System i there is at least one IP address defined, as defined by the scenario.

- Ensure that all of the clients in the network have network interface cards installed and properly configured:
  - In this book most of our client systems are PCs running Windows XP. But, in some cases we show you how to configure Linux and other non-System i servers. Before you attempt to re-create the scenario environment, make sure you have all of the necessary hardware and software as defined in the scenario.

12

Defining adaptable TCP/IP interfaces and routes

There is an ever-increasing demand for the performance and availability of the System i. Its interfaces and routes can be used to achieve this demand and can be configured to provide fault tolerance and load balancing for LAN and WAN adapter cards.

In this chapter we provide these scenarios:

12.1 Fault tolerance: virtual IP with RIPv2

This scenario describes how RIPv2 and a virtual IP address (VIPA) may be used to provide fault tolerance in the event of an adapter failure.

Note: Proxy ARP for VIPA was made available at V5R2 of OS/400. This feature enables a VIPA to select a physical interface as a proxy agent. If there is a failure in that physical interface the VIPA will automatically move to another physical interface (if available). In this case, RIPv2 would not have to be used. An example scenario using proxy ARP can be seen in 12.2, “Fault tolerance: proxy ARP for the virtual IP address” on page 189.

Problem definition

The availability of your System i is of great importance, as it is used 24 hours a day, seven days a week. The System i handles a high volume of TCP/IP traffic. You would like to keep your system available to remote users in the event of a physical interface failure.

Figure 12-1 shows two physical interfaces residing on the same subnet. Because TCP/IP connections are made (bound) to specific interfaces, if that interface fails, then all TCP/IP connections (and the applications they support) will also fail.

How can we keep your system availability through the loss of a physical interface?

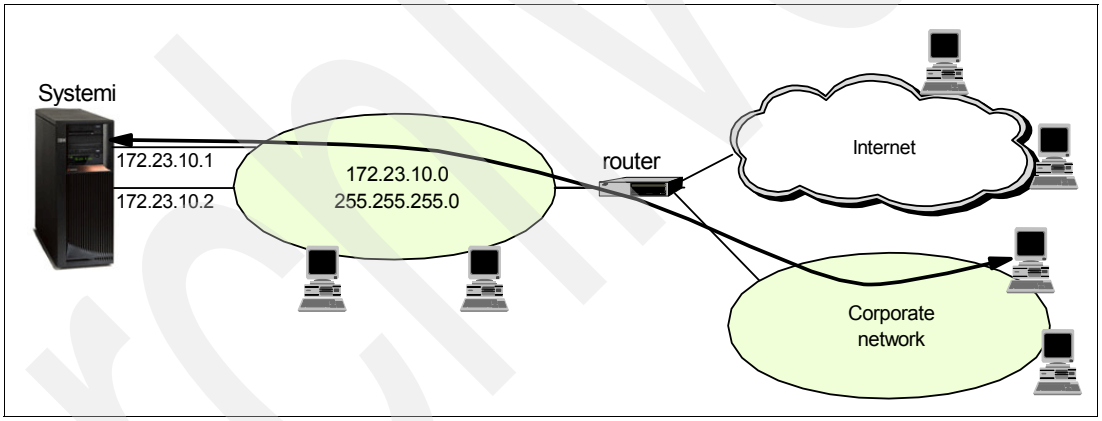


Figure 12-1 If one of the physical interfaces fails, so will all TCP/IP connections bound to the interface

Solution definition

The System i virtual IP address and RIPv2 may be used in combination to provide network adapter fault tolerance.

VIPA provides an address that is system-based and is reached via any physical interface.

RIPv2 (RouteD) provides a means of informing other hosts in the network of the best route to the virtual IP address. The RouteD server on the System i will broadcast route information for the VIPA. If a physical interface goes down, the RouteD server advertises that the route via the failed physical interface is no longer available and shows that the VIPA is accessible through the other interface. The routers on the network adds a RIP route for the other interface as a way to access the VIPA. Local hosts on the 172.23.10.0 subnet can take advantage of this fault tolerance by configuring routes for the VIPA that point to the router as the next hop or gateway host.

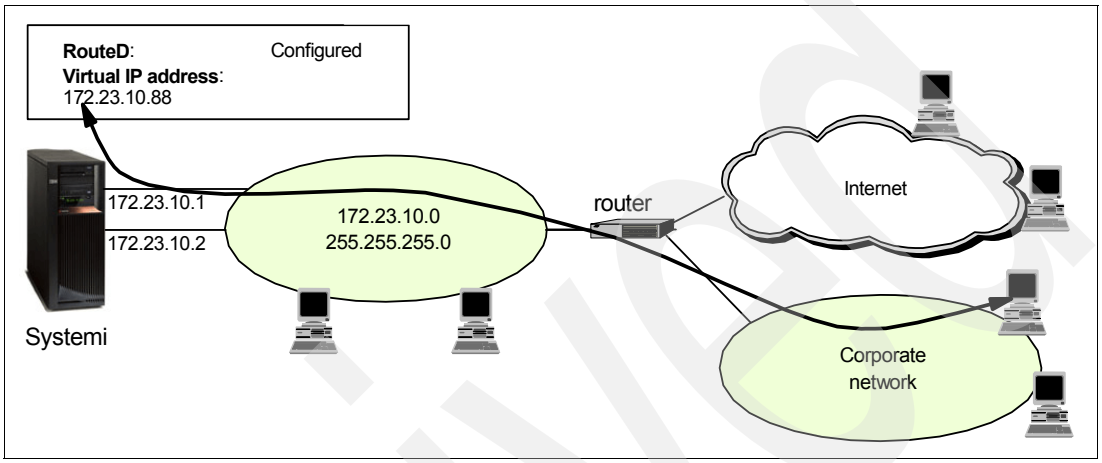


Figure 12-2 If one of the physical interfaces fails - the connection will be re-routed through the other

Alternative solution

The ability to use proxy ARP for Virtual IP is available in V5R2. This simplifies the use of VIPA for the purpose of fault tolerance for local hosts. This is discussed in 12.2, “Fault tolerance: proxy ARP for the virtual IP address” on page 189.

How-to

For this scenario we assume that you have already:

1. Configured multiple physical interfaces for the System i on each system
2. Installed iSeries Navigator on a PC that is connected to each System i in your network

Here are the steps necessary to configure fault tolerance through the use of a VIPA and RouteD:

- Step 1: Create new Virtual IP interface.
- Step 2: Configure the RouteD server and activate it.
- Step 3: Test the configuration with another System i.

Step 1: Create new Virtual IP interface

Create a virtual IP address on the System i:

1. Start the iSeries Navigator by clicking **Start** → **Programs** → **IBM iSeries Access for Windows** → **iSeries Navigator**. The iSeries Navigator window appears.
2. Expand your System i. You may be asked to enter your user ID and password.
3. Expand **Network** → **TCP/IP Configuration** → **IPv4**.

4. Right-click **Interfaces** and select **New Interface** → **Virtual IP** as shown in Figure 12-3.

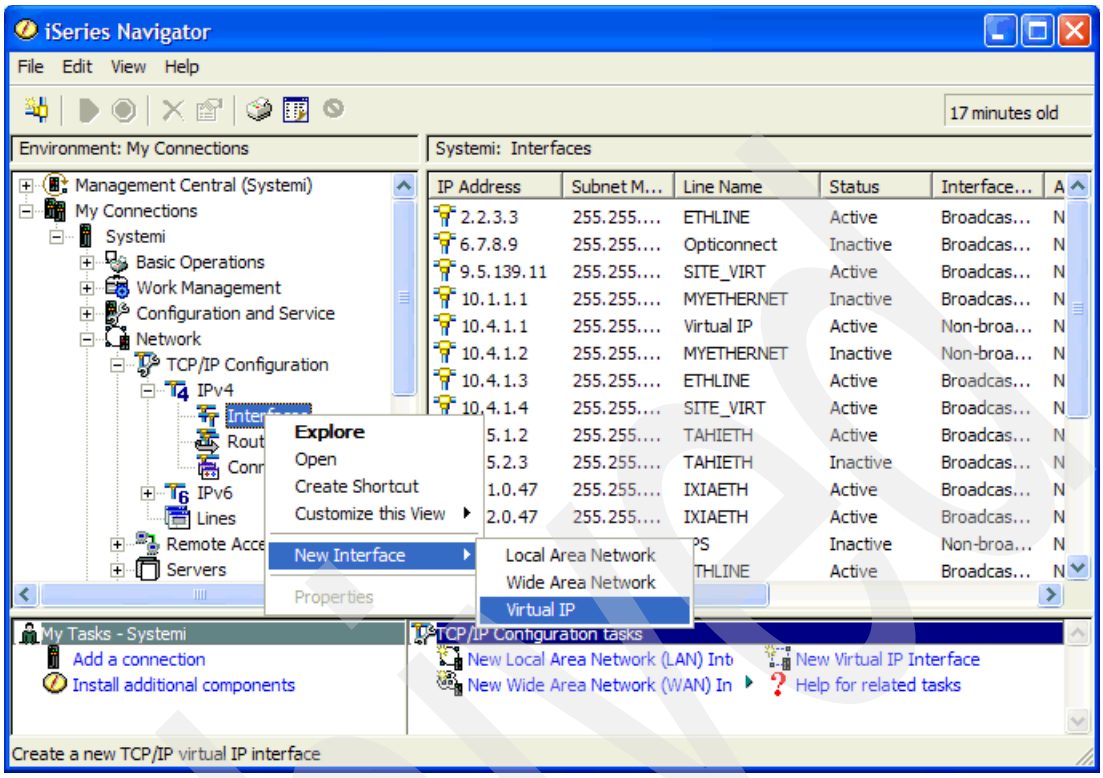


Figure 12-3 Interfaces: New Interface: Virtual IP

5. The Welcome window of the Interface Wizard appears as shown in Figure 12-4. Select **Next**.

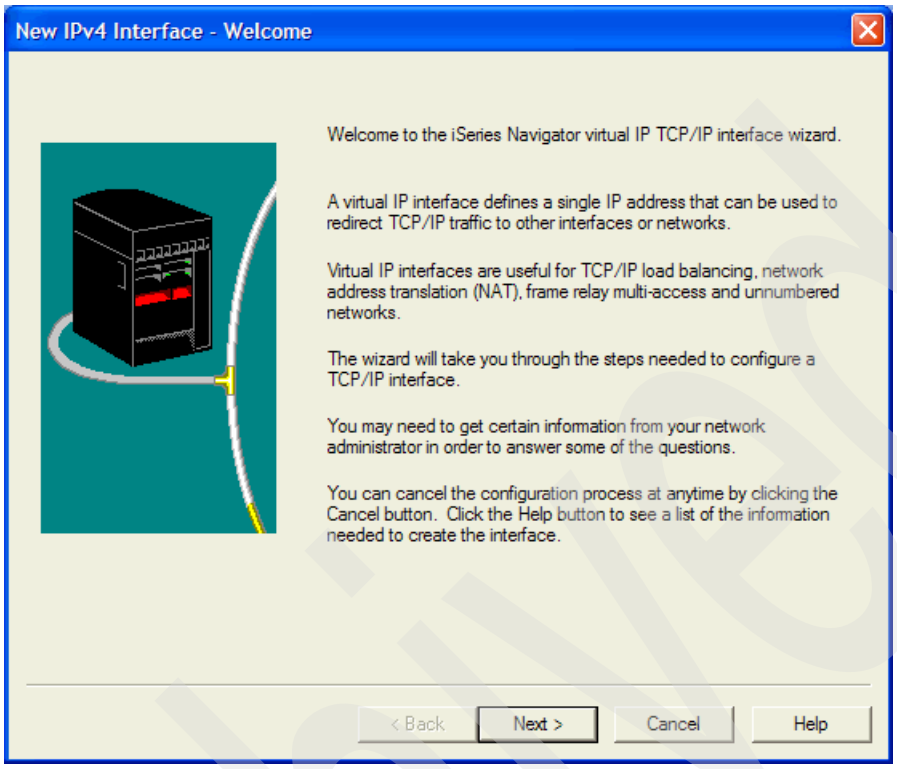


Figure 12-4 Virtual IP: Welcome window

6. From the Settings window, enter the VIPA `172.23.10.88`, a description for the interface (used in iSeries Navigator only) and the subnet mask `255.255.255.255` as shown in Figure 12-5. The subnet mask should be 255.255.255.255 to avoid routing ambiguity. Click **Next**.

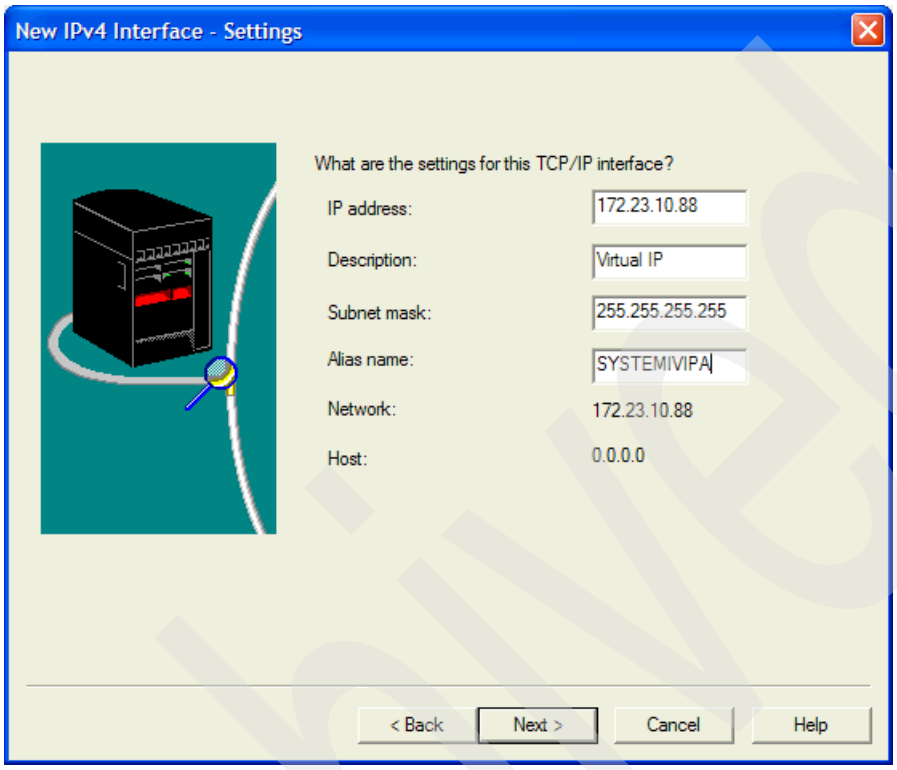


Figure 12-5 Virtual IP: Settings

7. The Start TCP/IP window appears as shown in Figure 12-6. Accept the default values and click **Next**.

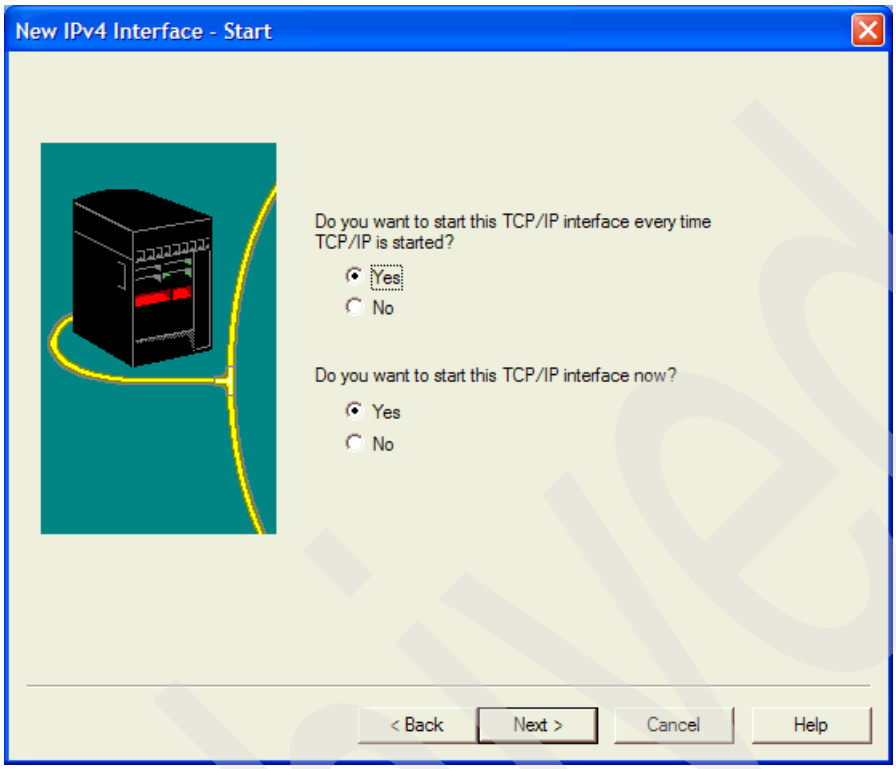


Figure 12-6 Virtual IP: Start

8. The Summary window appears as shown in Figure 12-7. It shows the values that have been selected in the creation process.
9. Click **Finish**.

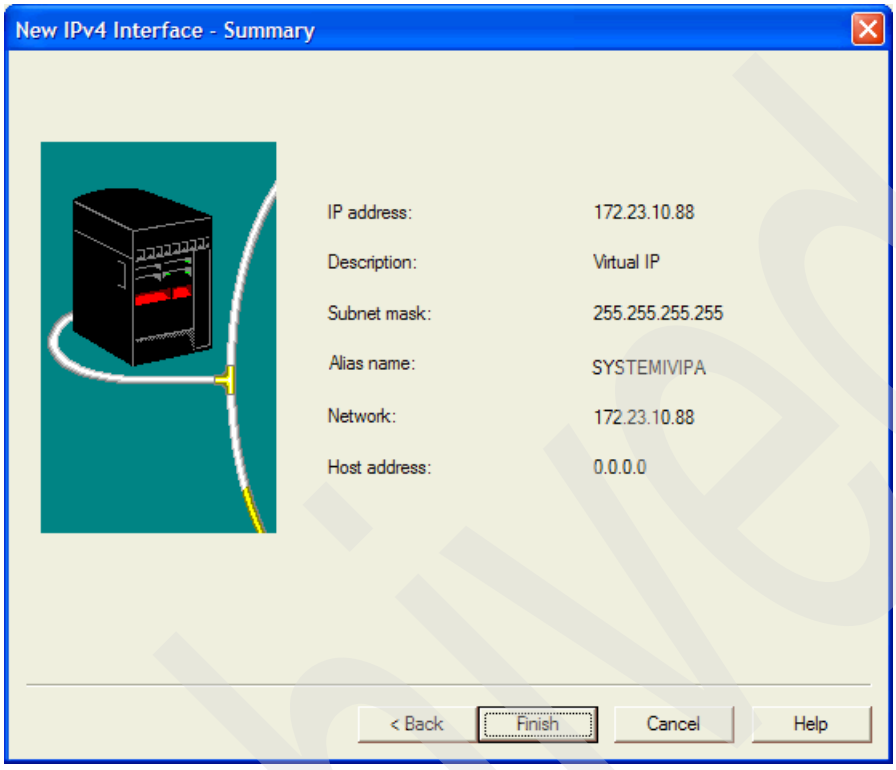


Figure 12-7 New IPv4 Interface Summary

10. The Test TCP/IP Interface window appears as shown in Figure 12-8. Click **Test now** to send a single ICMP Echo Request (PING) to the 172.23.10.88 interface. This test should be successful. Click **OK** to complete the steps to creating a new VIPA.

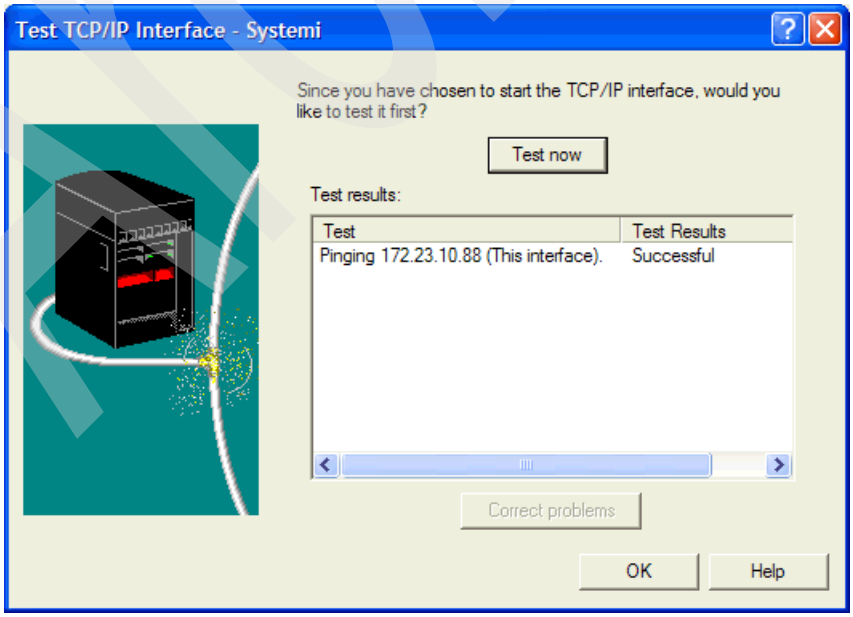


Figure 12-8 Virtual IP: Test TCP/IP Interface

Step 2: Configure the RouteD server and activate it

In this step we configure and start the RouteD server to enable the propagation of route information for the VIPA:

1. Starting from iSeries Navigator, expand **Network** → **Servers** and click **TCP/IP**.
2. Right-click **RouteD** and select **Properties**.
3. The RouteD Properties window appears as shown in Figure 12-9. Ensure that both boxes on the General tab are checked. The first one causes RouteD to start when TCP/IP is started. The second option enables RIP packets to be sent by the System i.

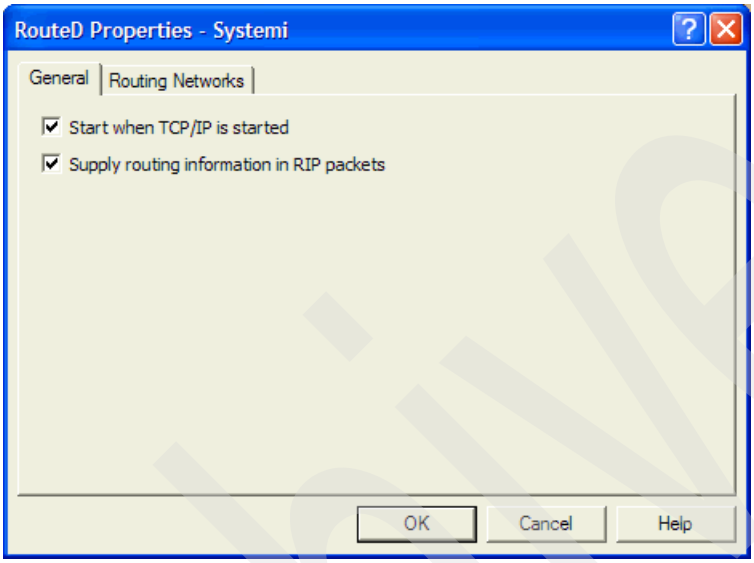


Figure 12-9 RouteD: RouteD Properties: General tab

4. Click the **Routing Networks** tab. The window in Figure 12-10 appears. We now define the physical interfaces that RouteD will use for sending out RIP updates. Click **New**.

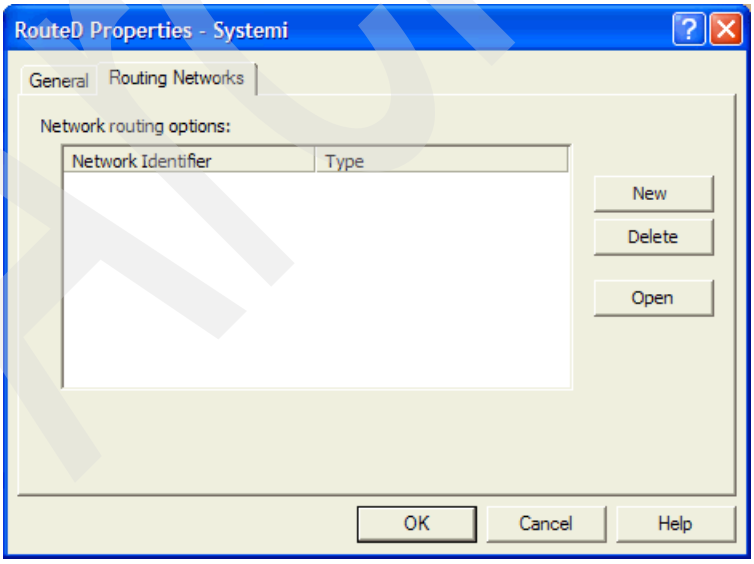


Figure 12-10 RouteD: RouteD Properties: Routing Networks

5. The New Routing Network Properties window appears as shown in Figure 12-11. Select the **Network** radio button. Enter your first physical interface address, `172.23.10.1`. You may leave the Subnet mask field blank. Click **OK** to continue. Perform this step again for the second interface, `172.23.10.2`. When finished, click **OK** to save the configuration.

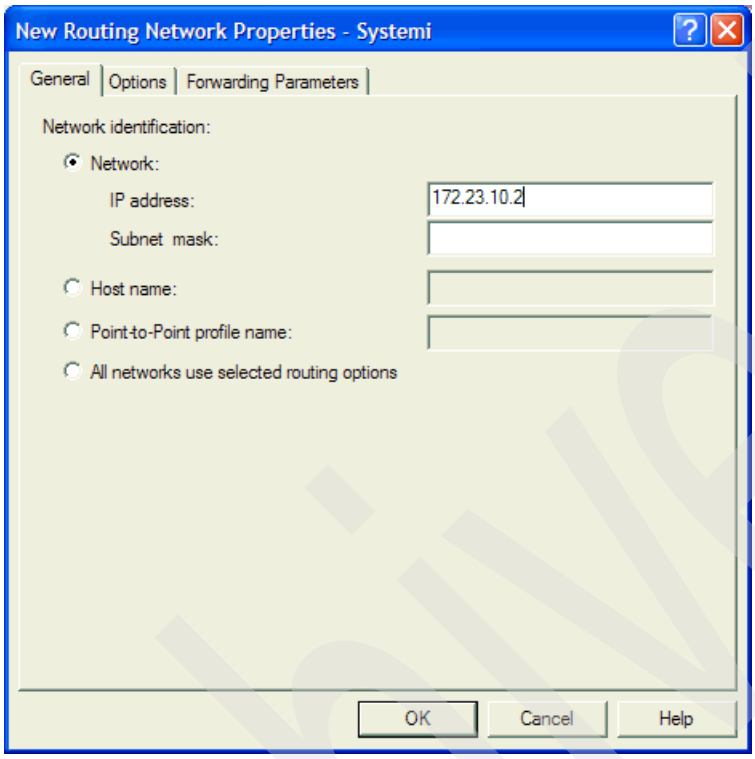


Figure 12-11 RouteD: New Routing Network Properties: Network: IP address: 172.23.10.2

6. Right-click **RouteD** and select **Start**. The RouteD server should go to a status of `started`.

Step 3: Test the configuration with another System i

The Virtual IP interface has been created and RouteD server has been configured and started. RIP entries sent by the System i RouteD server should now appear on other RouteD servers on the network. There should be a RIP host route for the VIPA. We have another System i with a RouteD server residing on the same network as the System i RouteD server. Using iSeries Navigator, we find that a RIP host route has been added for 172.23.10.88.

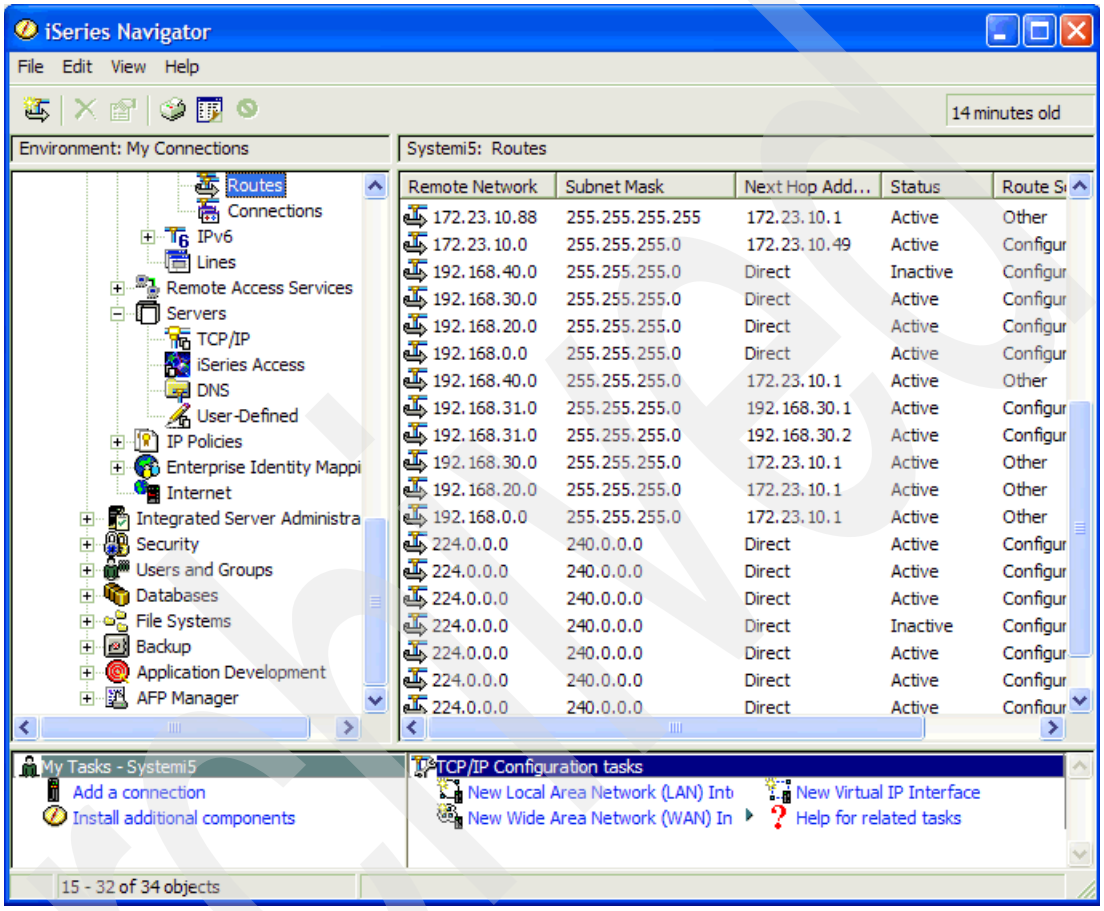


Figure 12-12 New RIP host route shown for 172.23.10.88

Review, conclusions, and references

The use of VIPA and RouteD together at pre-V5R2 allow for fault tolerance if a physical interface goes down.

12.2 Fault tolerance: proxy ARP for the virtual IP address

The ability to incorporate fault tolerance for locally attached hosts has been difficult to implement. Beginning at V5R2, the System i has made the process easier through the introduction of proxy ARP for virtual IP address. This expanded support also simplifies fault tolerance for remote hosts as well.

Problem definition

The system currently implements fault tolerance for the loss of a physical interface. Because the virtual IP address (VIPA) is not directly routable, the local hosts must manually configure

either host route to the VIPA or point to the local routers (as shown in Figure 11-15) as the means by which to access the VIPA. The routers are then configured with multiple indirect routes for the VIPA that have next hop addresses of the System i physical interfaces. Either way the implementation for local hosts is cumbersome and puts additional burden on the local routers.

Is there an easier way in which to implement fault tolerance?

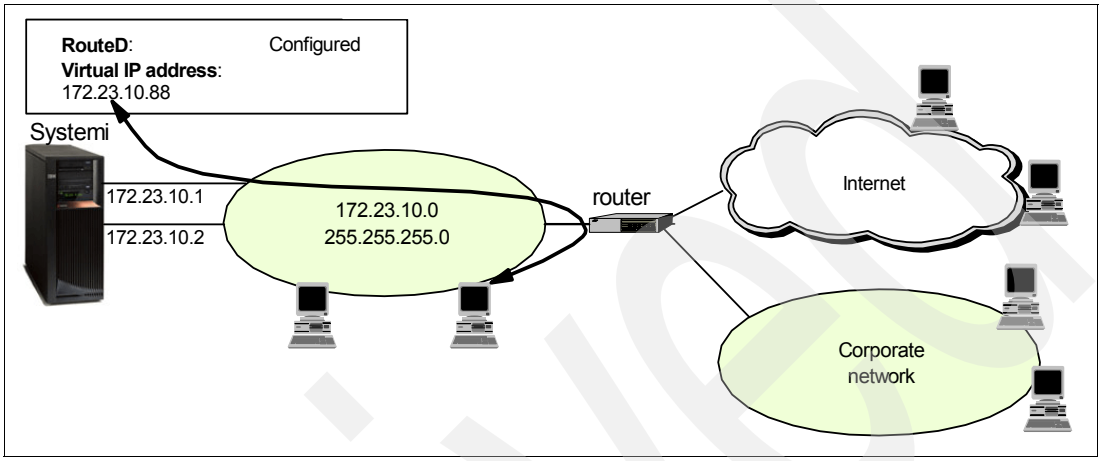


Figure 12-13 Local hosts must have extra indirect configuration to connect to Virtual IP 172.23.10.88

Solution definition

The implementation of fault tolerance for local and remote hosts is simplified through the use of proxy ARP. Proxy ARP enables the System i to respond to ARP requests made to the VIPA.

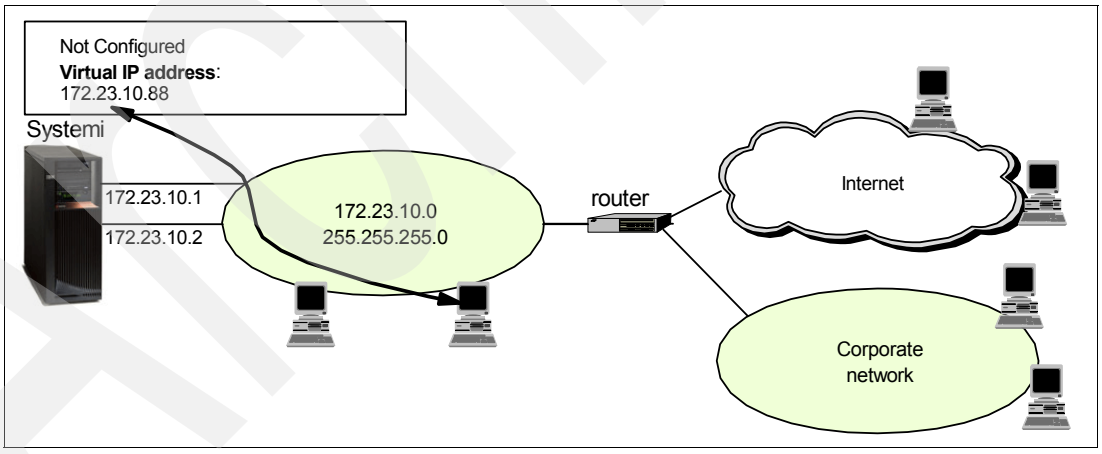


Figure 12-14 Local hosts ARP to discover virtual IP address172.23.10.88

How-to

Here are the steps to implement fault tolerance through the use of proxy ARP:

- Step 1: Enable proxy ARP for virtual IP address.
- Step 2: Test the configuration.
- Step 3: Using a preferred interface list.

Step 1: Enable proxy ARP for virtual IP address

These are the steps:

1. Start the iSeries Navigator by clicking **Start** → **Programs** → **IBM iSeries Access for Windows** → **iSeries Navigator**. The iSeries Navigator window appears.
2. Expand your System i. You may be required to enter a user ID and password.
3. Expand **Network** → **TCP/IP Configuration** → **IPv4**.
4. Click **Interfaces**.
5. Right-click your Virtual IP interface **172.23.10.88** and select **Stop**.

> **Tip:** While an interface is active, selecting Properties will only allow you to display values. You must *stop the interface*, then select **Properties** to be able to change values.

6. Right-click a Virtual IP interface and select **Properties**. The interface Properties window appears as shown in Figure 12-15.

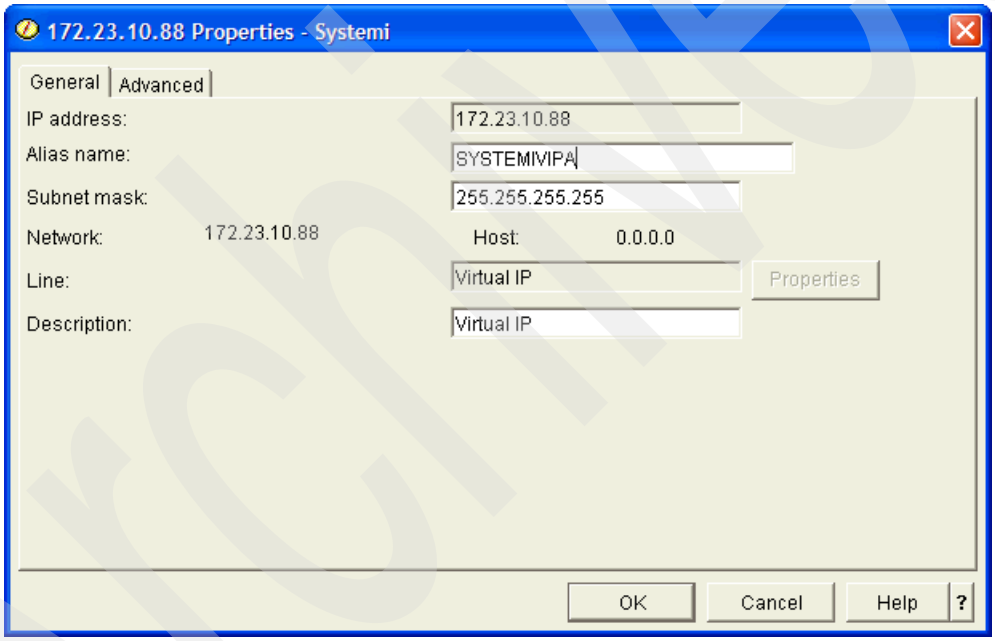


Figure 12-15 Virtual IP: Properties: General tab

7. Click the **Advanced** tab to open the window in Figure 12-16. Select the box next to **Enable proxy ARP**. Click **OK**.

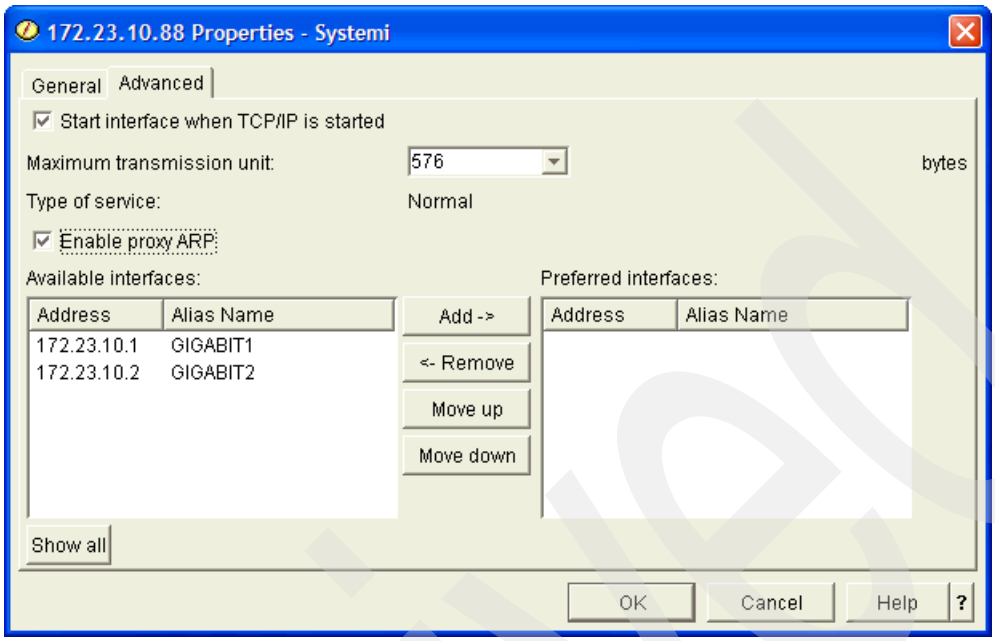


Figure 12-16 Virtual IP: Properties: Advanced tab

8. Right-click the Virtual IP interface and select **Start**. Proxy ARP has now been enabled for your virtual IP address. Refresh the display of all of the interfaces on your system. Figure 12-17 shows that your VIPA interface of 172.23.10.88 has picked an Associated Interface of either 172.23.10.1 (as pictured) or 172.23.10.2. Also note that proxy ARP is enabled for this VIPA.

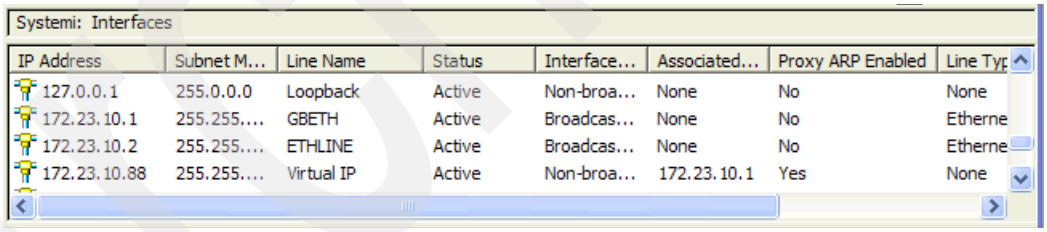


Figure 12-17 Proxy ARP for VIPA has selected an Associated Interface of 172.23.10.1

> **Tip:** See "How a MAC address is selected for a VIPA" on page 41 for a detailed description of how you can control which physical interface is selected to be your agent for proxy ARP by your VIPA.

Step 2: Test the configuration

It is now time to test the configuration. We will see whether another System i can contact the VIPA without a host route (as shown in Figure 12-14 on page 190):

1. Using iSeries Navigator on another system in the same network in which the VIPA resides, right-click **Networking** → **TCP/IP Configuration** and select the menu option **Utilities** → **Trace Route**. In this example, we use the system that corresponds to the IP address 172.23.10.49.

2. Enter the IP address of the VIPA and click **Trace**. The results should be similar to Figure 12-18.

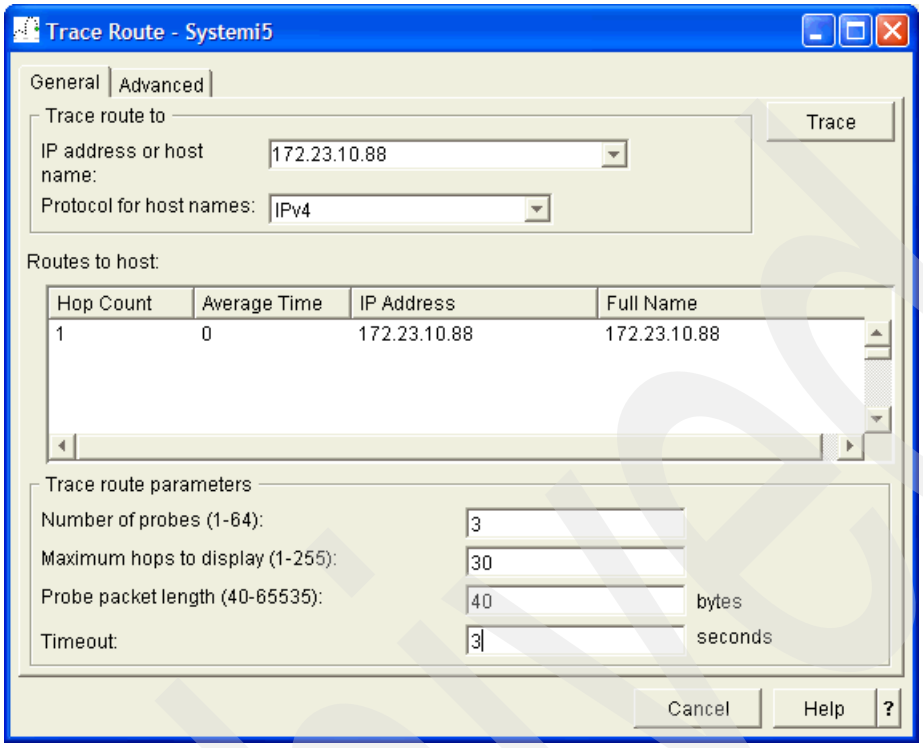


Figure 12-18 Trace Route for VIPA 172.23.10.88

3. The reply comes directly from the System i in which the VIPA resides. The System i is responding on behalf (proxy ARP) of the virtual IP address.
4. To test the fault tolerance nature of this proxy ARP for VIPA solution, start a Telnet session for IP address 172.23.10.49 and sign on to the System i.
5. From the i5/OS command line, start another Telnet session to the VIPA 172.23.10.88 and sign on.
6. On the System i for which the VIPA is configured, stop the interface that is currently acting as the agent for the proxy ARP. In our test this was 172.23.10.1.
7. You should now see that your VIPA interface of 172.23.10.88 has picked an Associated Interface of 172.23.10.2 (Figure 12-19).

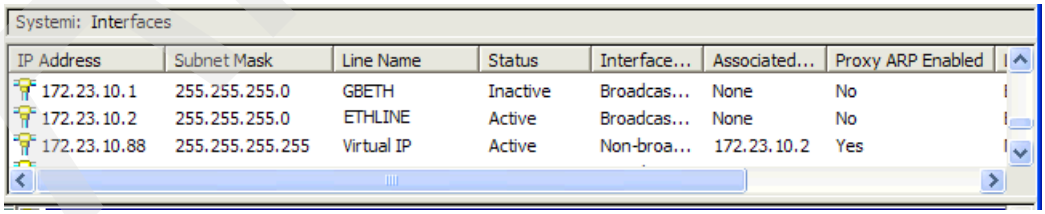


Figure 12-19 Proxy ARP for VIPA has selected an Associated Interface of 172.23.10.2

8. You will notice that even though your proxy agent changed, your Telnet session was not interrupted.

Step 3: Using a preferred interface list

These are the steps:

1. Start the iSeries Navigator by clicking **Start** → **Programs** → **IBM iSeries Access for Windows** → **iSeries Navigator**. The iSeries Navigator window appears.
2. Expand your System i. You may be required to enter a user ID and password.
3. Expand **Network** → **TCP/IP Configuration** → **IPv4**.
4. Click **Interfaces**.
5. Right-click your Virtual IP interface **172.23.10.88** and select **Stop**. Also, start the interface **172.23.10.1** interface if it is not active.
6. Right-click your Virtual IP interface **172.23.10.88** and select **Properties**.
7. Select the **Advanced** tab.
8. Move the interfaces 172.23.10.1 and 172.23.10.2 from the Available interfaces list to the Preferred interfaces list, as shown in Figure 12-20.

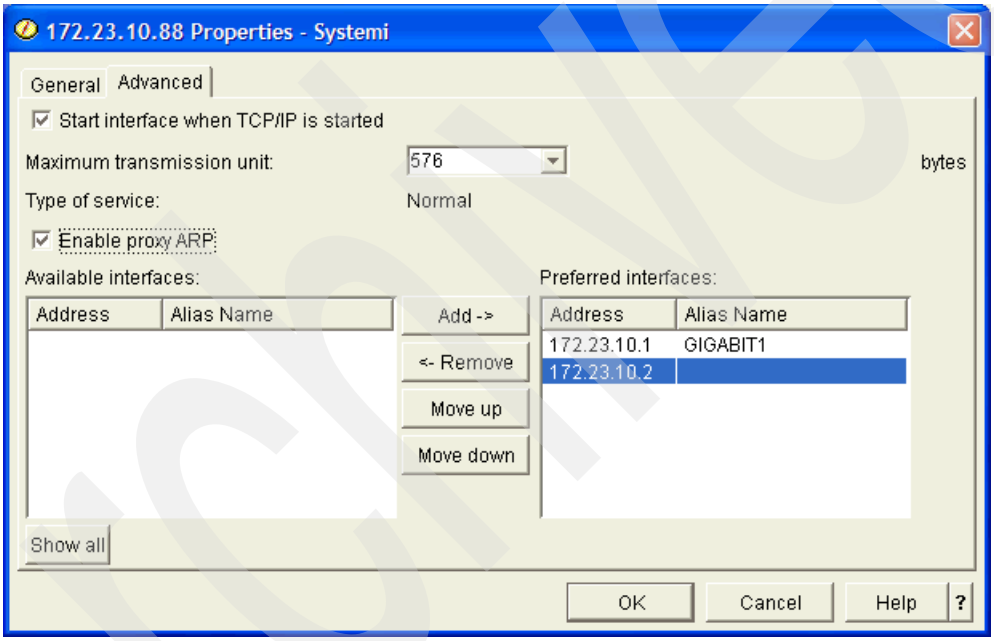


Figure 12-20 Configuring the preferred interface list.

9. Click **OK**. The VIPA 172.23.10.88 now has a preferred interface list configured. This means that the interface 172.23.10.1 will always be selected as the proxy agent if it is active. If 172.23.10.1 fails or is stopped for any reason, the interface 172.23.10.2 will become the proxy agent. Once 172.23.10.1 becomes active again, it will once again be the proxy agent.

Review, conclusions, and references

Proxy ARP for virtual IP addresses greatly simplifies the implementation of fault tolerance. It is a quick and easy way for applications on both local or remote hosts to stay up and running in the event of the failure of all but the last interface on the System i.

12.3 DNS-based inbound load balancing

Load balancing allows for splitting up TCP/IP traffic between different systems or adapters on a single system. DNS has a function that passes out multiple addresses for the same system name. This type of load balancing provides balancing by the connect request. DNS-based load balancing is oriented toward splitting the inbound traffic to the System i.

Problem definition

You have a System i to which both local and remote hosts are continuously connecting. Your System i has two physical interfaces. Your System i also has a DNS server configured to resolve the host name AS20 to the IP address 172.23.10.1, as shown in Figure 12-21.

Therefore, all of the traffic bound for the System i is received on the 172.23.10.1 interface. As you have two physical interfaces and you would like to load-balance the inbound traffic, how can this be done?

Important: This scenario's focus is very narrow in its scope as it only is about inbound load balancing. You should be very concerned about network security. See 16.6, "Split DNS: Private and Public DNS with masquerade NAT" on page 466 for a DNS scenario that also includes security.

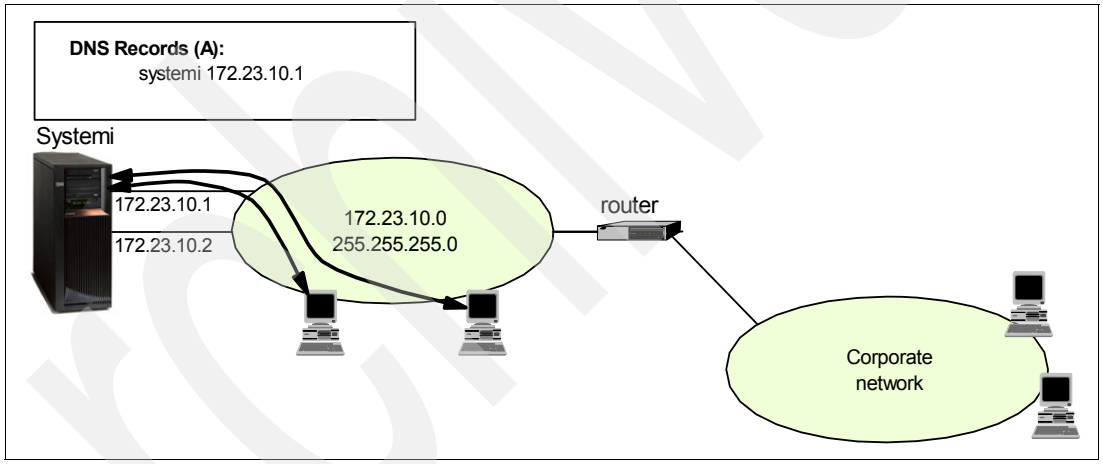


Figure 12-21 All clients resolve host name System i to IP address 172.23.10.1

Solution definition

The inbound traffic can be balanced through the use of the DNS server. The DNS can be configured to pass out multiple addresses for the same system name, and serves a different IP address each time a request is made for the Address (A) record for the system name. In most cases when a client has resolved the address, the client caches the address and will not ask again. This type of load balancing does not consider the amount of traffic going to each adapter, but only that each adapter receives an equal number of incoming connections. The addition of DNS load balancing can be seen in Figure 12-22.

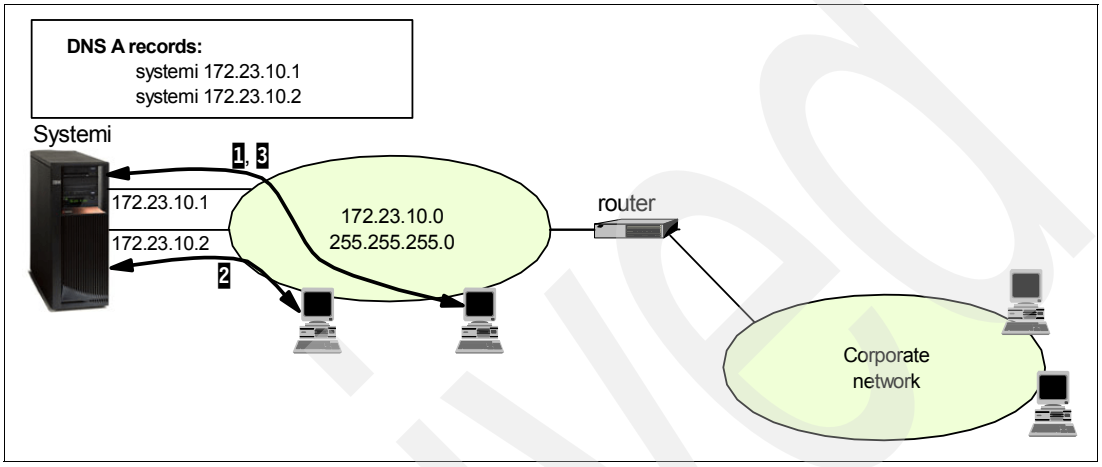


Figure 12-22 DNS first resolves System i to 172.23.10.1, then 172.23.10.2, then 172.23.10.1...

Note: It is possible that the first connection on 172.23.10.1 will be for the transfer of a very large file, the second connection on 172.23.10.2 for a light Telnet session, and the third on 172.23.10.1 for the transfer of another very large file. But, for connections numbering in the hundreds (if not thousands), the load on each of the interfaces should be relatively even.

How-to

Now we make the necessary changes to the System i DNS server to enable it to pass out multiple addresses for the same host name.

Attention: For this scenario, we assume that you have already configured a DNS server with a single A record for System i.

Here are the steps necessary to configure the System i DNS server so that it will give out addresses in a round-robin fashion to clients requesting name resolution. In doing this, inbound traffic to the System i will be balanced:

- Step 1: Configure System i DNS.
- Step 2: Test the configuration.

Step 1: Configure System i DNS

We add an additional A record for the system name:

1. Start the iSeries Navigator by clicking **Start** → **Programs** → **IBM iSeries Access for Windows** → **iSeries Navigator**. The iSeries Navigator window appears.
2. Expand your System i. This may require that you enter a user ID and password.
3. Expand **Network** → **Servers**.
4. Click **DNS**.

5. Right-click your existing DNS server, and select **Configuration** to open the window shown in Figure 12-23.

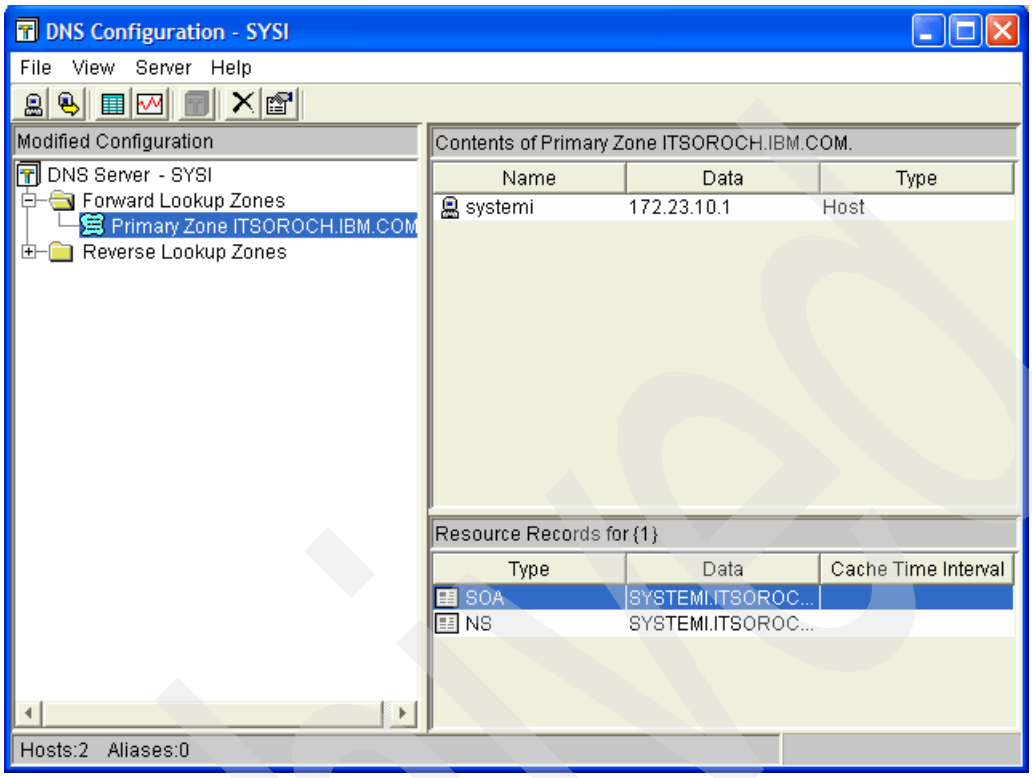


Figure 12-23 DNS configuration

6. Right-click **Primary Zone** and select **New** → **Host**.

7. The New Host wizard appears (Figure 12-24). In the Host domain name field, enter the system host name. Click **Next**.

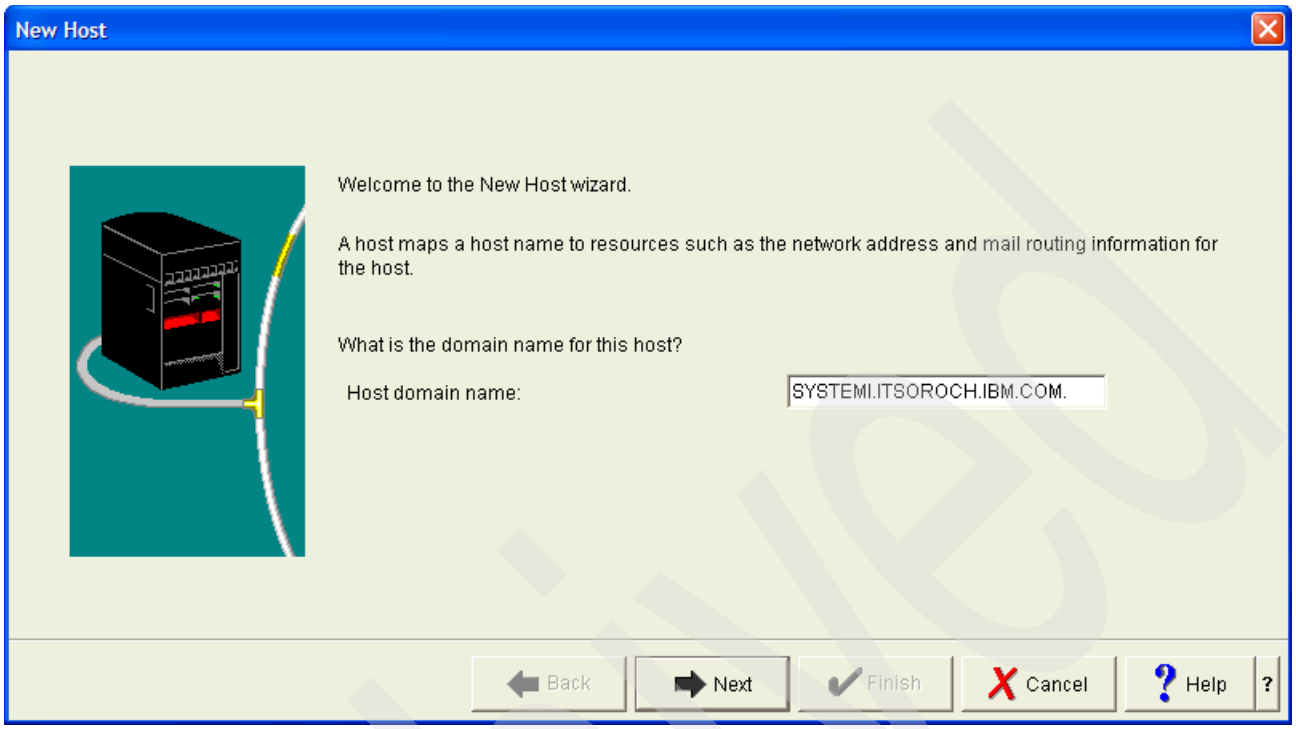


Figure 12-24 DNS Configuration: New Host

8. The New Host Resources window appears. Click **Add**.

9. In the Add/Edit Resource window (Figure 12-25), for Resource type, select **Address(A)**. In the IP address field, enter the IP address of the second physical interface that you want used. Select **Cache time interval**, and select a time period of **1 day**. Click **OK**.

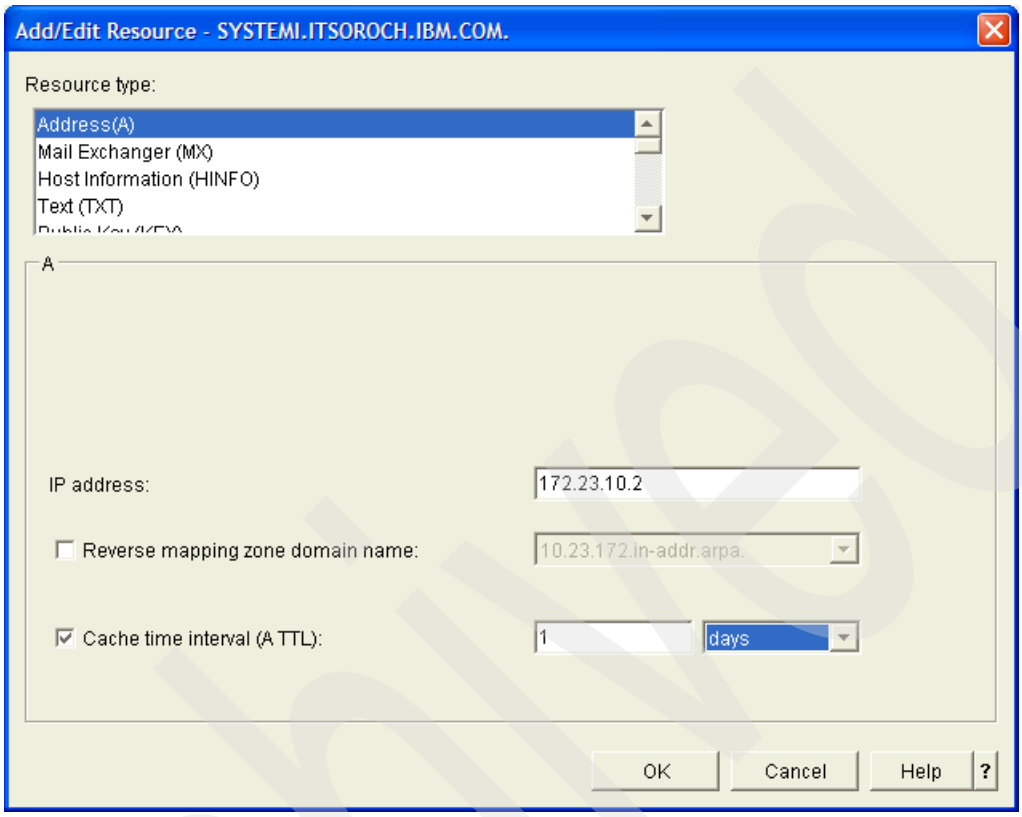


Figure 12-25 DNS Configuration: Add/Edit Resource: Resource Type

10. Click **Finish**.

11. Click the host name listed in the right-side pane. You should see two A records for the host name, as shown in Figure 12-26.

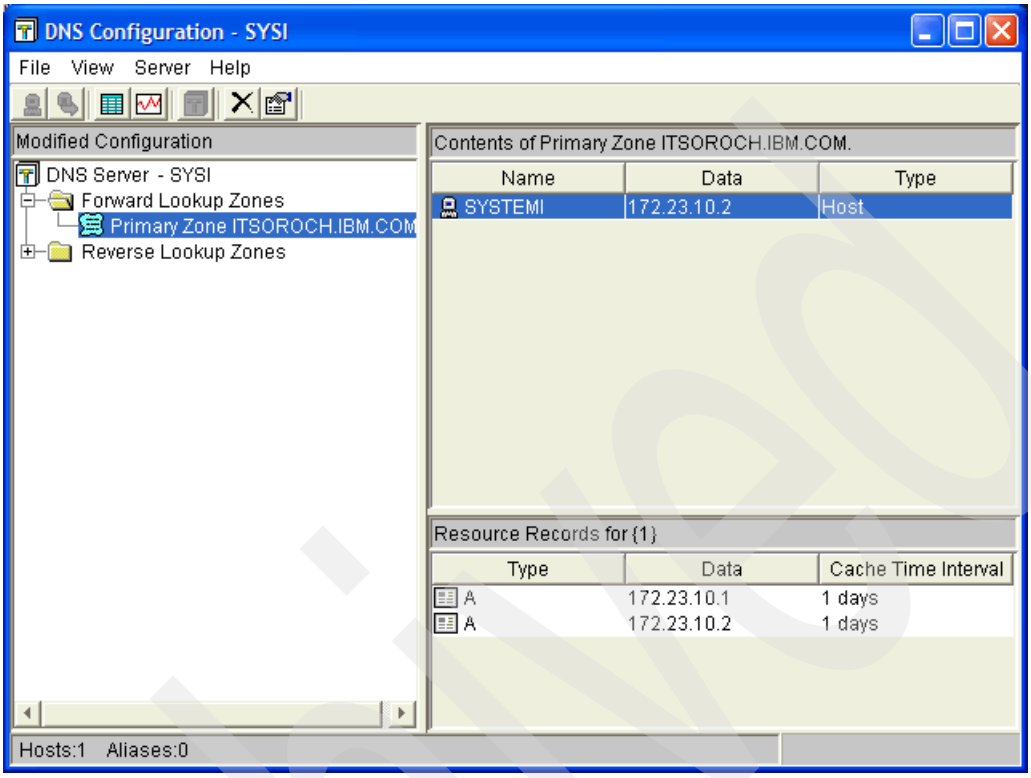


Figure 12-26 DNS Configuration: Primary zone itsoroch.ibm.com: as20

12. In the menu bar, select **File** → **Save Configuration** to save the changes.
13. Select **File** → **Close** to close the DNS Configuration window.
14. Right-click your DNS configuration and select **Start**. Your configuration is ready to test.

Step 2: Test the configuration

We verify that the System i DNS server is giving out multiple addresses in a round-robin process, as demonstrated in a series of commands from a Windows client:

1. From a Windows client command prompt panel, type `ipconfig /flushdns`. This purges the DNS resolver cache on the client.
2. Type `ping systemi.itsoroch.ibm.com`. The System i DNS server resolves the name to 172.23.10.1.
3. Type `ipconfig /flushdn`. This again purges the DNS resolver cache on the client.
4. Type `ping systemi.itsoroch.ibm.com`. The System i DNS server resolves the name to 172.23.10.2.

Review, conclusions, and references

We have shown how the System i DNS server can be configured to have multiple addresses for an A record. The System i DNS server will serve a different address each time a request is made for this A record in a round-robin fashion.

12.4 Outbound load balancing with duplicate route round-robin

In 12.3, “DNS-based inbound load balancing” on page 195, we demonstrated how load balancing can be done for inbound connections. Duplicate Route Round-Robin (DRRR) allows outbound load balancing across multiple physical interfaces on your System i.

Problem definition

One of the very important tasks for your System i is as an HTTP server that is being accessed by remote hosts who are downloading information about your company’s products. Currently, the HTTP traffic only occurs across one physical adapter, 173.23.10.1, as shown in Figure 12-27. Due to the nature of the HTTP requests, the outbound flow of data is heavier than the inbound data flow.

The System i has two additional physical interfaces. You would like to receive the data on one interface and send data on the other two interfaces, 173.23.10.2 and 173.23.10.3. You would like to have the outbound data flow balanced across the two outbound physical interfaces.

In addition, you would like to support in this manner traffic not only from the Internet and your corporate network through the router at 172.23.10.9, but also local traffic on the 172.23.10.0 subnet.

How can this be done?

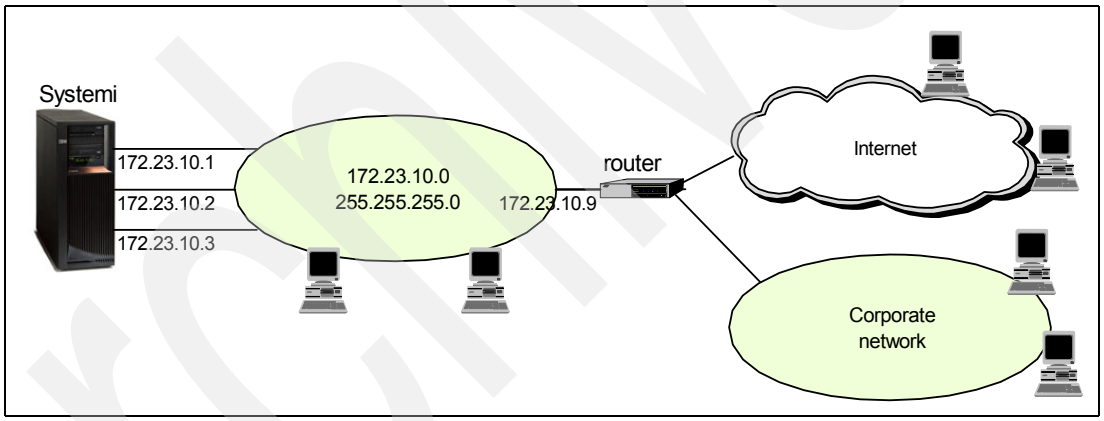


Figure 12-27 By default all traffic arriving on an interface will go back out the same interface

Solution definition

The problem can be solved by using DRRR. It enables outbound data to be balanced across two physical interfaces while data is received on a third. The diagram in Figure 12-28 shows the use of multiple duplicate routes with duplicate route priorities set to accomplish this.

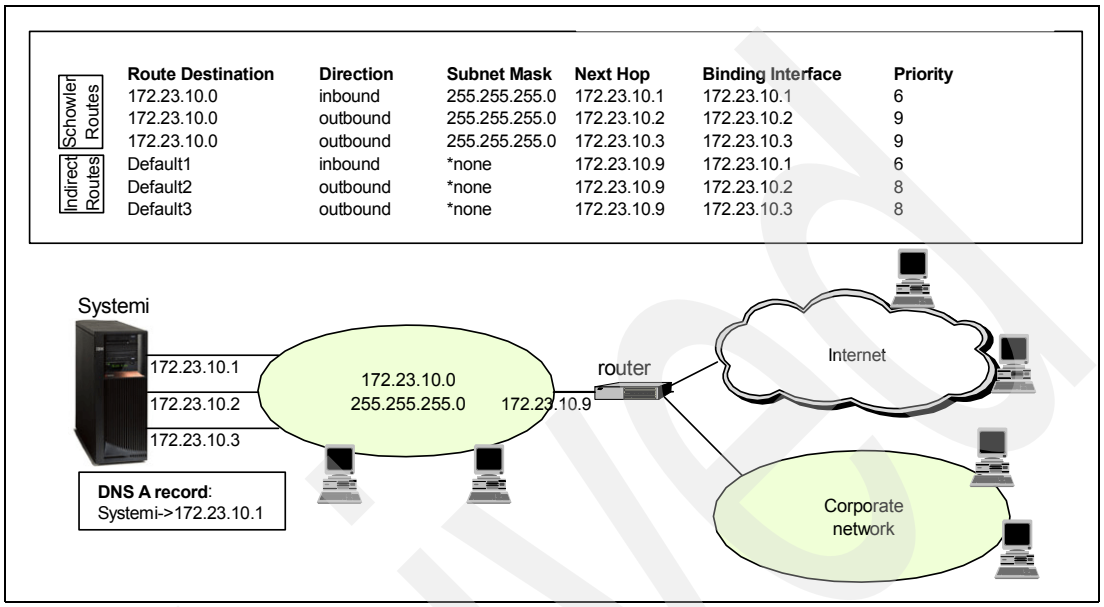


Figure 12-28 172.23.10.1 is for inbound traffic; 172.23.10.2 and 172.23.10.3 are for outbound

How-to

We assume that there is currently one default route and three direct routes that are automatically created for you when you created the interfaces for 172.23.10.1 through 172.23.10.3. We will delete this route and create three new duplicate default routes. The duplicate route priority and preferred binding interface parameter for these routes must correspond with one another.

As we also want to extend our outbound DRRR strategy for the local clients on the 172.23.10.0 subnet, you will create three Schowler routes to replace the direct routes for your three physical interfaces.

Here are the steps needed to set up DRRR on the System i for outbound traffic:

- Step 1: Create duplicate default routes for each physical interface.
- Step 2: Create duplicate Schowler routes for each physical interface.
- Step 3: Set the duplicate route priority for all routes.
- Step 4: Completion and test.

Step 1: Create duplicate default routes for each physical interface

You must to create a duplicate default route for each of the physical interfaces to allow IP datagrams to leave the System i and be sent to the primary gateway router, at 172.23.10.9.

The steps that follow create a new default route bound to the interface 172.23.10.1:

1. Start the iSeries Navigator by clicking **Start** → **Programs** → **IBM iSeries Access for Windows** → **iSeries Navigator**. The iSeries Navigator window appears.
2. Expand your System i. This may require that you enter a user ID and password.
3. Expand **Networ**k → **TCP/IP Configuration** → **IPv4**.

4. Right-click **Routes** and select **New Routes** from the context menu. The Welcome window for the route wizard appears. Select **Next**.
5. The Binding Interface window appears (Figure 12-29). In this step, the interface will be 172.23.10.1. We specify that we want to bind this new route to this physical interface. Click **Next** to continue.

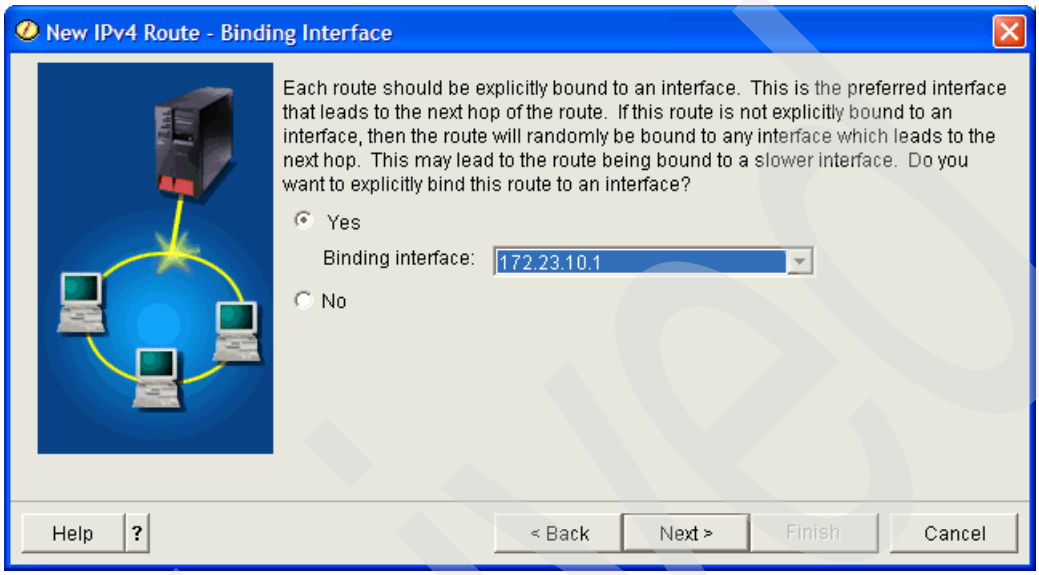


Figure 12-29 New IPv4 Route: Preferred Binding interface for 172.23.10.1

6. The Attributes window appears (Figure 12-30). We select the radio button for Default route. We also enter 172.23.10.9 for the Next hop address. This is the IP address of the gateway router (as seen in Figure 12-28 on page 202). Click **Next**.

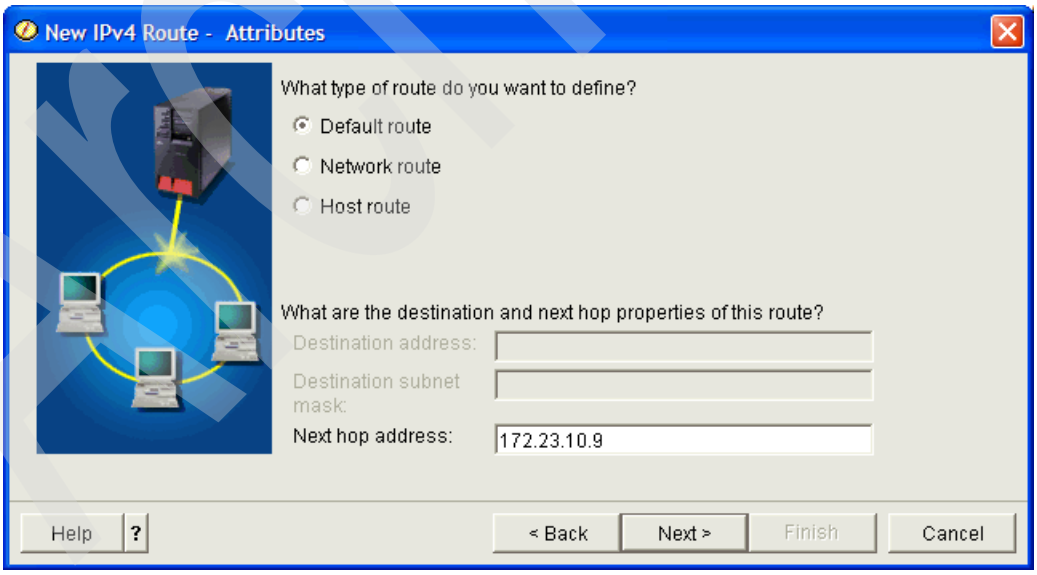


Figure 12-30 New IPv4 Routes: Attributes: Set New hop address to 172.23.10.9

7. The Summary window appears, as shown in Figure 12-31. Click **Finish**.

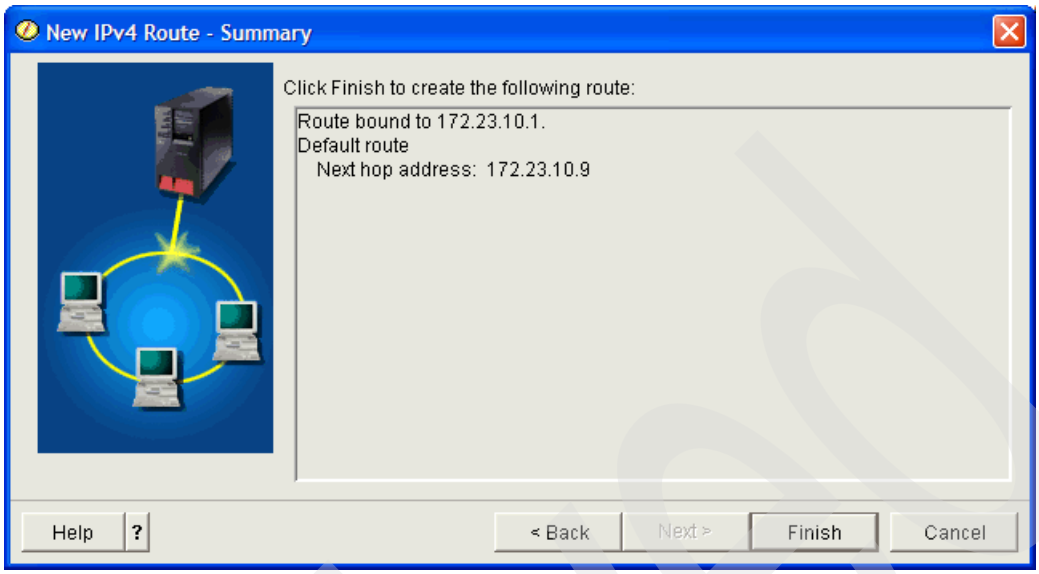


Figure 12-31 New IPv4 Route: Summary

These steps must be repeated for the other two interfaces, 172.23.10.2 and 172.23.10.3. That is, create two more duplicate default routes, each bound to the two interfaces 172.23.10.2 and 172.23.10.3.

Tip: Notice that the Create Route wizard does not include the option to specify the route priority. You have to change the route priorities after you create the routes as demonstrated in “Step 3: Set the duplicate route priority for all routes” on page 206.

Step 2: Create duplicate Schowler routes for each physical interface

Next create three duplicate Schowler routes for the three physical interfaces. These Schowler routes, when active, replace the direct routes already automatically created for you when you created the TCP/IP interfaces.

In the steps that follow, you will create a new network route bound to the interface 172.23.10.1:

1. Expand **Network** → **TCP/IP Configuration** → **IPv4**.
2. Right-click **Routes** and select **New Routes** from the context menu. The Welcome window for the route wizard appears. Select **Next**.

3. The Binding Interface window appears (Figure 12-32). In this step, the interface will be 172.23.10.1. We specify that we want to bind this new route to this physical interface. Click **Next** to continue.

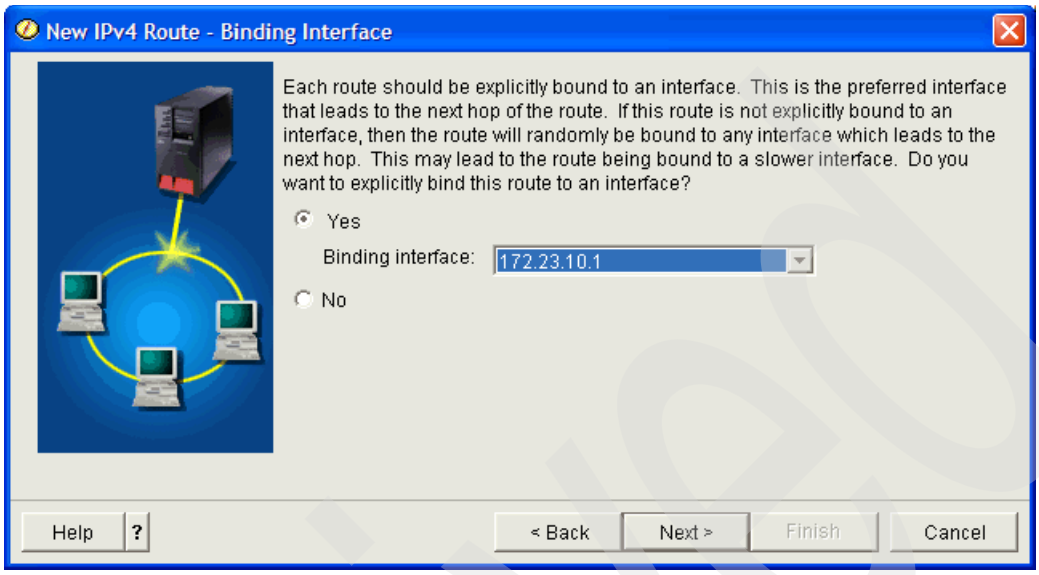


Figure 12-32 New IPv4 Route: Preferred Binding interface for 172.23.10.1

4. The Attributes window appears (Figure 12-33). Select the radio button for **Network route**.

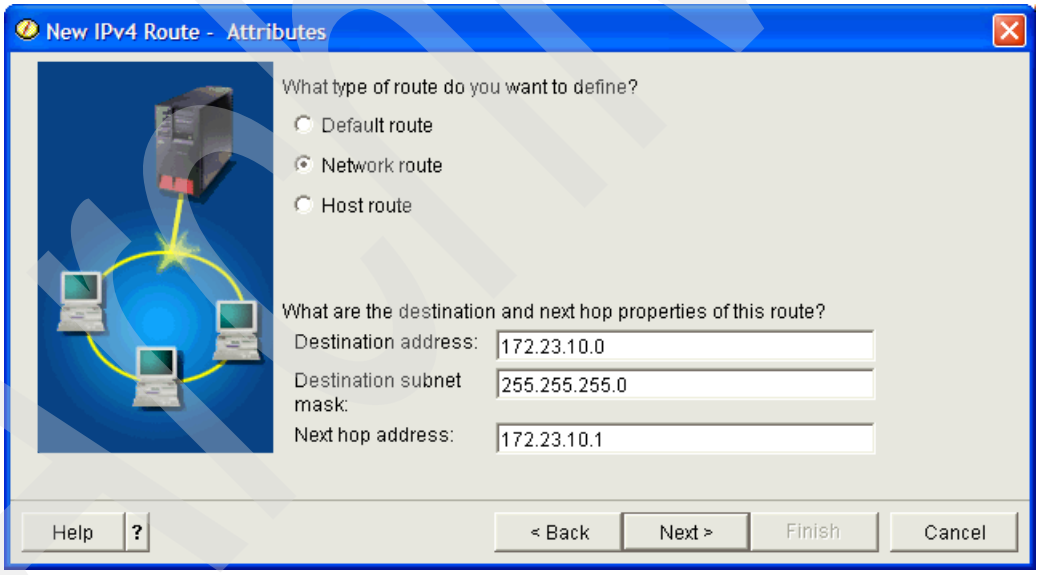


Figure 12-33 New IPv4 Routes: Attributes: Set New hop address to 172.23.10.1

The Destination address of `172.23.10.0` and Destination subnet mask of `255.255.255.0` represent the subnet to which the System i and all local clients are connected.

The Next hop address is the same address as the preferred binding interface specified in Figure 12-32.

This is a classic example of specifying a Schowler route on the System i, which is a network route that logically replaces a direct route. The advantage of using a Schowler route in this situation is that it enables both the remote and local hosts to take advantage

of the outbound load-balancing characteristics that you can specify in this new network route. For more information, see “Schowler routes” on page 52.

Click **Next**.

5. The Summary window appears (Figure 12-34). Click **Finish**.

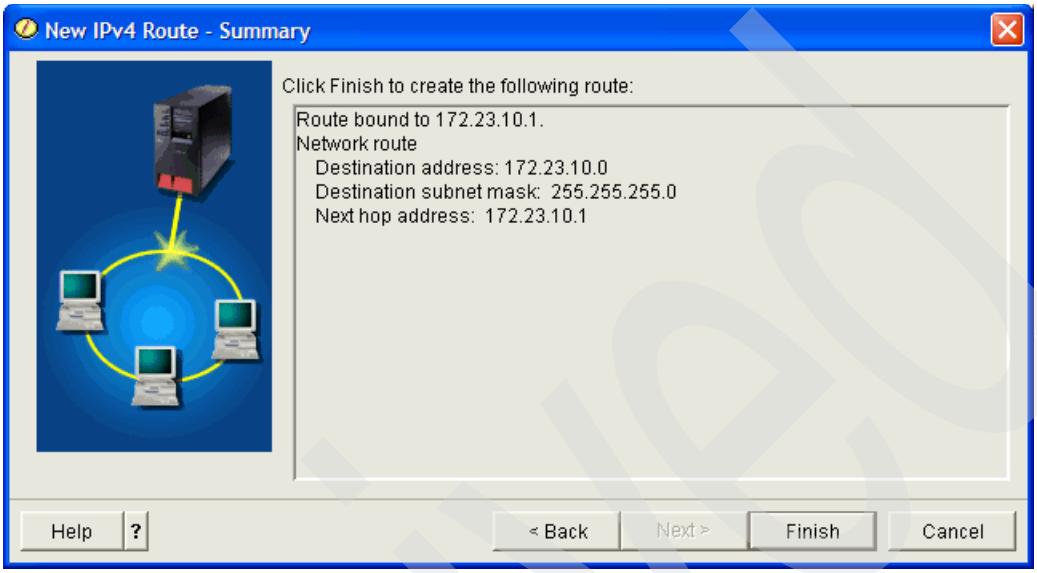


Figure 12-34 New IPv4 Route: Summary

Repeat these steps for the other two interfaces, 172.23.10.2 and 172.23.10.3. That is, create two more duplicate network routes, each bound to the two interfaces 172.23.10.2 and 172.23.10.3. All of these duplicate network routes should have the Next hop address as the same address as the preferred binding interface.

Step 3: Set the duplicate route priority for all routes

The physical interface to which the route is bound must be ended in order to change the route priority. We have three physical interfaces, all on the same subnet, with two routes bound to each of the physical interfaces.

Your iSeries Navigator and 5250 emulation might be connected to one of these physical interfaces. You therefore cannot perform the change from the iSeries Navigator or 5250 emulation, because ending the physical interfaces causes the iSeries Navigator connection to be terminated. In this scenario, you must perform the change from the System i console.

Tip: If there was an additional physical adapter that did not need to be ended and the iSeries Navigator was connected to this interface, then this function could be performed from iSeries Navigator.

Another option would be to configure a virtual IP address (VIPA) that is in the same subnet, 172.23.10.0, as your physical interfaces. Change the VIPA to support proxy ARP. Then configure a new iSeries Navigator connection to this VIPA address. As you stop then start the physical interfaces one-by-one, the System i will automatically be tolerant of these *faults* and your application connection will not be broken. See “VIPA V5R2 and beyond: directly routable with proxy ARP support” on page 39. Your System i must be at V5R2 for this support.

If you are able to make these changes to the inactive routes via iSeries Navigator, you should know that the procedure is to expand **Network** → **TCP/IP Configuration** → **IPv4** and click **Routes**. Right-click the inactive routes and select **Properties** from the context menu. On the Advanced tab is a *Route precedence* value that you can modify. Save your changes.

1. From an i5/OS command line, type `CFGTCP` and press Enter. Select option `1` from the Configure TCP/IP menu to work with TCP/IP interfaces. Press Enter.
2. Type `10` (iSeries Navigator terminology is Stop) next to the interface to which the default route is bound, and press Enter. This ends the interface. Press F12.
3. Select option `2` from the Configure TCP/IP menu to work with TCP/IP routes and press Enter.
4. You will now be at the Work with TCP/IP Routes menu. Type `2` (Change) next to the default route for the physical interface and press Enter.
5. The Change TCP/IP Route (CHGTCPRTE) command is displayed, as shown in Figure 12-35. We set the Duplicate route priority to `6` (as shown in Figure 12-28 on page 202) and press Enter. Press F12 to return to the main CFGTCP menu.

```
Change TCP/IP Route (CHGTCPRTE)

Type choices, press Enter.

Route destination  . . . . . > '*DFTROUTE'   Character value, *DFTROUTE...
Subnet mask  . . . . . > '*NONE'       Character value, *NONE, *HOST
Type of service  . . . . . > *NORMAL       *MINDELAY, *MAXTHRPUT...
Next hop . . . . . > '172.23.10.1'
Preferred binding interface  . . > '172.23.10.1'
Maximum transmission unit  . . .   *IFC          576-16388, *SAME, *IFC
Route metric . . . . . .   1             1-16, *SAME
Route redistribution . . . . . .   *NO           *SAME, *YES, *NO
Duplicate route priority . . . .   6             1-10, *SAME

                                                                      Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

Figure 12-35 Change TCP/IP Route (CHGTCPRTE)

6. Change the other Schowler route associated with this physical interface in the same manner. Change the Duplicate route priority to a value specified in the routing table in Figure 12-28 on page 202.
7. Type 1 (Work) with TCP/IP interfaces from the Configure TCP/IP menu and press Enter. Type 9 next to the interface that we had previously ended and press Enter. This starts the interface.

You must take the same steps for the other two physical interfaces and their associated routes. Set the Duplicate route priority based on the value as specified in the routing table in Figure 12-28 on page 202.

Step 4: Completion and test

For destination IP address that are within the subnet of 172.23.10.0, the Schowler routes bound to 172.23.10.2 and 172.23.10.3 will be used before the duplicate Schowler route created for 172.23.10.1. In addition, since the route priority is the same for 172.23.10.2 and 172.23.10.3, the System i will use DRRR to evenly distribute the outbound load across those two physical interface adapters.

The same logic is used for the three duplicate default routes for destination IP addresses that are outside of the 172.23.10.0 subnet.

To test this, we simply ping the System i from both a local client on the 172.23.10.0 subnet or from a client outside of that subnet (most likely from the corporate network). We used Start Communications Trace (STRCMNTRC) to trace the traffic on the physical interface 172.23.10.1. Note that all that you can see in the communications trace from the System i point of view is the incoming traffic. Because the outbound traffic goes over a different physical interface, it appears as though the System i is not responding. But it is, as the client confirmed the ICMP Echo Replies.

For our test system at ITSO, here are the communication trace commands that we used to take the trace:

```
1. STRCMNTRC CFGOBJ(ETHLIN1) CFGTYPE(*LIN) USRDTA(*MAX) TEXT('inbound traffic')
2. ENDCMNTRC CFGOBJ(ETHLIN1) CFGTYPE(*LIN)
3. PRTCMNTRC CFGOBJ(ETHLIN1) CFGTYPE(*LIN) CODE(*ASCII)
4. DLTCMNTRC CFGOBJ(ETHLIN1) CFGTYPE(*LIN)
```

You can use the Work with Job (WRKJOB) and option 4=Work with spooled files to find your communications trace spool file. The one we used for our test is shown in Figure 12-36. It shows that ICMP Echo Request records numbers 4 through 7 are received by the System i on this interface, but the ICMP Echo Reply does not travel back on the same interface.

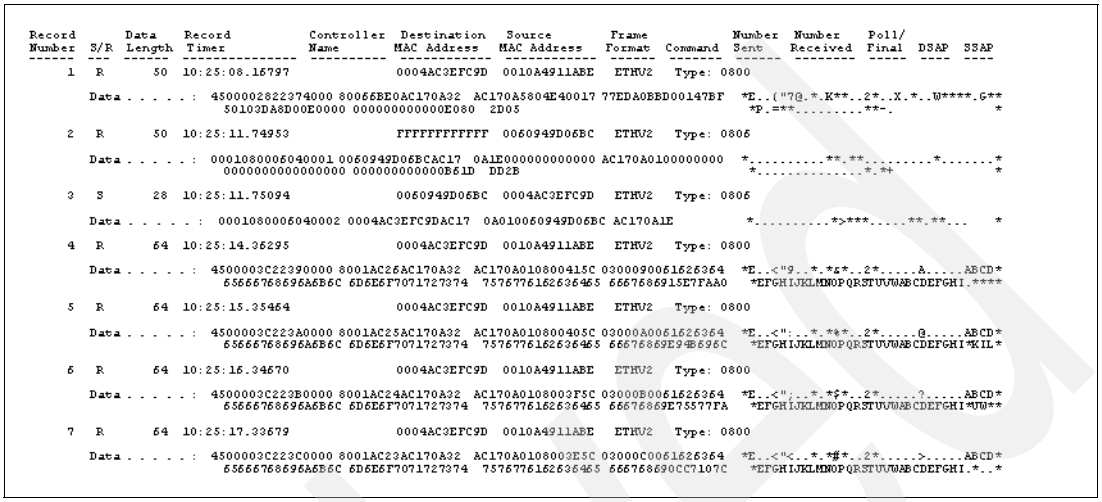


Figure 12-36 Communications trace showing ICMP Echo Requests arriving as records 4 through 7

12.5 Connect to a TCP/IP application while in restricted state

Starting in V5R2M0 of OS/400, the TCP/IP protocol stack and interfaces can be started while the operating system is in restricted state. One major restriction is that only your own user-written sockets application can be running on the system in this state. That is, none of the applications in the TCP/IP protocol suite, such as Telnet, FTP, SMTP, DNS, and so on can be started.

As an example, a network administrator can obtain status reports while you are running backup procedures. The operating system must be in restricted state to prevent users from changing any configuration. You can now access status reports remotely using a PDA or any TCP/IP networking device. The PDA uses a sockets-enabled application that requires having an active TCP/IP interface available to communicate with the server.

To allow this communication, you must first start TCP/IP using special parameters. After you start TCP/IP, start a specific TCP/IP interface to allow access to the system. More details follow.

Assumptions

Your System i must be running OS/400 V5R2M0 or later or i5/OS.

Restrictions

The following restrictions apply when the operating system is running in restricted state:

- You cannot start TCP/IP servers using the Start TCP/IP Server (STRTCPSVR) CL command, because they require active subsystems.
- You can start only interfaces for a specific line type (Ethernet, token-ring, or DDI) that is not attached to a network server description (NWSD) or a network interface description (NWID).

How-to

The steps for using TCP/IP when the system is in restricted state are:

- Step 1: Start TCP/IP using special parameters.
- Step 2: Start a specific TCP/IP interface.
- Step 3: Verify that the interface is active.

Step 1: Start TCP/IP using special parameters

When the System i is in restricted state, issue the Start TCP/IP (STRTCP) command on the 5250 command-line interface at the System i system console:

```
STRTCP STRSVR(*NO) STRIFC(*NO)STRPTPPRF(*NO)
```

These are the only parameters accepted when the operating system is in restricted state. This command starts TCP/IP, but it will not and cannot start TCP/IP application servers or IP interfaces.

Step 2: Start a specific TCP/IP interface

After you start TCP/IP in restricted state, you can start the specific interface needed for your sockets-enabled application. You also have to verify whether the interface you want to start is using a line description of *ELAN, *TRLAN, or *DDI. The interface may not be attached to an NWSD or an NWID.

Finally, start the interface. At a command line interface, enter the Start TCP/IP Interface (STRTCPIFC) command by replacing *a.b.c.d* with your interface IP address:

```
STRTCPIFC INTNETADR('a.b.c.d')
```

Note: Make sure that STRTCPIFC INTNETADR(*AUTOSTART) is not specified. This would cause all interfaces with the autostart parameter set to *YES to start.

Step 3: Verify that the interface is active

Ping the specific interface for your application. Very few TCP/IP-related utilities will operate in restricted state; however, the Verify TCP/IP Connection (PING) and the Work with TCP/IP Network Status (NETSTAT) commands can be used.

13

Virtual Ethernet within an LPAR environment

This chapter contains three scenarios in which we use the Virtual Ethernet LAN capability:

- "Virtual Ethernet and proxy ARP configuration" on page 212
- "Virtual Ethernet and NAT scenario" on page 224
- "Virtual Ethernet and routing scenario" on page 236

This capability was introduced in V5R1M0 of OS/400 and was known as Virtual LAN (local area network). Each Virtual LAN enables you to establish communication via TCP/IP between logical partitions. Each partition can define as many as 16 virtual LAN ports. Partitions defined to use the same port can communicate through that link.

In OS/400 V5R2M0, this capability was renamed to Virtual Ethernet. Virtual Ethernet enables you to establish communication via TCP/IP between logical partitions. Each partition can define up to 16 virtual local area networks. Partitions defined to use the same port can communicate through that link.

Starting with the System i5, you can define up to 4094 separate virtual LANs on each managed system using a Hardware Management Console (HMC). This allows i5/OS, AIX, and Linux logical partitions, and Windows environments integrated on System i to communicate with each other using TCP/IP over the virtual Ethernet communications ports.

Virtual Ethernet can be used without any additional hardware or software. The virtual Ethernet emulates a high-speed Ethernet environment and is used to establish multiple high-speed TCP/IP connections between logical partitions on the same System i without any additional communications hardware and software. For more information about the Virtual Ethernet setup, consult the IBM Redbooks publication *LPAR Configuration and Management Working with IBM eServer iSeries Logical Partitions*, SG24-6251.

13.1 Virtual Ethernet and proxy ARP configuration

This scenario describes the configuration of proxy ARP and Virtual Ethernet on the System i with four logical partitions in order to be able to access it from an external LAN.

Problem definition

In this scenario, a customer has the network shown in Figure 13-1. The System i has four different partitions:

- LPAR1: first partition running V5R4M0 of i5/OS
- LPAR2: second partition running V5R3M0 of i5/OS
- LPAR3: third partition running V5R4M0 of i5/OS
- LINUX: fourth partition is a guest partition running Linux hosted by LPAR1

We want to enable high-speed communications between all four partitions and extend that communication to an external LAN. As our hardware has a limited number of card slots available for installing LAN cards, we need a solution that does not require additional LAN cards to be installed in each of the logical partitions.

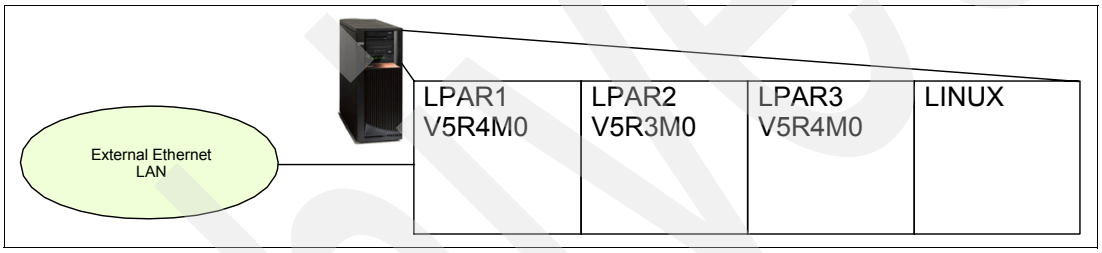


Figure 13-1 System i LPAR setup: four logical partitions

Solution definition

We create a Virtual Ethernet network (see Figure 13-2) to allow communications between the logical partitions on the System i. We will enable proxy ARP to connect the Virtual Ethernet network to the external LAN, and we will configure all necessary lines, interfaces, and routes.

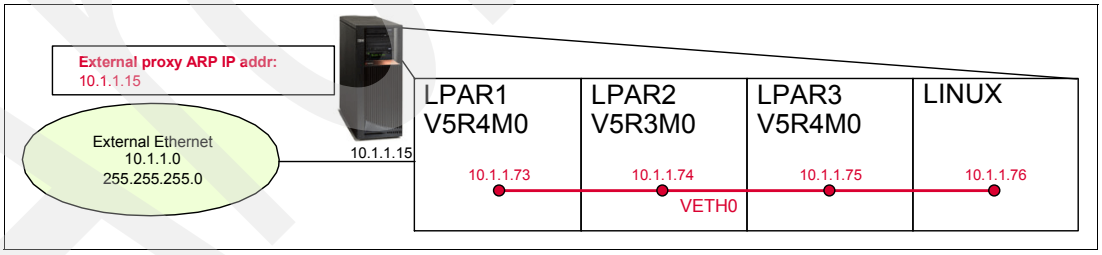


Figure 13-2 System i LPAR configuration setup

If a host on the 10.1.1.0 external LAN needs to resolve IP address 10.1.1.75 of LPAR3 to a MAC address, it will broadcast an ARP request. The partition PRIMARY responds with an ARP reply with the MAC address of the interface 10.1.1.15 on behalf of LPAR3.

For a complete explanation of proxy ARP, refer to “VIPA V5R2 and beyond: directly routable with proxy ARP support” on page 39.

Assumptions

Let us assume the following:

- We have a System i with four logical partitions already set up:
  - The first partition (LPAR1) runs i5/OS V5R4M0.
  - The second partition (LPAR2) runs i5/OS V5R3M0.
  - The third partition (LPAR3) runs i5/OS V5R4M0.
  - The fourth partition (LINUX) runs Linux.
- We have iSeries Access for Windows and iSeries Navigator (including the configuration and service component) installed.

How-to

To configure the System i accordingly, perform the following tasks:

- Step 1: Enable the logical partitions to participate in a Virtual Ethernet.
- Step 2: Create the Ethernet line descriptions.
- Step 3: Turn on IP datagram forwarding.
- Step 4: Create the TCP/IP interface to enable proxy ARP.
- Step 5: Create and start TCP/IP interface for Virtual Ethernet on LPAR1.
- Step 6: Create TCP/IP interface for Virtual Ethernet on LPAR2.
- Step 7: Create TCP/IP interface for Virtual Ethernet on LPAR3.
- Step 8: Create TCP/IP interface and default gateway for Virtual Ethernet.

Step 1: Enable the logical partitions to participate in a Virtual Ethernet

To enable Virtual Ethernet on systems prior to System i5, follow these steps:

1. On the 5250 command line on partition LPAR1 (the PRIMARY in this case), enter the Start Service Tools (STRSST) Sign On command.
2. Enter your service tools user ID and password.
3. As shown in Figure 13-3, in the System Service Tools (SST) display, select option 5 to work with system partitions.

```
                    System Service Tools (SST)

Select one of the following:

     1. Start a service tool
     2. Work with active service tools
     3. Work with disk units
     4. Work with diskette data recovery
     5. Work with system partitions
     6. Work with system capacity
     7. Work with system security
     8. Work with service tools user IDs


Selection
   5

F3=Exit         F10=Command entry         F12=Cancel
```

Figure 13-3 System Service Tools: work with system partitions (option 5)

4. As shown in Figure 13-4 from the Work with System Partitions display, select option 3 to work with the partition configuration.

```
                        Work with System Partitions
                                                                System:   AS20
Attention:  Incorrect use of this utility can cause damage
to data in this system. See service documentation.

  Number of partitions  . . . . . . . :  4
  Partition manager release . . . . . :  V5R4M0 L0

  Partition identifier  . . . . . . . :  0
  Partition name  . . . . . . . . . . :  LPAR1

Select one of the following:

     1. Display partition information
     2. Work with partition status
     3. Work with partition configuration
     4. Recover configuration data
     5. Create a new partition

Selection
    3

F3=Exit   F10=IPL system to activate changes   F12=Cancel
System IPL may be required to activate changes.
```

Figure 13-4 Work with System Partitions: Work with partition configuration (option 3)

5. Press F10 to work with Virtual Ethernet, as shown in Figure 13-5.

```
                    Work with Partition Configuration
                                                                System:   AS20
Type option, press Enter.
  1=Change partition name      2=Change partition processing resources
  3=Add I/O resources          4=Remove I/O resources
  5=Change bus ownership type  6=Select load source resource

Option  Par ID      Name
          0         LPAR1
          1         LPAR2
          2         LPAR3
          3         LINUX



< Indicates partition IPL may be required.
F3=Exit   F5=Refresh             F9=Work with shared processor pool
F10=Work with Virtual Ethernet   F11=Work with partition status
F12=Cancel   F23=More options
```

Figure 13-5 Work with Partition Configuration: Work with Virtual Ethernet (F10)

6. In the panel shown in Figure 13-6, type 2 in front of the primary partition (LPAR1) to set up the Virtual Ethernet configuration.

```
                    Work with Virtual Ethernet Configuration
                                                            System: AS20
Type options, press Enter.
  2=Change

    Par          -----------------Virtual Ethernet Identifiers----------------
Opt ID  Name     0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15
 2    0 LPAR1    .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .
      1 LPAR2    .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .
      2 LPAR3    .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .
      3 LINUX    .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .



'1' Indicates in use.  '.' Indicates not in use.
F3=Exit   F9=Show all partitions
F11=Display communication options   F12=Cancel
```

Figure 13-6 Work with Virtual Ethernet Configuration: change Virtual Ethernet configuration for PRIMARY

7. As shown in Figure 13-7, type 1 in the appropriate column for the primary partition and the secondary partitions to enable the partitions to communicate with one another over virtual Ethernet. In our example, we type a 1 in the column that refers to Virtual Ethernet Identifier 1 and we press Enter. By default, all of those are set to 2, telling that there is no Virtual Ethernet configured.

```
                   Change Virtual Ethernet Configuration
                                                            System:   AS20
  Partition identifier . . . . . . . . . . . . . :  0
  Partition name . . . . . . . . . . . . . . . . :  LPAR1

Type changes, press Enter.
  1=Yes  2=No

  -----------------Virtual Ethernet Identifiers----------------
  0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15
  2   1   2   2   2   2   2   2   2   2   2   2   2   2   2   2



F3=Exit   F12=Cancel
```

Figure 13-7 Change Virtual Ethernet Configuration: enable Virtual Ethernet Identifier 0 for LPAR1

8. Figure 13-8 shows the updated Work with Virtual Ethernet Configuration panel.

```
                  Work with Virtual Ethernet Configuration
                                                          System:   AS20
Type options, press Enter.
  2=Change

    Par          -----------------Virtual Ethernet Identifiers----------------
Opt ID  Name     0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15
      0 LPAR1    .   1   .   .   .   .   .   .   .   .   .   .   .   .   .   .
      1 LPAR2    .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .
      2 LPAR3    .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .
      3 LINUX    .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .



'1' Indicates in use.  '.' Indicates not in use.
F3=Exit   F9=Show all partitions
F11=Display communication options   F12=Cancel
```

Figure 13-8 Work with Virtual Ethernet Configuration: Updated configuration for primary partition

9. Repeat steps 6 on page 215 and 7 on page 215 for all secondary partitions running i5/OS as well as for the guest partition running Linux. Make sure to type a 1 in the appropriate column here as well, referring to Virtual Ethernet Identifier 1, and press Enter. Figure 13-9 shows the resulting panel with updated information.

```
                  Work with Virtual Ethernet Configuration
                                                          System:   AS20
Type options, press Enter.
  2=Change

    Par          -----------------Virtual Ethernet Identifiers----------------
Opt ID  Name     0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15
      0 LPAR1    .   1   .   .   .   .   .   .   .   .   .   .   .   .   .   .
      1 LPAR2    .   1   .   .   .   .   .   .   .   .   .   .   .   .   .   .
      2 LPAR3    .   1   .   .   .   .   .   .   .   .   .   .   .   .   .   .
      3 LINUX    .   1   .   .   .   .   .   .   .   .   .   .   .   .   .   .



'1' Indicates in use.  '.' Indicates not in use.
F3=Exit   F9=Show all partitions
F11=Display communication options   F12=Cancel
```

Figure 13-9 Work with Virtual Ethernet Configuration: updated configuration for all partitions

10. You can then exit System Service Tools (SST) to return to the 5250 command line.

To enable Virtual Ethernet on a System i5, follow these steps. These steps assume that the four partitions have already been created using the Hardware Management Console (HMC) and that we are using Dynamic Logical Partitioning from within the Web-based System Manager (WebSM):

1. From the Management Environment of WebSM, expand your HMC.
2. Expand **Server and Partition** → **Server Management**.

3. From the Server and Partition: Server Management panel, expand the appropriate **Managed System** → **Partitions**.
4. Right-click **LPAR1** → **Dynamic Logical Partitioning** → **Virtual Adapter Resources** → **Add/Remove**.
5. Select the **Ethernet** tab followed by the **Create Adapter** button. As shown in Figure 13-10, we will create the virtual adapter using Slot 2 on Virtual LAN ID 1.

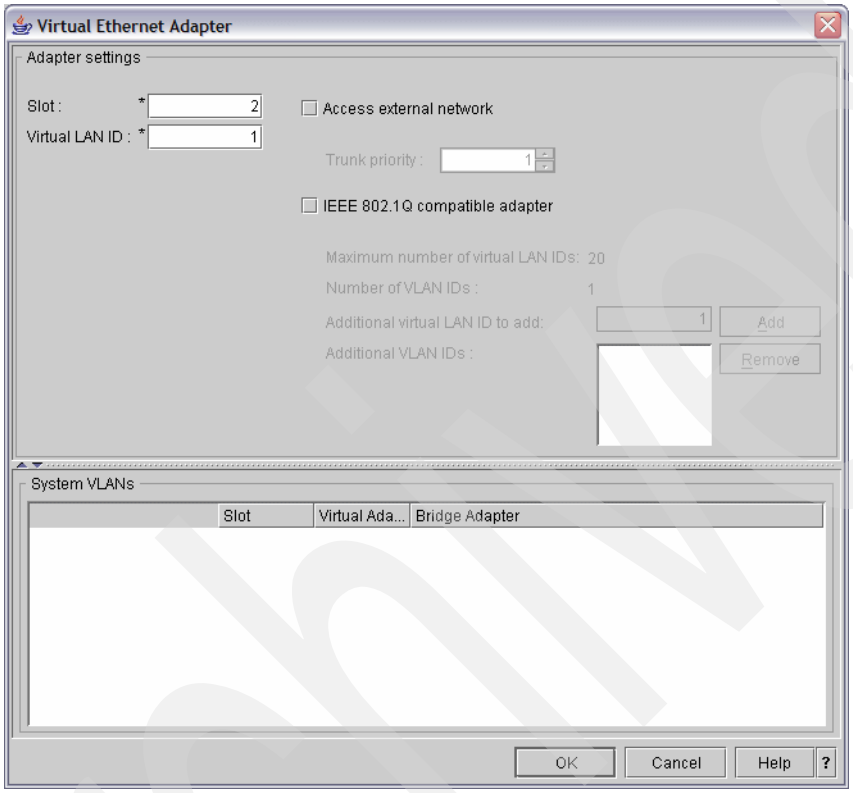


Figure 13-10 Create Virtual Ethernet Adapter

6. Selecting **OK** creates the virtual adapter, as shown in Figure 13-11.

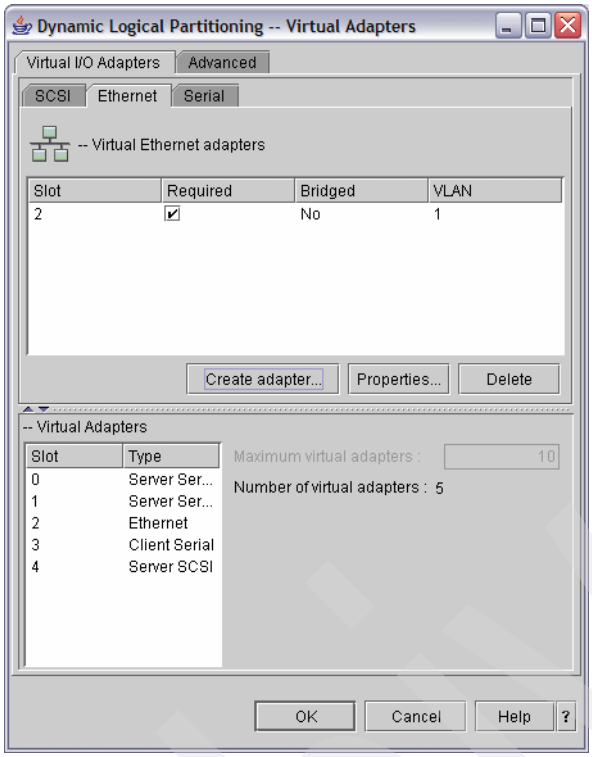


Figure 13-11 Virtual Ethernet Adapter added using Dynamic Logical Partitioning

7. Repeat steps 4 on page 217 through 6 for each of the additional partitions (LPAR2, LPAR3, and LINUX).

Step 2: Create the Ethernet line descriptions

To configure new Ethernet line descriptions to support Virtual Ethernet, follow these steps:

1. At the command line on partition LPAR1, use the Work with Hardware Resources (WRKHDWRSC) command:

```
WRKHDWRSC *CMN
```

2. As shown in Figure 13-12 from the Work with Communication Resources display, type option 7 (Display resource detail) next to the appropriate Virtual Ethernet port. A Virtual Ethernet can be identified as 268C in the Type column on the panel. There will be one 268C Ethernet Port resource for each Virtual Ethernet that is connected to the logical partition.

```
                    Work with Communication Resources
                                                          System:   AS20
Type options, press Enter.
  5=Work with configuration descriptions   7=Display resource detail

Opt  Resource        Type  Status               Text
     CMB01           286D  Operational          Combined function IOP
       LIN02         2793  Operational          Comm Adapter
         CMN02       2793  Operational          Comm Port
         CMN03       2793  Operational          Comm Port
       LIN01         2838  Operational          LAN Adapter
         CMN01       2838  Operational          Ethernet Port
     CMB02           268C  Operational          Combined function IOP
       LIN07         268C  Operational          LAN Adapter
 7       CMN09       268C  Operational          Ethernet Port
     CMB05           2843  Operational          Combined function IOP
       LIN05         2838  Operational          LAN Adapter
         CMN25       2838  Operational          Ethernet Port
                                                                  More...
F3=Exit   F5=Refresh   F6=Print   F12=Cancel
```

Figure 13-12 Work with Communication Resources display

3. In the Display Resource Detail display (Figure 13-13), scroll down to find the port address. This port address corresponds to the Virtual Ethernet you selected during the configuration of the logical partition. In our case Port 1 corresponds to Virtual LAN identifier 1, as shown in Figure 13-9 on page 216 or Figure 13-11 on page 218. Press Enter to continue.

```
                        Display Resource Detail
                                                          System:   AS20
Resource name  . . . . . :   CMN09
Text . . . . . :   Ethernet Port
Type-model . . . . . :   268C-001
Serial number  . . . . . :   00-00000
Part number  . . . . . :


  Port                          1


Press Enter to continue.

F3=Exit   F5=Refresh   F6=Print   F12=Cancel
```

*Figure 13-13 WRKHDWRSC *CMN: Display Resource Detail*

4. Figure 13-14 shows the Work with Communication Resources display. Type 5 (Work with configuration descriptions) next to the appropriate Virtual Ethernet port, and press Enter.

```
                    Work with Communication Resources
                                                              System:   AS20
Type options, press Enter.
  5=Work with configuration descriptions   7=Display resource detail

Opt  Resource        Type  Status               Text
     CMB01           286D  Operational          Combined function IOP
       LIN02         2793  Operational          Comm Adapter
         CMN02       2793  Operational          Comm Port
         CMN03       2793  Operational          Comm Port
       LIN01         2838  Operational          LAN Adapter
         CMN01       2838  Operational          Ethernet Port
     CMB02           268C  Operational          Combined function IOP
       LIN07         268C  Operational          LAN Adapter
 5       CMN09       268C  Operational          Ethernet Port
     CMB05           2843  Operational          Combined function IOP
       LIN05         2838  Operational          LAN Adapter
         CMN25       2838  Operational          Ethernet Port

                                                                      More...
F3=Exit   F5=Refresh   F6=Print   F12=Cancel
```

Figure 13-14 Work with Communication Resources panel: Work with configuration descriptions

5. In the Work with Configuration Descriptions display (Figure 13-15), type 1 (Create) and press Enter to see the Create Line Description Ethernet (CRTLINETH) display.

```
                    Work with Configuration Descriptions
                                                              System:   AS20
Resource name  . . . . . . . :   CMN09
Text . . . . . . . . . . . . :   Ethernet Port

Type options, press Enter.
  1=Create   5=Work with description   8=Work with configuration status

Opt     Description
 1


F3=Exit   F5=Refresh   F6=Print   F12=Cancel
```

Figure 13-15 Work with configuration description panel: creation of Ethernet line description

6. In the CRTLINETH panel (Figure 13-16), provide the following information:
   a. For the Line description prompt, type VETH1. VETH1 corresponds to the numbered column on the Virtual Ethernet page in which you enabled the logical partitions to communicate. If you use the same names for the line descriptions and their associated Virtual Ethernet, you can easily keep track of your Virtual Ethernet configurations.
   b. For the Line speed prompt, type 1G.
   c. For the Duplex prompt, type *FULL, and press Enter.
   d. For the Maximum frame size prompt, type 8996, and press Enter. Changing the frame size to 8996 improves the transfer of data across the Virtual Ethernet.

```
                    Create Line Desc (Ethernet) (CRTLINETH)

Type choices, press Enter.

Line description . . . . . > VETH1          Name
Resource name  . . . . . . > CMN09          Name, *NWID, *NWSD
Online at IPL  . . . . . .   *YES           *YES, *NO
Vary on wait . . . . . .     *NOWAIT        *NOWAIT, 15-180 seconds
Local adapter address  . .   *ADPT          020000000000-FEFFFFFFFFFF...
Exchange identifier  . . .   *SYSGEN        05600000-056FFFFF, *SYSGEN
Ethernet standard  . . . .   *ALL           *ETHV2, *IEEE8023, *ALL
Line speed . . . . . . . . > 1G             10M, 100M, 1G, *AUTO
Duplex . . . . . . . . . . > *FULL          *HALF, *FULL, *AUTO
Maximum frame size . . . .   8996           1496-8996, 1496, 8996
SSAP list:
  Source service access point  .   *SYSGEN      02-FE, *SYSGEN
  SSAP maximum frame . . . . .                  *MAXFRAME, 265-8996, 265...
  SSAP type  . . . . . .                        *CALC, *NONSNA, *SNA, *HPR
               + for more values

                                                                      More...
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

Figure 13-16 Create Line Description: CRTLINETH parameters

7. In the Work with Configuration Descriptions panel (Figure 13-17), type 8 (Work with configuration status) in front of the VETH1 line description that was just created.

```
                    Work with Configuration Descriptions
                                                          System:   AS20
Resource name  . . . . . :   CMN09
Text . . . . . . . . . . :   Ethernet Port

Type options, press Enter.
  1=Create   5=Work with description   8=Work with configuration status

Opt     Description

 8      VETH1



                                                                   Bottom
F3=Exit   F5=Refresh   F6=Print   F12=Cancel
```

Figure 13-17 'Work with Configuration Descriptions panel: Work with configuration status

8. In the Work with Configuration Status panel (Figure 13-18), type 1 (Vary on) for VETH1.

```
                        Work with Configuration Status                AS20
                                                              08/08/03  15:55:47
Position to  . . . . .               Starting characters

Type options, press Enter.
  1=Vary on   2=Vary off   5=Work with job   8=Work with description
  9=Display mode status    13=Work with APPN status...

Opt  Description       Status                -------------Job--------------
 1   VETH1             VARIED OFF




                                                                       Bottom
Parameters or command
===>
F3=Exit   F4=Prompt   F12=Cancel   F23=More options   F24=More keys
```

Figure 13-18 Work with Configuration Status panel: vary on of Virtual Ethernet VETH1

9. Repeat steps 1 on page 218 through 8, but perform the steps from the command lines on logical partitions LPAR2 and LPAR3 to create and vary on the Ethernet line description for the other partitions running i5/OS.

Although the names of your line descriptions are arbitrary, it is helpful to use the same names for all of the line descriptions associated with the Virtual Ethernet. In this scenario, all of the line descriptions are named VETH1.

Step 3: Turn on IP datagram forwarding

We want to turn on IP datagram forwarding so that the packets can be forwarded among different subnets.

At the command line on partition LPAR1, use the Change TCP/IP Attributes (CHGTCPA) command in the following way:

```
CHGTCPA IPDTGFWD (*YES)
```

Step 4: Create the TCP/IP interface to enable proxy ARP

Before you create the TCP/IP interfaces, decide how you want to connect your Virtual Ethernet to a physical LAN. To allow your logical partitions to communicate with systems on an external LAN, enable the TCP/IP traffic to travel between the Virtual Ethernet and the external LAN.

In this scenario we use the proxy ARP method. To create the TCP/IP interface to enable proxy ARP, complete the following steps:

1. Obtain a contiguous block of IP addresses that are routable by your network.

   Because you have a total of four logical partitions in this Virtual Ethernet, you need a block of eight addresses. The fourth segment of the first IP address in the block must be divisible by eight. The first and last IP addresses of this block are the subnet and broadcast IP addresses and are unusable. The second address can be used for a virtual TCP/IP interface on the primary partition, and the third, fourth, and fifth addresses can be used for the TCP/IP connections on each of the other logical partitions. For this scenario, the IP address block is 10.1.1.72 through 10.1.1.79 with a subnet mask of 255.255.255.248.

You also need a single IP address for your external TCP/IP address. This IP address does not have to belong to your block of contiguous addresses, but it must be within the same original subnet mask of 255.255.255.0.

2. Create an i5/OS TCP/IP interface for logical partition LPAR1. This interface is known as the external, proxy ARP IP interface. In order to create the interface, follow these steps:
   a. At the command line on the primary partition, type `CFGTCP` and press Enter to see the Configure TCP/IP display.
   b. Select option `1` = Work with TCP/IP Interfaces, and press Enter.
   c. Select option `1` = Add, and press Enter to see the Add TCP/IP Interface (ADDTCPIFC) display.
      i. For the Internet address prompt, type `10.1.1.15`.
      ii. For the Line description prompt, type the name of your line description, such as ETHLINE.
      iii. For the Subnet mask prompt, type `255.255.255.0`.
3. Start the interface. On the Work with TCP/IP Interfaces display, type `9` (Start) by the interface you want to start.

Step 5: Create and start TCP/IP interface for Virtual Ethernet on LPAR1

We need to create and then start the Virtual Ethernet interface on partition LPAR1:

1. Use the Add TCP/IP Interface (ADDTCPIFC) command to create the interface:

```
ADDTCPIFC INTNETADR('10.1.1.73') LIND(VETH1) SUBNETMASK('255.255.255.248')
LCLIFC('10.1.1.15')
```

> **Note:** The associated local interface (LCLIFC) parameter (10.1.1.15 in our example) associates the virtual interface to the external interface and enables proxy ARP to forward packets between the virtual interface 10.1.1.73 and the external interface 10.1.1.15.
>
> **Note:** Starting in i5/OS V5R4, you can gain additional fault tolerance by having more than one physical interface that is capable of being the proxy ARP agent for the Virtual Ethernet subnet. This is accomplished by using a Preferred Interface list instead of the associated local interface. This type of configuration can either be done using iSeries Navigator (via the interface properties) or the Change TCP/IP IPv4 Interface (QTOCC4IF) application programming interface (API).

2. Use the Start TCP/IP Interface (STRTCPIFC) command to start the interface we just created:

```
STRTCPIFC INTNETADR('10.1.1.73')
```

Step 6: Create TCP/IP interface for Virtual Ethernet on LPAR2

Now we create and then start the Virtual Ethernet interface on partition LPAR2:

1. Use the Add TCP/IP Interface (ADDTCPIFC) command to create the interface:

```
ADDTCPIFC INTNETADR('10.1.1.74') LIND(VETH1) SUBNETMASK('255.255.255.248')
```

2. Use the Start TCP/IP Interface (STRTCPIFC) command to start the interface we just created:

```
STRTCPIFC INTNETADR('10.1.1.74')
```

Step 7: Create TCP/IP interface for Virtual Ethernet on LPAR3

Now we create and then start the Virtual Ethernet interface on partition LPAR3:

1. Use the Add TCP/IP Interface (ADDTCPIFC) command to create the interface:

   ```
   ADDTCPIFC INTNETADR('10.1.1.75') LIND(VETH1) SUBNETMASK('255.255.255.248')
   ```

2. Use the Start TCP/IP Interface (STRTCPIFC) command to start the interface we just created:

   ```
   STRTCPIFC INTNETADR('10.1.1.75')
   ```

Step 8: Create TCP/IP interface and default gateway for Virtual Ethernet on LINUX

On the Linux guest partition, set up networking using the instructions or tools provided by your Linux distribution. Make sure to use the correct IP address, subnet mask, port, and router IP address. In our example, we specify the following:

Interface IP address	10.1.1.76
Subnet mask	255.255.255.248
Default gateway IP address	10.1.1.73
Network device	VETH1

For more information about adding interfaces and default routes within Linux, refer to the Redbooks publication *Implementing POWER Linux on IBM System i Platform*, SG24-6388.

Step 9: Create the necessary routes on partitions LPAR2 and LPAR3

To create the default routes to enable the packets to exit the Virtual Ethernet, use the Add TCP/IP Route (ADDTCPRTE) command on LPAR2 *and* LPAR3:

```
ADDTCPRTE RTEDEST(*DFTROUTE) SUBNETMASK(*NONE) NEXTHOP('10.1.1.73') MTU(1492)
```

Packets from each of these logical partitions travel over the Virtual Ethernet to the 10.1.1.73 interface using these default routes. Because 10.1.1.73 is associated with the external proxy ARP interface 10.1.1.15 (as defined in Step 5: Create and start TCP/IP interface for Virtual Ethernet on LPAR1), the packets continue out of the Virtual Ethernet using the proxy ARP interface. Notice that we have specified an MTU of 1492. We want to do this if our external Ethernet network is using 1492 so that packets destined for the external network will not have to be fragmented as they travel out of the Virtual Ethernet. Traffic travelling between LPAR1, LPAR2, LPAR3, or LINUX will use the *DIRECT routes, which are using the larger frame sizes defined in our virtual Ethernet line descriptions.

Step 10: Verify the network communications

We can verify our network communications by using the Verify TCP/IP Connection (PING) command:

- From partitions LPAR2, LPAR3, and LINUX, ping the Virtual Ethernet interface 10.1.1.73, then 10.1.1.15 and also an external host.
- From an external host, ping each of the Virtual Ethernet interfaces 10.1.1.73, 10.1.1.74, 10.1.1.75, and 10.1.1.76.

13.2 Virtual Ethernet and NAT scenario

This scenario describes the configuration of NAT and Virtual Ethernet on the System i with four logical partitions in order to be able to access it from an external LAN.

Problem definition

For this scenario, a customer has a network, as presented in Figure 13-19. The System i has four different partitions, and we need to allow high-speed communications between all four partitions and we want to extend that communication to an external LAN. As our hardware has a limited number of card slots available for installing LAN cards, we need to find a solution that does not require additional LAN cards to be installed.

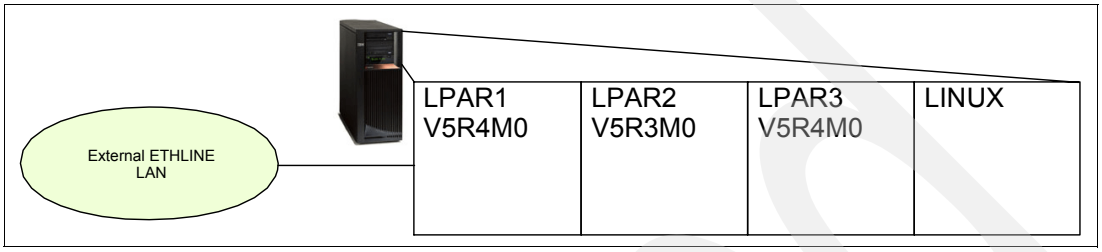


Figure 13-19 System i LPAR setup: four logical partitions

Problem solution

We create a Virtual Ethernet network (Figure 13-20) to allow communications between the logical partitions on the System i. We enable NAT to connect the Virtual Ethernet network to the external LAN, and configure all necessary lines, interfaces, and routes.

The 10.1.1.0 network represents an external network, and the 192.168.1.0 network represents the Virtual Ethernet network.

Attention: We use network 10.1.1.0 for public addresses as we assume that there is no need to get routable IP addresses on the ETHLINE network in our scenario.

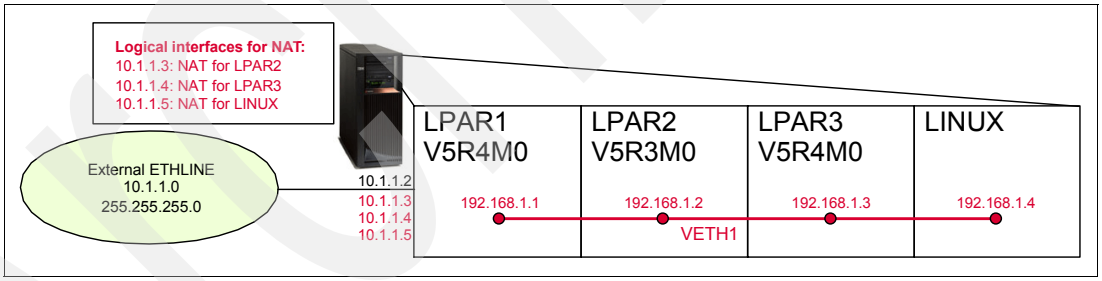


Figure 13-20 System i LPAR configuration setup

We can route traffic between your partitions with Virtual Ethernet and the external network by using NAT. This particular form of NAT is called static NAT: it allows both inbound and outbound IP traffic to and from the partitions. As we will use IP packet rules, we use iSeries Navigator to create and apply those rules.

Assumptions

Let us assume the following:

- We have a System i with four logical partitions already set up:
  - The first partition (LPAR1) runs i5/OS V5R4M0.
  - The second partition (LPAR2) is running i5/OS V5R3M0.
  - The third partition (LPAR3) is running i5/OS V5R4M0.
  - The fourth partition (LINUX) is running Linux.

- We have iSeries Access for Windows and iSeries Navigator (including the configuration and service component) installed.
- We added TCP/IP interface 10.1.1.2 for the external LAN ETHLINE on partition LPAR1.
- We enabled the logical partitions to participate in a Virtual Ethernet.
- We created the Virtual Ethernet line descriptions VETH1 on LPAR1, LPAR2, and LPAR3.
- We turned on IP datagram forwarding on partition LPAR1.

How-to

To configure the System i accordingly, perform the following tasks:

- Step 1: Add and start the TCP/IP interface on Virtual Ethernet in LPAR1.
- Step 2: Add and start the TCP/IP interface on Virtual Ethernet in LPAR2.
- Step 3: Add and start the TCP/IP interface on Virtual Ethernet in LPAR3.
- Step 4: Add and start TCP/IP interface and default gateway on Virtual Ethernet.
- Step 5: Create the necessary routes on LPAR2 and LPAR3.
- Step 6: Add and start TCP/IP interfaces on external LAN to be used by NAT.
- Step 7: Set up NAT for LPAR2, LPAR3, and LINUX with iSeries Navigator.

Step 1: Add and start the TCP/IP interface on Virtual Ethernet in LPAR1

Add a TCP/IP interface for your Virtual Ethernet in LPAR1 by using the Add TCP/IP Interface (ADDTCPIFC) command:

```
ADDTCPIFC INTNETADR('192.168.1.1') LIND('VETH1') SUBNETMASK('255.255.255.0')
```

Start the TCP/IP interface for your Virtual Ethernet in LPAR1 by using the Start TCP/IP Interface (STRTCPIFC) command:

```
STRTCPIFC INTNETADR('192.168.1.1')
```

Step 2: Add and start the TCP/IP interface on Virtual Ethernet in LPAR2

Add a TCP/IP interface for your Virtual Ethernet in LPAR2 by using the Add TCP/IP Interface (ADDTCPIFC) command:

```
ADDTCPIFC INTNETADR('192.168.1.2') LIND('VETH1') SUBNETMASK('255.255.255.0')
```

Start the TCP/IP interface for your Virtual Ethernet in LPAR2 by using the Start TCP/IP Interface (STRTCPIFC) command:

```
STRTCPIFC INTNETADR('192.168.1.2')
```

Step 3: Add and start the TCP/IP interface on Virtual Ethernet in LPAR3

Add a TCP/IP interface for your Virtual Ethernet in LPAR3 by using the Add TCP/IP Interface (ADDTCPIFC) command:

```
ADDTCPIFC INTNETADR('192.168.1.3') LIND('VETH1') SUBNETMASK('255.255.255.0')
```

Start the TCP/IP interface for your Virtual Ethernet in LPAR3 by using the Start TCP/IP Interface (STRTCPIFC) command:

```
STRTCPIFC INTNETADR('192.168.1.3')
```

Step 4: Add and start TCP/IP interface and default gateway on Virtual Ethernet in LINUX

On partition LINUX, set up networking by using the instructions or tools provided by your Linux distribution. Be sure to use the correct IP address, subnet mask, port, and gateway IP address.

In our example, we use the following:

Interface IP address 192.168.1.4
Subnet mask 255.255.255.0
Gateway IP address 192.168.1.1
Network device VETH1

Step 5: Create the necessary routes on LPAR2 and LPAR3

To create the default routes to enable the packets to exit the Virtual Ethernet, use the Add TCP/IP Route (ADDTCPRTE) command on LPAR2 *and* LPAR3:

```
ADDTCPRTE RTEDEST(*DFTROUTE) SUBNETMASK(*NONE) NEXTHOP('192.168.1.1')
```

Step 6: Add and start TCP/IP interfaces on external LAN to be used by NAT

We create three TCP/IP interfaces on LPAR1 that connect to the external LAN ETHLINE by using the Add TCP/IP Interface (ADDTCPIFC) command:

```
ADDTCPIFC INTNETADR('10.1.1.3') LIND('ETHLINE') SUBNETMASK('255.255.255.0')
ADDTCPIFC INTNETADR('10.1.1.4') LIND('ETHLINE') SUBNETMASK('255.255.255.0')
ADDTCPIFC INTNETADR('10.1.1.5') LIND('ETHLINE') SUBNETMASK('255.255.255.0')
```

Note: Be aware those are using the same line description ETHLINE as the existing 10.1.1.2 TCP/IP interface.

Now start the three interfaces we just created on partition LPAR1 by using the Start TCP/IP Interface (STRTCPIFC) command:

```
STRTCPIFC INTNETADR('10.1.1.3')
STRTCPIFC INTNETADR('10.1.1.4')
STRTCPIFC INTNETADR('10.1.1.5')
```

These addresses will be used for NAT purposes:

- 10.1.1.3 is used by LPAR2.
- 10.1.1.4 is used by LPAR3.
- 10.1.1.5 is used by LINUX.

Step 7: Set up NAT for LPAR2, LPAR3, and LINUX with iSeries Navigator

We perform the following steps from a PC with iSeries Access for Windows on the 10.1.1.0 network and connect to LPAR1 through the ETHLINE 10.1.1.2 interface.

1. Start iSeries Navigator and expand your iSeries server → **Network** → **IP Policies**.
2. Right-click **Packet Rules** and select **Rules Editor** from the context menu.

3. The Packet Rules Editor opens and a Welcome Packet Rules Configuration window (Figure 13-21) automatically pops up. Make sure that the Create a new packet rules file has been selected and click **OK** to continue.

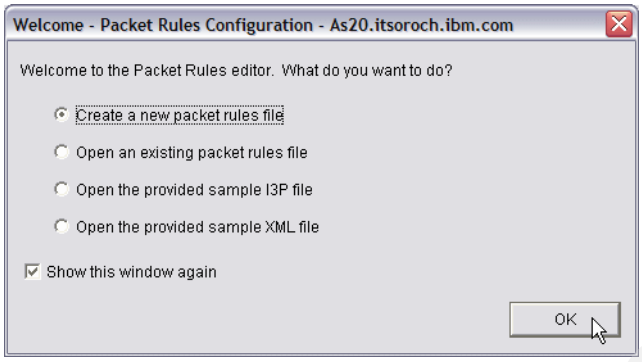


Figure 13-21 Packet Rules Editor: Welcome Packet Rules Configuration window

4. The Getting Started window (Figure 13-22) opens. Click **OK** to continue.

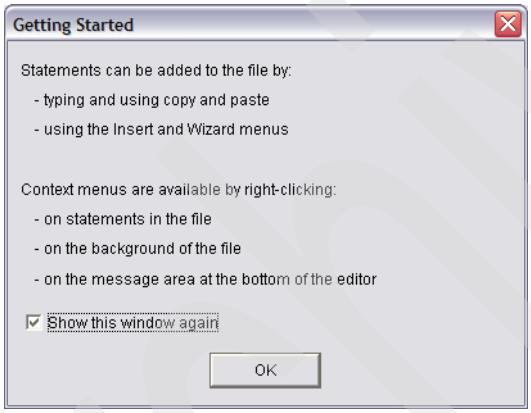


Figure 13-22 Packet Rules editor: Getting Started window

5. From the menu bar on the Packet Rules Editor window in Figure 13-23, select **Wizards** → **Address Translation**.

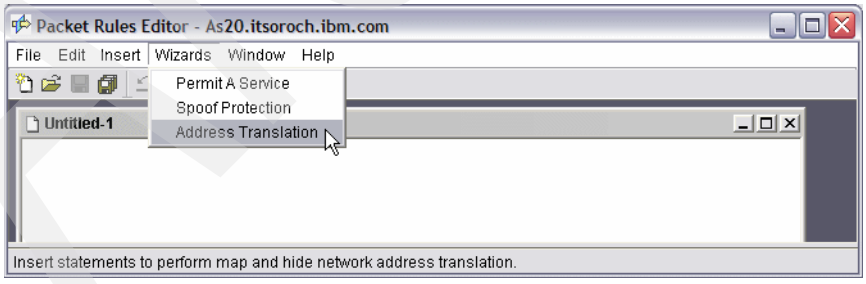


Figure 13-23 Packet Rules Editor: Address Translation wizard

6. The Address Translation Wizard Welcome window appears (Figure 13-24). Click **Next**.

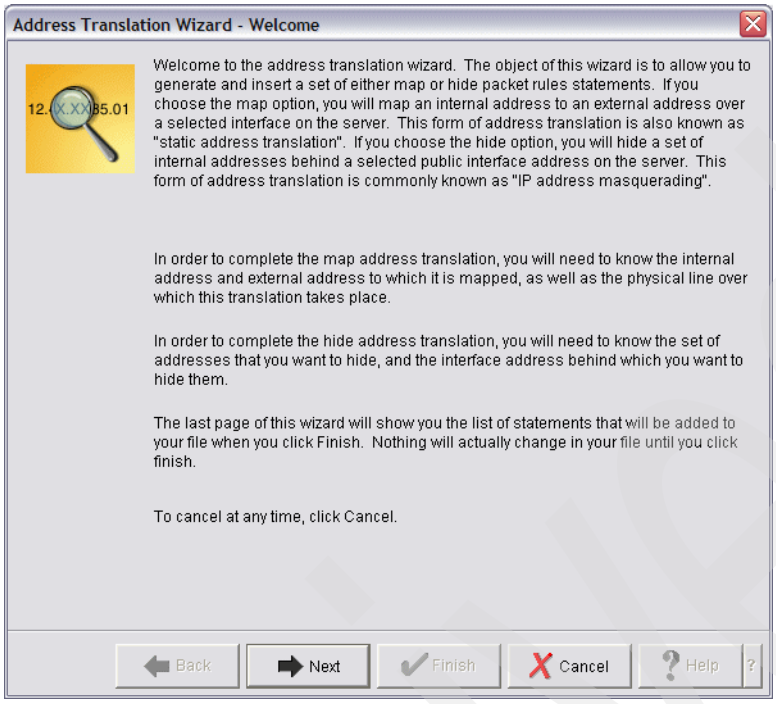


Figure 13-24 Address Translation Wizard Welcome window

7. The Address Translation Selection window appears (Figure 13-25). Make sure that **Map address translation** is selected and click **Next**.

Figure 13-25 Address Translation Selection window: Select Map address translation

8. On the Private Address window (Figure 13-26), we type `192.168.1.2` for the private address and click **Next**.

Figure 13-26 Address Translation wizard: Specify private IP address

9. On the Public Address window shown in Figure 13-27, we type `10.1.1.3` for the public address and click **Next**.

> **Attention:** We specified 10.1.1.3 for the public address as we assume that there is no need to get a routable IP address on the ETHLINE network in our scenario.

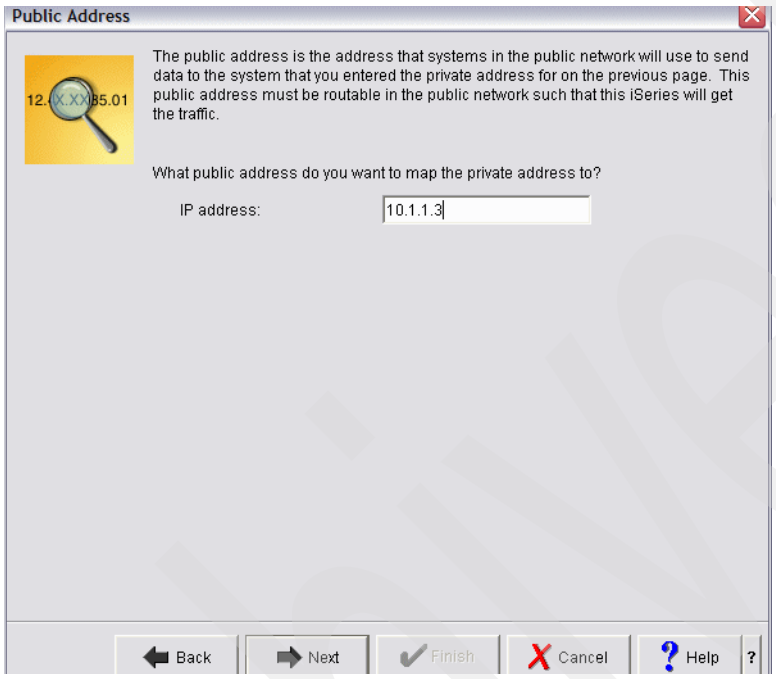


Figure 13-27 Address Translation wizard: Specify public IP address

10. On the Line window shown in Figure 13-28, we select **ETHLINE** from the list to be used for NAT. Click **Next**.

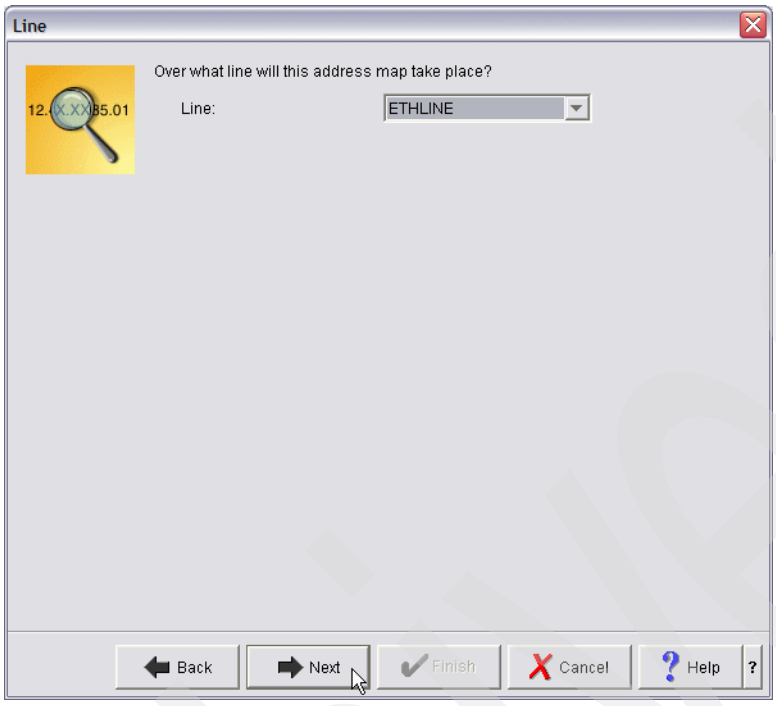


Figure 13-28 Address Translation wizard: physical line selection for NAT

11. The window shown in Figure 13-29 opens with a summary of all of the details that have been specified. Click **Finish**.

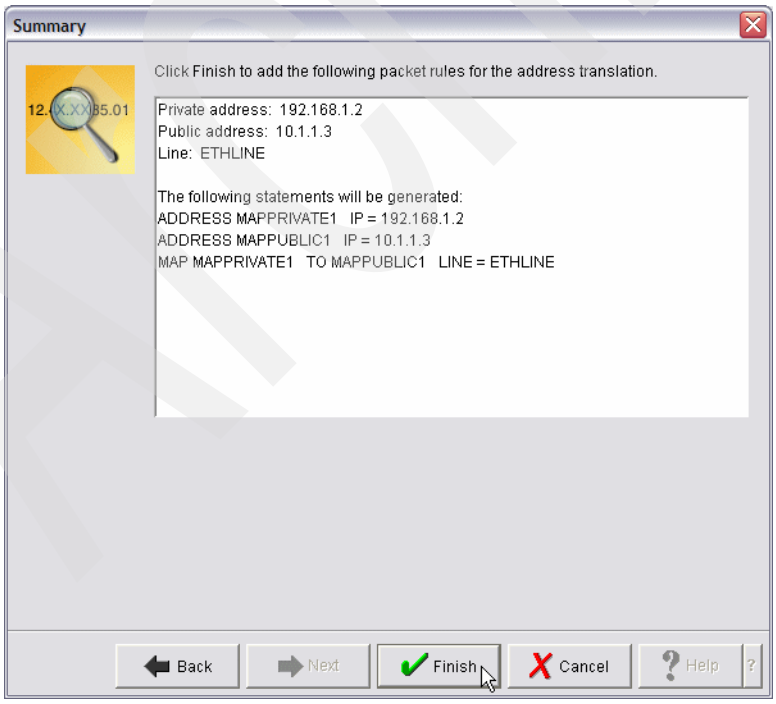


Figure 13-29 Address Translation wizard: Summary window for NAT on ETHLINE

12. The window shown in Figure 13-30 opens, displaying our NAT filter rule setup for LPAR2 where we map IP address 192.168.1.2 to 10.1.1.3 IP address.

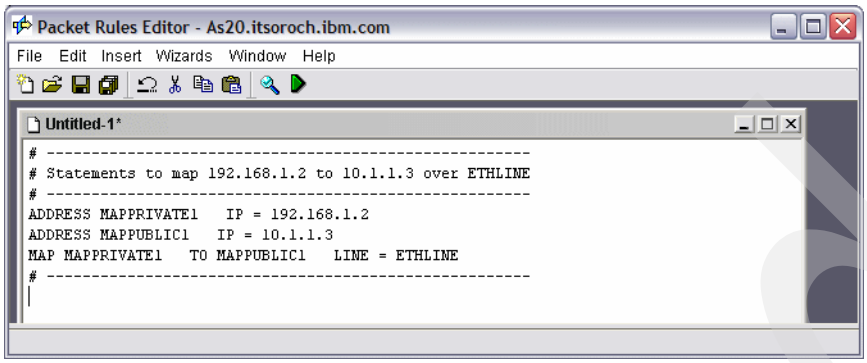


Figure 13-30 Address Translation wizard: NAT setup for LPAR2

13. We repeat steps 5 through 10 for LPAR3 and LINUX partitions. We use, respectively:
   - 192.168.1.3 and 10.1.1.4 for LPAR3
   - 192.168.1.4 and 10.1.1.5 for LINUX

   This brings us to a final setup window, as shown in Figure 13-31.

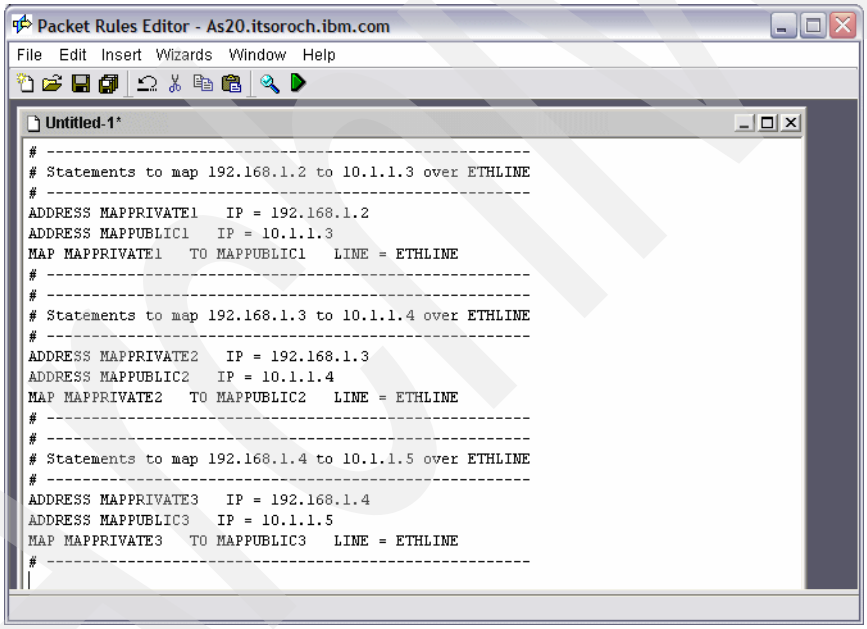


Figure 13-31 Address Translation wizard: NAT setup for LPAR2, LPAR3, and LINUX partitions

14. From the Menu bar in the Packet Rules Editor window, select **File** → **Save** and type a name for the file, as shown in Figure 13-32. We specify NAT for the filename in our example. Click **Save**. It is saved in the IFS on the System i as NAT.I3P in /qibm/UserData/OS400/TCPIP/PacketRules.

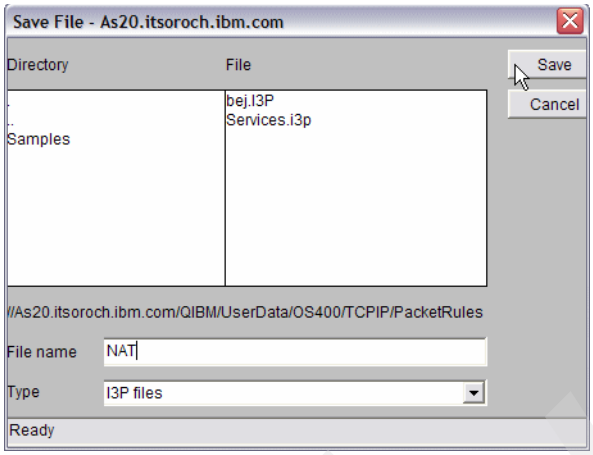


Figure 13-32 Address Translation wizard: Saving the NAT packet rule configuration

15. From the Menu bar in the Packet Rules Editor window, select **File** → **Verify Rules**. The Verify Packet Rules window is shown in Figure 13-33. Click **OK**.

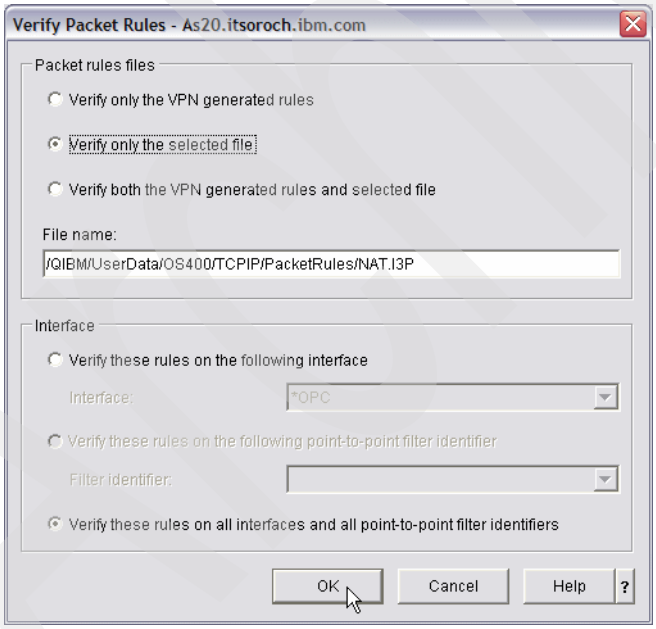


Figure 13-33 Packet Rules Editor: Verify Packet Rules

16. This opens the window shown in Figure 13-34. Look for successful verification in the lower pane.

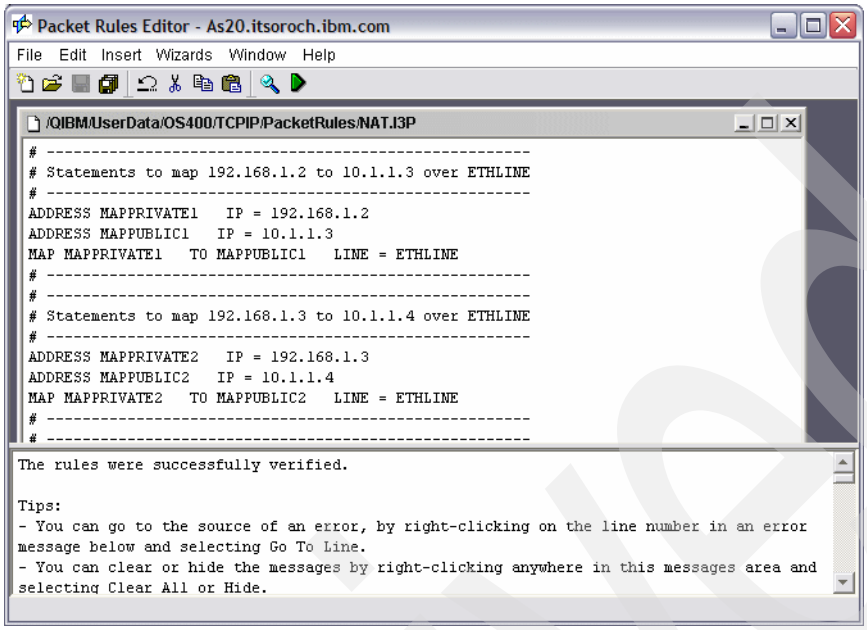


Figure 13-34 Packet Rules Editor: Verify Rules

17. Now select **File** → **Activate Rules** from the menu bar of the Packet Rules Editor window to open the window shown in Figure 13-35. Click **OK** to activate them.

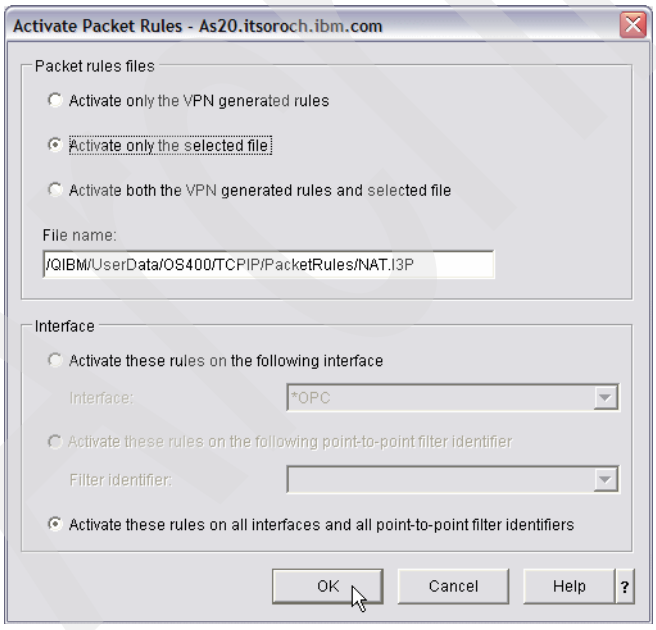


Figure 13-35 Packet Rules Editor: Activate Rules

18. The window shown in Figure 13-36 opens. The lower pane displays a message that the rules are successfully activated.

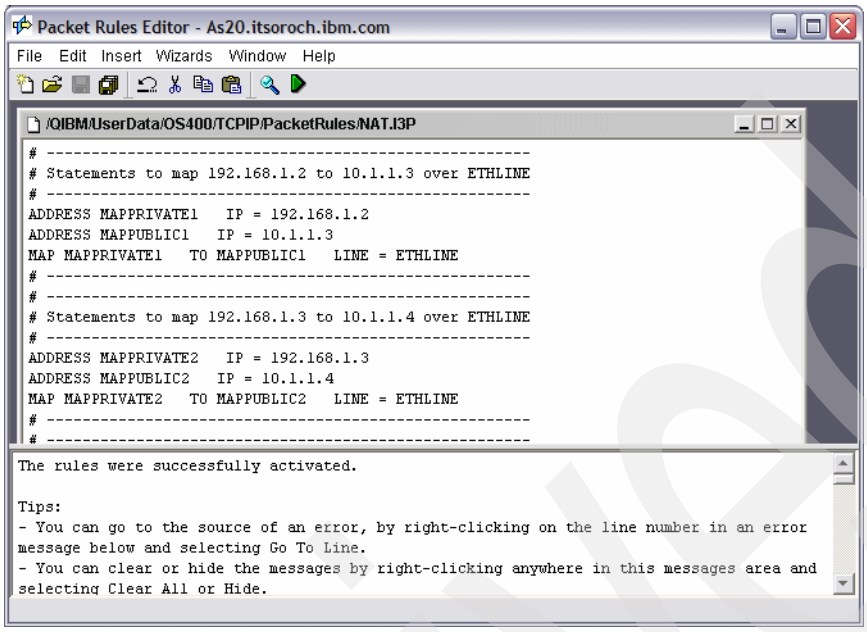


Figure 13-36 Packet Rules Editor: successful activation of packet rules

Tip: If the IP Filtering was not set up correctly using iSeries Navigator, you can deactivate the IP rules from a green screen by using the Remove TCP/IP Table (RMVTCPTBL) command.

Step 10:Verify the network communications

We can test our network communications by using the Verify TCP/IP Connection (PING) command:

- To test the outbound communications, we ping an external host on the ETHLINE 10.1.1.0 network from LPAR2, LPAR3, and LINUX.
- To test the inbound communications, we ping LPAR2, LPAR3, and LINUX from an external host on the ETHLINE 10.1.1.0 network.

13.3 Virtual Ethernet and routing scenario

This scenario describes the use of basic TCP/IP routing and Virtual Ethernet LAN on the System i with four logical partitions in order to be able to access it from an external LAN and from a remote network behind a router.

Problem definition

For this scenario, a customer has the network shown in Figure 13-37 on page 237. The System i has four different partitions:

- LPAR1: first partition running V5R4M0 of i5/OS
- LPAR2: second partition running V5R3M0 of i5/OS
- LPAR3: third partition running V5R4M0 of i5/OS
- LINUX: fourth partition is a guest partition running Linux hosted by LPAR1

We want to allow high-speed communications between all four partitions and to extend that communication to an external LAN. As our hardware has a limited number of card slots available for installing LAN cards, we need to find a solution that does not require additional LAN cards to be installed. We need access to all partitions on the System i from the external LAN as well as from the remote network behind the router.

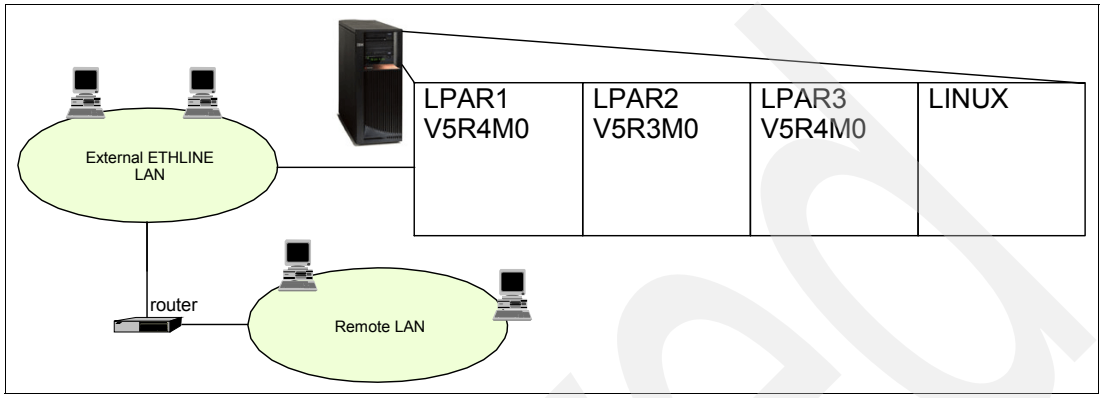


Figure 13-37 System i LPAR setup with four logical partitions and routing

Problem solution

We create the Virtual Ethernet network shown in Figure 13-38 to allow communications between the logical partitions on the System i. We use basic TCP/IP routing techniques to connect the Virtual Ethernet network to the external LAN as well as to the remote LAN. We will configure all necessary lines, interfaces, and routes to be able to access all partitions from the external LAN as well as from the remote network behind the router.

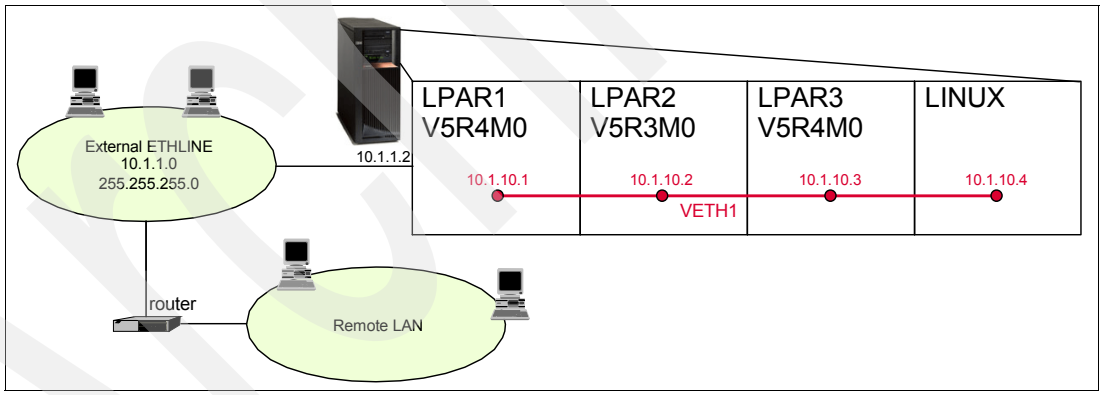


Figure 13-38 System i LPAR configuration setup with basic TCP/IP routing

The external TCP/IP interface 10.1.1.2 connects the System i to the external ETHLINE LAN. This is connected to a remote network through a router.

Assumptions

Let us assume the following:

- We have a System i with four logical partitions already set up:
  - The first partition (LPAR1) runs i5/OS V5R4M0.
  - The second partition (LPAR2) is running i5/OS V5R3M0.
  - The third partition (LPAR3) is running i5/OS V5R4M0.
  - The fourth partition (LINUX) is running Linux and has LPAR1 as the hosting partition.

- We have iSeries Access for Windows and iSeries Navigator (including the configuration and service component) installed.
- We added TCP/IP interface 10.1.1.2 for the external LAN ETHLINE on partition LPAR1.
- We enabled the logical partitions to participate in a Virtual Ethernet.
- We created the Virtual Ethernet line descriptions VETH1 on LPAR1, LPAR2, and LPAR3.
- We turned on IP datagram forwarding on partition LPAR1.

How-to

To configure the System i server accordingly, perform the following tasks:

- Step 1: Add and start the TCP/IP interface on Virtual Ethernet in LPAR1.
- Step 2: Add and start the TCP/IP interface on Virtual Ethernet in LPAR2.
- Step 3: Add and start the TCP/IP interface on Virtual Ethernet in LPAR3.
- Step 4: Add and start TCP/IP interface and default gateway on Virtual Ethernet.
- Step 5: Create the necessary routes on LPAR2 and LPAR3.
- Step 6: Configure the router for remote LAN clients.

Step 1: Add and start the TCP/IP interface on Virtual Ethernet in LPAR1

Add a TCP/IP interface for your Virtual Ethernet in LPAR1 by using the Add TCP/IP Interface (ADDTCPIFC) command:

```
ADDTCPIFC INTNETADR('10.1.10.1') LIND('VETH1') SUBNETMASK('255.255.255.0')
```

Start the TCP/IP interface for your Virtual Ethernet in LPAR1 by using the Start TCP/IP Interface (STRTCPIFC) command:

```
STRTCPIFC INTNETADR('10.1.10.1')
```

Step 2: Add and start the TCP/IP interface on Virtual Ethernet in LPAR2

Add a TCP/IP interface for your Virtual Ethernet in LPAR2 by using the Add TCP/IP Interface (ADDTCPIFC) command:

```
ADDTCPIFC INTNETADR('10.1.10.2') LIND('VETH1') SUBNETMASK('255.255.255.0')
```

Start the TCP/IP interface for your Virtual Ethernet in LPAR2 by using the Start TCP/IP Interface (STRTCPIFC) command:

```
STRTCPIFC INTNETADR('10.1.10.2')
```

Step 3: Add and start the TCP/IP interface on Virtual Ethernet in LPAR3

Add a TCP/IP interface for your Virtual Ethernet in LPAR3 by using the Add TCP/IP Interface (ADDTCPIFC) command:

```
ADDTCPIFC INTNETADR('10.1.10.3') LIND('VETH1') SUBNETMASK('255.255.255.0')
```

Start the TCP/IP interface for your Virtual Ethernet in LPAR3 by using the Start TCP/IP Interface (STRTCPIFC) command:

```
STRTCPIFC INTNETADR('10.1.10.3')
```

Step 4: Add and start TCP/IP interface and default gateway on Virtual Ethernet in LINUX

On this guest partition LINUX, set up networking by using the instructions or tools provided by your Linux distribution. Be sure to use the correct IP address, subnet mask, port, and gateway IP address.

In our example, we use the following:

Interface IP address `10.1.10.4`

Subnet mask 255.255.255.0
Gateway IP address 10.1.10.1
Network device VETH1

Step 5: Create the necessary routes on LPAR2 and LPAR3

To create the default routes to enable the packets to exit the Virtual Ethernet, use the Add TCP/IP Route (ADDTCPRTE) command on LPAR2 *and* LPAR3:

```
ADDTCPRTE RTEDEST(*DFTROUTE) SUBNETMASK(*NONE) NEXTHOP('10.1.10.1')
```

Step 6: Configure the router for remote LAN clients

In this scenario, we add a static route definition on the router so that it passes packets destined for the 10.1.10.0 network to the 10.1.1.2 interface.

The System i then routes the traffic to the partitions LPAR2, LPAR3, and LINUX across the Virtual Ethernet.

This will work fine for the remote clients on the remote network. This also works for the local clients on the external ETHLINE LAN as long as those clients recognize the router as their next hop. If that is not the case, then each of those clients needs a route in order to direct the 10.1.10.0 traffic to the 10.1.1.2 interface on the System i. This makes this an impractical method because if you have hundreds of clients, you may have to define hundreds of routes for them.

For information regarding the configuration of the router, refer to its documentation material.

14

Multilink in action

The PPP Multilink Protocol (MP) enables multiple PPP links to be grouped together to form a single virtual link or bundle. MP is used to provide additional bandwidth, fault tolerance, and dynamic bandwidth adjustment based on demand.

Tip: Prior to any work with an internal modem on the System i (such as the 2 V.90 port #2772 and the 4 V.92 port #2805 internal modem cards), you should change the Modem country or region ID using the Change Network Attributes (CHGNETA) command. As an example, the following is the 5250 command line to change the modem country or region ID to Belgium:

```
CHGNETA MDMCNTRYID(BE)
```

If you do not do this, you will see CPF message CPPC025.

14.1 Multilink: dynamic bandwidth allocation

MP allows the addition and subtraction of links based on utilization. Bandwidth utilization monitoring, which is only supported on the System i originator profile, monitors the line usage. Bandwidth Allocation Protocol (BAP) and Bandwidth Allocation Control Protocol (BACP) enable links to be added and removed dynamically based on demand.

Problem definition

Your System i initiates a PPP asynchronous connection on a daily basis to your corporate office. Most of the traffic on this link is casual Telnet and SMTP. What started out as a convenience link between the two sites is now being used more and more. Now, once a day, FTP is used across the connection to send sales information to the corporate office. When this occurs, Telnet response times become unpredictable. The current average is 40 Kbps to the corporate office. You would like a way to automatically increase the available bandwidth when your bandwidth utilization is high. How can this be done?

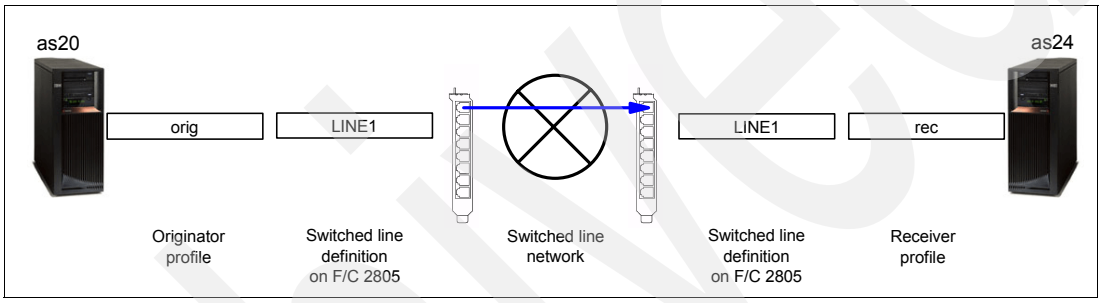


Figure 14-1 AS20 as call originator and AS24 as receiver: single link to support required bandwidth

Solution definition

The use of multilink enables multiple links to be used for one connection. These links can be added dynamically based on the utilization of the connection. We will configure multilink to bring a second line active (or more, if needed) when utilization of the 40 Kbps line exceeds 50% for more than 15 seconds.

Tip: A side benefit is that if both lines become active, and a failure occurs to one of the lines, the connection will remain active.

The network that you will configure will be similar to that shown in Figure 14-2.

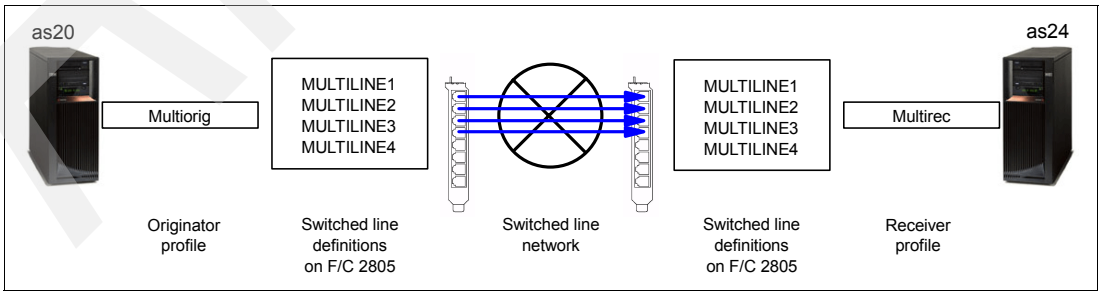


Figure 14-2 AS20 as call originator and AS24 as receiver: multiple links to support required bandwidth

Alternative solution

An alternative solution is to use QoS Differentiated Services to restrict the consumption of bandwidth by the FTP application. See Chapter 10, “Quality of Service (QoS)” on page 141 for more information.

How-to

For this scenario we assume that you have already:

1. Configured multiple asynchronous lines on each of your System i servers

   Any System i interface card that supports PPP can be used by MP. This scenario uses a four-port V.92 #2805.

   The lines we have configured are multiline1, multiline2, multiline3, and multiline4, as shown in Figure 14-2 on page 242.

2. Installed iSeries Navigator on a PC that is connected to each System i in your network

Here are the steps for creating both a System i Originator profile and a System i Receiver profile for a PPP connection that will make use of the MP to dynamically add and remove extra links based on our scenario description:

- Step 1: Create Originator profile to support MP and bandwidth utilization monitoring.
- Step 2: Create Receiver profile to support MP.
- Step 3: Test the configuration.

Step 1: Create Originator profile to support MP and bandwidth utilization monitoring

Create a System i PPP Originator profile that is configured for MP and bandwidth utilization monitoring. This is done for the System i as20.itsoroch.ibm.com (Figure 14-2 on page 242).

1. Start the iSeries Navigator by clicking **Start** → **Programs** → **IBM iSeries Access for Windows** → **iSeries Navigator**. The iSeries Navigator window appears.
2. Expand your System i connection. This may require that you enter a user ID and password.
3. Expand **Network** → **Remote Access Services**.
4. Right-click **Originator Connection Profiles** → **New Profile** (Figure 14-3).

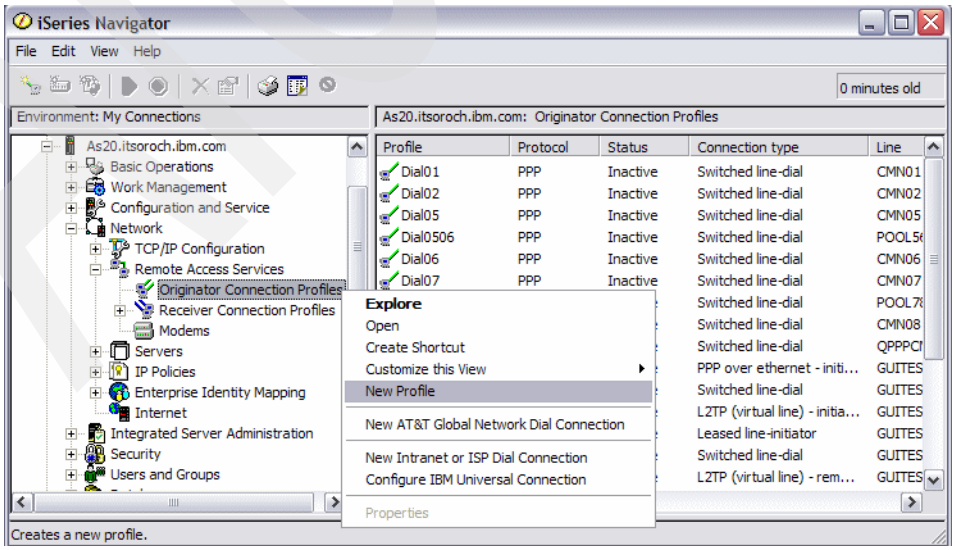


Figure 14-3 Originator Connection Profiles: New Profile

5. The New Point-to-Point Connection Profile Setup window appears (Figure 14-4). The only parameter that must be changed on this window is Type of line service. Select **Line pool** and then click **OK** to continue.

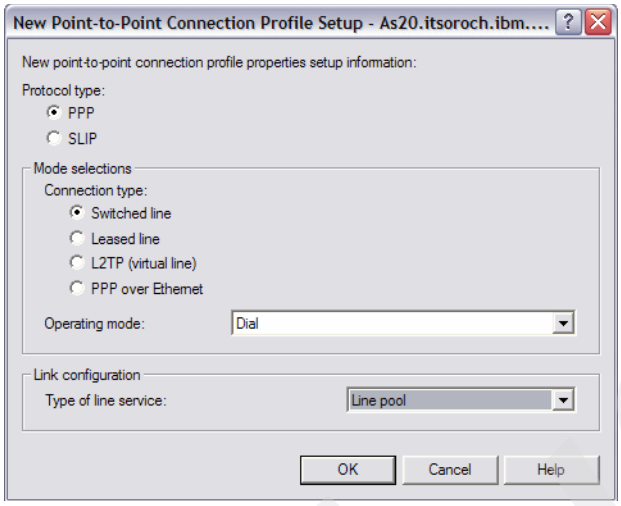


Figure 14-4 Originator Connection Profile: Type of line service: Line pool

6. The New Point-to-Point Profile Properties window displays the General tab, (Figure 14-5). We give the profile the name **Multiorig** as a shorthand note to ourselves that this is a multilink originator profile, which we are configuring. Optionally, enter a description. Click the **Connection** tab.

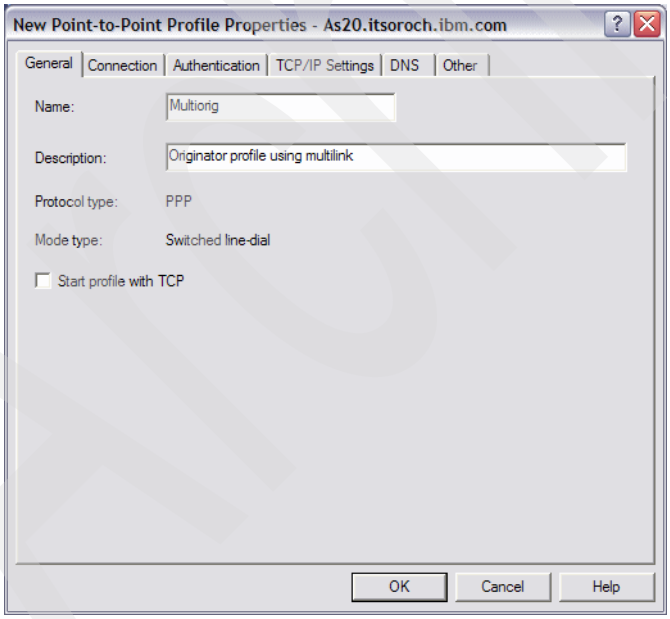


Figure 14-5 Originator Connection Profile: Name: Multiorig

7. On the Connection tab, we give our line pool the name `Multiline` (Figure 14-6). Click **New**.

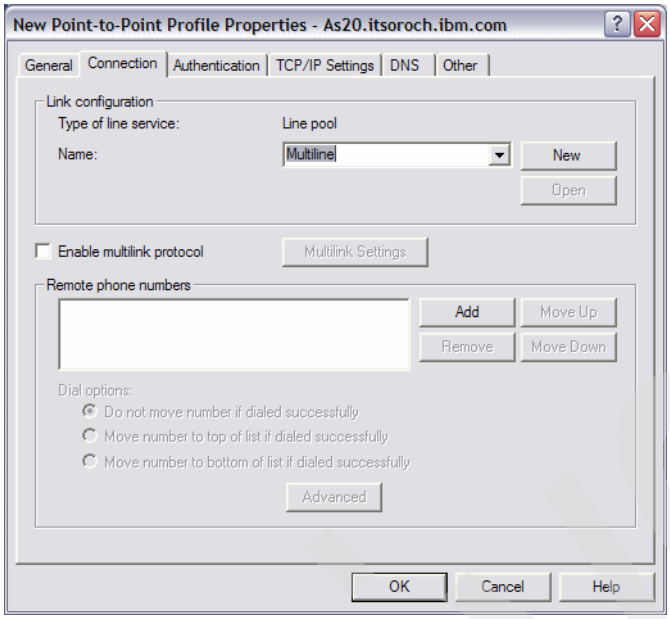


Figure 14-6 Connection tab: Name: Multiline

8. The New Line Pool Properties window appears (Figure 14-7). New lines are created by clicking the **New Line** button, and they appear in the left pane. Move lines from the left pane to the line pool in the right pane by using the buttons between the two panes. We have selected multiline1, multiline2, multiline3, and multiline4 to be in our line pool and to be used for MP. Click **OK**.

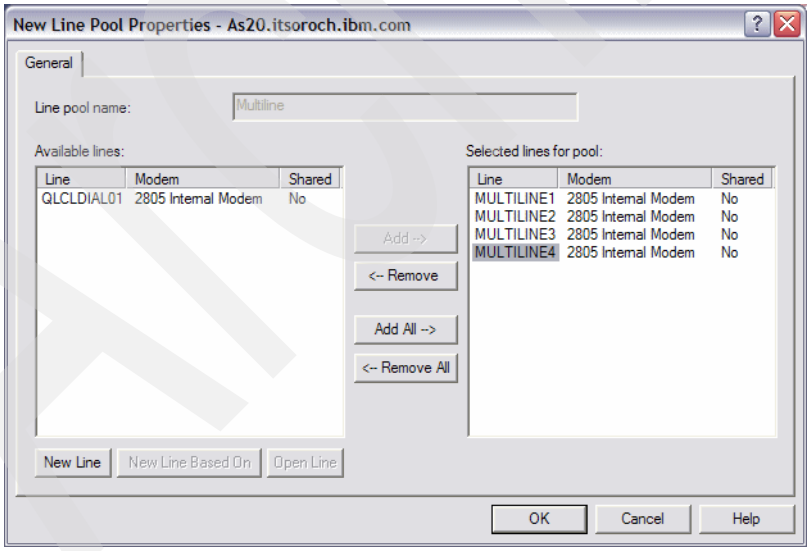


Figure 14-7 New Line Pool Properties: Selected lines for pool

9. This returns to the Connection tab (Figure 14-8). Click **Enable multilink protocol** to enable MP. If the Define Multilink Configuration window does not pop up (as shown in Figure 14-9), click **Multilink Settings** to proceed.

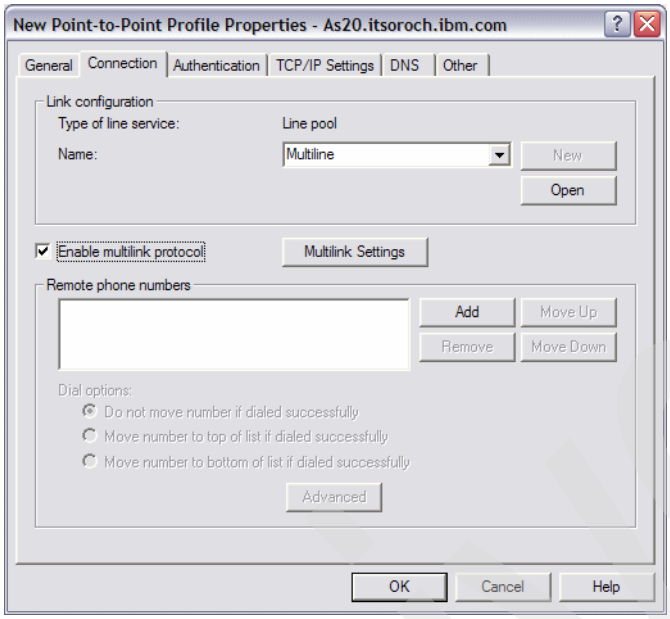


Figure 14-8 Connection tab: Enable multilink protocol

The Define Multilink Configuration window appears (Figure 14-9). Check the box for **Enable bandwidth utilization monitoring**. This enables you to set the utilization parameters that you desire.

As defined in this scenario, we want to start a second line when utilization of our 40 Kbps average link is above 50%. As explained in 4.2.2, “Bandwidth utilization monitoring” on page 83, we use the formula:

```
(40 Kbps / 28.8 Kbps) * 50% = 69%
```

The closest percentage to 69% that can be selected on the Add link pull-down menu is 75% (Figure 14-9).

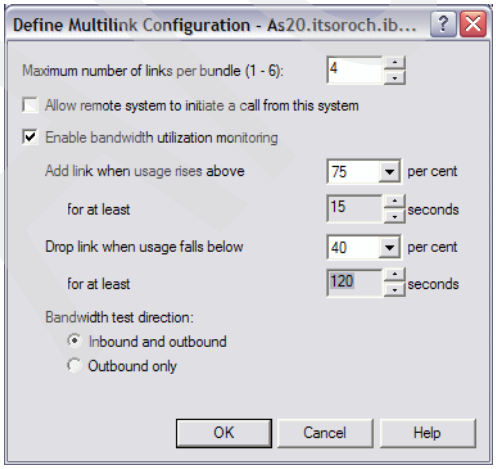


Figure 14-9 Define Multilink Configuration: Enable bandwidth utilization monitoring

We have set the link to be dropped if the utilization drops below 40% for 120 seconds. It is better to set the drop time value higher than the default of 15 seconds, because if the link is dropped and needed again, it will take time for the connection to reestablish. Click **OK**.

10. Click the **Add** button and enter the phone numbers for the remote location (Figure 14-10).

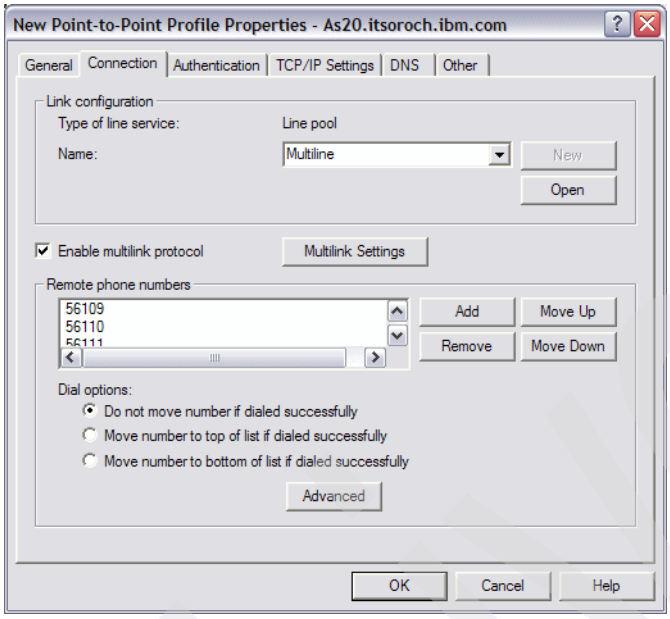


Figure 14-10 Connection tab: Remote phone numbers

Optionally, click the **Advanced** button for additional dial options for the list of phone numbers defined on the Connection tab. One very powerful option that we do not use in this scenario is the automatic redial of a line that was previously disconnected due to some kind of transient error.

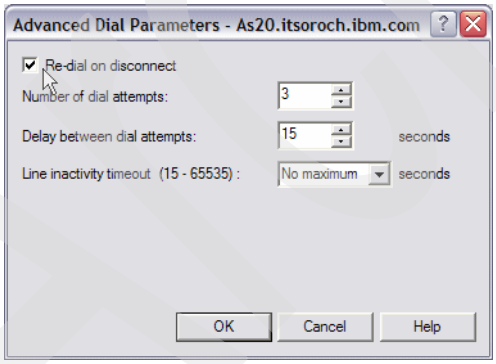


Figure 14-11 Advanced Dial Parameters: Redial on disconnect

Click **OK** or **Cancel**.

Click the **TCP/IP Settings** tab.

11. We allow the as24.itsoroch.ibm.com to assign addresses, and therefore take the defaults as shown in Figure 14-12. Click **OK** to finish.

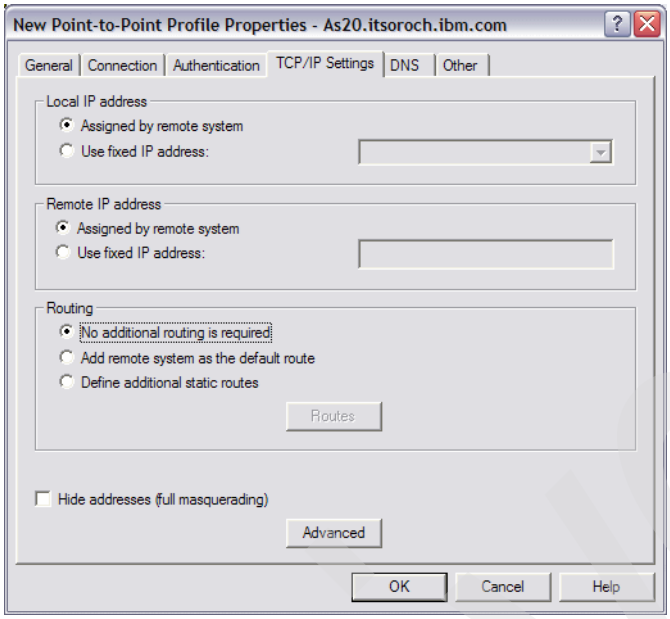


Figure 14-12 Originator Connection Profile: TCP/IP Settings

Step 2: Create Receiver profile to support MP

The steps for creating a Receiver profile and enabling MP, as demonstrated on our System i as24.itsoroch.ibm.com, are:

1. Start the iSeries Navigator by clicking **Start** → **Programs** → **IBM iSeries Access for Windows** → **iSeries Navigator**. The iSeries Navigator window appears.
2. Expand your System i connection. This may require that you enter a user ID and password.
3. Expand **Network** → **Remote Access Services**.
4. Right-click **Receiver Connection Profiles** → **New Profile** (Figure 14-13).

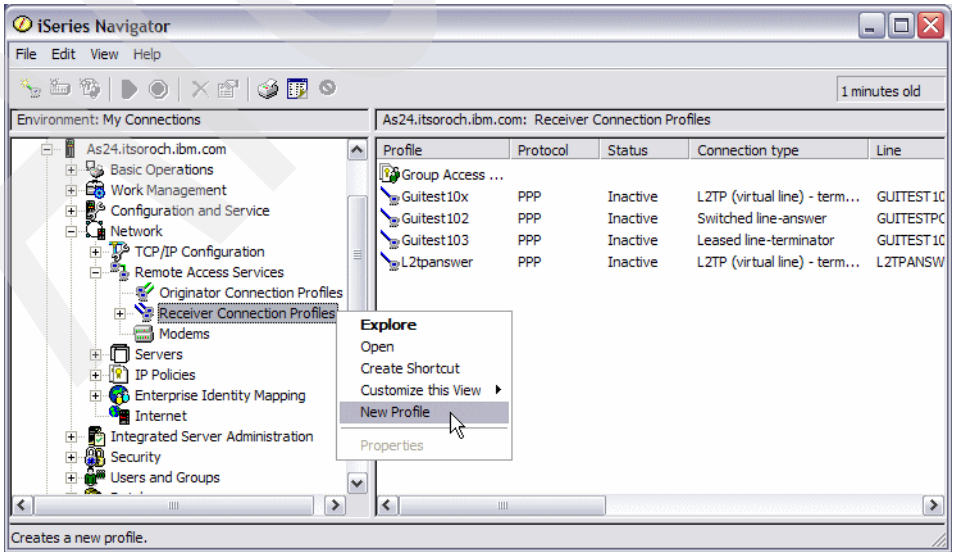


Figure 14-13 Receiver Connection Profiles: New Profile

5. The New Point-to-Point Connection Profile Setup window appears (Figure 14-14). The only parameter that has to be changed on this window is “Type of line service”. Select **Line pool** and then select **OK**.

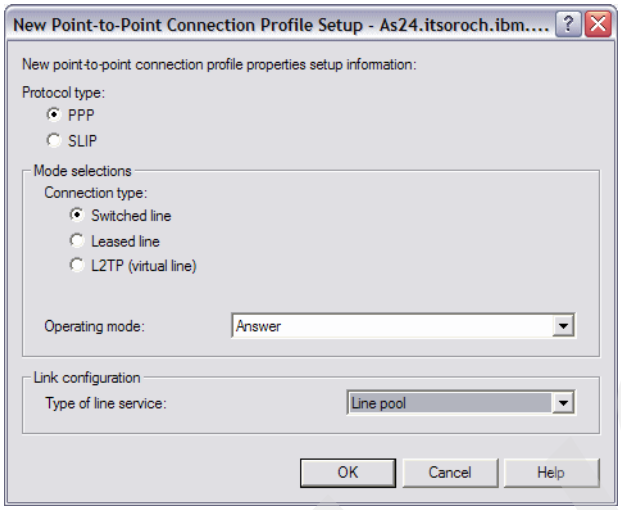


Figure 14-14 Receiver Connection Profile: Type of line service: Line pool

6. The New Point-to-Point Profile Properties window General tab is displayed, (Figure 14-15). We gave the profile the name **Multirec** as a shorthand note to ourselves that this is a multilink originator profile that we are configuring. Optionally, enter a description. If you would like the profile to start with TCP/IP to ensure that it is always ready to accept incoming calls, then check the box for **Start profile with TCP**. Click the **Connection** tab.

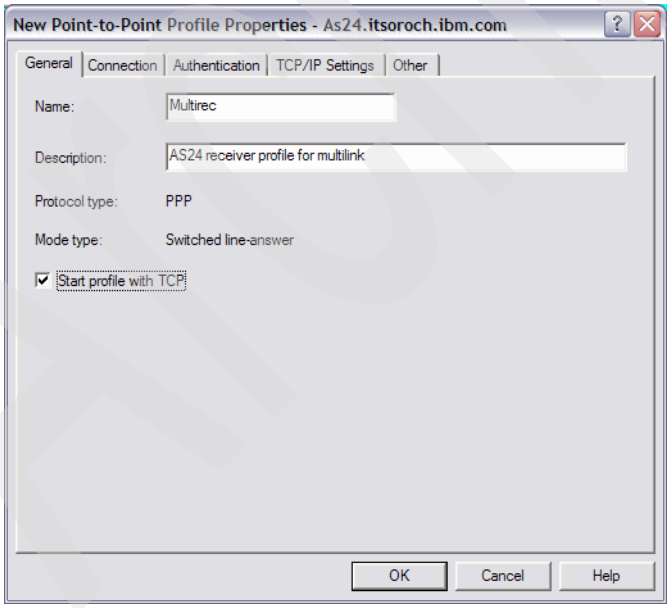


Figure 14-15 Receiver Connection Profile: General tab: Name: Multirec

7. On the Connection tab, we define the lines to be used for MP, giving our line pool the name `multiline`, as shown in Figure 14-16. Click **New** or **Open**.

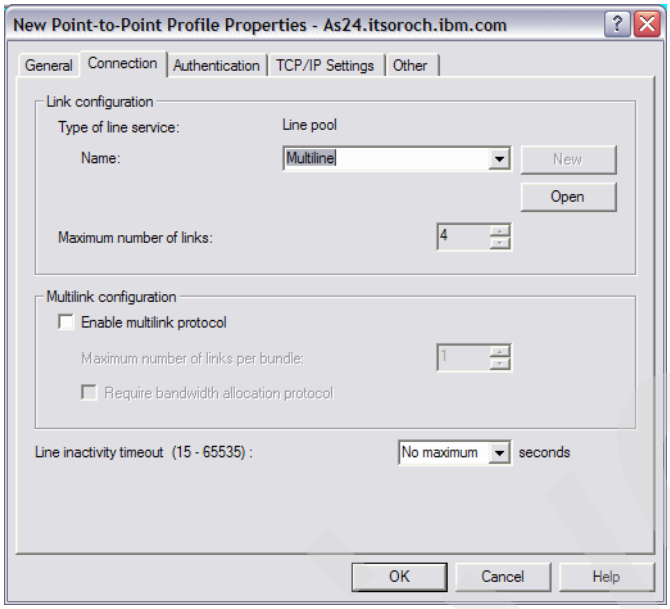


Figure 14-16 Receiver Connection Profile: Name: multiline

8. The New Line Pool Properties window appears (Figure 14-17). New lines are created by clicking the **New Line** button, and they appear in the left pane. Move lines from the left pane to the line pool in the right pane by using the buttons between the two panes. We have selected multiline1, multiline2, multiline3, and multiline4 to be in our line pool and to be used for MP. Click **OK.**

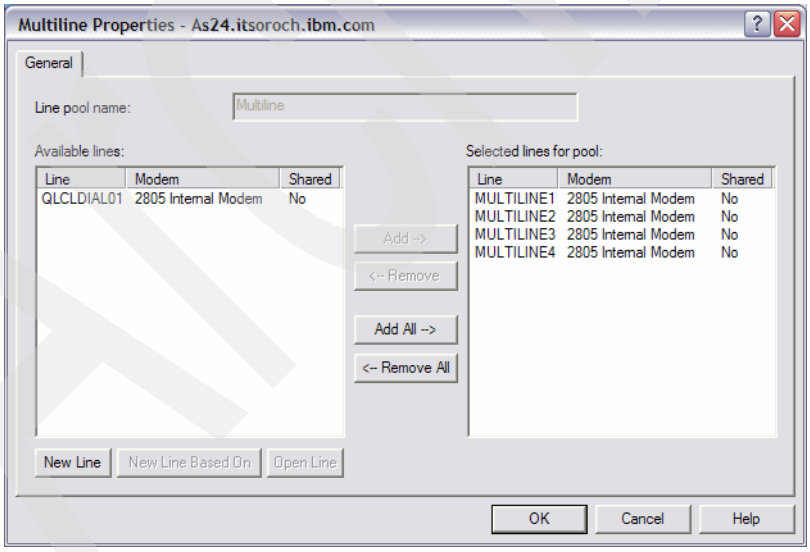


Figure 14-17 Receiver Connection Profile: multiline properties: Selected lines for pool

9. Returning to the Connection tab, click **Enable multilink protocol** (Figure 14-18). Four lines are available in our pool so we specify 4 as the maximum number of links per bundle.

> **Tip:** The default entry for Maximum number of links per bundle (1) is grayed out, but you can, of course, change that value using the arrows. You can increase this number up to the number of lines you specified in the line pool multiline.

We did not enable **Require bandwidth allocation protocol** for this profile because the Originator profile we are using supports BAP so there was no reason to require it to be used. You would select this option if you wanted to allow only BAP-enabled connections. If you have both BAP and non-BAP clients connecting to the same profile, do not enable this.

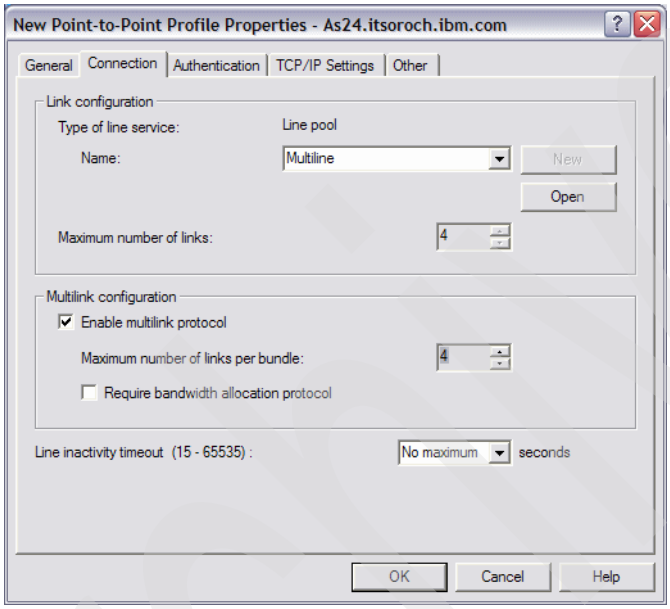


Figure 14-18 Receiver Connection Profile: Multilink configuration: Enable multilink protocol

10. Click the **TCP/IP** tab and specify a local and remote IP address. The local address must be an existing interface on the System i. We selected **172.23.10.2**. The remote IP address can be anything that you want it to be. We entered `15.15.15.15` for the remote, as shown in Figure 14-19. This enables the remote device to connect to the System i but prevents it from accessing the network that the System i resides on. Selecting an address in the 172.23.10.xxx range would allow access to the rest of the 172.23.10.xxx network if IP datagram forwarding is enabled. Click **OK**.

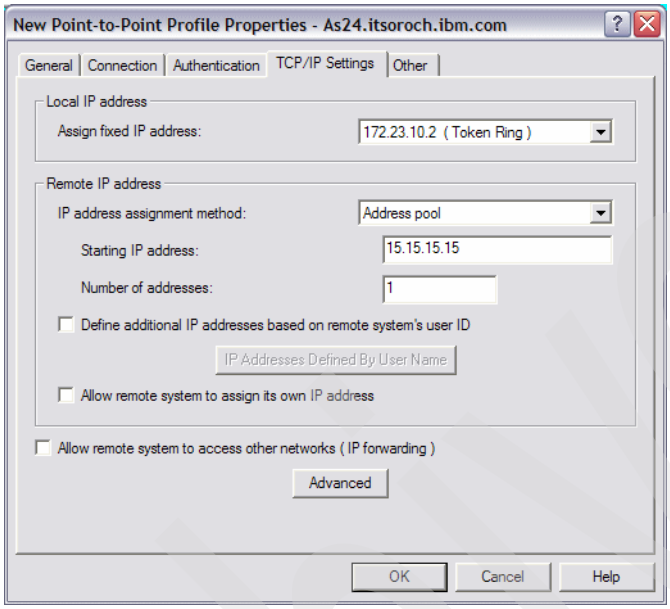


Figure 14-19 Receiver Connection Profile: TCP/IP settings

Step 3: Test the configuration

We test the configuration by using FTP to put a load on the initial line:

1. To start the Receiver profile on as24.itsoroch.ibm.com, right-click the profile and select **Start** (see Figure 14-20).

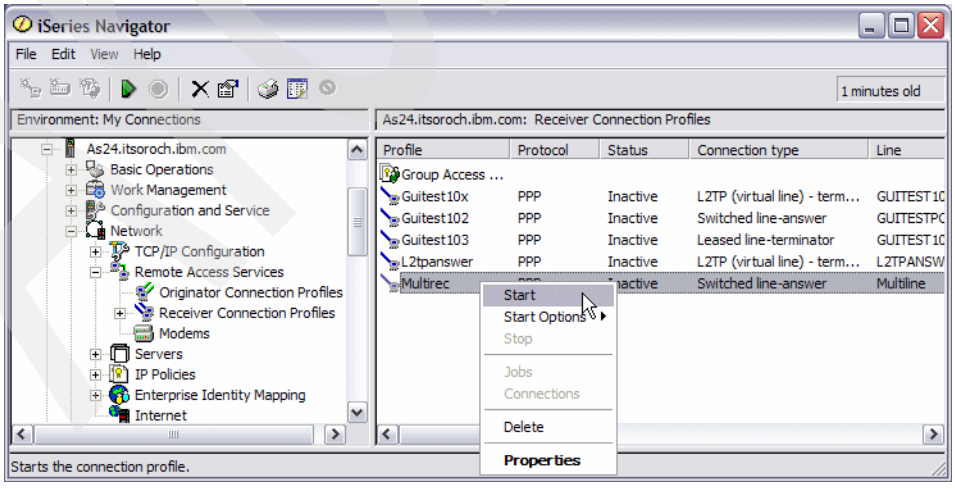


Figure 14-20 Receiver Connection Profile: Start

Tip: After starting your Receiver profile, the Status changes to Active Connections. This is an odd choice of words as you know that you have not yet made any calls from the other System i Originator profile. If you right-click the **Multirec Receiver** profile and select **Connections** from the context menu, you will see that the more detailed Primary Status is waiting for incoming calls. This is closer to what you would expect.

2. To start the Originator profile on as20.itsoroch.ibm.com, right-click the desired profile and select **Start** (see Figure 14-21).

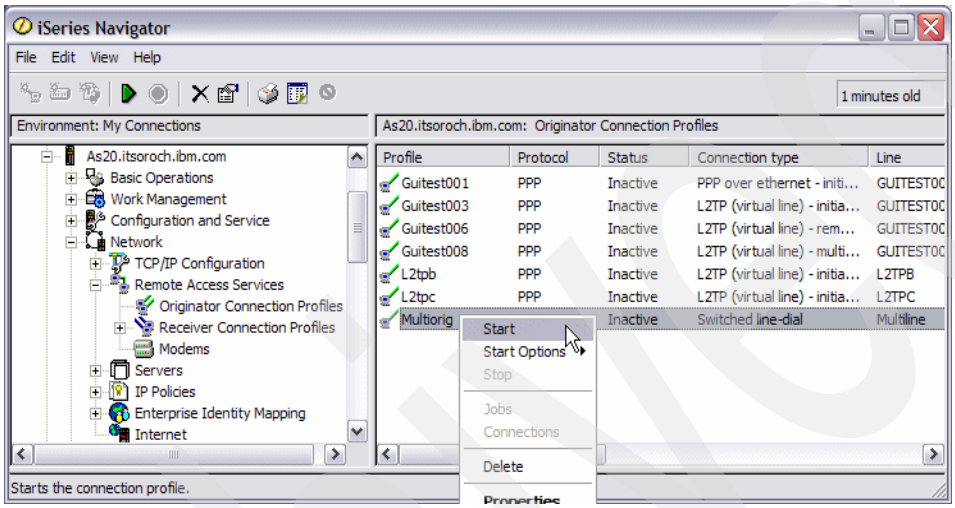


Figure 14-21 Originator Connection Profile: Start

3. Right-click an Originator profile on as20.itsoroch.ibm.com and select **Connections** to view the status of the connection (Figure 14-22).

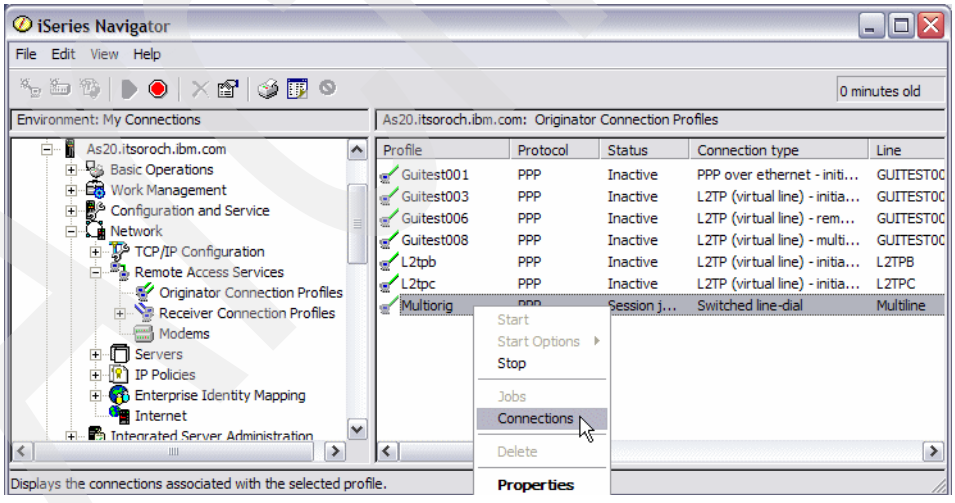


Figure 14-22 Originator Connection Profile: Connections

4. The Connections window is displayed (Figure 14-23). The first line shows the active connection between the two systems. The second line is the modem connection. We only see one link active at this time.

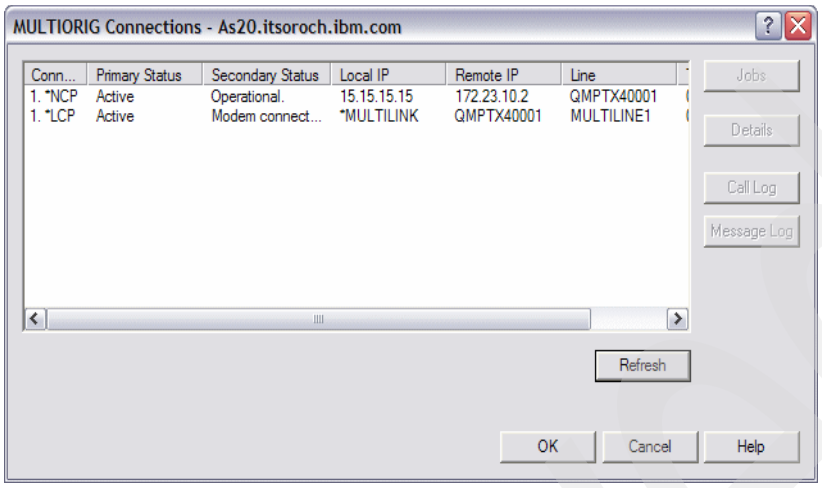


Figure 14-23 Originator Connection Profile: Multiorig connections

5. FTP is used to transfer a large file from AS20 to AS24. This will increase the utilization of the initial link and cause additional links to become active.
6. A refresh of the Connections window shows that a second link is active (Figure 14-24). The utilization of the line increased above 75% (assuming an effective modem-to-modem speed of 28.8 Kbps) for at least 15 seconds, so this activated a second line.

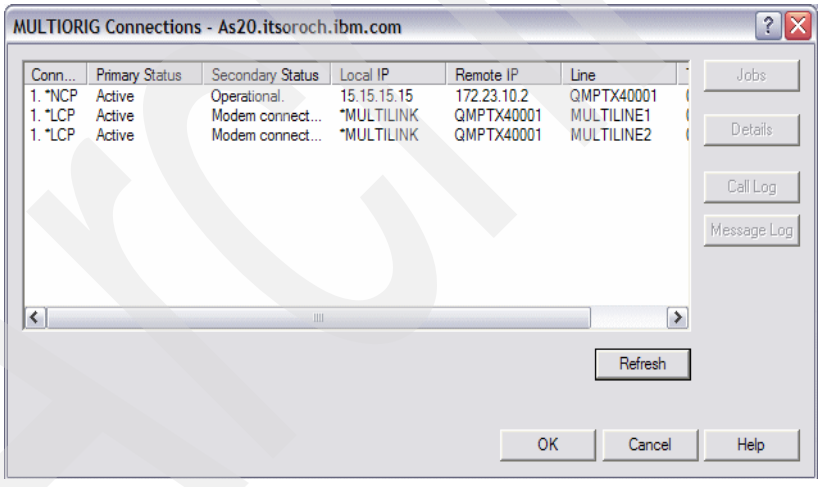


Figure 14-24 Originator Connection Profile: two lines active

7. We now cause a failure to one of the active lines. This is done in our test environment by simply disconnecting one of the active RJ-11 connectors from the F/C 2805 shown in Figure 14-2 on page 242. The Connections window shows that one modem connection remains active and keeps the connection functional (Figure 14-25).

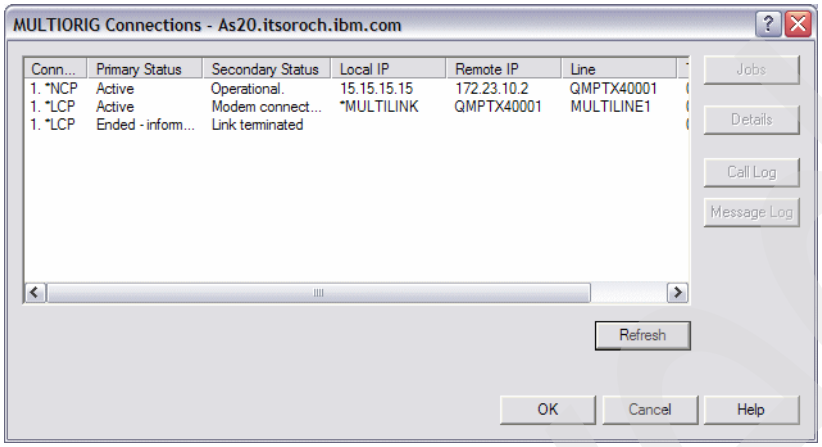


Figure 14-25 Originator Connection Profile: line failure

8. If the second line becomes functional and additional bandwidth is still required, the originator profile will reactivate the second line, as shown in Figure 14-26. This is discussed further in “Review, conclusions, and references” on page 268.

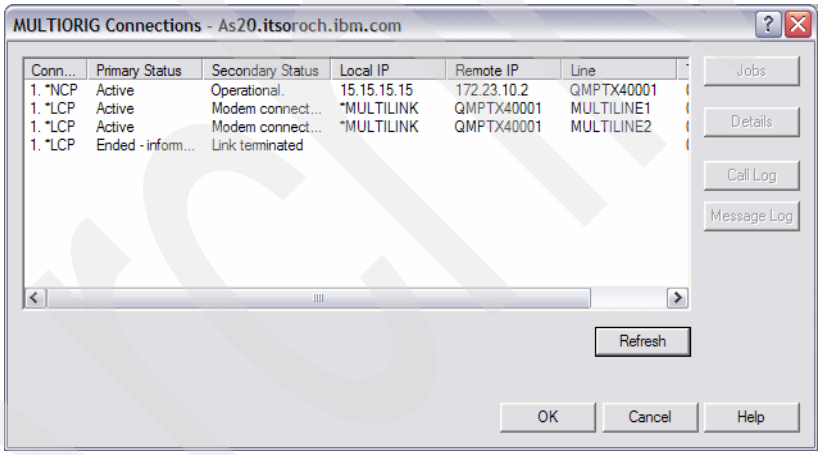


Figure 14-26 Originator Connection Profile: line re-activated

Review, conclusions, and references

Multilink allows for dynamic changes in bandwidth based on link utilization. We showed how an increase in the line utilization caused a second line to be brought active. In addition, Multilink offers fault tolerance at the link level when multiple links are active. See 14.2, “Multilink: Fault tolerance” on page 255 for an example that focuses more on the fault tolerance nature of MP.

14.2 Multilink: Fault tolerance

Multilink gives us the ability to incorporate fault tolerance at the link layer into PPP on the System i.

Problem definition

You have a very important transaction that takes place every evening across a single PPP connection (see Figure 14-27). Your System i transmits data asynchronously to another System i in your company. If the transaction does not complete due to a PPP line failure, business the next day is held up for hours. What can be done to minimize the affect of a line failure during transmission?

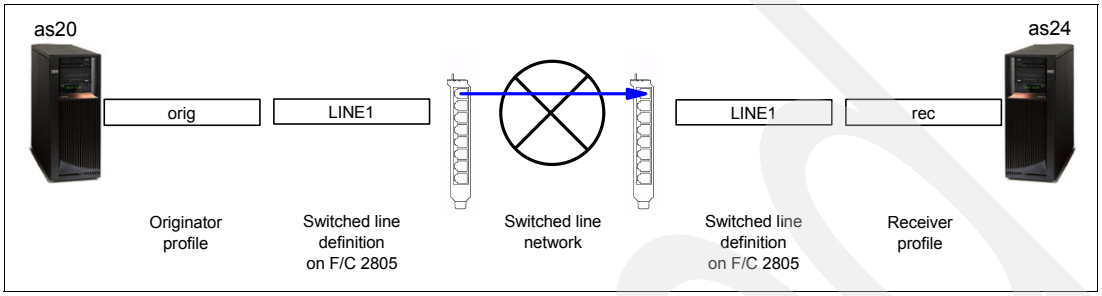


Figure 14-27 AS20 as call originator and AS24 as receiver: single link as single point of failure

Solution definition

Multilink gives the ability to start multiple lines at one time. If one line goes down, the connection will continue to run over the remaining lines. Having multiple lines active also results in a sustained increase of bandwidth.

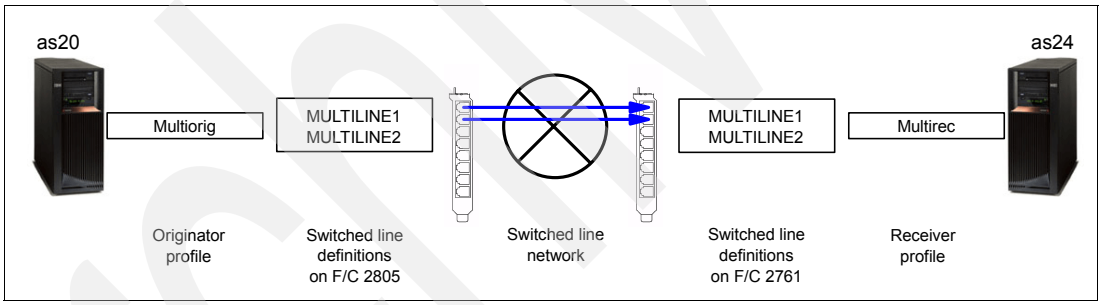


Figure 14-28 AS20 as call originator and AS24 as receiver: multiple links

How-to

For this scenario we assume that you have already:

1. Configured multiple asynchronous lines on each of your i5/OS servers.

   Any System i interface card that supports PPP can be used by MP. This scenario uses a four-port V.92 #2805.

   The lines that we have configured are multiline1 and multiline2, as shown in Figure 14-28.

2. Installed iSeries Navigator on a PC that is connected to each System i in your network

Here are the steps for creating System i Originator and Receiver profiles for a PPP connection, which will make use of the MP to provide fault tolerance by having multiple links active for a connection:

- Step 1: Create an Originator profile to support fault tolerance.
- Step 2: Create a Receiver profile to support fault tolerance.
- Step 3: Test the configuration.

Step 1: Create an Originator profile to support fault tolerance

We create a System i PPP Originator profile that is configured for MP for our System i as20.itsoroch.ibm.com, as shown in Figure 14-28 on page 256:

1. Start the iSeries Navigator by clicking **Start** → **Programs** → **IBM iSeries Access for Windows** → **iSeries Navigator**. The iSeries Navigator window opens.
2. Expand your i5/OS server. This may require that you enter a user ID and password.
3. Expand **Network** → **Remote Access Services**.
4. Right-click **Originator Connection Profiles** → **New Profile**, as shown in Figure 14-29.

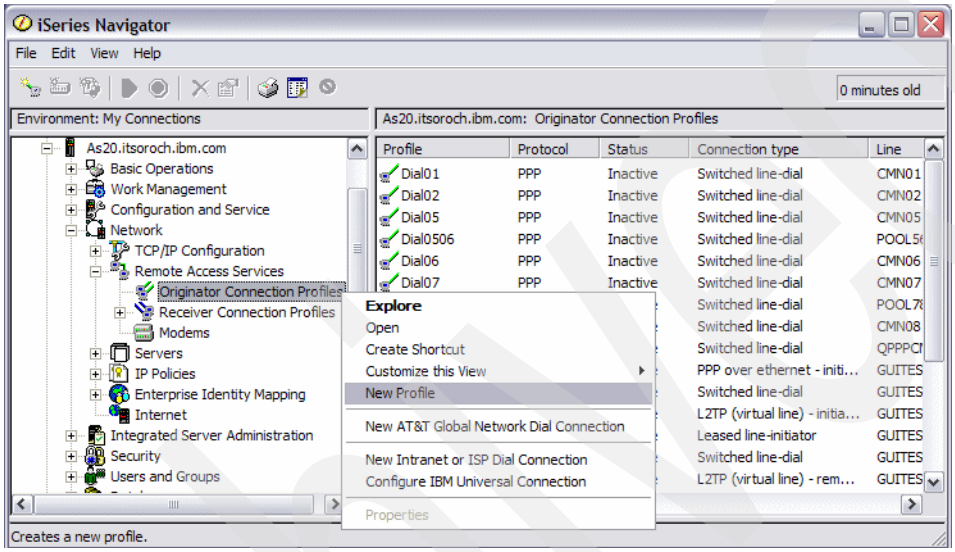


Figure 14-29 Originator Connection Profile: New Profile

5. The New Point-to-Point Connection Profile Setup window appears as shown in Figure 14-30. The only parameter that has to be changed on this window is **Type of line service**. Select **Line pool** and click **OK**.

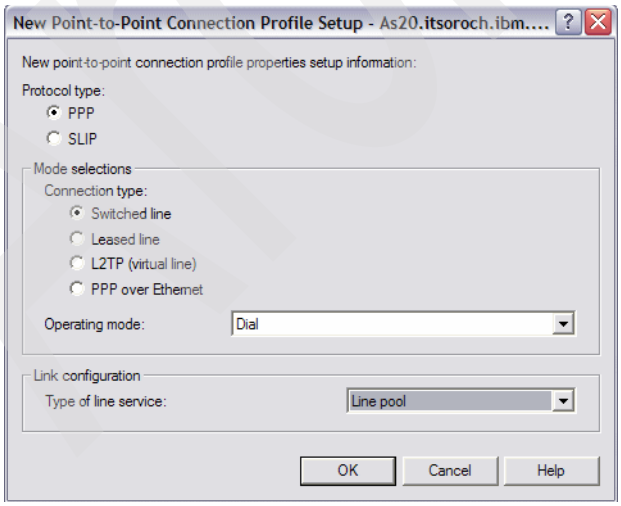


Figure 14-30 Originator Connection Profile: Type of line service: Line pool

6. The General tab of the New Point-to-Point Profile Properties window is displayed (Figure 14-31). We named the profile `Multiorig` as a shorthand note to ourselves that this we are configuring a multilink originator profile. Optionally, enter a description. Click the **Connection** tab.

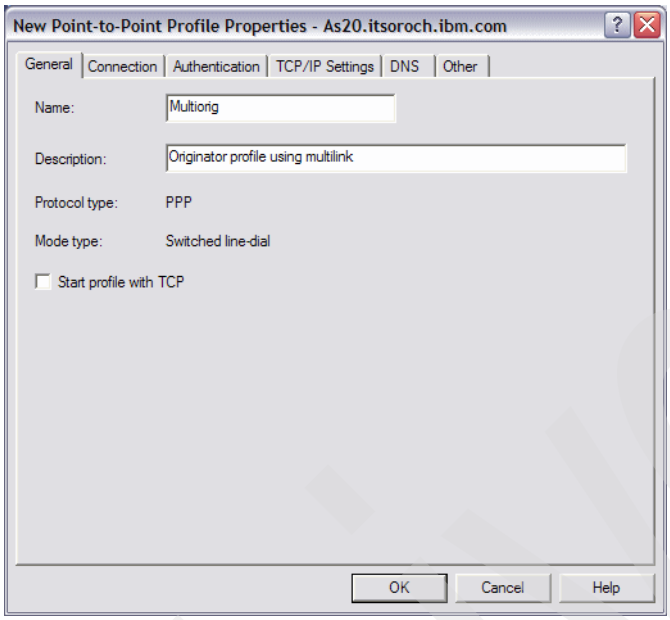


Figure 14-31 Originator Connection Profile: Name: Multiorig

7. From the Connection tab, we name our line pool `Multiline`, as shown in Figure 14-32. Click the **New** button.

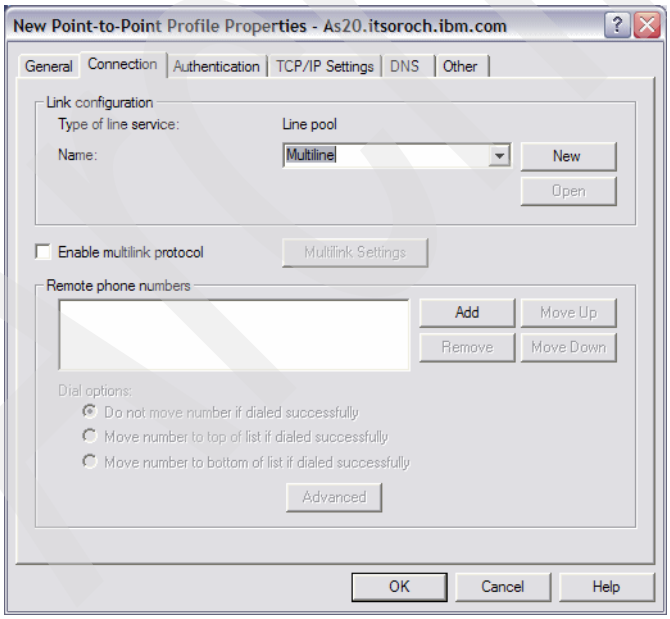


Figure 14-32 Originator Connection Profile: Connection tab: Name: Multiline

8. This opens the New Line Pool Properties window (Figure 14-33). New lines are created by clicking the **New Line** button, and they appear in the left pane. Move lines to the line pool in the right pane by using the buttons between the two panes. We have selected multiline1 and multiline2 to be in our line pool, and to be used for MP. Click **OK**.

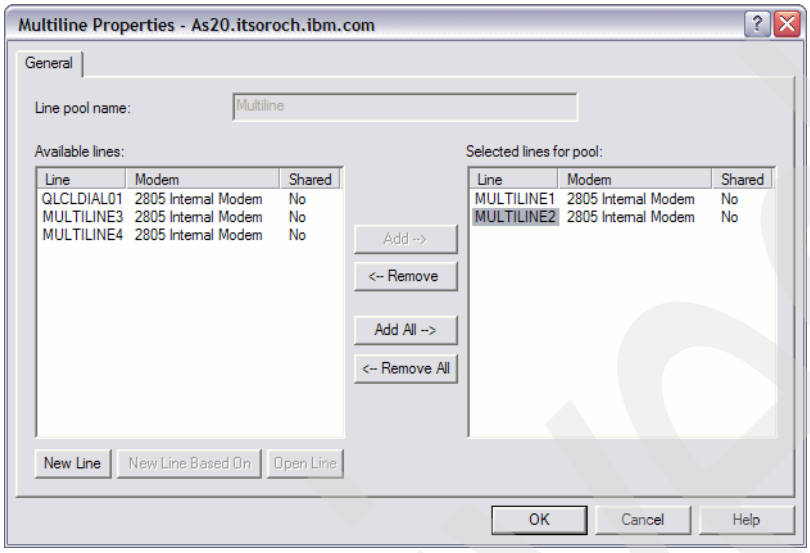


Figure 14-33 Originator Connection Profile: New Line Pool Properties: Selected lines for pool

9. This returns us to the Connection tab (Figure 14-34). Select **Enable multilink protocol** to enable MP.

   If the Define Multilink Configuration window (Figure 14-35 on page 260) does not pop up, click **Multilink Settings** to proceed.

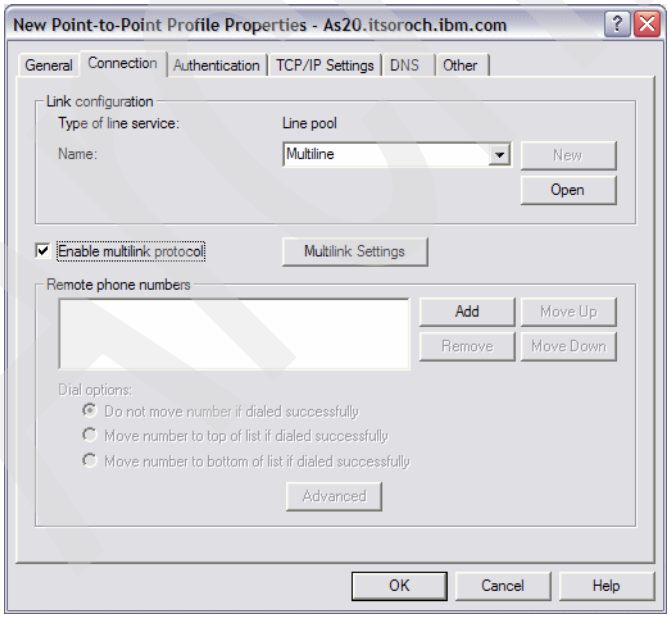


Figure 14-34 Originator Connection Profile: Connection tab: Enable multilink protocol

10. The Define Multilink Configuration window appears as shown in Figure 14-35. The maximum number of links per bundle should remain at 2 because we plan to use two links for the connection. Click **OK**.

> **Important:** To force the Originating profile to establish a multilink connection with the maximum number of links (thus maximizing the multilink bundle's chance at surviving many single link failures), *do not* select **Enable bandwidth utilization monitoring**.

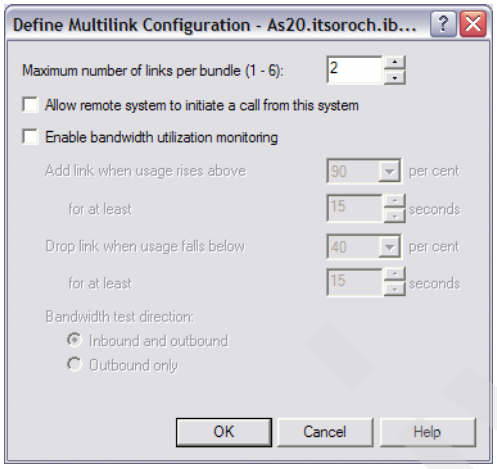


Figure 14-35 Define multilink configuration

11. Click the **Add** button and input the phone numbers for the remote location (Figure 14-36).

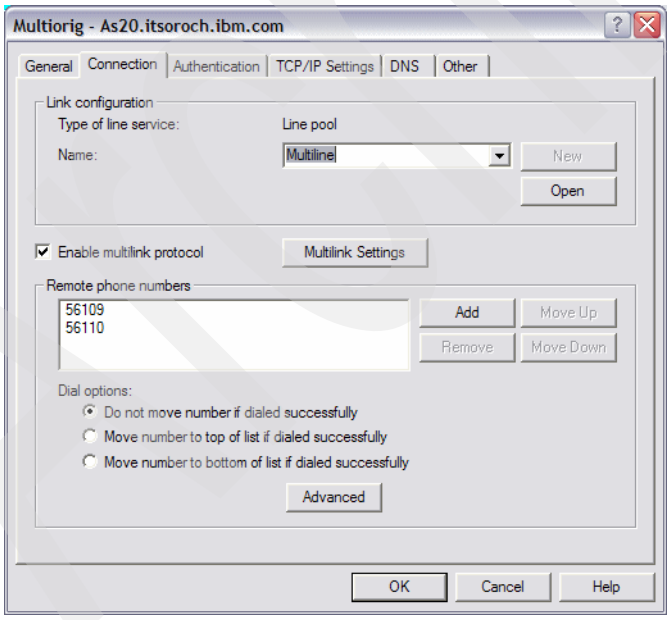


Figure 14-36 Originator Connecting Profile: Connection tab: Remote phone numbers

12. Click the **Advanced** button. This offers additional dial options for the list of phone numbers defined on the Connection page. One very powerful option that we use in this scenario is

the automatic redial of a line that was previously disconnected due to some kind of transient error.

Select **Re-dial on disconnect** and optionally change any of the other parameters. Click **OK** to return to the Connection tab. Click the **TCP/IP Settings** tab.

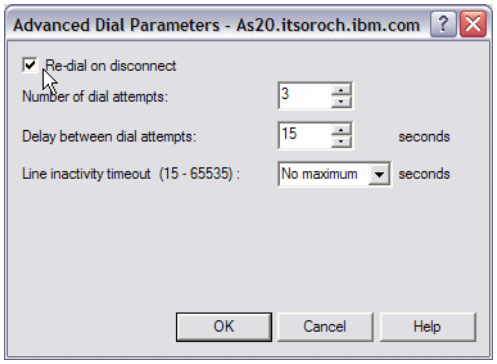


Figure 14-37 Advanced Dial Parameters: Redial on disconnect

13. We want the as24.itsoroch.ibm.com to be able to assign addresses, so we accept the defaults shown in Figure 14-38. Click **OK** to finish.

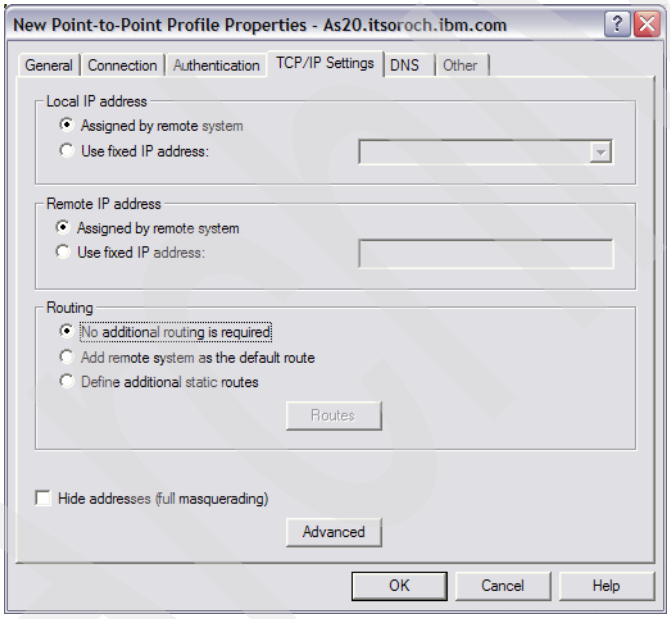


Figure 14-38 Originator Connection Profile: TCP/IP Settings

Step 2: Create a Receiver profile to support fault tolerance

Create a Receiver profile for multilink connection for the purpose of fault tolerance. This is done on System i as24.itsoroch.ibm.com (as shown in Figure 14-28 on page 256):

1. Start the iSeries Navigator by clicking **Start** → **Programs** → **IBM iSeries Access for Windows** → **iSeries Navigator**. The iSeries Navigator window appears.
2. Expand your iSeries server. This may require that you enter a user ID and password.
3. Expand **Network** → **Remote Access Services**.

4. Right-click **Receiver Connection Profiles** and click **New Profile** (Figure 14-39).

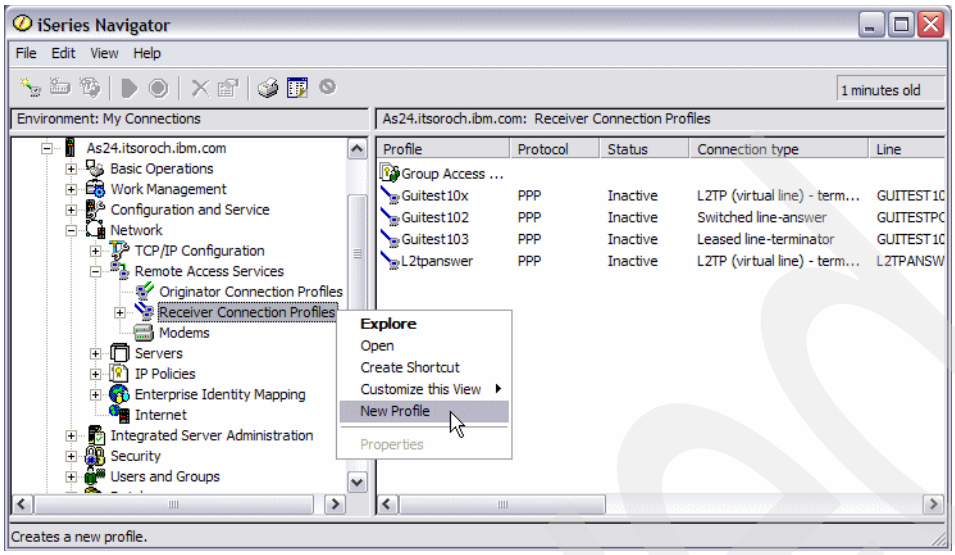


Figure 14-39 Receiver Connection Profiles: New Profile

5. The New Point-to-Point Connection Profile Setup window appears (Figure 14-40), and the only parameter to change on this window is **Type of line service**. Select **Line pool** and then click **OK**.

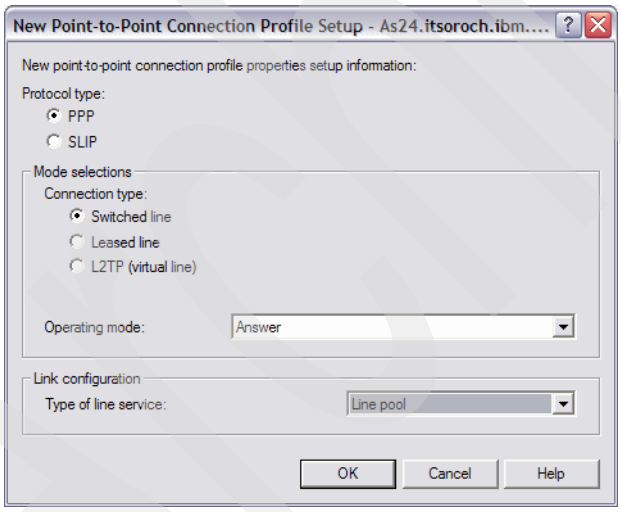


Figure 14-40 Receiver Connection Profile: Type of line service: Line pool

6. The General tab of the New Point-to-Point Profile Properties window is displayed (Figure 14-41). We gave the profile the name of `Multirec` as a shorthand note to ourselves that we are configuring a multilink originator profile. Optionally, enter a description. Click the **Connection** tab.

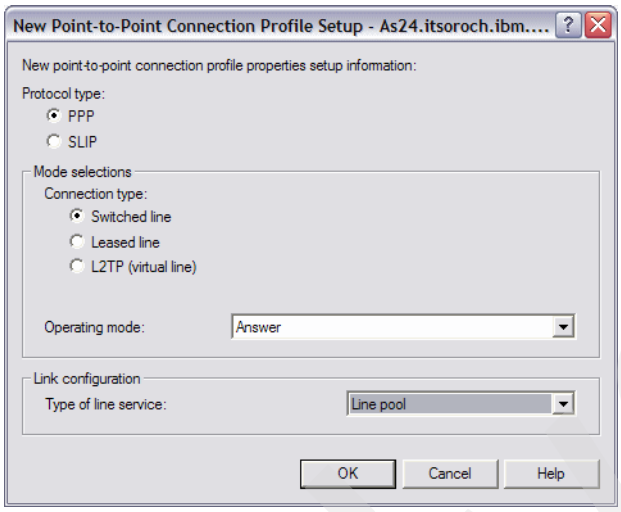


Figure 14-41 Receiver Connection Profile: General tab: Name: Multirec

7. From the Connection tab, we define the lines to be used for MP. We have given our line pool the name `multiline` (Figure 14-42). Click **Next**.

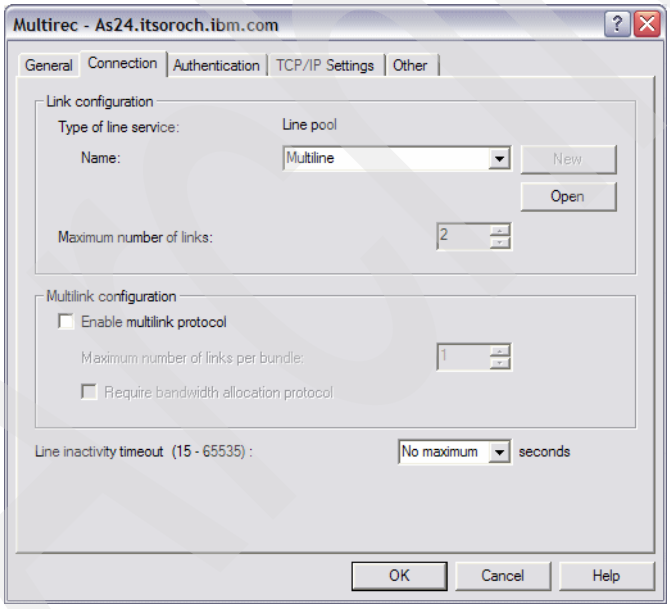


Figure 14-42 Receiver Connection Profile: Line pool name: multiline

8. The new line pool properties window appears (Figure 14-43). New lines are created by clicking the **New Line** button, and they appear in the left pane. Move lines from the left pane to the line pool in the right pane by using the buttons between the two panes. We have selected multiline1 and multiline2 to be in our line pool, and to be used for MP. Click **OK**.

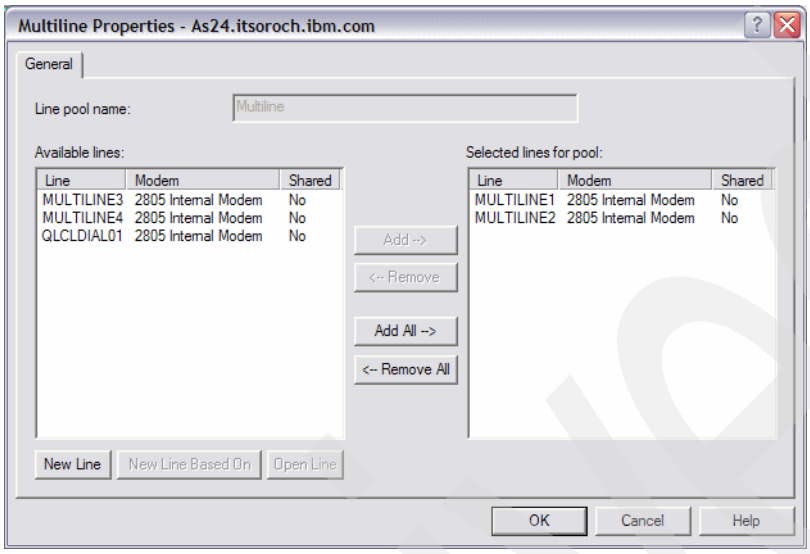


Figure 14-43 Receiver Connection Profile: multiline Properties: Selected lines for pool

9. This returns us to the Connection tab. Click **Enable multilink protocol** as shown in Figure 14-44. Two lines are available in our pool, so we specify 2 for Maximum number of links per bundle.

We did not enable the Require bandwidth allocation protocol setting for this profile. This is because our Originator profile supports BAP, so there was no reason to require it to be used here. You would enable this option if you wanted to allow only BAP-enabled connections. If you have both BAP and non-BAP clients connecting to the same profile, you would not enable this.

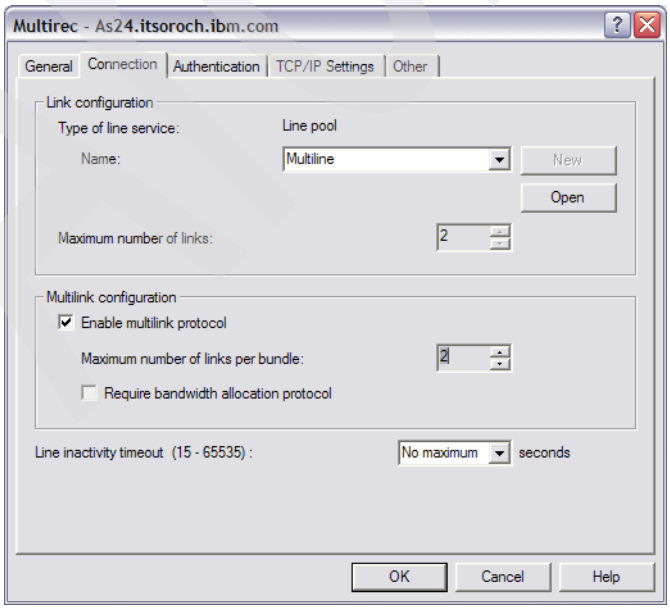


Figure 14-44 Receiver Connection Profile: Multilink configuration: Enable multilink protocol

10. Click the **TCP/IP Settings** tab. We need to specify a local and remote IP address. The local address must be an existing interface on the System i, and the remote IP address can be anything that you want it to be. We specified 172.23.10.110 for the remote as shown in Figure 14-45. We also selected the check box to **Allow remote system to access other networks** so that the client will have access to the networks attached to AS24. Click **OK**.

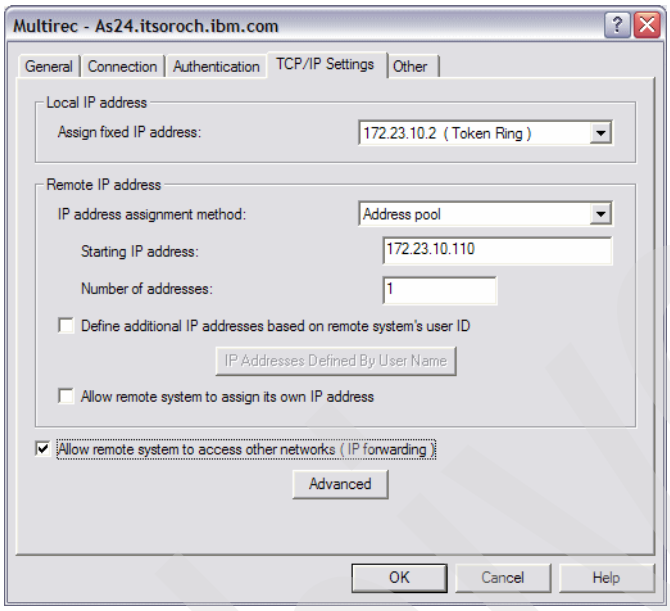


Figure 14-45 Receiver Connection Profile: TCP/IP settings

Step 3: Test the configuration

We test the configuration by first establishing a connection over both links in the MP bundle, and then simply disconnecting one of the active RJ-11 connectors from the F/C 2805 (shown in Figure 14-28 on page 256).

1. To start the Receiver profile on as24.itsoroch.ibm.com, right-click the Profile and select **Start** (Figure 14-46).

> **Tip:** After Start completes for your Receiver profile, the status changes to Active Connections. This is an odd choice of words as you have not yet made any calls from the other System i Originator profile. If you right-click the **Multirec Receiver** profile and select **Connections** from the context menu, you see that the more detailed Primary Status is waiting for an incoming call. This is closer to what you would expect.

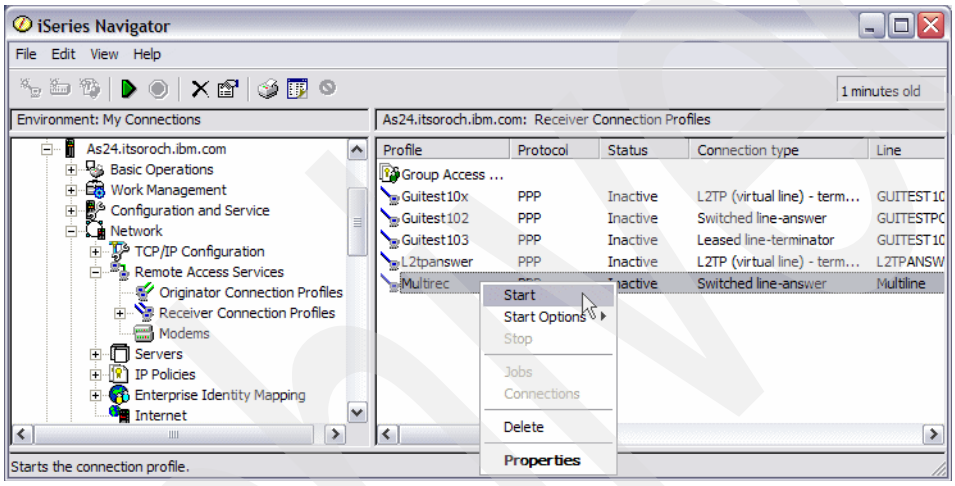


Figure 14-46 Receiver Connection Profile: Start

2. To start the Originator profile on as20.itsoroch.ibm.com, right-click the profile and select **Start** (Figure 14-47).

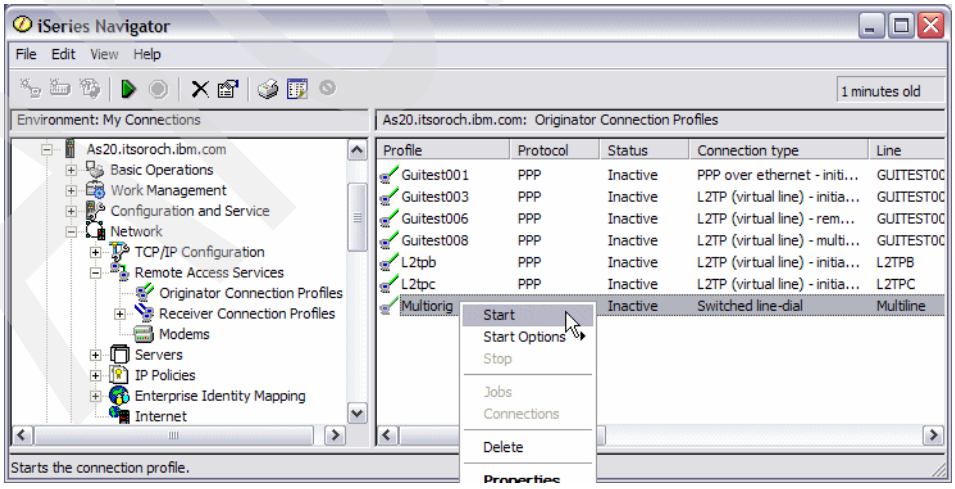


Figure 14-47 Originator Connection Profile: Start

3. Right-click the profile on as20.itsoroch.ibm.com again and select **Connections** to view the status of the connection (Figure 14-48).

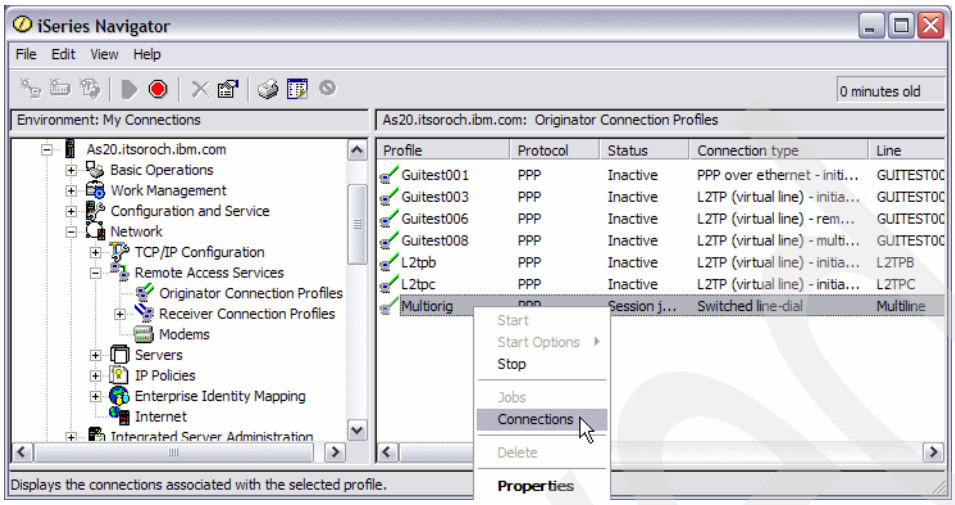


Figure 14-48 Originator Connection Profile: Connections

4. This displays the profile's Connections window (Figure 14-49). The first line shows the active connection between the two systems. The second and third lines show the modem connections. We currently have two active links.

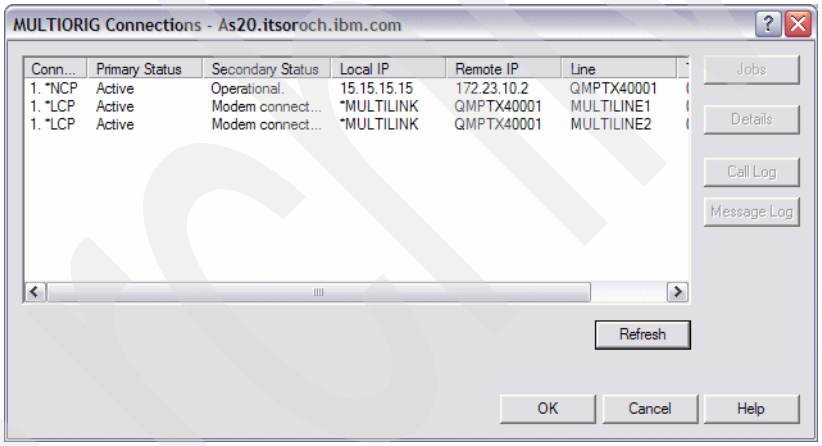


Figure 14-49 Originator Connection Profile: Multiorig connections: two lines active

5. We now cause a failure to one of the lines. You can do this simply by pulling out one of the RJ-11 connectors from either the System i interface or the PBX or phone switch. The Connections window shows that one modem connection remains active and keeps the connection functional, as seen Figure 14-50.

Note: If bandwidth utilization monitoring is not used, the System i PPP originator profile attempts to activate the number of links specified for the bundle when the connection is established. The point-to-point connection will remain active as long as one of the links is active. If a link goes down, the System i PPP originator profile will not attempt to bring the link active again unless re-dial on disconnect has been configured.

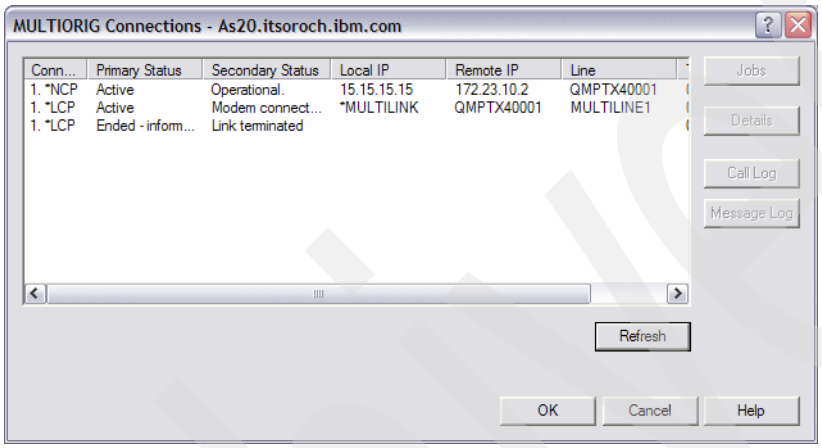


Figure 14-50 Originator Connection Profile: Multiorig connections: Link failure

6. To test the System i automatic recovery of the failed link, we connect the *failed* RJ-11 connector back into the System i or the PBX or phone switch. Eventually, the System i Originator profile will redial the remote System i and re-establish the connection on the second link. The status will look similar to Figure 14-49 on page 267 before one of the links was broken.

Review, conclusions, and references

Multilink Protocol gives provides the ability to have fault tolerance for a Point-to-Point (PPP) connection. Multiple links must be established in order to have fault tolerance protection. To force the System i Originating profile to establish a multilink connection with the maximum number of links (thereby maximizing the multilink bundle's chance at surviving many single link failures), you can specify not to use bandwidth utilization monitoring.

In addition, you should click the **Advanced** button in the Originator Connecting Profile's Connection tab for additional dial options for the list of phone numbers defined on the Connection page. One very powerful option that we have used is the automatic redial of a line that was previously disconnected due to some kind of transient error.

An additional side benefit of not using bandwidth utilization monitoring, and thus bringing multiple links active at one time, is that a sustained increase in bandwidth is achieved.

15

DHCP: Dynamic allocation of IP addresses

This chapter presents scenarios for configuring the DHCP server on System i to dynamically allocate IP addresses to clients in a local area network (LAN). Each scenario shows detailed steps for setting up the DHCP server on System i and describes how to configure the Windows 2000 client to act as a DHCP client.

This chapter contains a group of scenarios that focus on just DHCP:

- "DHCP: One physical network, one logical network, one DHCP server" on page 270
- "DHCP: One physical network, multiple logical networks, one DHCP server" on page 292
- "DHCP: One physical subnet, one logical subnet, multiple DHCP servers" on page 307
- "DHCP: multiple physical networks, logical networks, and DHCP servers" on page 322
- "DHCP: multiple physical, logical networks, and DHCP servers using Relay Agents" on page 343

This next group demonstrates the power of the System i DHCP dynamically updating your domain's DNS server:

- "Single DDNS and DHCP server on the same server" on page 368
- "Single DDNS and DHCP servers without secured updates" on page 402
- "Single DDNS and DHCP servers with secured updates" on page 438
- "Primary DDNS and DHCP servers on one server, secondary server as backup" on page 447
- "Primary DDNS and DHCP servers, secondary DNS server Red Hat Linux 7.2" on page 460

This last scenario demonstrates the System i DHCP server's ability to dynamically assign IP addresses to remote PPP clients: 17.5, "Assigning an IP address to PPP client from DHCP server" on page 610.

15.1 DHCP: One physical network, one logical network, one DHCP server

This scenario describes the configuration of the System i DHCP server in a simple TCP/IP network. This DHCP server will lease IP addresses to clients in a subnet.

Problem definition

For this scenario, a customer has the network shown in Figure 15-1. The customer uses static IP address allocation, which means that the IP addresses are hard coded in the client configuration. As the number of clients in the network increases, the allocation of the IP addresses to clients becomes very hard to manage. For example, adding a new client in the network requires first configuring the client with all of the IP options that it needs (IP address, network mask, gateway IP address, and so on).

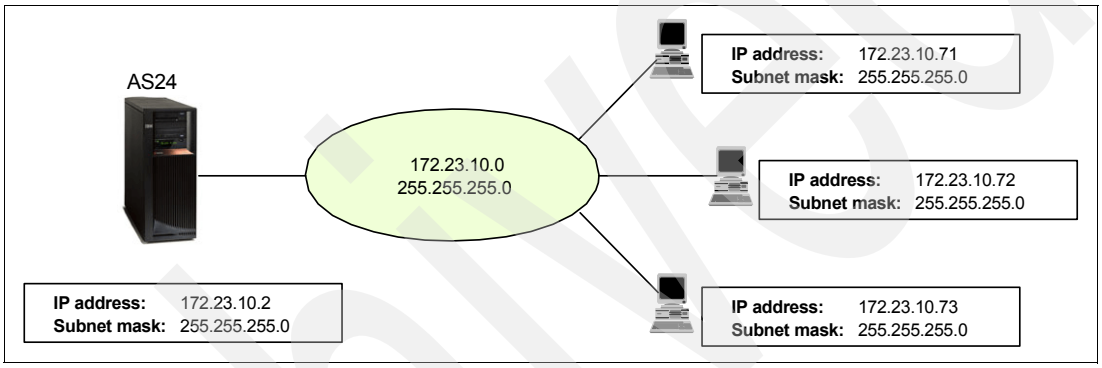


Figure 15-1 Simple network: one physical subnet, static IP allocation

Also, a modification in the IP network configuration (network address changed, network mask changed, gateway IP address changed, and so on) means the modification of each client with the new configuration (Figure 15-2). This represents a lot of hard work for you or your staff, and the opportunity for hard-to-debug errors.

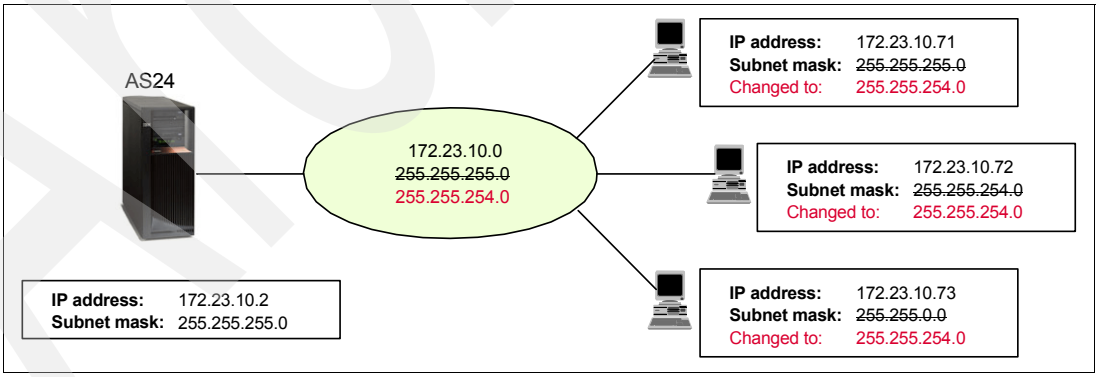


Figure 15-2 Simple Network: implementing a configuration change (network mask)

The best solution is to centralize the configuration of many of these network options at a DHCP server as shown in Figure 15-3. A change to the subnet mask (1) will automatically (dynamically.) propagate to the DHCP clients (3) via the next DHCP Offer (2).

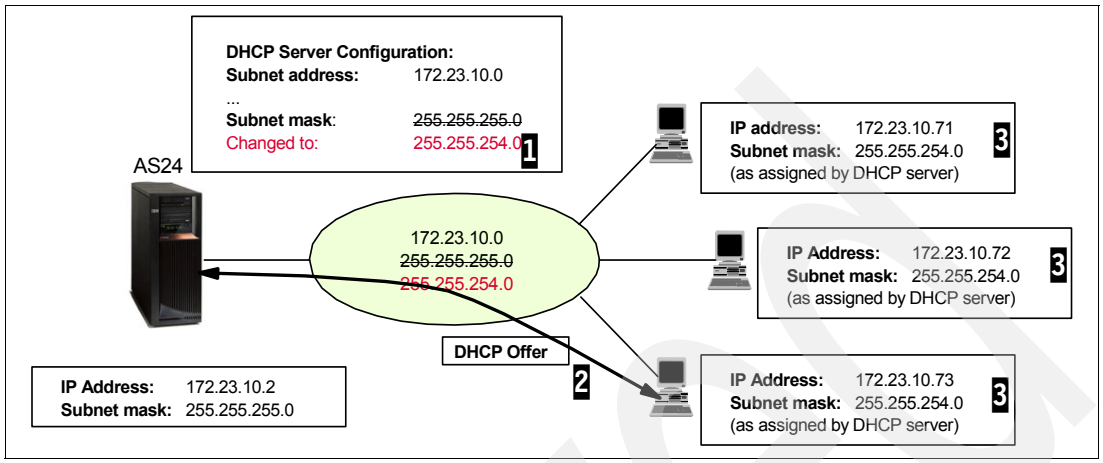


Figure 15-3 Simple network: implementing a configuration change (network mask)

Solution definition

To simplify the IP administration and monitoring of IP addresses in a medium-to-large TCP/IP network, a DHCP server should be used instead of static IP allocation. The DHCP server listens for client requests and leases IP addresses to clients when required. So, using a DHCP server, any modification in the IP configuration is performed only on the DHCP server, and then propagated in the network as clients request IP configuration options from the DHCP server (Figure 15-4). Also, adding a new client in a network with a System i DHCP server requires only physically connecting the client in the network, because all of the IP options the client needs are automatically downloaded by the client from the DHCP server.

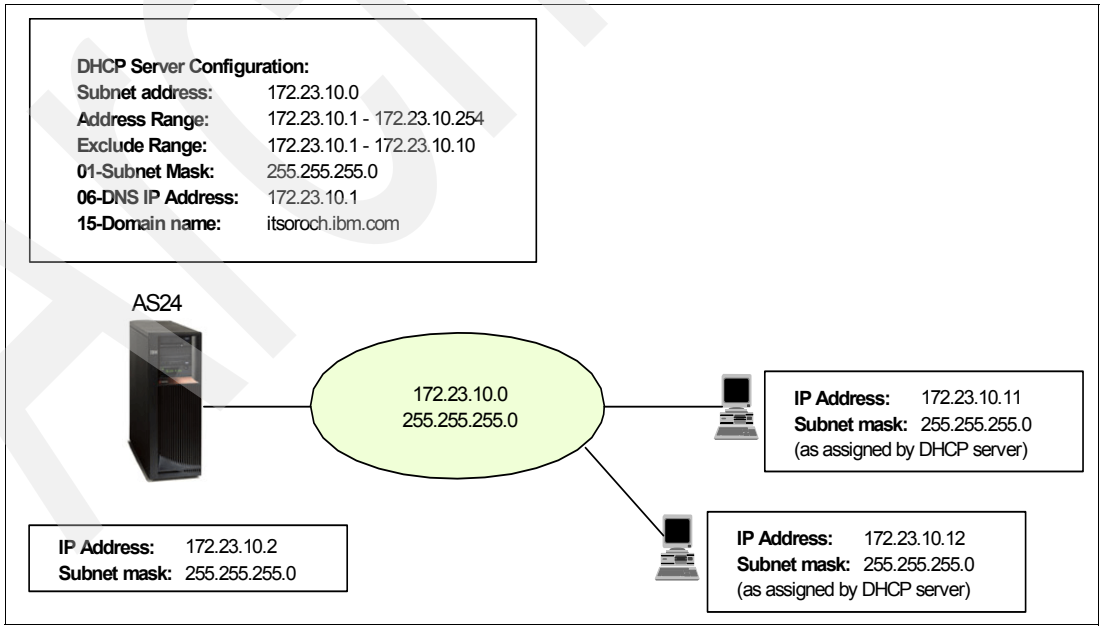


Figure 15-4 Simple DHCP network: One physical subnet with dynamic IP allocation and options

The System i DHCP server optionally provides the clients with these IP-related options:

- IP address

- Network mask
- IP address of the DNS
- Domain name

> **Note:** DHCP can provide many more options to the clients. See RFC 2132, *DHCP Options and BOOTP Vendor Extensions* for details about the DHCP options. Also, Figure 6-3 on page 104, an image from iSeries Navigator, shows some of the options that are available for your use.

Assumptions

The network used in this scenario has the following characteristics:

- There is a single physical subnet.
- The System i has only one network adapter in LAN, with only one IP interface.
- The subnet IP address is 172.23.10.0, the subnet mask is 255.255.255.0. The subnet mask enables the DHCP server to service 254 clients. The IP address range 172.23.10.1 to 172.23.10.10 is reserved for other servers, and is excluded from the addressing pool.
- There is a single System i DHCP server that allocates the IP addresses in the network.
- There are no routers or bridges in this network.

How-to

To configure the DHCP server and clients in this scenario, perform the following tasks:

- Step 1: Configure the System i AS24 network interface.
- Step 2: Plan the configuration of the System i DHCP server AS24.
- Step 3: Configure the DHCP server AS24.
- Step 4: Start the DHCP server.
- Step 5: Configure your Windows 2000 DHCP client.
- Step 6: Test the configuration.

Step 1: Configure the System i AS24 network interface

In this scenario we assume that you have not yet configured your System i with an IP address associated with a physical LAN interface. This step creates a new TCP/IP interface. To configure the TCP/IP interface, perform the following steps:

1. Sign on to the system as a user with *IOSYSCFG and *ALLOBJ special authorities.
2. On the command line, type the command:

```
CFGTCP
```

3. In the Configure TCP/IP menu, select option **1** to open the Work with TCP/IP Interfaces panel (Figure 15-5).

```
                         Work with TCP/IP Interfaces
                                                                System:   AS24
Type options, press Enter.
  1=Add   2=Change   4=Remove   5=Display   9=Start   10=End

     Internet         Subnet              Line
Opt  Address          Mask             Description  Type

     127.0.0.1        255.0.0.0        *LOOPBACK    *NONE

                                                                          Bottom
F3=Exit      F5=Refresh   F6=Print list   F11=Display interface status
F12=Cancel   F17=Top      F18=Bottom
```

Figure 15-5 Work with TCP/IP Interfaces display

4. Select option **1** (Add) to add a TCP/IP interface and press Enter to continue.
5. Specify the Internet address, line description, and the subnet mask (Figure 15-6).

```
                      Add TCP/IP Interface (ADDTCPIFC)

Type choices, press Enter.

Internet address . . . . . > '172.23.10.2'
Line description . . . . . .   ETHLIN1       Name, *LOOPBACK...
Subnet mask  . . . . . .   '255.255.255.0'
Associated local interface . . .   *NONE
Type of service  . . . . . .   *NORMAL       *MINDELAY, *MAXTHRPUT...
Maximum transmission unit  . . .   *LIND         576-16388, *LIND
Autostart  . . . . . .   *YES          *YES, *NO
PVC logical channel identifier                 001-FFF
            + for more values
X.25 idle circuit timeout  . . .   60           1-600
X.25 maximum virtual circuits  .   64           0-64
X.25 DDN interface . . . . . .   *NO           *YES, *NO
TRLAN bit sequencing . . . . . .   *MSB          *MSB, *LSB

                                                                        Bottom
```

Figure 15-6 Add TCP/IP Interface display

6. Press Enter to create the IP interface.

7. Press F11 to view the status of the interface and verify that the status is active (Figure 15-7).

```
                        Work with TCP/IP Interfaces
                                                                System:   AS24
Type options, press Enter.
  1=Add   2=Change   4=Remove   5=Display   9=Start   10=End

     Internet         Subnet           Interface
Opt  Address          Mask             Status

     127.0.0.1        255.0.0.0        Active
     172.23.10.2      255.255.255.0    Active

                                                                        Bottom
F3=Exit      F5=Refresh   F6=Print list   F11=Display line information
F12=Cancel   F17=Top      F18=Bottom
```

Figure 15-7 Work with TCP/IP Interfaces display: The new created interface

Step 2: Plan the configuration of the System i DHCP server AS24

To configure the DHCP server through iSeries Navigator, first we respond to a series of questions about the configuration of the network in which we implement the DHCP server, such as what address range or ranges we use for leasing and which addresses are reserved for special hosts (routers, DNS servers, mail servers, and so on). All of these questions are included in Table 15-1, Table 15-2, and Table 15-3 on page 275. The answers are based on the network configuration in Figure 15-4 on page 271.

Table 15-1 contains information about the System i IP configuration.

Table 15-1 Planning the DHCP server AS24: AS24 TCP/IP information

Configuration parameter	Value
Host name	AS24
Description	DHCP server
IP address	172.23.10.2
Network mask	255.255.255.0
Line description	ETHLIN1
Domain name	itsoroch.ibm.com

Table 15-2 contains information about the DHCP server on the System i. The third column in this table indicates the place in iSeries Navigator where you can configure the specified parameter. Many of these configuration options can be specified through the DHCP configuration wizard the first time you configure the DHCP server.

Table 15-2 Planning the DHCP server AS24: DHCP server overview

#	Question	Answer	Configuration reference
1	Start the DHCP server when TCP/IP starts?	Yes	DHCP Server → Properties → General → Start when TCP/IP is started
2	Is the BOOTP server already configured on the system?	No	DHCP configuration wizard

#	Question	Answer	Configuration reference
3	Do you want to migrate the BOOTP configuration to DHCP?	N/A	File → Migrate BOOTP
4	What is the default lease time for this server?	1 day	Global → Properties → Leases → Duration
5	Do you want the DHCP server to support BOOTP clients?	No	Global → Properties → Client Support → Support BOOTP clients
6	Do you want the DHCP server to support any client from any subnet?	Yes	Global → Properties → Client Support → Support unlisted clients → DHCP clients
7	Do you want to log the DHCP server activity?	Yes	DHCP server → Properties → Logging → Enable logging
8	Can your DHCP clients (other than Network Stations) identify the class they belong to?	No	
9	If the answer to 8 is Yes, do you want to add a new class to serve the DHCP clients that belong to that class?	N/A	Global → New Class
10	Which are the IP interfaces the server listens on?	172.23.10.2	See Figure 15-4 on page 271
11	Which subnet will be administered by the DHCP server?	172.23.10.0	See Figure 15-4 on page 271
12	Do you want to add a new subnet to be administered by the DHCP server?	Yes	Global → New Subnet - Advanced

In order to allocate IP addresses to the clients in the subnet, a new subnet must be defined on the DHCP server. The properties of this subnet and the place in iSeries Navigator where these properties can be set are presented in Table 15-3.

Table 15-3 Planning the DHCP server AS24: properties for subnet 172.23.10.0

Property	Value	Configuration reference
Subnet name	172.23.10.0	Subnet Properties → General
Subnet description	ITSO subnet 1	Subnet Properties → General
Subnet address	172.23.10.0	Subnet Properties → Address Pool → Subnet address
Subnet mask	255.255.255.0	Subnet Properties → Address Pool → Subnet mask
Address range for leasing	172.23.10.1 - 172.23.10.254	Subnet Properties → Address Pool → Range to assign
Lease time	Inherit from server (1 day)	Subnet Properties → Leases → Inherit lease time
IP addresses excluded from the address pool	Range 172.23.10.1 - 172.23.10.10	Subnet Properties → Address Pool → IP addresses excluded

Property	Value	Configuration reference
Options offered to DHCP clients 01 - Subnet mask 06 - DNS IP address 15 - Domain name	 255.255.255.0 172.23.10.1 itsoroch.ibm.com	Subnet Properties → Options

Step 3: Configure the DHCP server AS24

The first time you configure the DHCP configuration, iSeries Navigator launches the DHCP configuration wizard. Using this wizard, you can specify the basic options for the DHCP server. If a configuration already exists, the configuration wizard will not be launched.

Tip: To reset an existing DHCP configuration, perform the following steps:

1. From the DHCP Server Configuration window, select **File** → **New Configuration**. This pops up an informational message window asking whether you are sure that you want to create a new configuration.
2. Click the **OK** button. This starts the New DHCP Configuration wizard.
3. After performing all of the steps in the configuration wizard, a new DHCP configuration is created.

Another way to do that is to delete the file dhcpsd.cfg in /QIBM/UserData/OS400/DHCP and run the program QSYSDIR/QTODDINS. This program creates an empty file dhcpsd.cfg in /QIBM/UserData/OS400/DHCP, which can be edited by using the DHCP configuration from iSeries Navigator.

To configure the DHCP server, perform these steps using the values in Table 15-2 on page 274:

1. Start iSeries Navigator.
2. Expand your System i server. You may be asked to enter your user ID and password.
3. Expand **Network** → **Servers** and click **TCP/IP**.

4. Double-click **DHCP** to start the DHCP server configuration wizard (Figure 15-8). Click **Next** to continue.

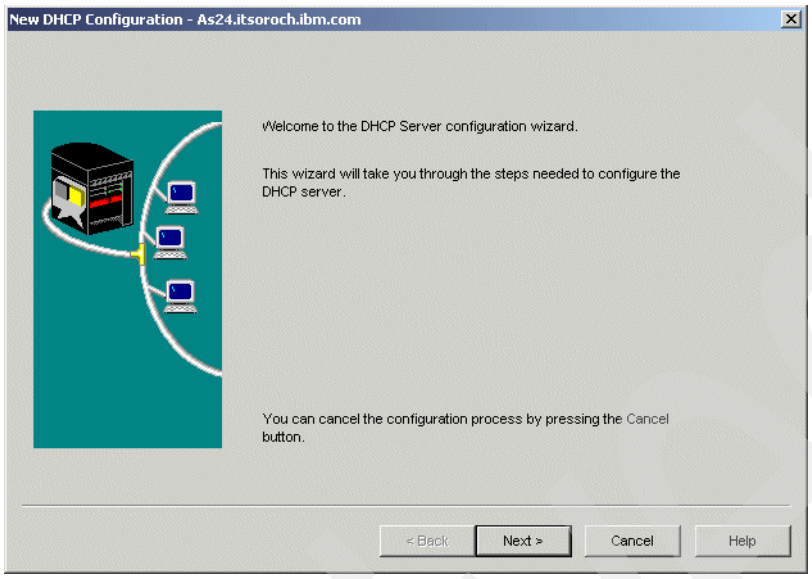


Figure 15-8 The DHCP Configuration wizard

5. Because we do not want the DHCP server to update the DNS with client information, select **No** in the Dynamic DNS window (Figure 15-9). See 16.2, “Single DDNS and DHCP servers without secured updates” on page 402 for an example that demonstrates using your System i DHCP server to automatically update the Dynamic DNS server with IP address and host name information. Click **Next**.

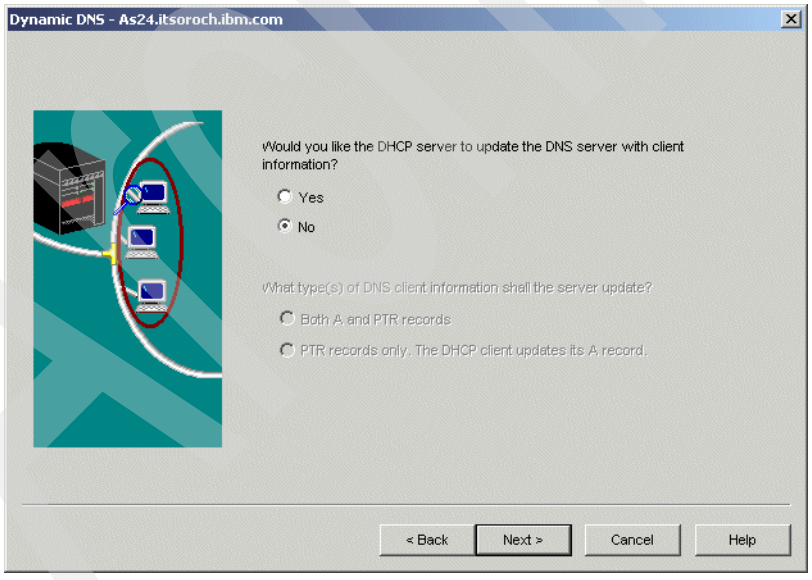


Figure 15-9 DHCP Configuration wizard: Dynamic DNS

6. In the Default Lease Time window, specify the default lease time for the IP addresses that DHCP allocates to clients (Figure 15-10). By default, lease time is 24 hours. Click **Next**. To learn more about the lease time, refer to 6.3.2, “Lease renewal” on page 109.

> **Note:** To see how the lease renewal works while you are testing, you can specify a smaller value for lease time (one to two minutes) and take a communication trace. Be sure to change this value back to a longer lease time when you are done.

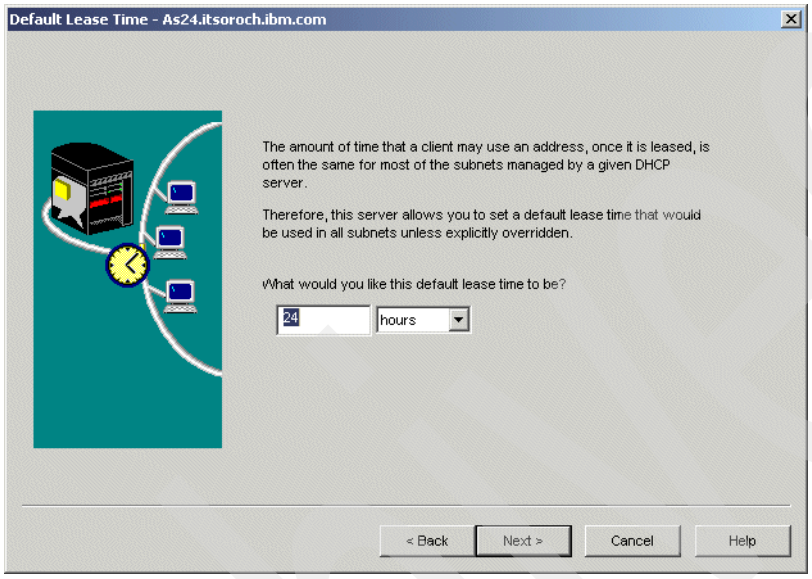


Figure 15-10 DHCP Configuration wizard: Default Lease Time

7. In the Default Domain Name Server window, specify the IP address of the DNS that is provided by default by the DHCP server to clients as a DHCP option (Figure 15-11). Click **Next**.

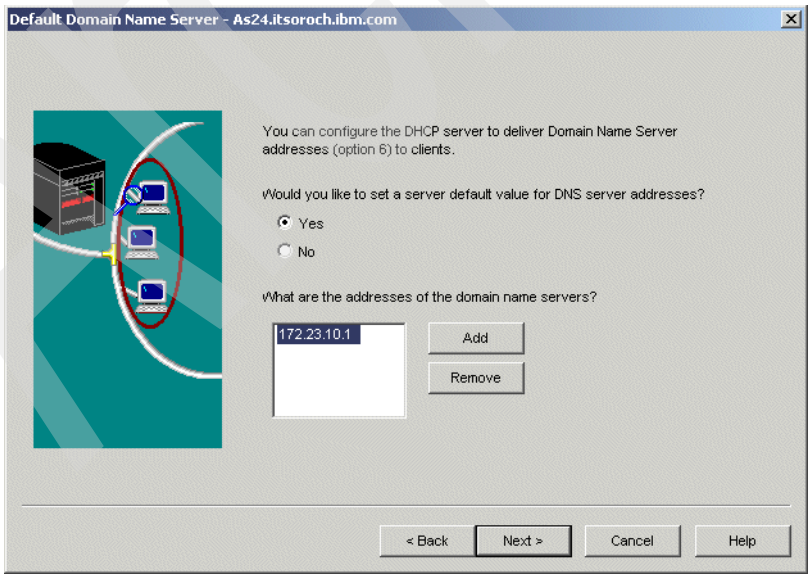


Figure 15-11 DHCP Configuration wizard: Default Domain Name Server

8. In the Default Domain Name window, specify the domain name that will be provided by default by the DHCP server to clients as a DHCP option (Figure 15-12). Click **Next**.

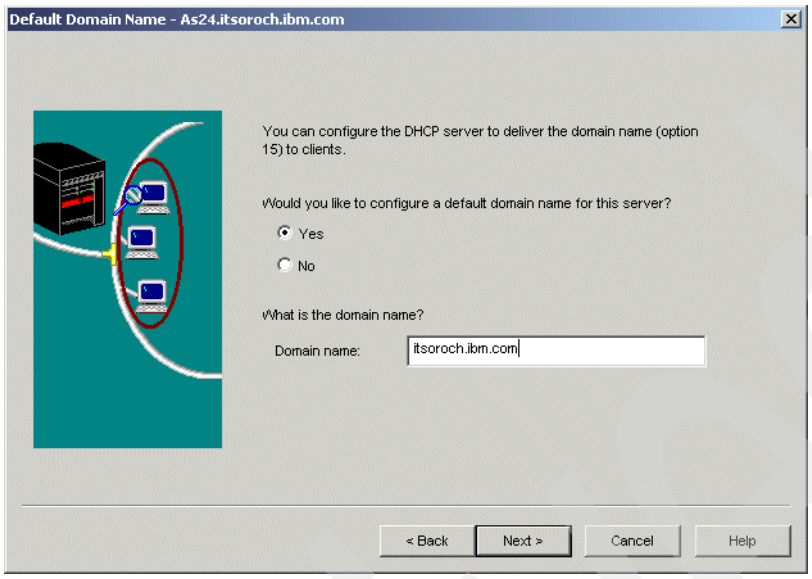


Figure 15-12 DHCP Configuration wizard: Default Domain Name

9. In the Start DHCP window, select **Yes** to the question Do you want the DHCP server to start when TCP/IP starts? and **No** to the question Do you want the DHCP server started now? (Figure 15-13). In general, it is a good idea to not start the DHCP server right away because you must still configure at least one DHCP subnet. Click **Next**.

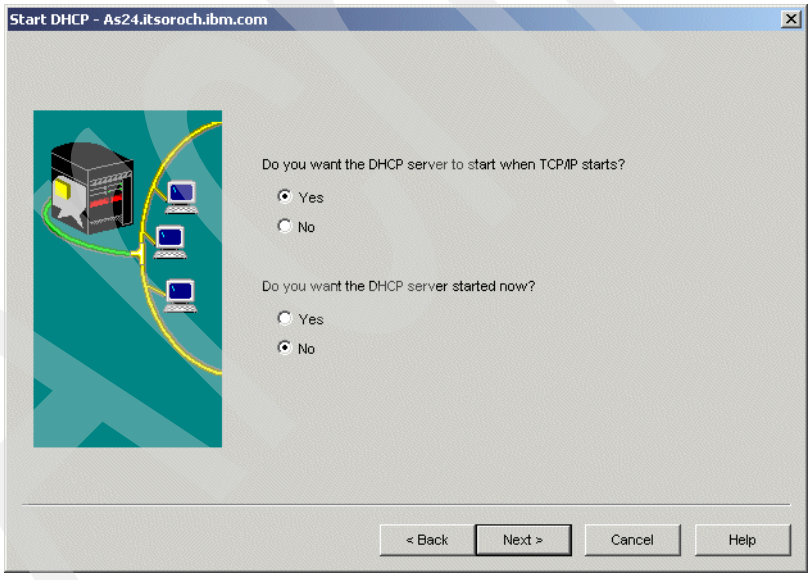


Figure 15-13 DHCP Configuration wizard: Start DHCP

10. The DHCP server configuration is displayed (Figure 15-14). Click **Finish** to create the new configuration.

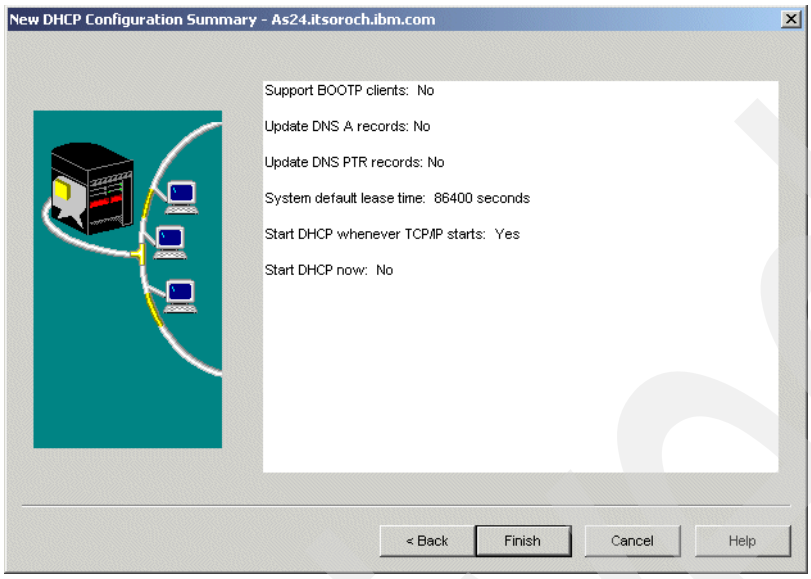


Figure 15-14 DHCP Configuration wizard: New DHCP Configuration Summary

11. After the new DHCP configuration is created, a message is displayed asking whether to create a new subnet (Figure 15-15). We must add the subnet 172.23.10.0 to the DHCP configuration using the values in Table 15-3 on page 275, so click **Yes** to start the configuration of a new subnet.

Figure 15-15 Create a new subnet

12. The New DHCP Subnet wizard is started (Figure 15-16). Click **Next**.

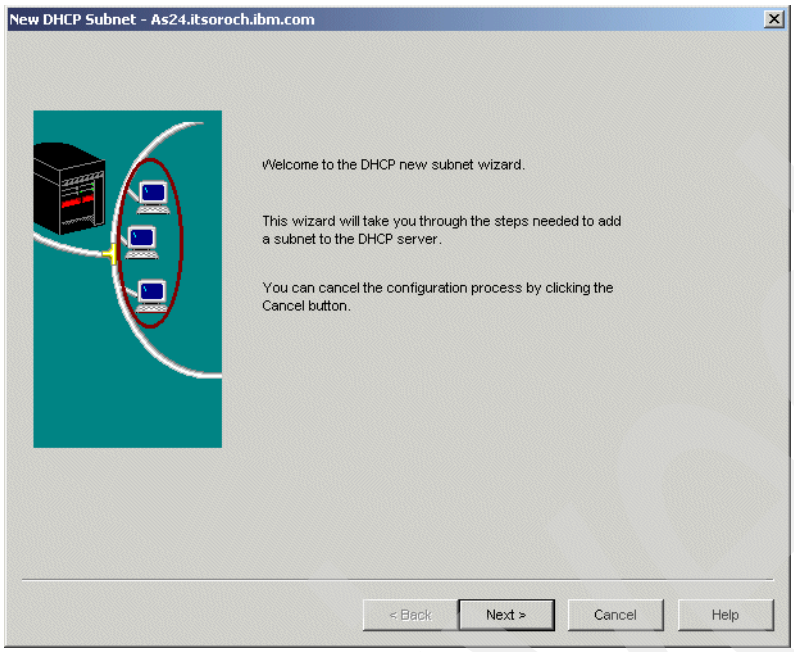


Figure 15-16 New DHCP Subnet wizard

13. In the Subnet Manages Twinaxial Devices window, select **No** (Figure 15-17). Click **Next**.

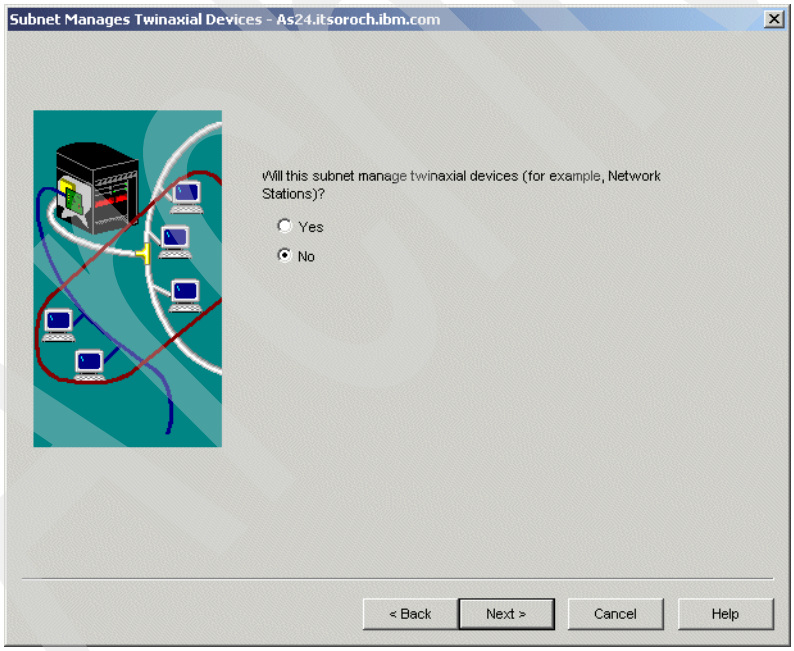


Figure 15-17 New DHCP Subnet wizard: Subnet Manages Twinaxial Devices

14. In the Address Range or Subnet window, select **Define subnet based on an address range within a subnet** (Figure 15-18). Click **Next**.

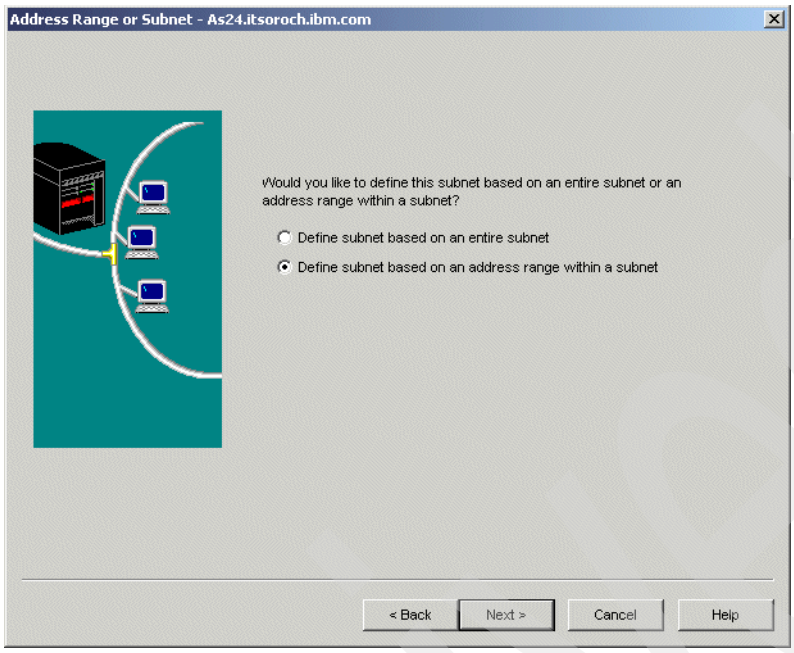


Figure 15-18 New DHCP Subnet wizard: Address Range or Subnet

15. In the Define Subnet Based on an Address Range window, specify the name and the description of the subnet being created, and define the range of addresses to use within the subnet (Figure 15-19). The addresses within this range will be allocated to the DHCP clients for this subnet. Click **Next**.

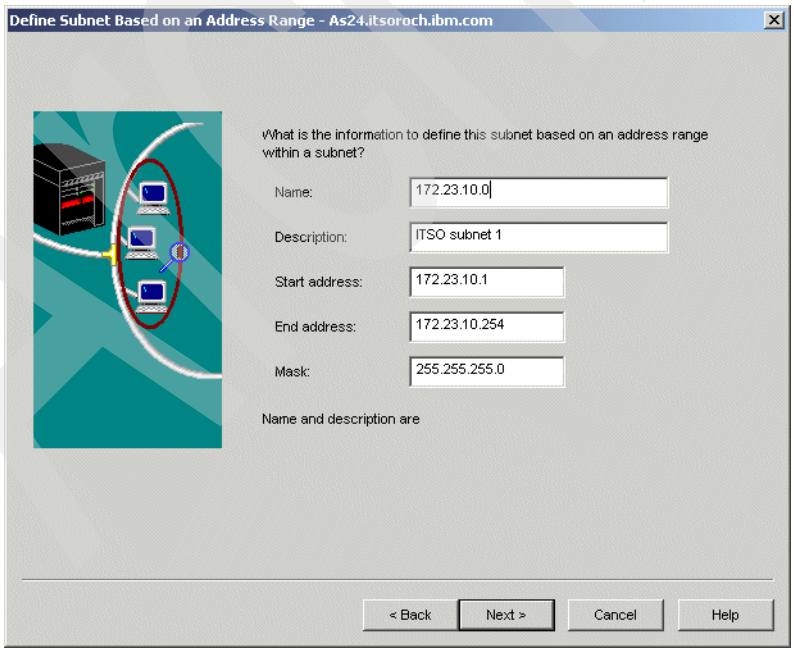


Figure 15-19 New DHCP Subnet wizard: Define Subnet Based on an Address Range

16. In the Exclude Addresses window, specify the IP address range that is excluded from the IP address pool. The DHCP server does not deliver these addresses to clients

(Figure 15-20). In our scenario, these are the IP addresses that we are reserving for servers in this subnet range (172.23.10.1 through 172.23.10.254). Click **Add** to place the IP address range into the display list and then click **Next**.

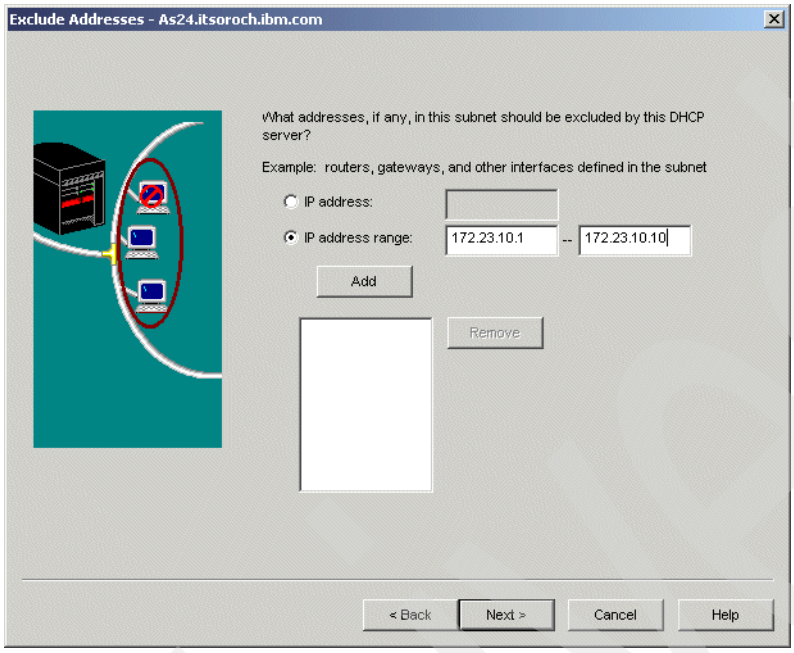


Figure 15-20 New DHCP Subnet wizard: Exclude Addresses

17. In the Subnet Domain Name Server window, select **Use the global value** (Figure 15-21). Click **Next**.

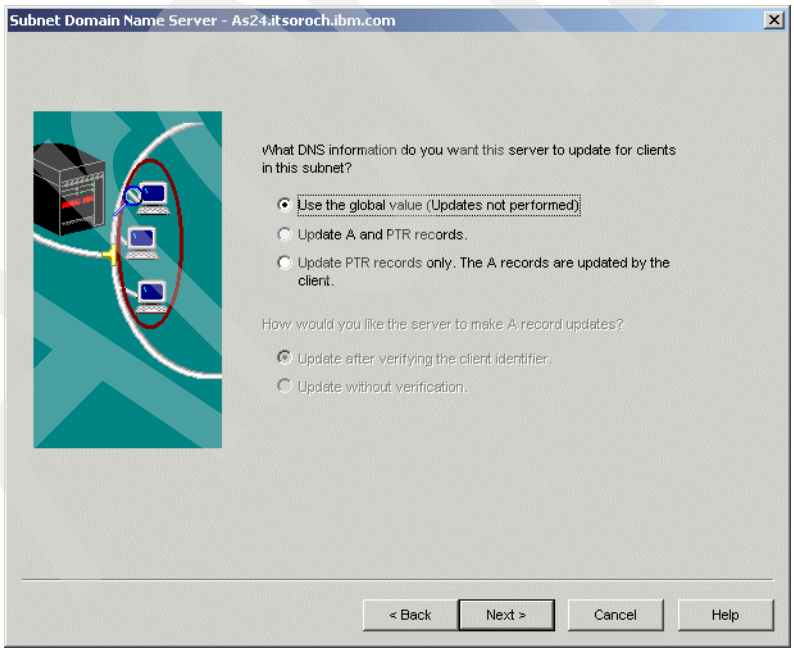


Figure 15-21 New DHCP Subnet wizard: Subnet Domain Name Server

18. In the Subnet Lease Time window, select **Inherit the server's default lease time** (Figure 15-22). Click **Next**.

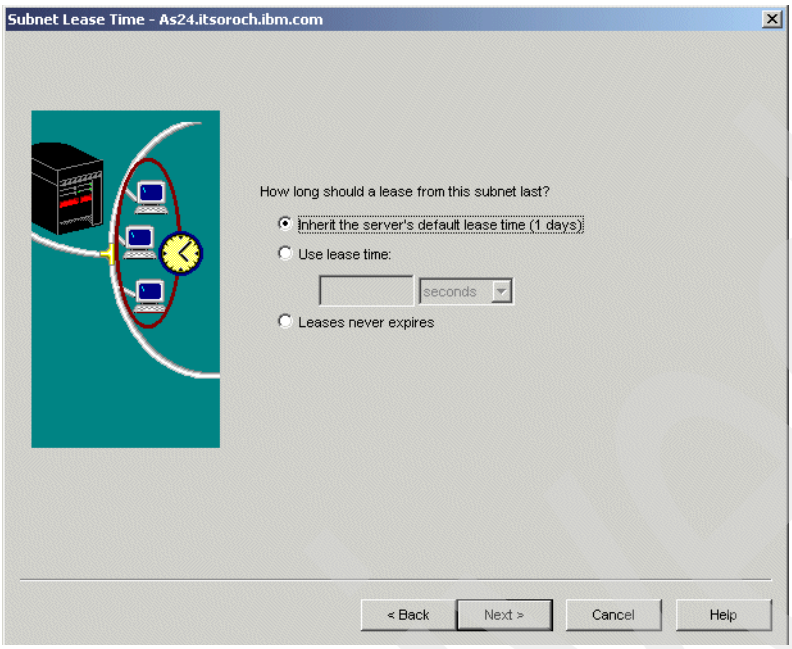


Figure 15-22 New DHCP Subnet wizard: Subnet Lease Time

19. Because we have a single subnet, there is no need to provide the clients with a default gateway. In the Subnet Gateways window, select **No** (Figure 15-23). Click **Next**.

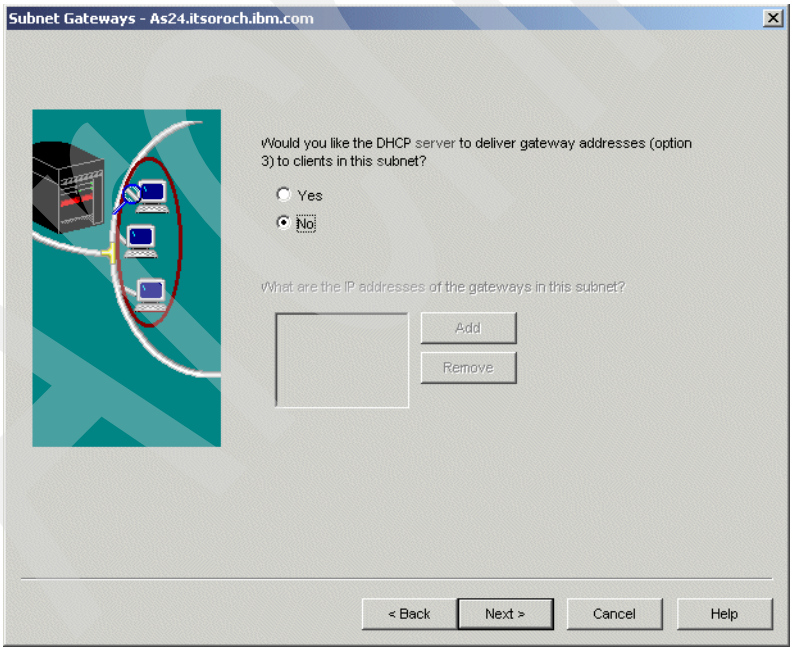


Figure 15-23 New DHCP Subnet wizard: Subnet Gateways

20. In the Subnet Domain Name Server window, select **Use the global value** (Figure 15-24). The DHCP server will send to the DHCP clients the IP address of the DNS specified at the global level. Click **Next**.

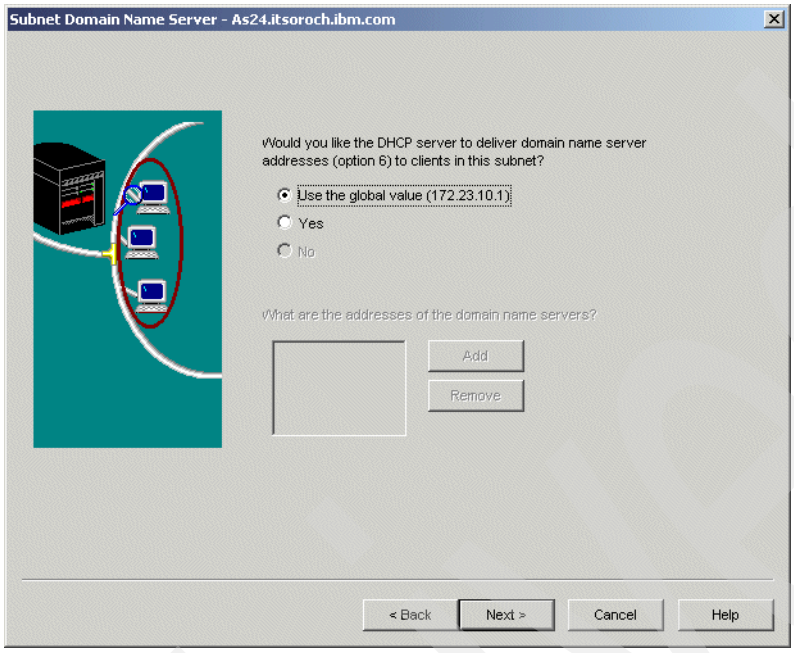


Figure 15-24 New DHCP Subnet wizard: Subnet Domain Name Server

21. In the Subnet Domain Name window, select the **Use the global value** (Figure 15-25). The DHCP server will send to the DHCP clients the domain name specified at the global level. Click **Next**.

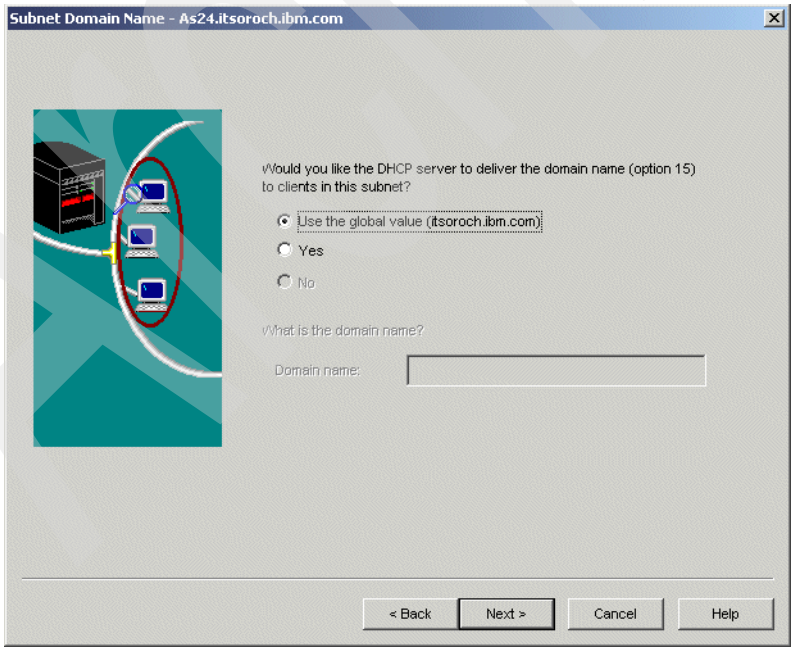


Figure 15-25 New DHCP Subnet wizard: Subnet Domain Name

22. In the More Subnet Options window, select **No** (Figure 15-26).

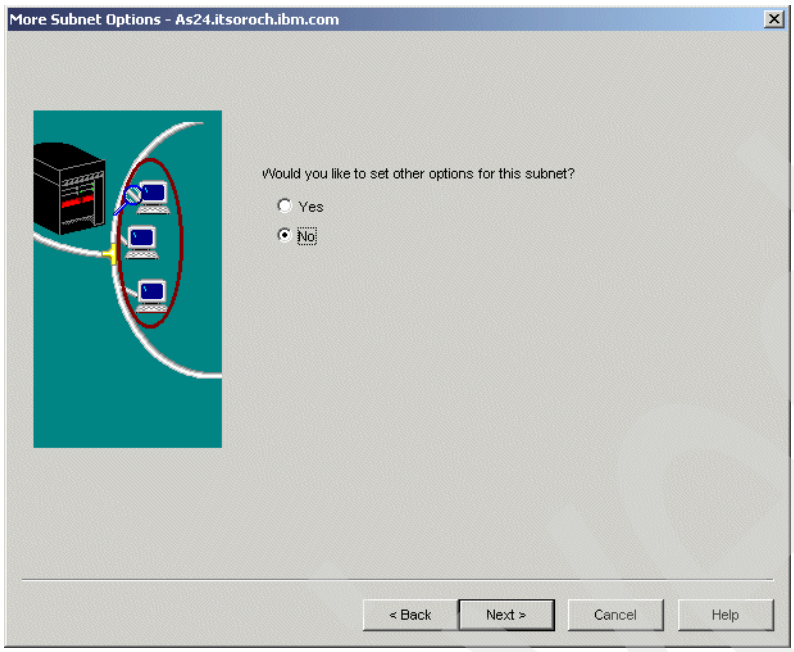


Figure 15-26 New DHCP Subnet wizard: More Subnet Options

23. Now the configuration properties for the new DHCP subnet are displayed (Figure 15-27). Click **Finish** to create the subnet.

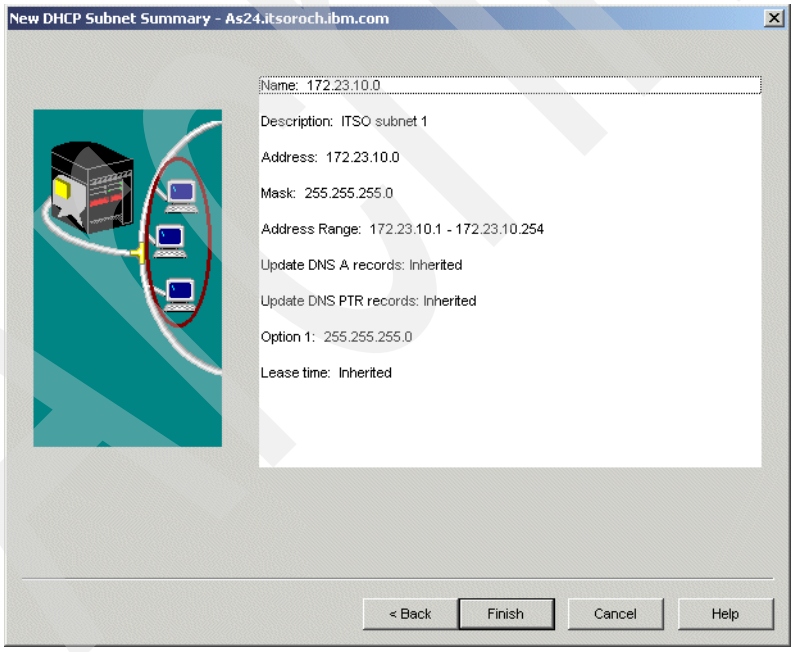


Figure 15-27 New DHCP Subnet wizard: New DHCP Subnet Summary

The configuration of a simple network to use DHCP is complete.You have created one subnet from a class B IP address using the subnet mask 255.255.255.0. The subnet 172.23.10.0 was added to the DHCP configuration. As you can see in Figure 15-28, when you select the subnet 172.23.10.0, the status bar displays the basic subnet properties. In addition, the bottom-right pane displays the DHCP options that are sent to the clients who request an IP address from this subnet.

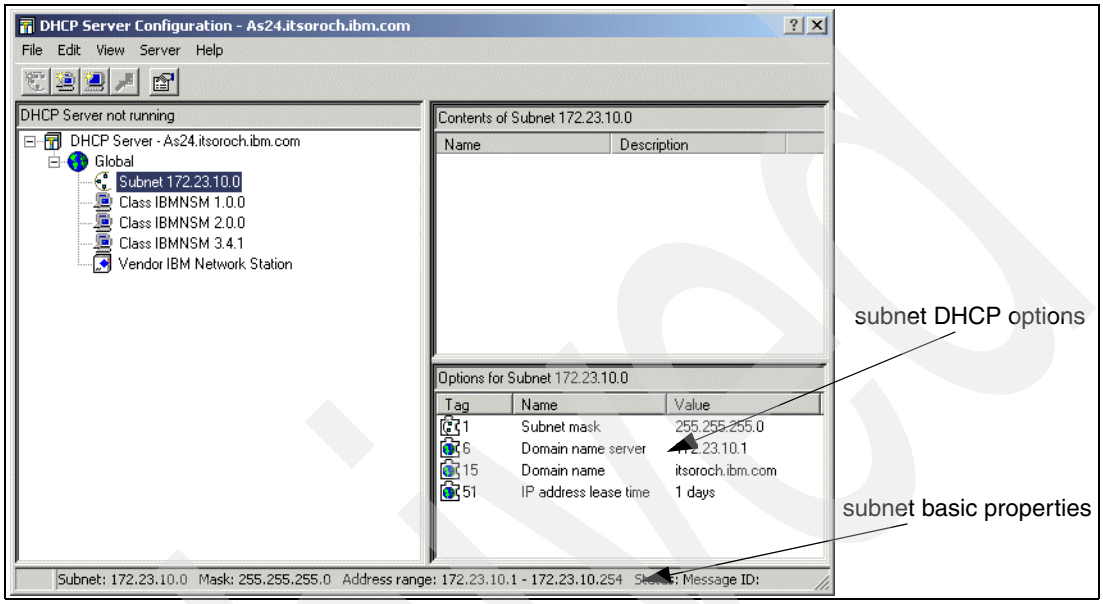


Figure 15-28 DHCP Server Configuration

Although only option 1 (Subnet mask) is specified in the subnet configuration, the subnet 172.23.10.0 inherits option 6 (Domain name server), option 15 (Domain name), and option 51 (IP address lease time), which were specified at the Global level, when the new DHCP configuration was created.

This ends the DHCP server configuration.

Step 4: Start the DHCP server

To start the DHCP server, perform the following steps:

1. From iSeries Navigator, expand your System i. You may be asked to enter your user ID and password.
2. Expand **Network** → **Servers** and then click **TCP/IP**.

3. Right-click **DHCP** and select **Start** (Figure 15-29).

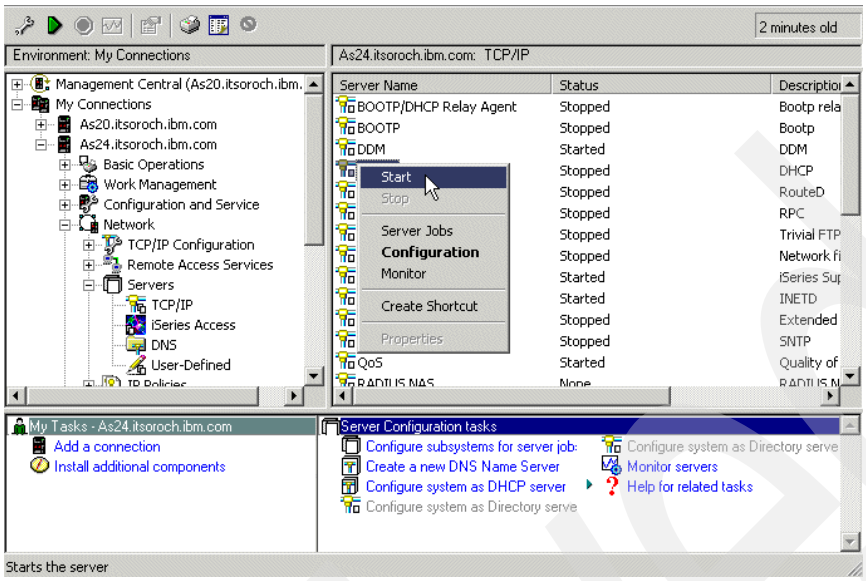


Figure 15-29 iSeries Navigator: Start DHCP server

Step 5: Configure your Windows 2000 DHCP client

To use the DHCP server, clients must support DHCP and be appropriately configured. Many DHCP clients are available on the market, but for the tests in this book we used the Windows 2000 Professional Edition. Refer to your DHCP client documentation for information about your client's DHCP support.

To enable DHCP on your Windows 2000 Professional Edition workstation:

1. Click **Start** → **Settings** → **Control Panel.**
2. In Control Panel, double-click **Network and Dial-up Connections.**

3. Right-click the network interface you want to configure, and select **Properties** from the context menu. A window similar to Figure 15-30 is displayed. Select **Internet Protocol (TCP/IP)** and click **Properties**.

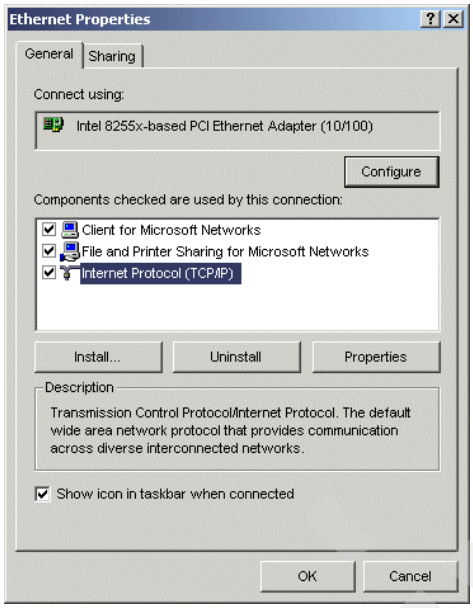


Figure 15-30 DHCP client configuration: network adapter properties

4. In the Internet Protocol (TCP/IP) Properties window (Figure 15-31), select **Obtain an IP address automatically** and **Obtain DNS server address automatically.** Click **Advanced**.

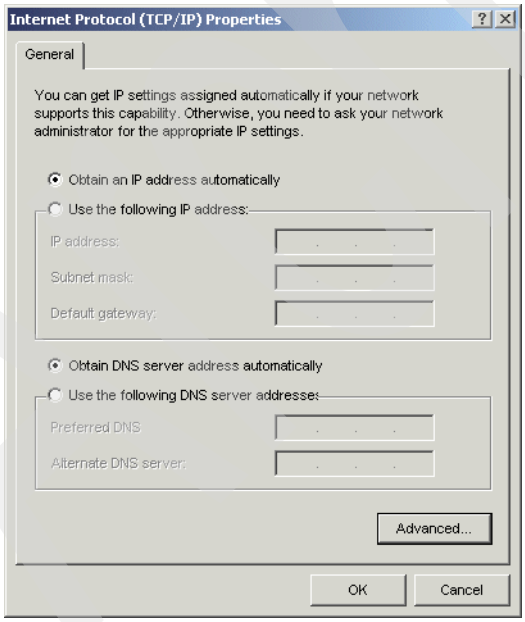


Figure 15-31 DHCP client configuration: Internet Protocol (TCP/IP) Properties

5. The Advanced TCP/IP Settings window opens (Figure 15-32). Select the **DNS** tab. Select **Append primary and connection specific DNS suffixes**. Uncheck **Register this connection's addresses in DNS**.

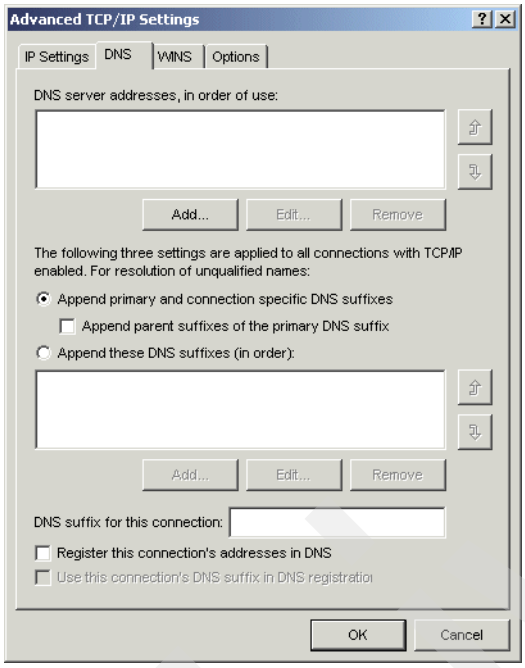


Figure 15-32 DHCP client configuration: Advanced TCP/IP Settings (DNS tab)

> **Tip:** The configuration option Register this connection's addresses in DNS is the way the Windows 2000 client indicates whether it (or something) is going to update the A record (which is the mapping from name to IP address) in the DNS. The host client, by convention, owns the responsibility of updating the A record in the DNS.
>
> But it is very common for the DHCP server to take on both the update of the A and PTR records. The configuration here should match the System i DHCP server configuration. See Figure 15-27 on page 286 to see that the System i configuration for this scenario has the DHCP server updating both the A and PTR records.

6. Click **OK** in the Advanced TCP/IP Settings window. Click **OK** in the Internet Protocol (TCP/IP) Properties window. Click **OK** in the Network Adapter Properties window.

This ends the DHCP client configuration.

Step 6: Test the configuration

In this scenario, we use one DHCP client located in the subnetwork 172.23.10.0. To test the configuration, perform the following steps:

1. Power on the DHCP-enabled Windows 2000 workstation. During the boot of the PC, the DHCP client will request an IP address from the System i DHCP server.

Note: If your DHCP-enabled Windows 2000 workstation is already powered up, you can request an IP address by using the following command:

```
ipconfig /renew
```

To release an IP address that was dynamically obtained by a Windows 2000 client, use the following command:

```
ipconfig /release
```

2. After the workstation starts, open a command prompt panel.
3. Run the following command:

```
C:\> ipconfig /all
```

4. The IP configuration of the client is displayed on the command prompt panel. The configuration in Figure 15-33 shows that the client obtained the IP address from the System i DHCP server, along with the other DHCP options the server provides such as subnet mask, DNS IP address, and domain name.

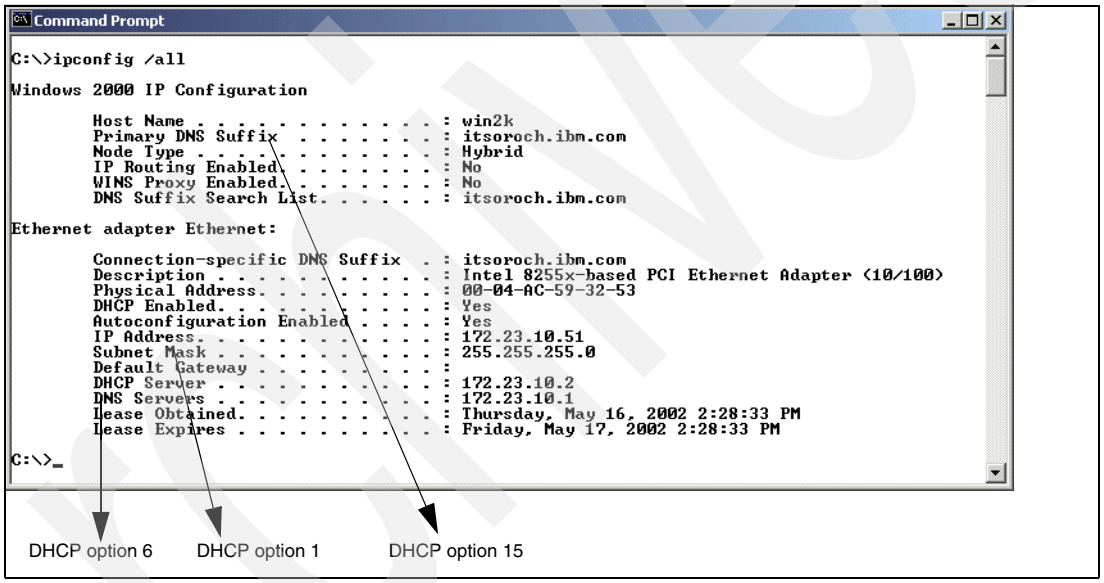


Figure 15-33 Client IP configuration after IP address allocation

On the System i side, perform the following:

1. Start iSeries Navigator.
2. Expand your System i → **Network** → **Servers** and click **TCP/IP**.
3. Right-click the **DHCP**. Select **Monitor** from the context menu. The DHCP Monitor window is displayed.

4. Expand **Status** and select **Leased**. The right pane shows the address that was allocated from the address pool and the client that obtained it (Figure 15-34).

This ends the DHCP configuration test.

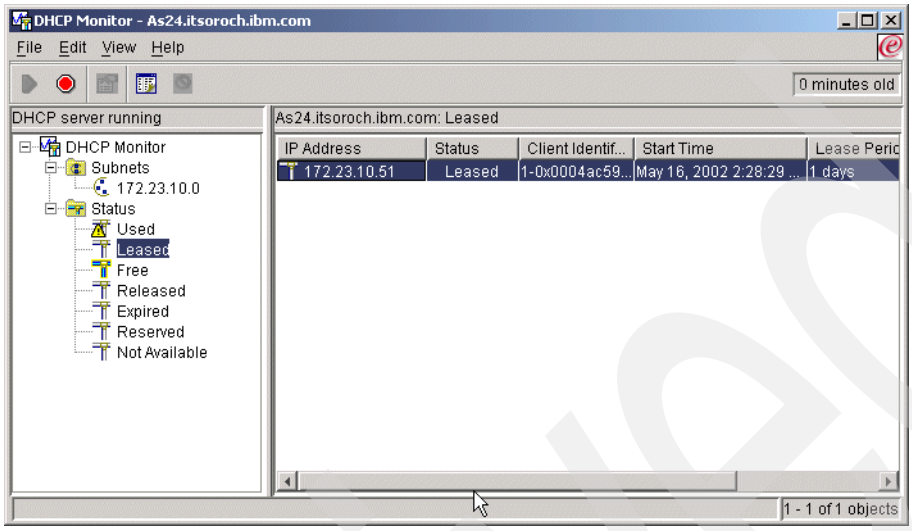


Figure 15-34 DHCP Monitor: Leased Addresses

Review, conclusions, and references

This scenario demonstrated how to get started with the DHCP in a simple network. We configured one DHCP server in a network with one physical subnet and one logical subnet.

First, we presented the reasons for using a DHCP server instead of static IP address allocation. Then, we helped you understand your network features and set up a DHCP configuration. Table 15-2 on page 274, and Table 15-3 on page 275 helped gather information about your network.

Next, you learned how to configure the DHCP server on System i by using iSeries Navigator DHCP configuration wizard, which took you through a series of steps. This chapter also explained how to configure the DHCP client on Windows 2000 Professional Edition.

Finally, we presented you how to test the DHCP configuration.

15.2 DHCP: One physical network, multiple logical networks, one DHCP server

In this scenario we demonstrate how to configure the DHCP server in a TCP/IP network with one physical subnet and multiple logical subnets. Also, we install two DHCP clients that request IP addresses from the DHCP server.

Problem definition

This scenario is based on the scenario in 15.1, “DHCP: One physical network, one logical network, one DHCP server” on page 270, in which the System i acts as a DHCP server in a simple TCP/IP network. In this scenario, the customer wants to expand his network by adding two new logical subnets. As you can see in Figure 15-35, the three logical subnets 172.23.10.0, 172.23.11.0, and 172.23.12.0 are located in the same physical network.

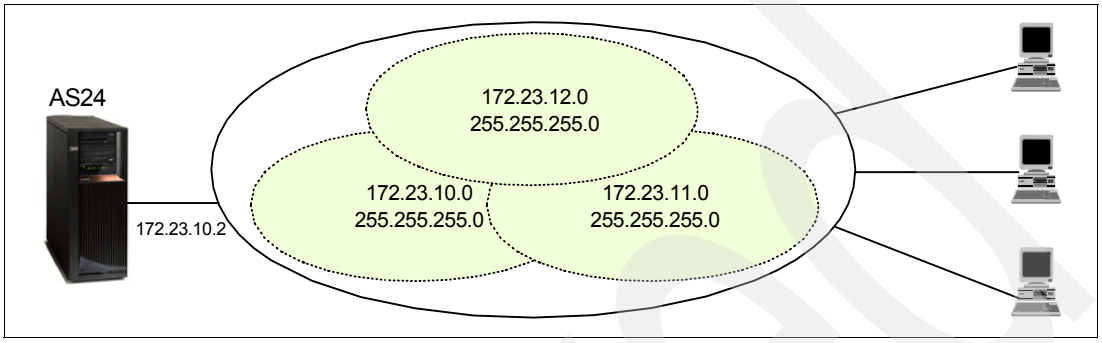


Figure 15-35 One physical subnet, multiple logical subnets, one DHCP server

The existing DHCP server already services IP addresses in subnet 172.23.10.0, so it must be configured to service the subnets 172.23.11.0 and 172.23.12.0 as well.

The DHCP server uses the interface the packet is received on to know which subnet pool to get an address from. The System i in our example has only one IP interface, 172.23.10.2. That means that it can offer IP addresses only from subnet 172.23.10.0, not from subnets 172.23.11.0 and 172.23.12.0.

Solution definition

To solve the problem, a subnet group must be defined on the DHCP server. A subnet group joins two or more subnets that are logically separated but physically located on the same wire. By creating the subnet group, the DHCP server will be able to service IP addresses that are on a different logical IP subnet (in our example, IP addresses from subnets 172.23.11.0 and 172.23.12.0).

The subnet group will include subnet 172.23.10.0, which is already defined on the DHCP server, and the subnets 172.23.11.0 and 172.23.12.0, which are created during this scenario.

When you create an address group in the DHCP server configuration, you must decide on the way the IP addresses from the subnet group should be assigned to the DHCP clients. There are two possibilities:

In order The DHCP server exhausts the subnet address pool with the highest priority within that group before it uses the subnet address pool with the next highest priority for assigning addresses.

Balanced Specifies that you want the DHCP server to assign addresses from the address pools of all subnets on an equal basis. The DHCP server assigns the first IP address from the subnet that is first in the list. Then it uses the next subnet in the list to assign the next IP address, and so on. The DHCP server repeats this cycle until addresses are used equally from all subnets.

In this scenario, we configure the subnet to use the balanced address assignment method.

Assumptions

The network used in this scenario has the following characteristics:

- There is a single physical subnet.
- The System i has only one network adapter in LAN, with only one IP interface (172.23.10.2).
- There are three logical subnets: 172.23.10.0, 172.23.11.0, and 172.23.12.0. In all three subnets, the addresses from 1 to 10 are reserved for other servers and will be excluded from the addressing pool.
- There is a single System i DHCP server (AS24) that allocates the IP addresses from all the logical subnets.
- There are no routers or bridges in this network.

How-to

Because this scenario is based on the configuration in scenario number one (15.1, "DHCP: One physical network, one logical network, one DHCP server" on page 270). In this section, we plan and perform only the modifications that occur in the DHCP configuration.

To configure the DHCP server AS24 and clients in this scenario, perform the following steps:

- Step 1: Plan the new subnets and the subnet group configuration.
- Step 2: Configure the new subnet on the DHCP server.
- Step 3: Configure the subnet group on the DHCP server.
- Step 4: Configure the Windows 2000 DHCP clients.
- Step 5: Start the DHCP server.
- Step 6: Test the configuration.

Step 1: Plan the new subnets and the subnet group configuration

To configure the new subnet and the subnet group with iSeries Navigator, first respond to a series of questions about the new subnets and the subnet group. All of these questions are included in Table 15-4, Table 15-5 on page 295, and Table 15-6 on page 295. The answers are based on the network configuration in Figure 15-35 on page 293.

In order to allocate IP addresses to DHCP clients from subnets 172.23.11.0 and 172.23.12.0, two new subnets must be defined on the DHCP server.

Table 15-4 shows the properties of subnet 172.23.11.0 and the place in iSeries Navigator where these properties can be set.

Table 15-4 Planning the DHCP server AS24: properties for subnet 172.23.11.0

Property	Value	Configuration reference
Subnet name	172.23.11.0	Subnet Properties → General
Subnet description	ITSO subnet 2	Subnet Properties → General
Subnet address	172.23.11.0	Subnet Properties → Address Pool → Subnet address
Subnet mask	255.255.255.0	Subnet Properties → Address Pool → Subnet
Address range for leasing	172.23.11.1 to 172.23.11.254	Subnet Properties → Address Pool → Range to assign
Lease time	Inherit from server (1 day)	Subnet properties → Leases → Inherit lease time

Property	Value	Configuration reference
IP addresses excluded from the pool	172.23.11.1 to 172.23.11.10	Subnet Properties → Address Pool → IP addresses excluded
Options offered to DHCP clients 01 - Subnet mask 02 - DNS IP address 15 - Domain name	 255.255.255.0 172.23.11.1 itsoroch.ibm.com	Subnet Properties → Options

Table 15-5 shows the properties of subnet 172.23.12.0 and the place in iSeries Navigator where these properties can be set.

Table 15-5 Planning the DHCP server AS24: properties for subnet 172.23.12.0

Property	Value	Configuration reference
Subnet name	172.23.12.0	Subnet Properties → General
Subnet description	ITSO subnet 3	Subnet Properties → General
Subnet address	172.23.12.0	Subnet Properties → Address Pool → Subnet address
Subnet mask	255.255.255.0	Subnet Properties → Address Pool → Subnet
Address range for leasing	172.23.12.1 to 172.23.12.254	Subnet Properties → Address Pool → Range to assign
Lease time	Inherit from server (1 day)	Subnet properties → Leases → Inherit lease time
IP addresses excluded from the pool	172.23.12.1 to 172.23.12.10	Subnet Properties → Address Pool → IP addresses excluded
Options offered to DHCP clients 01 - Subnet mask 02 - DNS IP address 15 - Domain name	 255.255.255.0 172.23.12.1 itsoroch.ibm.com	Subnet Properties → Options

All the three subnets 172.23.10.0, 172.23.11.0, and 172.23.12.0 are grouped into a subnet group. Table 15-6 shows the properties of this subnet group and the place in iSeries Navigator where these properties can be set.

Table 15-6 Planningg the DHCP server AS24: properties for the subnet group

Property	Value	Configuration reference
Subnet group name	ITSOSubnetGroup1	Subnet Group Properties → General → Name
Subnets to include	172.23.10.0 172.23.11.0 172.23.12.0	Subnet Group Properties → General → Subnets to include
Address assignment method	Balanced	Subnet Group Properties → Address order → Address assignment method
Subnet order	1) 172.23.10.0 2) 172.23.11.0 3) 172.23.12.0	Subnet Group Properties → Address order → Subnet order

Step 2: Configure the new subnet on the DHCP server

To configure subnet 172.23.11.0, perform the following steps, using the values specified in Table 15-4 on page 294:

1. Start iSeries Navigator.
2. Expand your System i → **Network** → **Servers** and then click **TCP/IP**.
3. Double-click **DHCP**. This starts the DHCP Server Configuration window.
4. Right-click **Global** and select **New Subnet - Advanced** option from the context menu (Figure 15-36).

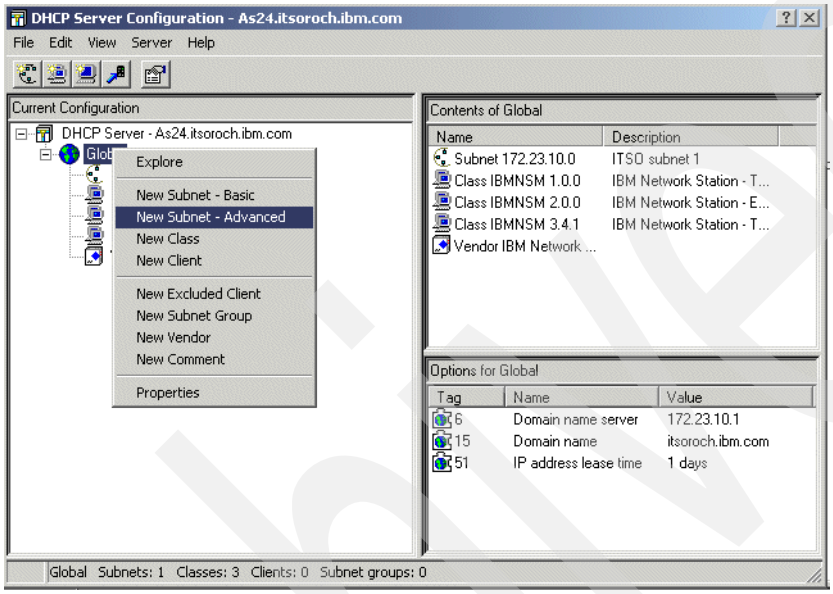


Figure 15-36 DHCP Server Configuration: creating a new subnet

5. The New Subnet properties window is opened. In the General tab, specify the name and the description of the new subnet. Select **Enabled** state (Figure 15-37).

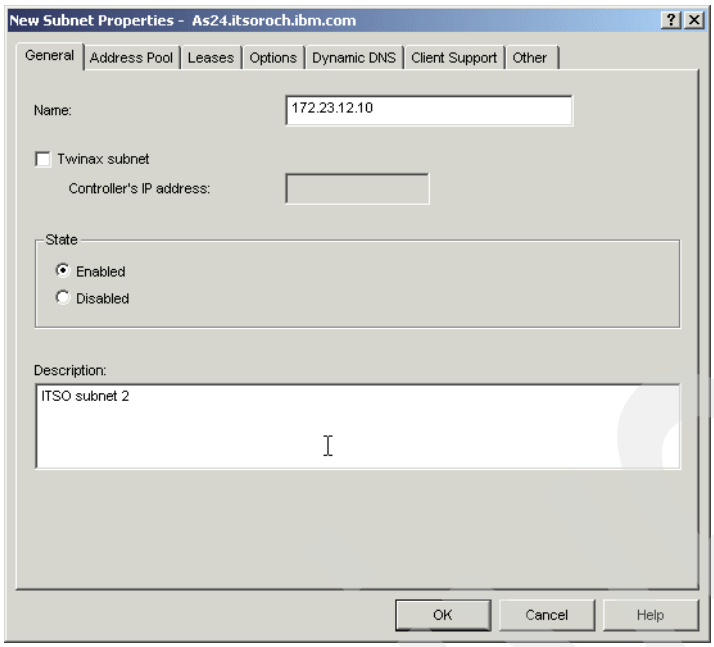


Figure 15-37 Configure subnet 172.23.11.0: General tab

6. Select the **Address Pool** tab. Specify the subnet address and the subnet mask. Specify the address range of IP addresses excluded from the pool (Figure 15-38). Click **Add**.

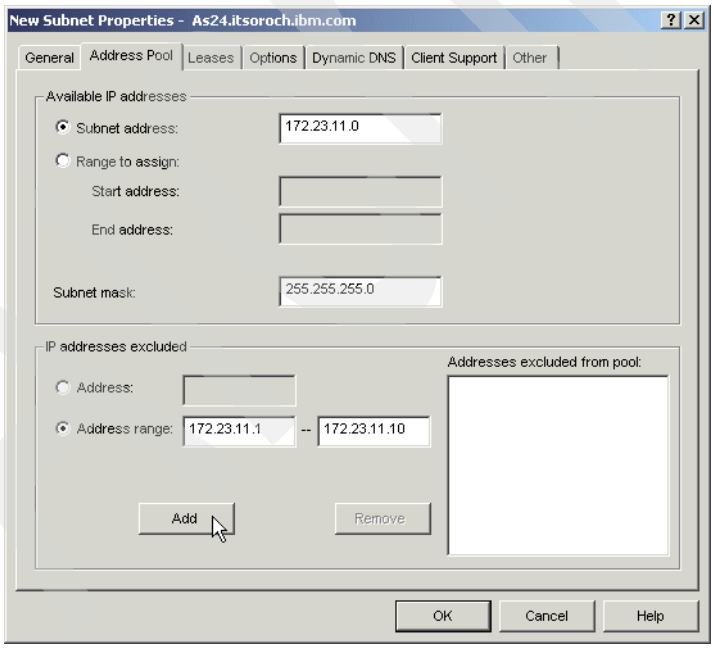


Figure 15-38 Configure subnet 172.23.11.0: Address Pool tab

7. Select the **Leases** tab. Select **Inherit lease time** (Figure 15-39). The lease time specified at the global level will be used for this subnet. If, for testing purposes, you want to override the global level default to something shorter, you would do that here.

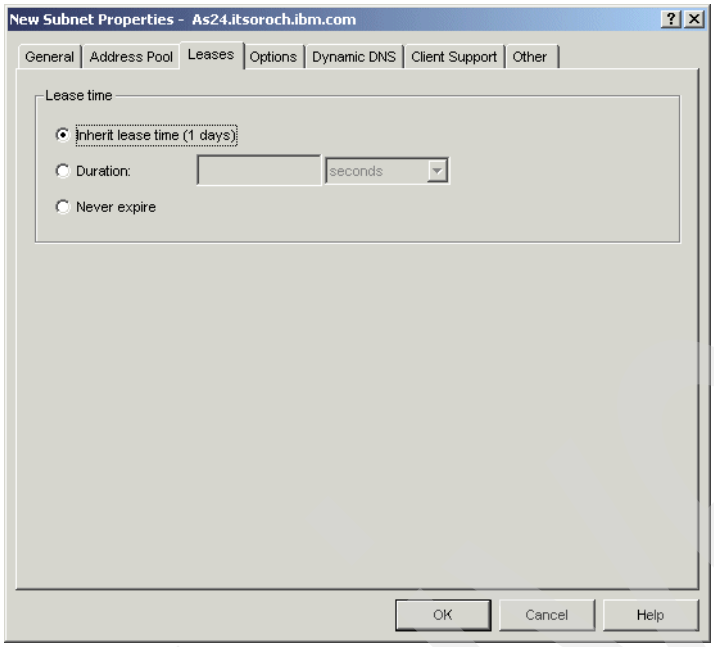


Figure 15-39 Configure subnet 172.23.11.0: Leases tab

8. Select the **Options** tab. From the Available options list, select option **1 - Subnet mask** and click **Add**. In the lower pane, specify the value for this option (Figure 15-40).

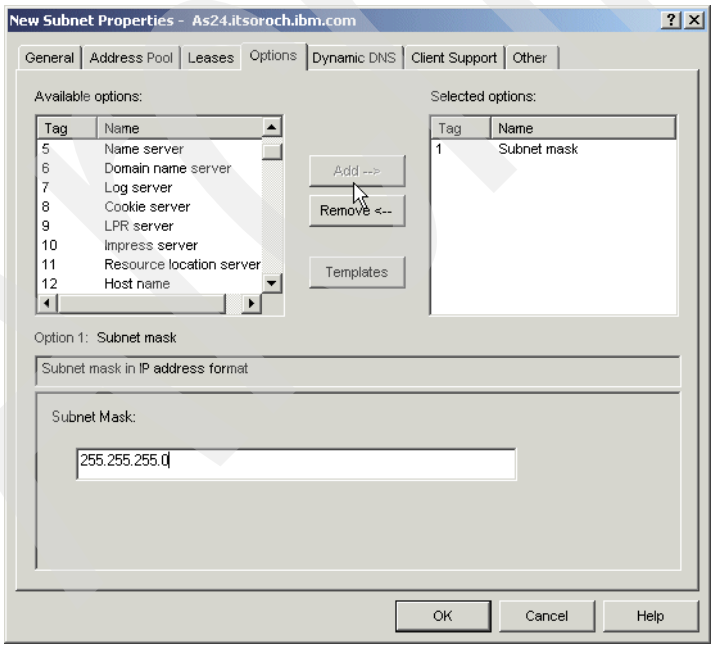


Figure 15-40 Configure subnet 172.23.11.0: Options tab (subnet mask)

9. From the Available options list, select option **6 - Domain name server** and click **Add**. In the lower pane, click **Add**. Specify the IP address of the domain name server and press Enter (Figure 15-41).

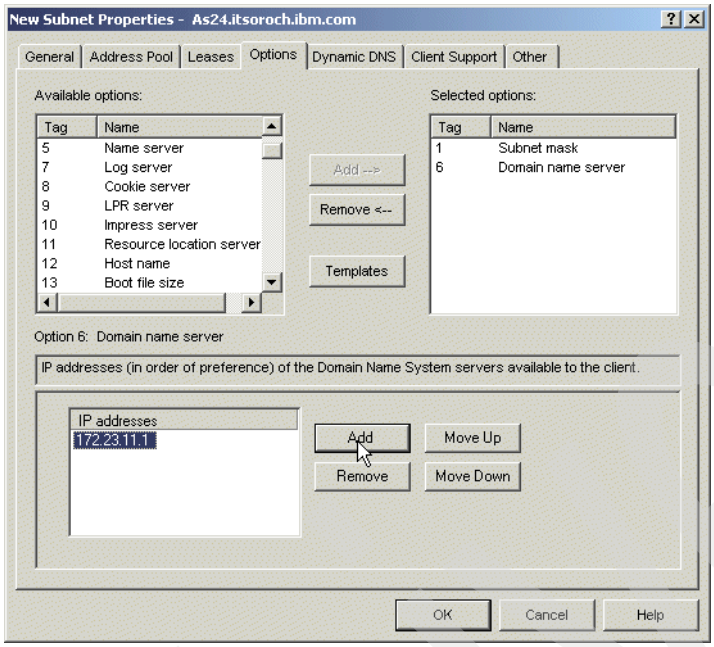


Figure 15-41 Configure subnet 172.23.11.0: Options tab (domain name server)

10. From the Available options list, select option **15 - Domain name** and click **Add**. In the lower pane, specify the domain name (Figure 15-42).

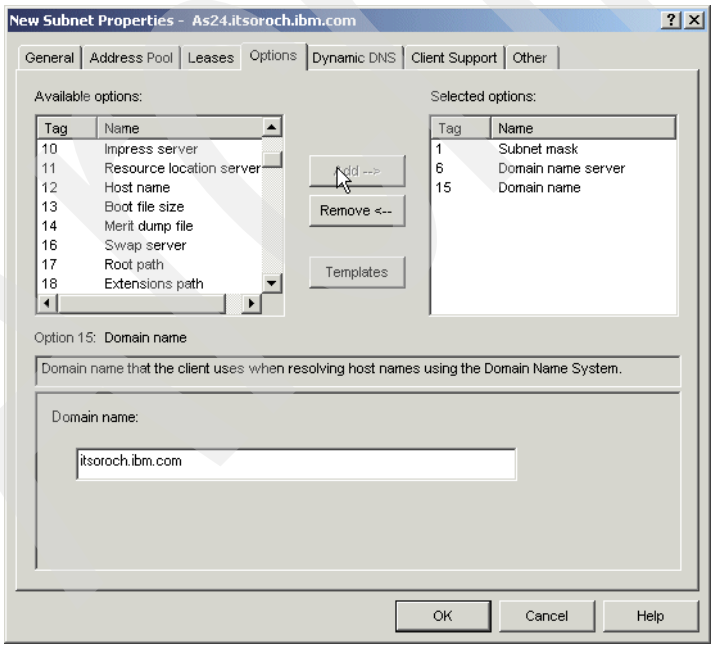


Figure 15-42 Configure subnet 172.23.11.0: Options tab (domain name)

11. Select the **Dynamic DNS** tab. Select **Inherited (Updates not performed)** under Update client records, and **Inherited** under Append domain name to host name (Figure 15-43).

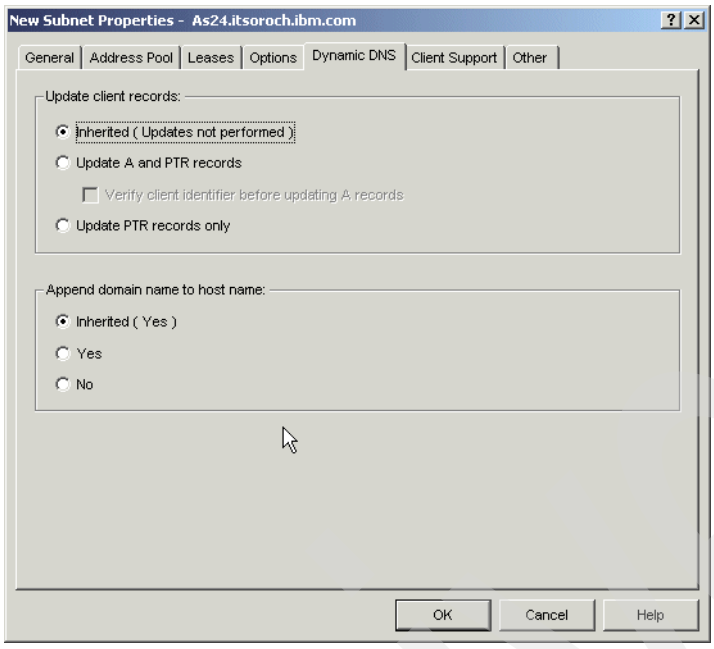


Figure 15-43 Configure subnet 172.23.11.0: Dynamic DNS tab

12. Select the **Client Support** tab. Select **Inherited** (Figure 15-44).

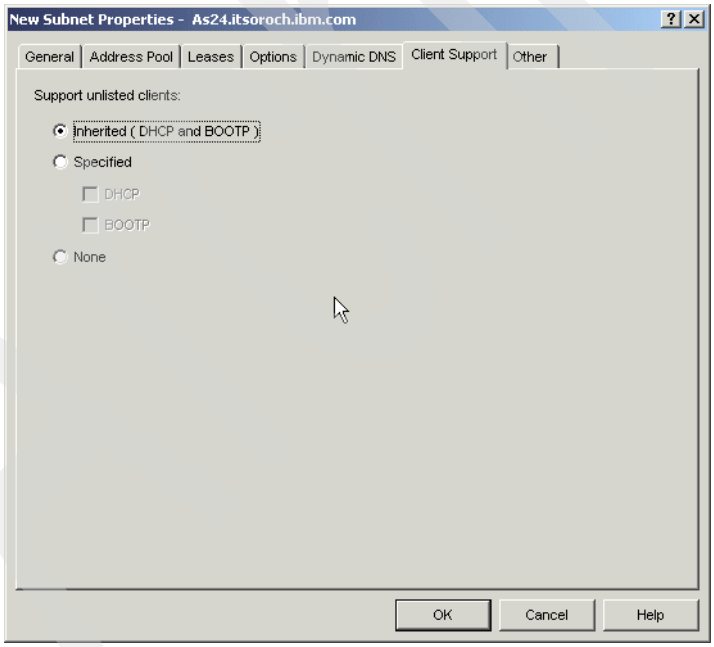


Figure 15-44 Configure subnet 172.23.11.0: Client Support tab

13. Select the **Other** tab. Select **Inherited** for the Bootstrap server (Figure 15-45).

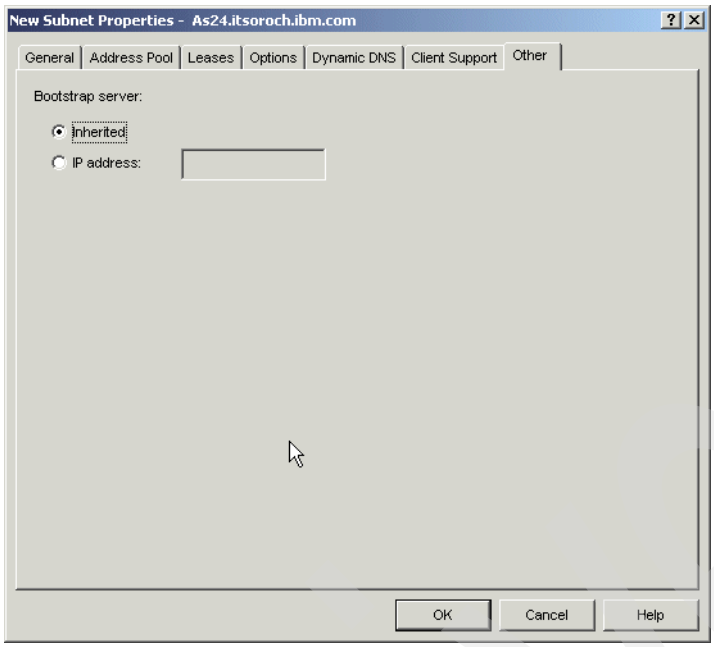


Figure 15-45 Configure subnet 172.23.11.0: Other tab

14. Click **OK**. This ends the configuration of subnet 172.23.11.0.

To continue by configuring subnet 172.23.12.0, repeat the steps in this section using the values specified in Table 15-5 on page 295.

This ends the configuration of the new subnets.

As you can see in Figure 15-46, the subnets 172.23.11.0 and 172.23.12.0 have been added to the DHCP configuration.

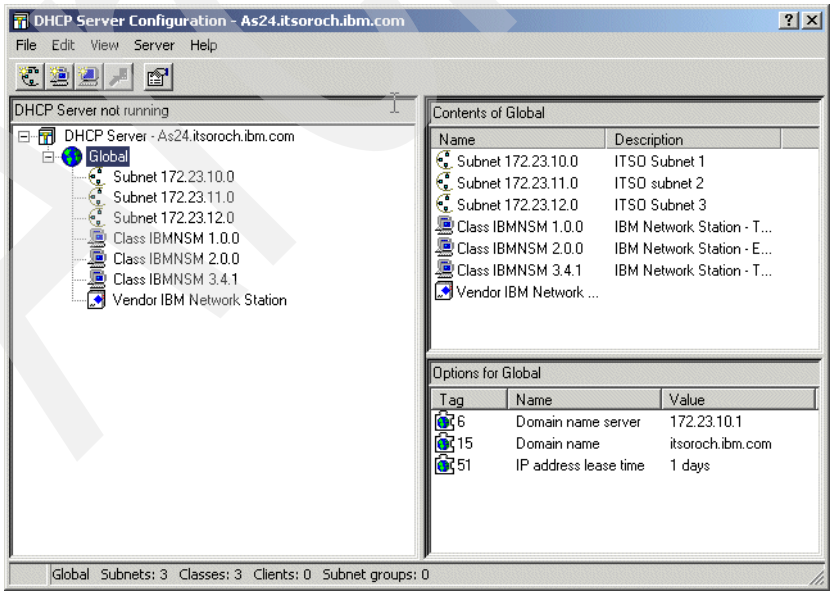


Figure 15-46 DHCP Configuration: the new created subnets

Step 3: Configure the subnet group on the DHCP server

To configure the subnet group, perform the following steps, using the values specified in Table 15-6 on page 295:

1. Start iSeries Navigator.
2. Expand your System i → **Network** → **Servers** and click **TCP/IP**.
3. Double-click **DHCP**. This starts the DHCP server configuration window.
4. Right-click **Global** and select **New Subnet Group** from the context menu (Figure 15-47).

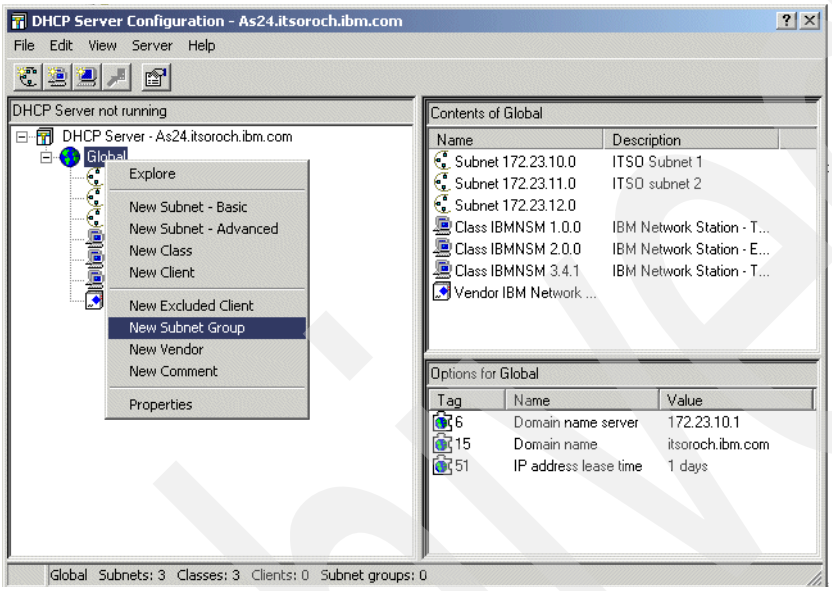


Figure 15-47 DHCP Server Configuration: create a new subnet group

5. The New Subnet Group Properties window is opened. Specify the name of the subnet group. From the Available subnets group, select subnet **172.23.10.0** and click **Add**, then select subnet **172.23.11.0** and click **Add**, then select subnet **172.23.12.0** and click **Add** (Figure 15-48).

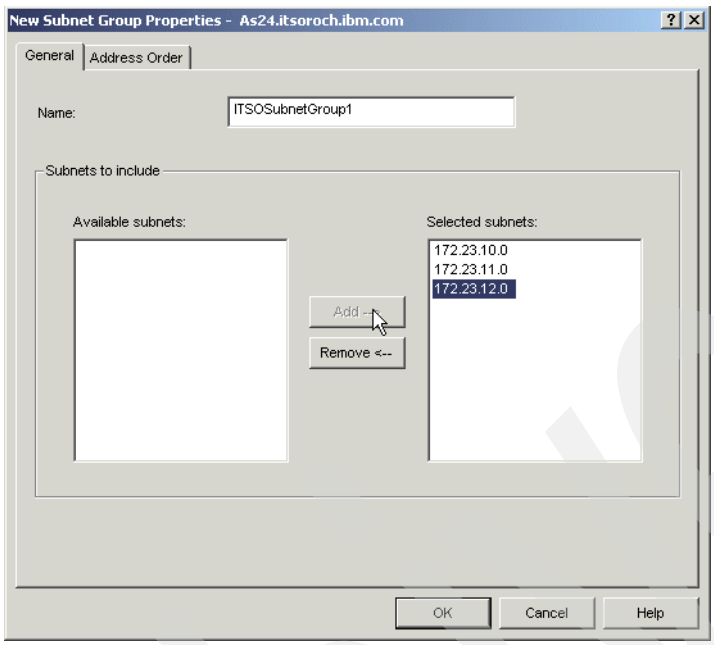


Figure 15-48 New Subnet Group Properties: General tab

6. Select the **Address Order** tab. Select **Balanced** for the Address assignment method. In the Subnet order list, specify the order to be used for the assignment of IP addresses (Figure 15-49).

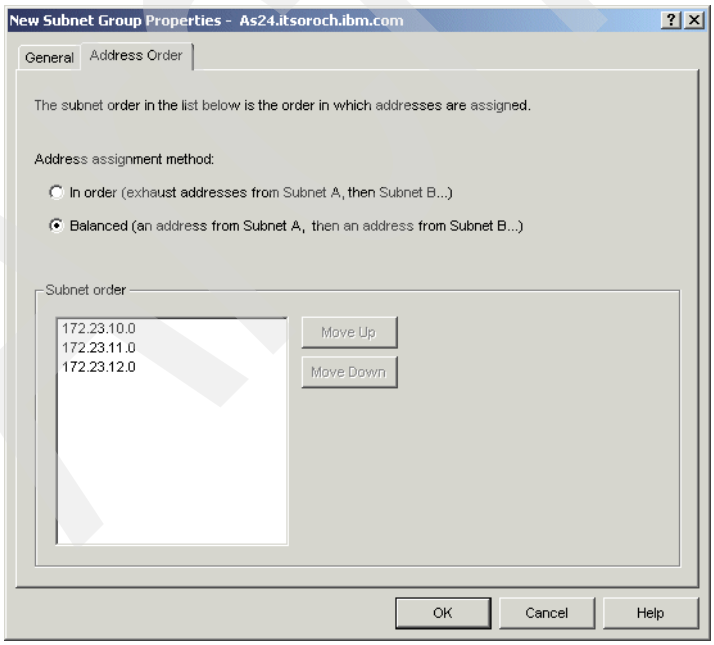


Figure 15-49 New Subnet Group Properties: Address Order tab

7. Click **OK**.

The subnet group ITSOSubnetGroup1 was created in the DHCP configuration. As shown in Figure 15-50, this subnet now includes the three subnets we created before.

This ends the subnet group configuration.

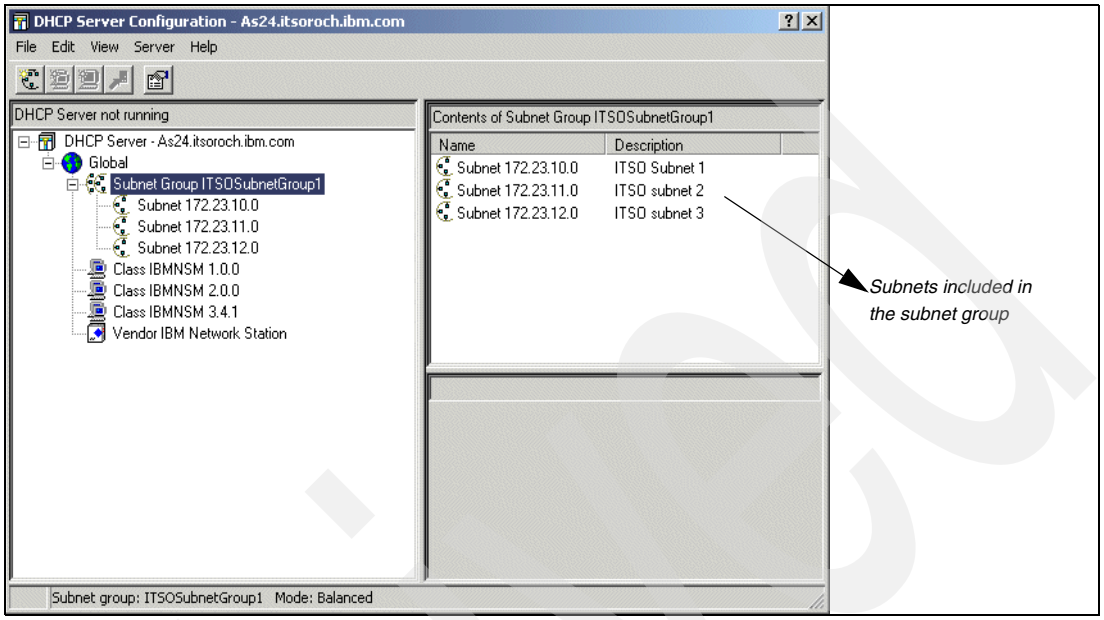


Figure 15-50 DHCP Server Configuration: the subnet group

Step 4: Configure the Windows 2000 DHCP clients

If you have not already configured your Windows 2000 clients, refer to “Step 5: Configure your Windows 2000 DHCP client” on page 288.

Step 5: Start the DHCP server

To start your DHCP server, refer to “Step 4: Start the DHCP server” on page 287.

Step 6: Test the configuration

In this scenario we use two DHCP clients (Client1 and Client2) for testing to see how the IP addresses are allocated by the server using balanced IP address allocation method.

To test the configuration, perform the following steps:

1. Power on workstation Client1. During the boot of the PC, the DHCP client will request an IP address from the System i DHCP server.
2. After the workstation starts, open a command prompt panel. Run the following command:

```
C:\> ipconfig /all
```

3. The IP configuration of the client is displayed on the command prompt panel. The configuration in Figure 15-51 shows us that the client obtained the IP address from the subnet with the highest priority in the subnet group, 172.23.10.0.

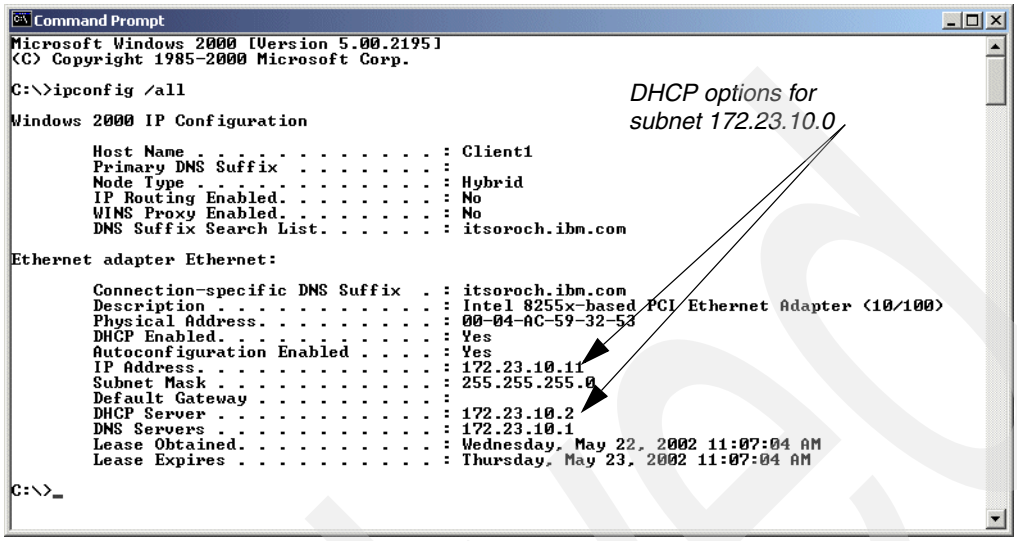


Figure 15-51 Client1 IP configuration after IP address allocation

4. Power on workstation Client2.
5. At the boot time, the DHCP client will request an IP address from the DHCP server.
6. After the workstation starts, open a command prompt panel.
7. Run the following command:

```
C:\> ipconfig /all
```

8. The IP configuration of the client is displayed on the command prompt panel. The configuration in Figure 15-52 shows us that the workstation Client2 obtained the IP address from the subnet 172.23.12.0.

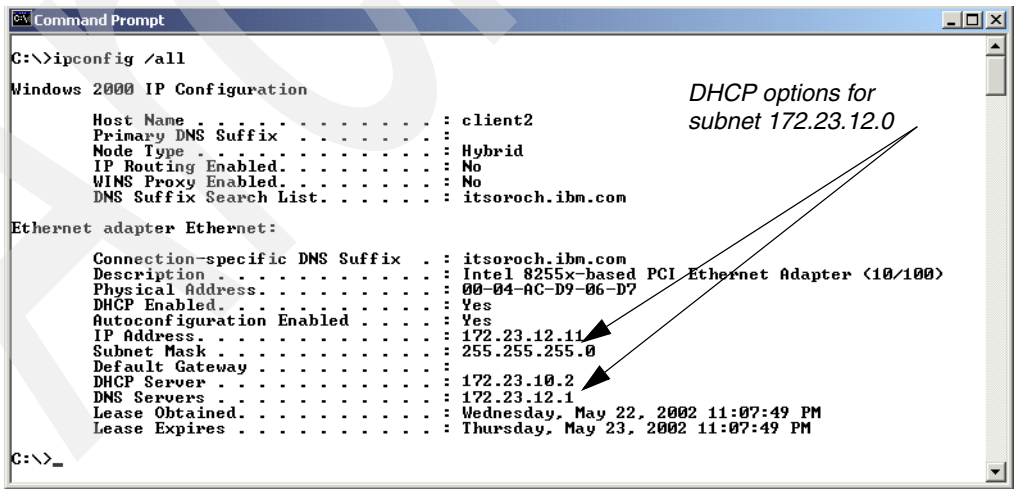


Figure 15-52 Client2 IP configuration after IP address allocation

This test shows how the balanced address assignment method works: The DHCP server, after allocating an IP address from the subnet 172.23.10.0 to the first client that requested an

IP address, changed the subnet used to allocate IP addresses to the next DHCP client, and so on.

Tip: Why did the first client reserve an IP address from subnet 172.23.10.0 and the second from 172.23.12.0? That is, why did the System i DHCP server seemingly skip allocating the second address from subnet 172.23.11.0?

We found the answer when we took a communication trace. It turns out that in our test environment Client1 sent two DHCPDISCOVER packets to the System i (the second packet was sent when a timeout occurred on the DHCP client, after 2 seconds from the first DHCPDISCOVER packet). This second DHCPDISCOVER causes the System i to rotate the subnets used to allocate IP addresses within the ITSOSubnetGroup1 subnet group.

So, when the third DHCPDISCOVER arrives at the System i DHCP server, it is actually from the second client, Client2. And, the DHCP server allocates an IP address from the next available subnet, 172.23.12.0.

At first, this interaction between the Windows client and the System i DHCP server does not look *balanced*. But balance will come when a large number of DHCP clients start spreading the IP address assignments across all the available subnets.

On the System i side perform the following:

1. Start iSeries Navigator.
2. Expand your System i → **Network** → **Servers** then click **TCP/IP**.
3. Right-click the **DHCP**. Select **Monitor** from the context menu. The DHCP Monitor window opens.
4. Expand **Status** and select **Leased**. The right-side pane shows the address that was allocated from the address pool and the client that obtained it (Figure 15-53).

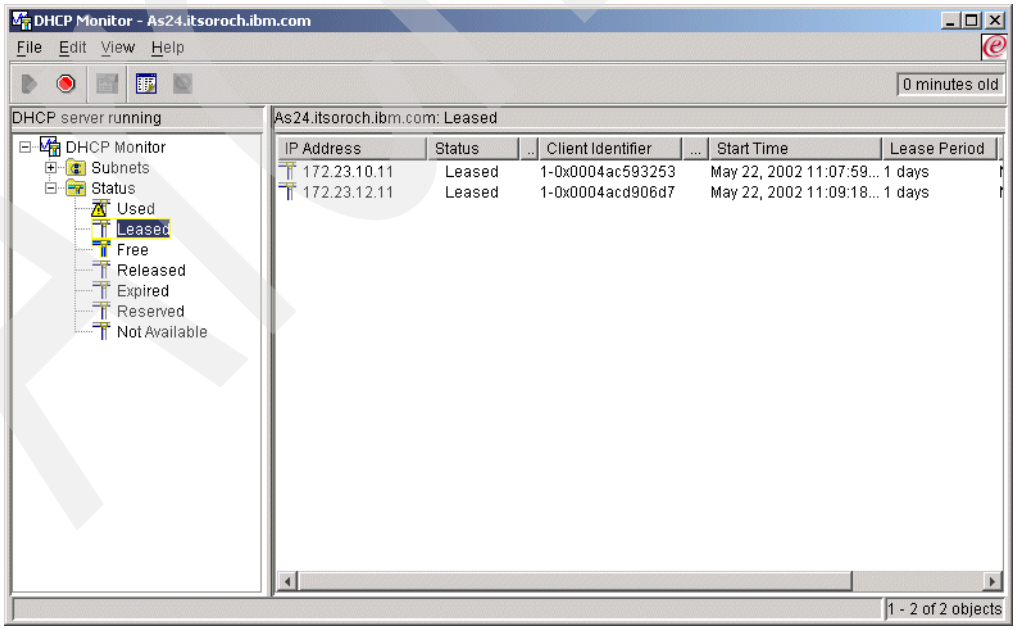


Figure 15-53 DHCP Monitor: leased addresses

This ends the DHCP configuration test.

Review, conclusions, and references

This scenario demonstrated how to configure the DHCP server to serve multiple logical subnets located in the same physical subnet.

First we showed how the DHCP server manages multiple logical subnets in the same physical subnet: by using the subnet group. Then we presented two methods of assigning IP addresses from a subnet group: in order and balanced.

We planned the configuration of the subnets and the subnet group. Table 15-4 on page 294, Table 15-5 on page 295, and Table 15-6 on page 295 helped you gather information about subnets and subnet group.

Then you learned how to configure the DHCP server on System i using iSeries Navigator.

Finally we showed how to test the configuration on the System i using the DHCP Monitor and the Windows 2000 client.

15.3 DHCP: One physical subnet, one logical subnet, multiple DHCP servers

In this scenario, we configure multiple DHCP servers in a network to minimize DHCP server-related failures.

Problem definition

In a TCP/IP network, a single DHCP server represents a single point of failure. That is, if the DHCP server fails, the new DHCP clients that want to enter the LAN will be unable to obtain an IP address.

Solution definition

To avoid a situation in which the DHCP clients are unable to obtain an IP address because the DHCP server failed, more than one DHCP server must be configured to serve the same subnet. Using multiple DHCP servers decreases, but does not eliminate, the probability of having a DHCP-related network access failure. If one server fails, the other can continue to serve the clients in the subnet. To serve all of the clients in the network, the DHCP servers must be accessible by any client in the network, by direct attachment or by using a BOOTP/DHCP relay agent.

One system can run only one DHCP server, so multiple DHCP servers mean multiple systems.

Two DHCP servers cannot serve the same addresses. Therefore, when you configure multiple DHCP systems to serve the same subnet, you must split the subnet IP addressing space across the DHCP servers. Two techniques are used to split the address pool across DHCP servers:

- *The 70/30 split technique*. In this technique, the primary DHCP server services 70% of the IP addressing space, and the other (backup) DHCP server services the remaining (30%) addressing space.

  This technique does not provide full DHCP client support because if the primary DHCP fails the backup DHCP server might be unable to service all requests from the clients due to its limited address pool.

When you use this technique, you might want DHCP clients to choose the IP addresses from the primary server, thus this server will exhaust its pool of addresses first and the backup DHCP server will have more IP addresses available in its address pool.

Some DHCP clients tend to select the DHCP server that offers more options. Other clients select the first DHCP server that responded, regardless of the number of options offered by the DHCP servers.

For more information about this technique, refer to *AS/400 TCP/IP Autoconfiguration: DNS and DHCP Support*, SG24-5147.

Note: In the testing environment, we attempted to get the Windows 2000 client to favor the DHCP server that offered more configuration options than the others. We performed the test using two System i DHCP servers:

- DHCP server SERVER1 was configured to offer DHCP clients only the IP address, subnet mask, and a lease time of 1 hour.
- DHCP server SERVER2 was configured to offer DHCP clients the IP address, subnet mask, router IP address, the domain name server IP address, the domain name, and a lease time of 24 hours.

When the DHCP client issued the DHCPDISCOVER message, SERVER1 responded first with the DHCPOFFER message. The DHCP server SERVER2 responded after 100 milliseconds.

The attempt to favor SERVER2 by sending more options and setting a longer lease time did not work. The Windows 2000 client did not appear to wait long for all incoming DHCPOFFERS to arrive. It appear to take the first offer received (the offer sent by SERVER1).

- Increase the IP addresses pool by modifying the subnet mask. By increasing the addresses pool, each DHCP server can manage a large enough pool of addresses to satisfy DHCP requests from all of the DHCP clients in the network. So, if the primary server fails, the backup server would be able to service all of the clients in the network.

  If enough IP addresses are available this is clearly the best option.

How-to

This scenario is based on the scenario in 15.1, “DHCP: One physical network, one logical network, one DHCP server” on page 270, in which a System i acts as a DHCP server in a simple TCP/IP network. However, to eliminate the single point of failure, a new DHCP server running on a second System i (AS20) will be added to the network and must be configured. In this scenario, we present both techniques used to split the address pool between the DHCP servers:

- Option A: dividing the address pool across two DHCP servers
- Option B: providing full DHCP client support

This scenario does not discuss the DHCP client configuration. It only provides techniques to reduce the possible DHCP server outages.

We plan and perform only the modifications made to the configurations implemented in 15.1, “DHCP: One physical network, one logical network, one DHCP server” on page 270.

Option A: dividing the address pool across two DHCP servers

The network configuration that is implemented in this scenario is presented in Figure 15-54 and has the following characteristics:

- There is a single physical subnet.
- There are two System i's: AS24 and AS20. Each one has a single network interface in the network with a single IP address. Both System i's act as DHCP servers in the network.
- The DHCP server on AS24 is the primary server and holds 70% of the available IP addresses (address range 172.23.10.10 to 172.23.10.181).
- The DHCP server on AS20 is the secondary server and holds 30% of the address pool (address range 172.23.10.182 to 172.23.10.254).
- The primary DHCP server offers more options than the secondary DHCP server.

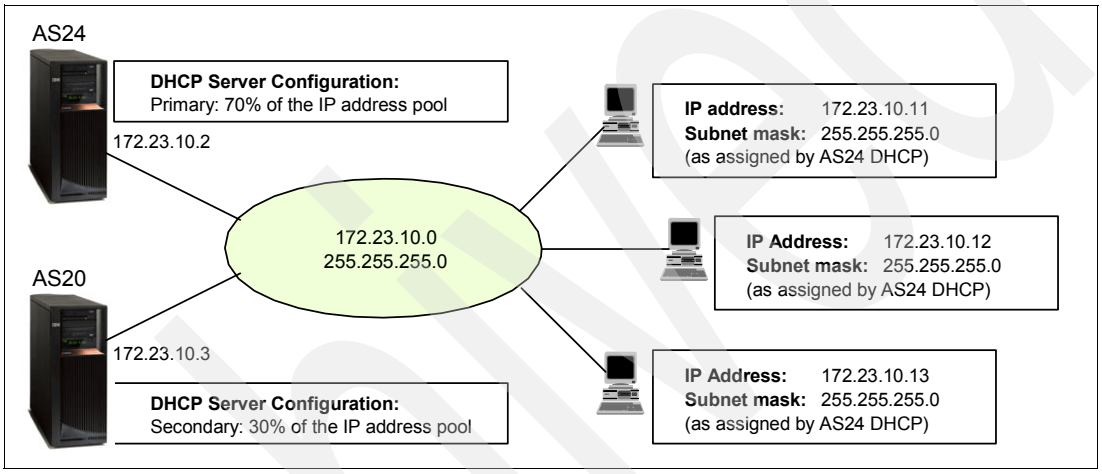


Figure 15-54 Multiple DHCP servers in a single network: 70/30 split technique

Because this scenario is based on the configuration in the scenario in 15.1, "DHCP: One physical network, one logical network, one DHCP server" on page 270, we plan and perform only the modifications that occur in the network configuration.

To implement the configuration specified above, perform the following steps:

- Step 1: Configure the IP interface on System i AS20.
- Step 2: Plan the configuration of the DHCP server AS20.
- Step 3: Configure the DHCP server AS20.
- Step 4: Reduce the primary DHCP server IP address pool; add DHCP options.
- Step 5: Add the remaining IP addresses to the backup DHCP server.
- Step 6: Start the primary and backup DHCP servers.

Step 1: Configure the IP interface on System i AS20

To configure the IP interface on System i AS20, perform the steps specified in "Step 1: Configure the System i AS24 network interface" on page 272, using the values specified below:

IP interface 172.23.10.3
Network mask 255.255.255.0
Line description ETHLIN1

Step 2: Plan the configuration of the DHCP server AS20

To configure the DHCP server AS20 through iSeries Navigator, first respond to a series of questions about the configuration of the DHCP server. All of these questions are included in

Table 15-7, Table 15-8, and Table 15-9 on page 311. The answers are based on the network configuration in Figure 15-54 on page 309.

Table 15-7 contains information about the IP configuration of System i AS20.

Table 15-7 Pplanning the DHCP server AS20: TCP/IP information

Configuration parameter	Value
Host name	AS20
Description	DHCP server
IP address	172.23.10.3
Network mask	255.255.255.0
Line description	ETHLIN1
Domain name	itsoroch.ibm.com

Table 15-8 contains information about the DHCP server on the System i AS20. The third column in this table indicates the place in iSeries Navigator where you can configure the specified parameter. Many of these configuration options can be specified through the DHCP configuration wizard the first time you configure the DHCP server.

Table 15-8 Planning the DHCP server AS20: DHCP server overview

#	Question	Answer	Configuration reference
1	Start the DHCP server when TCP/IP starts?	Yes	DHCP Server → Properties → General → Start when TCP/IP is started
2	Is the BOOTP server already configured on the system?	No	DHCP configuration wizard
3	Do you want to migrate the BOOTP configuration to DHCP?	N/A	File → Migrate BOOTP
4	What is the default lease time for this server	1 day	Global → Properties → Leases → Duration
5	Do you want the DHCP server to support BOOTP clients?	No	Global → Properties → Client Support → Support BOOTP clients
6	Do you want the DHCP server to support any client from any subnet?	Yes	Global → Properties → Client Support → Support unlisted clients → DHCP clients
7	Do you want to log the DHCP server activity?	Yes	DHCP server → Properties → Logging → Enable logging
8	Can your DHCP clients (other than Network Stations) identify the class they belong to?	No	
9	If the answer to 8 is Yes, do you want to add a new class to serve the DHCP clients that belong to that class?	N/A	Global → New Class
10	Which are the IP interfaces the server listens on?	172.23.10.3	See Figure 15-54 on page 309
11	Which subnet will be administered by the DHCP server?	172.23.10.0	See Figure 15-54 on page 309

#	Question	Answer	Configuration reference
12	Do you want to add a new subnet to be administered by the DHCP server?	Yes	Global → New Subnet - Advanced

To allocate 30% of the IP address pool to the backup DHCP server, a new subnet must be configured on the DHCP server AS20. The properties of this subnet and the place in iSeries Navigator where these properties can be set are presented in Table 15-9.

Table 15-9 Planning the DHCP server AS20: properties for subnet 172.23.10.0

Property	Value	Configuration reference
Subnet name	172.23.10.0	Subnet Properties → General
Subnet description	ITSO subnet 1 (30% of pool)	Subnet Properties → General
Subnet address	172.23.10.0	Subnet Properties → Address Pool → Subnet address
Subnet mask	255.255.255.0	Subnet Properties → Address Pool → Subnet
Address range for leasing	172.23.11.182 to 172.23.11.254	Subnet Properties → Address Pool → Range to assign
Lease time	Inherit from server (1 day)	Subnet properties → Leases → Inherit lease time
IP addresses excluded from the pool		Subnet Properties → Address Pool → IP addresses excluded
Options offered to DHCP clients 01 - Subnet mask 02 - DNS IP address 15 - Domain name	 255.255.255.0 172.23.10.1 itsoroch.ibm.com	Subnet Properties → Options

Step 3: Configure the DHCP server AS20

To configure the DHCP server on AS20, perform the steps from "Step 3: Configure the DHCP server AS24" on page 276 using the values from Table 15-8 on page 310.

Step 4: Reduce the primary DHCP server IP address pool; add DHCP options

In this example, we use the 70/30 split technique. The existing DHCP server on AS24 becomes the primary DHCP server and holds 70% of the address pool. The new DHCP server implemented in the network on the System i AS20 is the backup DHCP server and holds 30% of the IP address pool. Depending on your IP addressing scheme, you might choose another split technique.

To reduce the IP address pool of the primary DHCP server AS24, perform the following:

1. Start iSeries Navigator.
2. Expand your System i → **Network** → **Servers** then click **TCP/IP**.
3. Double-click **DHCP**. This starts the DHCP server configuration window.

4. Right-click subnet **172.23.10.0** and select **Properties** from the context menu (Figure 15-55).

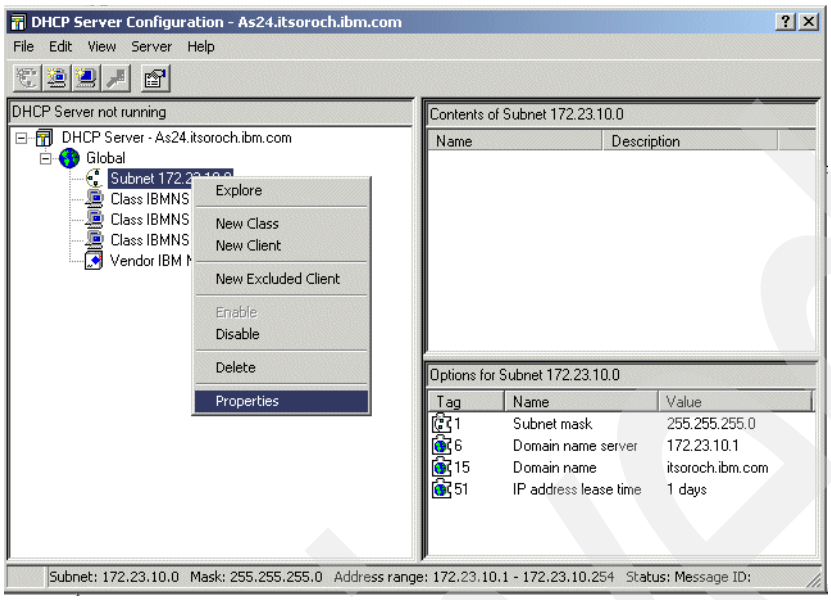


Figure 15-55 DHCP Server Configuration: selecting subnet properties

5. The subnet properties window is opened. Select the **Address Pool** tab. In the Available IP addresses group, modify the End address from 172.23.10.254 to `172.23.10.181` (Figure 15-56). This reduces the available IP address pool to 70% of its maximum.

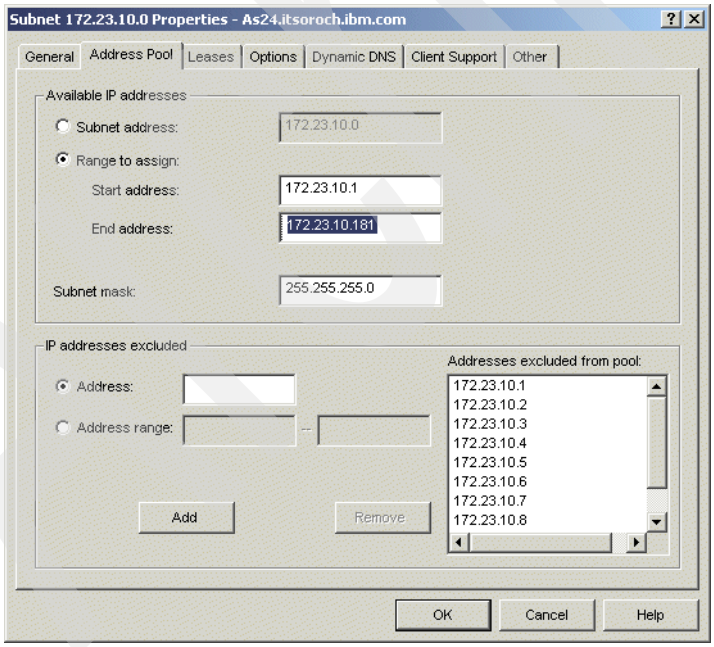


Figure 15-56 Subnet Properties: modifying the IP address pool

6. In this step, we add a new option to the options provided by the primary DHCP server to the DHCP clients. We are attempting to cause the DHCP clients to select the IP address from the primary DHCP server instead of the IP address from the backup DHCP server.

   However, our tests revealed that the Windows 2000 clients do not wait for offers from more servers to compare them and to choose the best offer. They simply pick the first offer they receive. So, depending on the behavior of your clients, adding a new option in the primary DHCP server might not influence the clients to choose this DHCP server.

   To add an option to the subnet configuration, click the **Options** tab. In this tab, there are already three DHCP options configured (1 - Subnet mask, 6 - Domain name server, 15 - Domain name). Add an option from the Available options list by selecting it and clicking the **Add** button (Figure 15-57). Then specify the value of the option added.

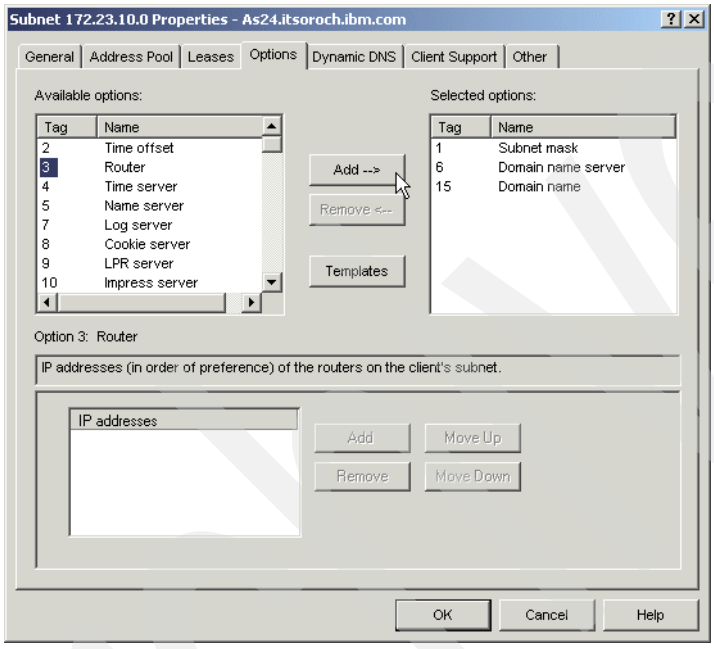


Figure 15-57 Subnet Properties: adding a new option to the subnet configuration

7. Click **OK** to close the Subnet properties window.

Step 5: Add the remaining IP addresses to the backup DHCP server

To add the remaining IP address to the backup DHCP server, a subnet must be created in its DHCP configuration.

To create a subnet on the backup DHCP server AS20, perform the following steps using the values specified in Table 15-9 on page 311:

1. Start the iSeries Navigator.
2. Expand **as20.itsoroch.ibm.com** to select this system.
3. Expand **Network** → **Servers** and click **TCP/IP**.
4. Double-click **DHCP**. The DHCP Server Configuration window opens.
5. Right-click **Global** and select **New Subnet - Advanced** from the context menu. The New Subnet Properties window opens.

6. In the General tab, specify the Name and Description for the new subnet. Select State **Enabled** (Figure 15-58).

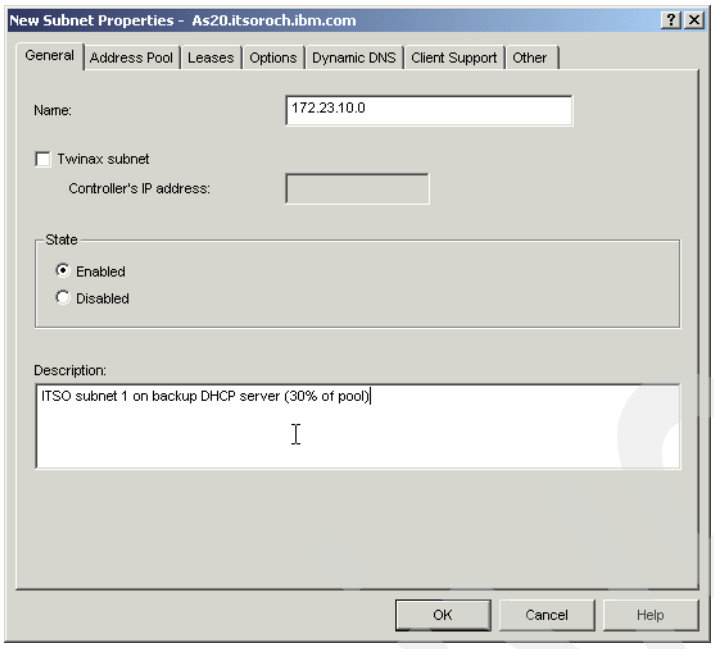


Figure 15-58 New Subnet Properties on backup DHCP server (AS20): General tab

7. Click the **Address Pool** tab. Select **Range to assign** and specify Start address `172.23.10.182` and End address `172.23.10.254`, as in Figure 15-59. This address range represents 30% of the total available IP address pool (172.23.10.11 - 172.23.10.254).

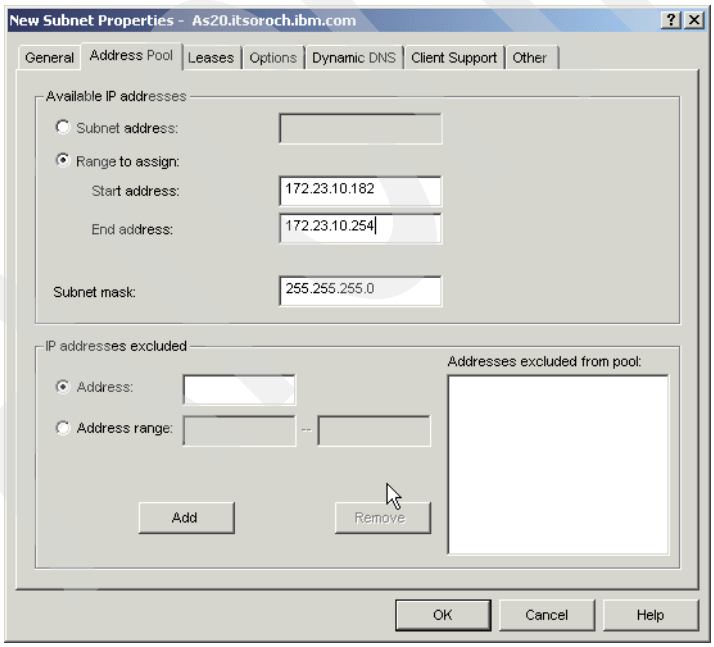


Figure 15-59 New Subnet Properties on backup DHCP server (AS20): Address Pool tab

8. Click the **Leases** tab and select **Inherit lease time** (1 day) (Figure 15-60).

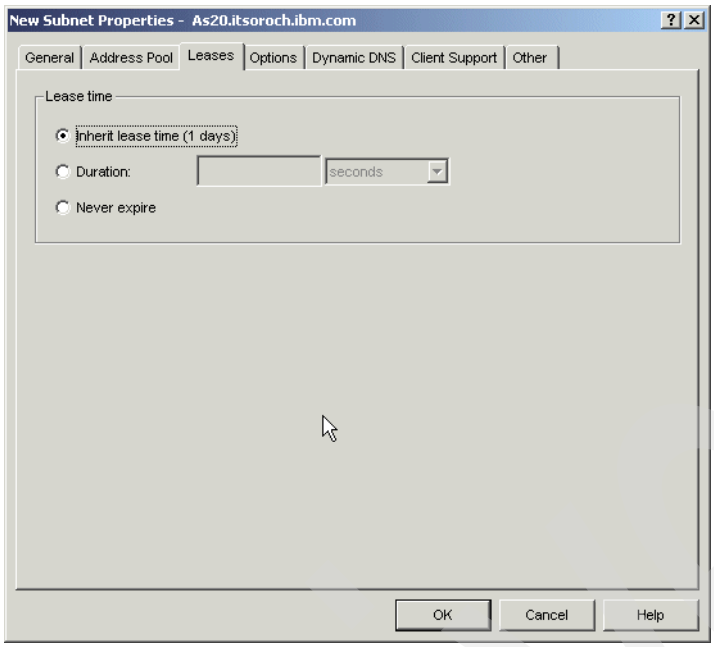


Figure 15-60 New Subnet properties on backup DHCP server (AS20): Leases tab

9. Select the **Options** tab. From the Available options list, select options **1 - Subnet mask**, **6 - Domain name server**, and **15 - Domain name**, and click **Add** (Figure 15-61). Specify the value of each option, as follows:

1 - Subnet mask	255.255.255.0
6 - Domain name server	172.23.10.1
15 - Domain name	itsoroch.ibm.com

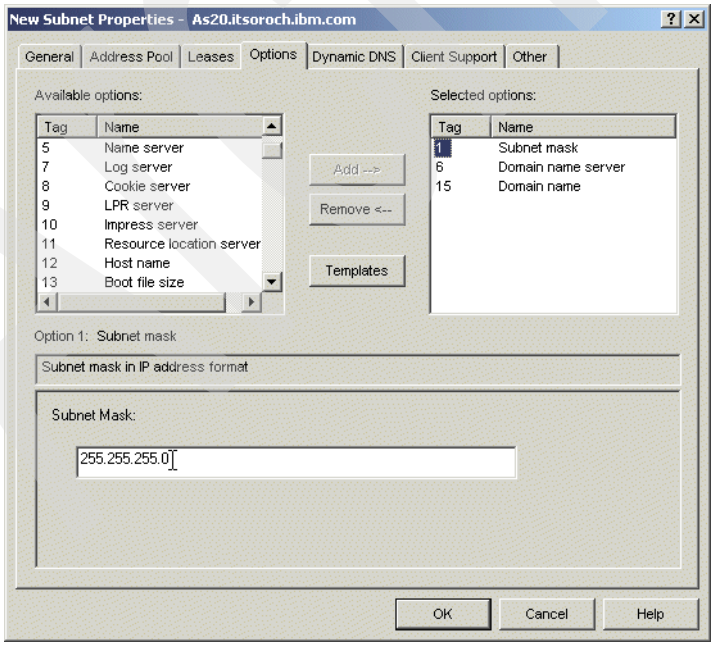


Figure 15-61 New Subnet properties on backup DHCP Server (AS20): Options tab

10. Select the **Dynamic DNS** tab. Select **Inherited** (Updates not performed) in the Update client records, and **Inherited** in the Append domain name to host name.
11. Select the **Client support** tab. Select **Inherited for Support unlisted clients**.
12. Select the **Other** tab. Select **Inherited for the Bootstrap server**.
13. Click **OK** to create the new subnet on the backup DHCP server, shown in Figure 15-62.

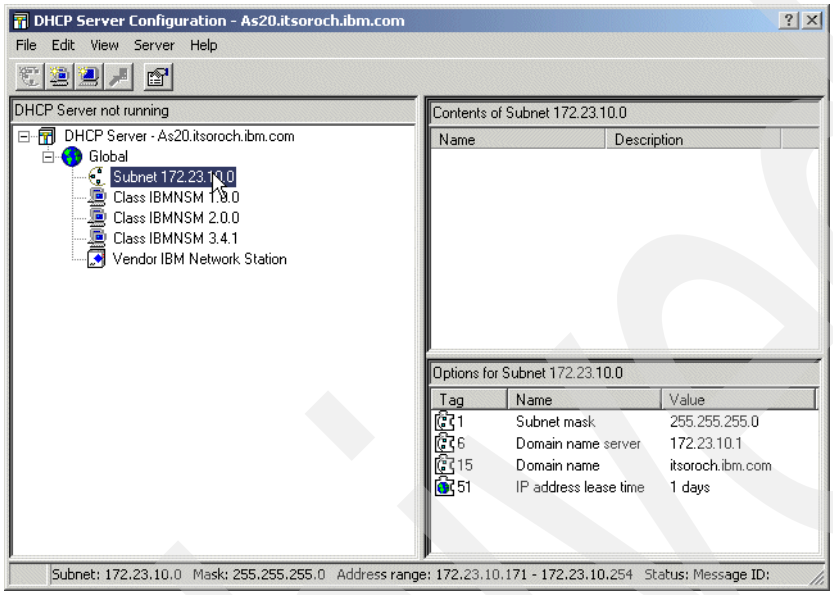


Figure 15-62 Backup DHCP server (AS20) configuration

This ends the configuration of the subnet on the backup server.

Step 6: Start the primary and backup DHCP servers

Because some DHCP clients choose the first DHCP offer they receive, not necessarily the best DHCP offer, consider starting the primary DHCP server first and leave it running until it has exhausted nearly all of its IP range, then start the backup DHCP server. Or, you can start the backup DHCP server only when the primary DHCP server fails.

To start the DHCP servers, refer to “Step 4: Start the DHCP server” on page 287.

Option B: providing full DHCP client support

In this section we modify the DHCP servers configured in the previous scenario (“Option A: dividing the address pool across two DHCP servers” on page 309) to provide full DHCP client support.

Full DHCP client support means that each DHCP server can service all of the clients, so if one DHCP server fails there is no danger of having DHCP clients that cannot obtain an IP address because the other DHCP server has run out of available IP addresses. This scenario assumes that there are no constraints with IP addresses.

The network used in our scenario has a maximum 240 clients. The subnet mask used is 255.255.255.0. This subnet mask allows an IP address pool of 254 addresses.

In order to implement the full-DHCP client support, each DHCP server must hold 240 IP addresses, thus an IP address space of at least 480 addresses.

To increase the IP address space enough, the subnet mask must be modified from 255.255.255.0 to 255.255.254.0. This new subnet mask has 9 bits for host-addressing scope, which means:

```
2^9 - 2 = 510 IP addresses
```

Having an IP address space of 510 IP addresses, we can allocate to each DHCP server a pool of 240 addresses to implement the full-DHCP client support.

The network configuration implemented in this scenario is presented in Figure 15-63 and has the following characteristics:

- There is a single physical subnet.
- There is a single logical network. The network mask is 172.23.10.0, the subnet mask used is 255.255.254.0. The first 10 IP addresses are reserved for servers and are not available to DHCP clients.
- There are two DHCP servers. Each DHCP server holds an IP address pool of 255 addresses, large enough to service all DHCP clients in the network.

To enable the System i servers to communicate with the clients after the IP address allocation, the network mask of the System i servers' IP interfaces must be changed from 255.255.255.0 to 255.255.254.0.

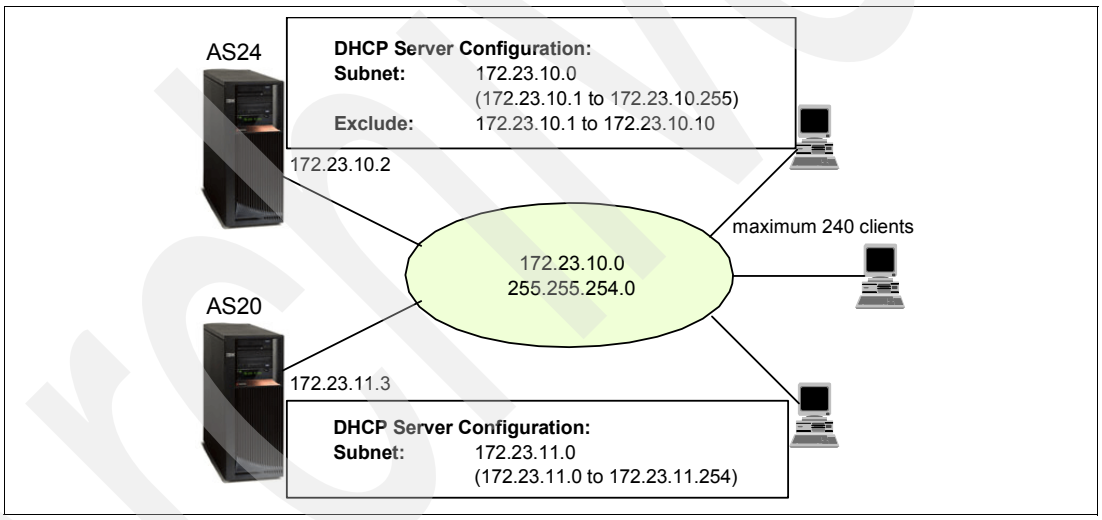


Figure 15-63 Multiple DHCP servers in a single network: full DHCP client support

To implement the configuration mentioned above, perform the following steps:

- Step 1: Modify the network mask of the IP interface on AS24 and AS20.
- Step 2: Extend the primary DHCP server IP address pool.
- Step 3: Add the remaining IP addresses to the backup DHCP server.
- Step 4: Start the primary and backup DHCP servers.

These steps represent only modifications to the configuration implemented in the scenario in "Option A: dividing the address pool across two DHCP servers" on page 309.

Step 1: Modify the network mask of the IP interface on AS24 and AS20

To modify the network mask of the System i AS24 IP interface, perform the following steps:

1. Sign on to the system as a user with *IOSYSCFG and *ALLOBJ special authorities. Make sure this connection is not using the interface because one of the steps is to end this interface.

2. On the command line enter the Configure TCP/IP (CFGTCP) command:

```
CFGTCP
```

3. In the Configure TCP/IP menu, select option **1** (Work with TCP/IP interfaces) to display the Work with TCP/IP Interfaces display (Figure 15-64).

```
                        Work with TCP/IP Interfaces
                                                              System:   AS24
Type options, press Enter.
  1=Add   2=Change   4=Remove   5=Display   9=Start   10=End

     Internet         Subnet           Interface
Opt  Address          Mask             Status

     127.0.0.1        255.0.0.0        Active
     172.23.10.2      255.255.255.0    Active



                                                                      Bottom
F3=Exit      F5=Refresh   F6=Print list   F11=Display line information
F12=Cancel   F17=Top      F18=Bottom
```

Figure 15-64 Work with TCP/IP Interfaces panel

4. Select option **10** to end the interface 172.23.10.2 and press Enter. Press F5 to refresh the panel until you see that the interface is inactive (Figure 15-65).

```
                        Work with TCP/IP Interfaces
                                                              System:   AS24
Type options, press Enter.
  1=Add   2=Change   4=Remove   5=Display   9=Start   10=End

     Internet         Subnet           Interface
Opt  Address          Mask             Status

     127.0.0.1        255.0.0.0        Active
     172.23.10.2      255.255.255.0    Inactive



                                                                    Bottom
F3=Exit      F5=Refresh   F6=Print list   F11=Display line information
F12=Cancel   F17=Top      F18=Bottom
```

Figure 15-65 Work with TCP/IP Interfaces panel

5. Select option **2** to change the interface 172.23.10.2 and press Enter to continue. The Change TCP/IP Interface panel is displayed. Modify the subnet mask to value 255.255.254.0 (Figure 15-66). Press Enter.

```
                     Change TCP/IP Interface (CHGTCPIFC)

Type choices, press Enter.

Internet address . . . . . > '172.23.10.2'
Line description . . . . . .   ETHLIN1       Name, *SAME, *VIRTUALIP...
Subnet mask  . . . . . .   '255.255.254.0'
Associated local interface . . .   '*NONE'
Type of service  . . . . . .   *NORMAL       *SAME, *MINDELAY...
Maximum transmission unit  . . .   *LIND         576-16388, *SAME, *LIND
Autostart  . . . . . .   *YES          *SAME, *YES, *NO
PVC logical channel identifier     *SAME         001-FFF, *SAME, *NONE
              + for more values
X.25 idle circuit timeout  . . .   *SAME         1-600, *SAME
X.25 maximum virtual circuits  .   *SAME         0-64, *SAME
X.25 DDN interface . . . . . .   *SAME          *SAME, *YES, *NO
TRLAN bit sequencing . . . . . .   *SAME         *SAME, *MSB, *LSB




                                                                      Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

Figure 15-66 Change TCP/IP Interface panel

6. In the Work with TCP/IP Interfaces panel, select option **9** (Start) to start the interface 172.23.10.2 (Figure 15-67).

```
                       Work with TCP/IP Interfaces
                                                          System:   AS24
Type options, press Enter.
  1=Add   2=Change   4=Remove   5=Display   9=Start   10=End

     Internet         Subnet           Interface
Opt  Address          Mask             Status

     127.0.0.1        255.0.0.0        Active
     172.23.10.2      255.255.254.0    Active

                                                                    Bottom
F3=Exit      F5=Refresh   F6=Print list   F11=Display line information
F12=Cancel   F17=Top      F18=Bottom
```

Figure 15-67 Work with TCP/IP Interfaces panel

Perform these same steps on System i AS20 to modify IP interface 172.23.10.3 on it.

Step 2: Extend the primary DHCP server IP address pool

To enlarge the IP address space, the network mask of the subnet 172.23.10.0 must be modified from 255.255.255.0 to 255.255.254. After that, the IP address range must be reduced to half.

To extend the primary DHCP server IP address pool, perform the following:

1. Start the iSeries Navigator.
2. Click **as24.itsoroch.ibm.com** to select the system.
3. Expand **as24.itsoroch.ibm.com** → **Network** → **Servers** → **TCP/IP**.
4. Double-click **DHCP**. This opens the DHCP Server Configuration window.
5. Right-click the subnet **172.23.10.0** and select **Properties** from the context-menu.
6. The subnet Properties window opens. Select the **Address Pool** tab. Modify the Subnet mask from 255.255.255.0 to `255.255.254.0`. Modify the End address from 172.23.10.181 to `172.23.10.255` as shown in Figure 15-68.

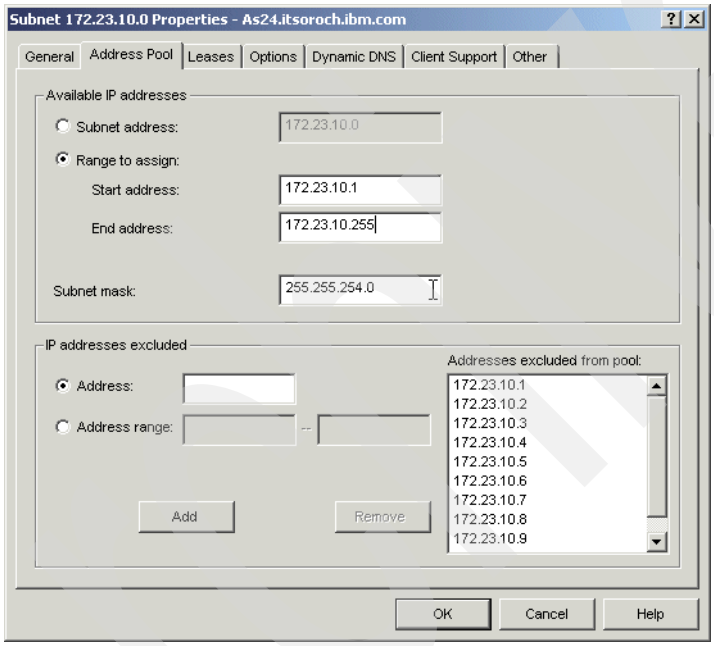


Figure 15-68 Primary DHCP server: increasing the IP address space

7. Click **OK** to finish and close the DHCP Server Configuration wizard.

This ends the procedure.

Step 3: Add the remaining IP addresses to the backup DHCP server

To add the remaining IP addresses to the backup DHCP server, perform the following:

1. Start the iSeries Navigator.
2. Click **as20.itsoroch.ibm.com** to select the system.
3. Expand **as20.itsoroch.ibm.com** → **Network** → **Servers** → **TCP/IP**.
4. Double-click **DHCP**. This opens the DHCP Server Configuration window.
5. Right-click the subnet **172.23.10.0** and select **Properties** from the context menu. The Subnet Properties window opens.

6. Select the **Address Pool** tab. Modify the subnet mask from 255.255.255.0 to `255.255.254.0`. Modify the Start address from 172.23.10.182 to `172.23.11.0`. Modify the End address from 172.23.10.254 to `172.23.11.254` (Figure 15-69).

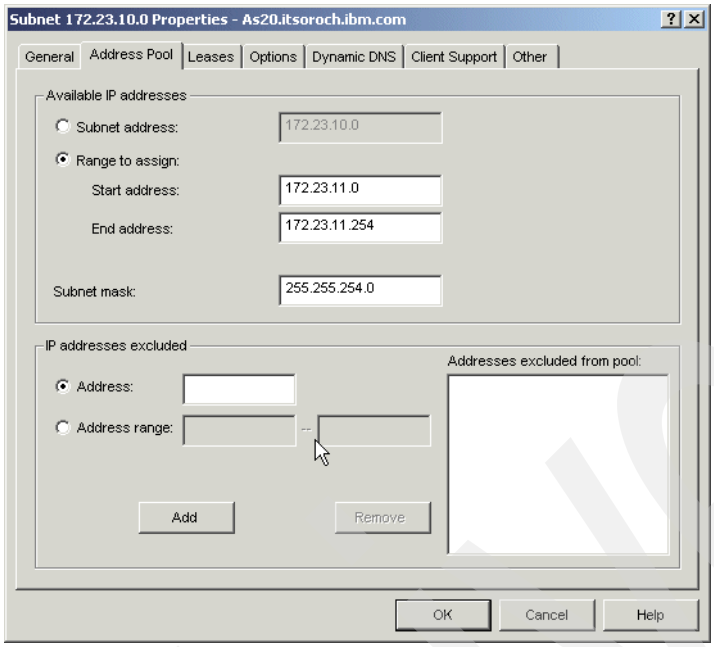


Figure 15-69 Backup DHCP server: modifying the IP address pool

7. Click **OK** to close the Subnet Properties window.
8. Close the DHCP Server Properties window.

Step 4: Start the primary and backup DHCP servers

To start the DHCP servers, refer to “Step 4: Start the DHCP server” on page 287.

Review, conclusions, and references

In this section we showed how to reduce the probability of having DHCP-related network access failure by adding a new DHCP server in the network.

We presented two ways of splitting the IP address pool across two DHCP servers.

First, if you are constrained by the IP addressing scheme, you can provide only a partial fall-back support, also called the 70/30 technique. This means that you split the existing address space across the DHCP servers. In this case, you can favor which DHCP server the DHCP client chooses by configuring more DHCP options in the configuration of the DHCP server. However, there are some DHCP clients that select the first DHCP server that responds, regardless the options offered.

Second, if you do not have any constraints on your IP addressing scheme, you can implement full-DHCP client support by enabling the DHCP servers to support all of the clients in the network. This is accomplished by increasing the IP address space. Using this method results in most of the IP addresses not being used unless one of the DHCP servers fails.

For more information about implementing multiple DHCP servers in a network, refer to *AS/400 TCP/IP Autoconfiguration: DNS and DHCP Support*, SG24-5147.

15.4 DHCP: multiple physical networks, logical networks, and DHCP servers

In this scenario we configure a multi-homed DHCP server to service clients from multiple physical networks.

Problem definition

This scenario is based on the scenario in “Option B: providing full DHCP client support” on page 316. In this scenario, we extend the customer’s network by adding a new physical subnet B, as you can see in Figure 15-70.

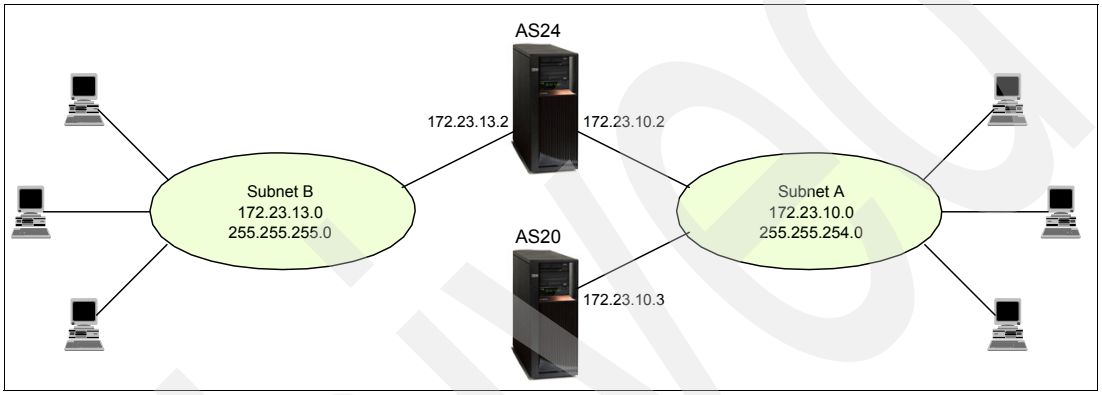


Figure 15-70 One DHCP server servicing multiple physical subnetworks

The IP network in subnet B is 172.23.13.0 and the subnet mask is 255.255.255.0.

The two subnets are connected through the System i AS24, which has a network interface in each subnet. The system AS24 will act as a router, routing IP packets from one network to another.

The clients in the new subnet must obtain the IP configuration dynamically from a DHCP server.

Solution definition

The DHCP servers on System i’s AS24 and AS20 are already configured to serve clients from subnet A.

However, when the System i DHCP server on AS24 receives a DHCPDISCOVER packet on interface 172.23.13.2, it verifies in its configuration whether there is a subnet that has the same network address as the interface on which the server receives the DHCPDISCOVER packet. If the DHCP server finds no subnet configured, it will record the request in the log but does not respond with a DHCP offer. So, in order to enable the DHCP server on AS24 to serve clients from subnet B, a new subnet must be added to the DHCP server configuration.

Because the clients must communicate across networks, the DHCP clients must be configured with a default gateway, which is the IP address of the AS24. Therefore, a new option must be added to the DHCP server configuration. This is option 3 - Router. To enable the server AS24 to route the IP packets, IP forwarding must be activated on this server.

Assumptions

The network configuration used in this scenario has the following characteristics:

- There are two physical networks, subnet A and subnet B, connected through the System i AS24. The network addresses of subnet A is 172.23.10.0 and the network mask is 255.255.254.0. The network address of subnet B is 172.23.13.0 and the network mask is 255.255.255.0.
- There are two System i's: AS24 and AS20. AS20 is located in subnet A and is configured as DHCP server for subnet 172.23.10.0.
- The System i AS24 has two network adapters, one in each physical subnet. It routes packets between subnet A and B, and is configured as a DHCP server for subnet A.

In this scenario, the DHCP server on AS24 is configured to serve DHCP clients in subnet 172.23.13.0. The DHCP holds all the address pool, 172.23.13.1 to 172.23.13.254. However, the address range 172.23.13.1 to 172.23.13.10 is reserved for other servers and is excluded from the addressing pool.

How-to

Because this scenario is based on the configuration in the scenario "Option B: providing full DHCP client support" we plan and perform only the modifications that occur in the network configuration.

To configure the DHCP server and the clients in this scenario, perform the following steps:

- Step 1: Add a new interface on System i AS24.
- Step 2: Activate IP forwarding on System i AS24.
- Step 3: Plan the new subnet configuration.
- Step 4: Add the new subnet to the AS24 DHCP configuration.
- Step 5: Add the new option to existing subnet on DHCP server AS24.
- Step 6: Add the new option to existing subnet on DHCP server AS20.
- Step 7: Configure the DHCP clients.
- Step 8: Start the DHCP server.
- Step 9: Test the configuration.

Step 1: Add a new interface on System i AS24

The new IP interface added on AS20 has the following characteristics:

IP interface 172.23.13.2
Network mask 255.255.255.0
Line description ETHLIN2

To configure the interface on a System i using iSeries Navigator, perform the following steps:

1. Start iSeries Navigator.
2. Expand your System i → **Network** → **TCP/IP Configuration** → **IPv4**.

3. Right-click **Interfaces** and select **New Interface** → **Local Area Network** from the context menu (Figure 15-71).

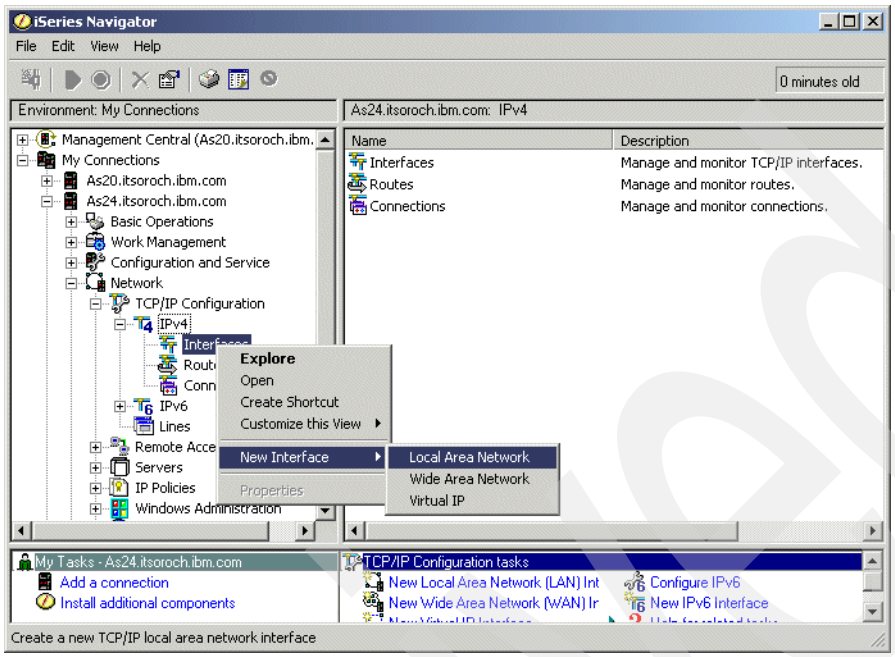


Figure 15-71 iSeries Navigator: adding a new IPv4 interface

4. The New IPv4 Interface wizard opens. Click **Next** in the Welcome window (Figure 15-72).

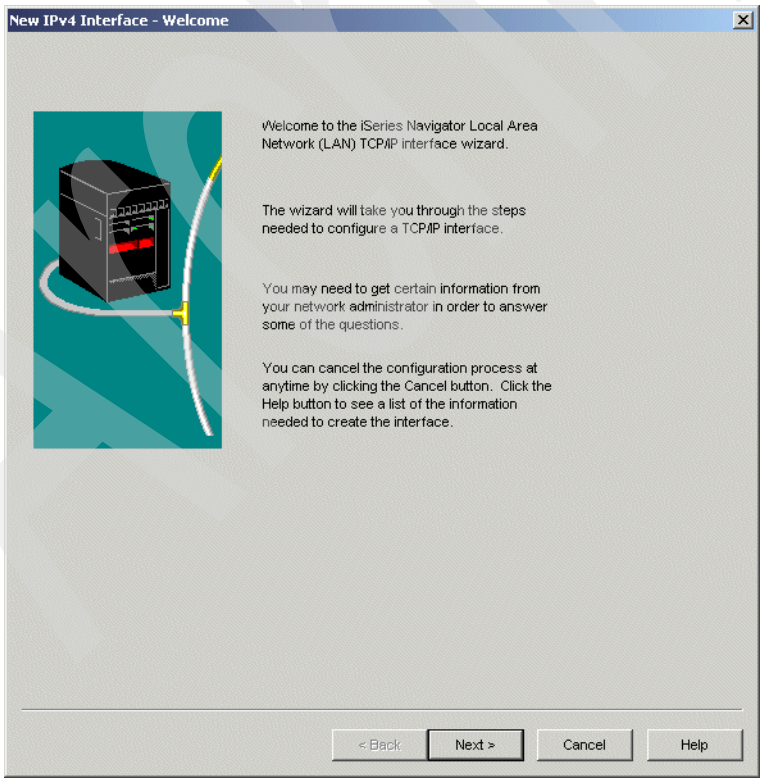


Figure 15-72 New IPv4 Interface wizard: Welcome window

5. In the Line Type window, select the type of line you want to configure the IP interface on (Figure 15-73). In our scenario, we use an Ethernet line. Click **Next**.

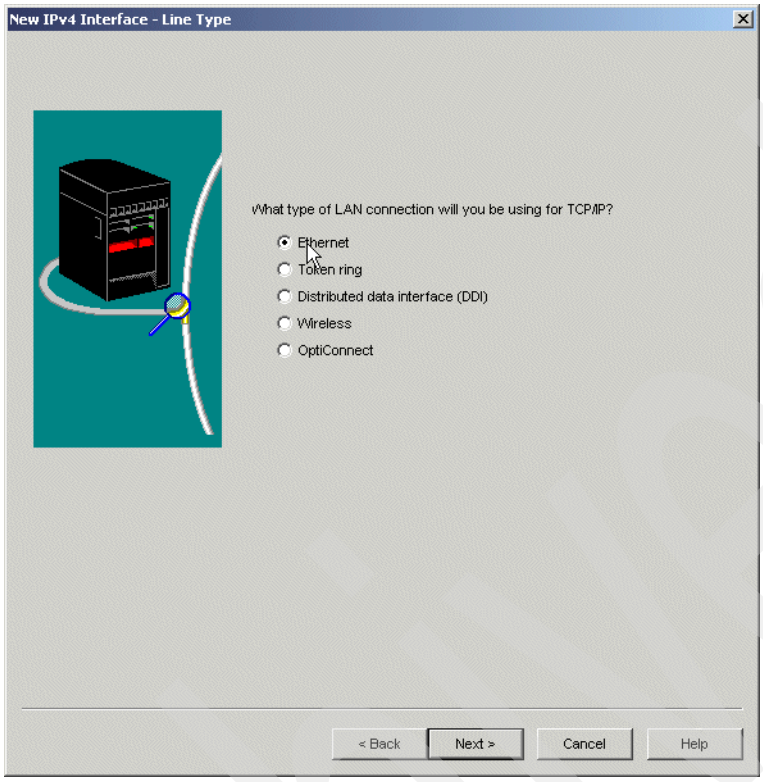


Figure 15-73 New IPv4 Interface wizard: Line Type window

6. In the Resource window, select **List by lines**. The list displays the line descriptions and the corresponding hardware resources. Select the line description where you want to install the new IP interface (Figure 15-74). Click **Next**.

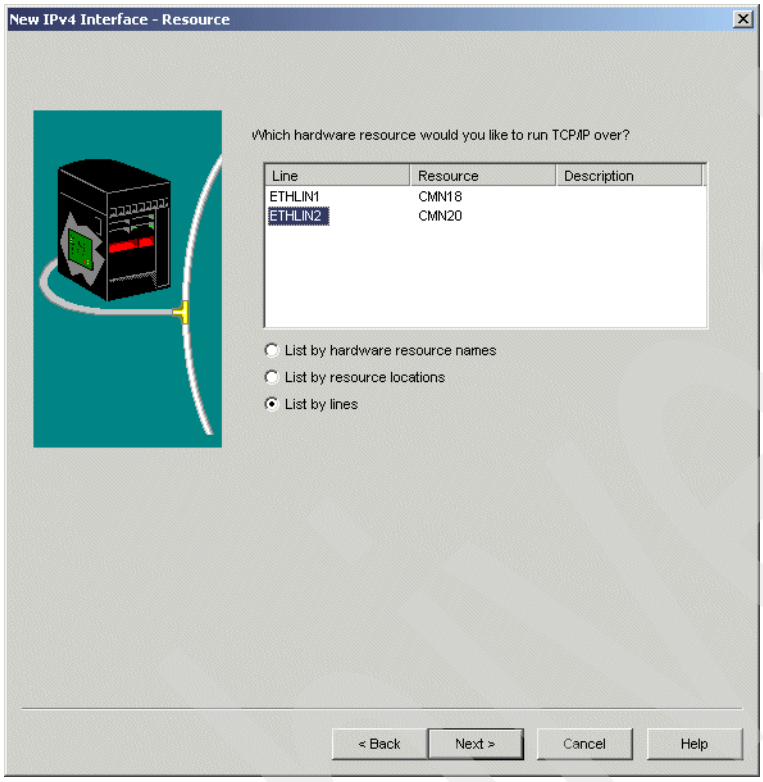


Figure 15-74 New IPv4 Interface wizard: Resource window

7. In the Settings window, specify the IP address of the interface, the IP interface description, and the subnet mask. Leave the default value for the Maximum transmission unit and specify **No** to the question “Do you want to work with TCP/IP settings that affect the entire system?” (Figure 15-75). Click **Next**.

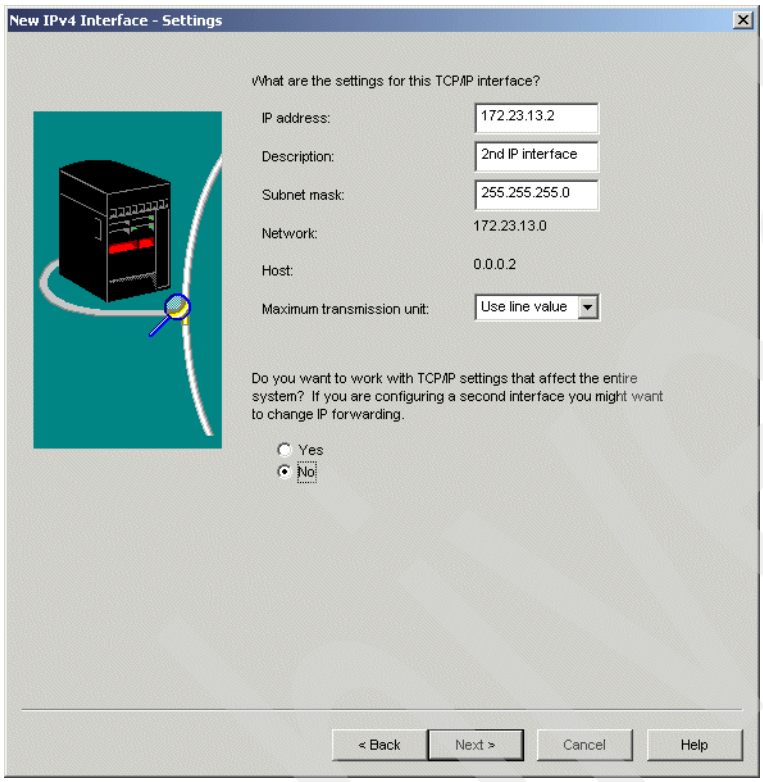


Figure 15-75 New IPv4 Interface wizard: Settings window

8. In the TCP/IP Routing window, click **Next** (Figure 15-76).

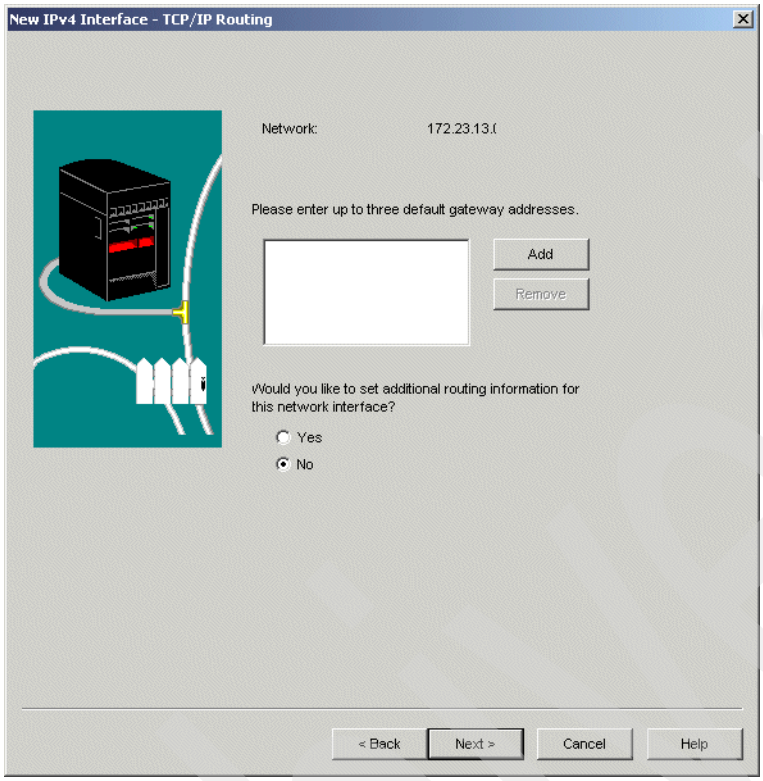


Figure 15-76 New IPv4 Interface wizard: TCP/IP Routing window

9. In the Start window, respond **Yes** to the question “Do you want to start this TCP/IP interface every time TCP/IP in started?” Respond **Yes** to the question “Do you want to start this interface now?” (Figure 15-77) Click **Next**.

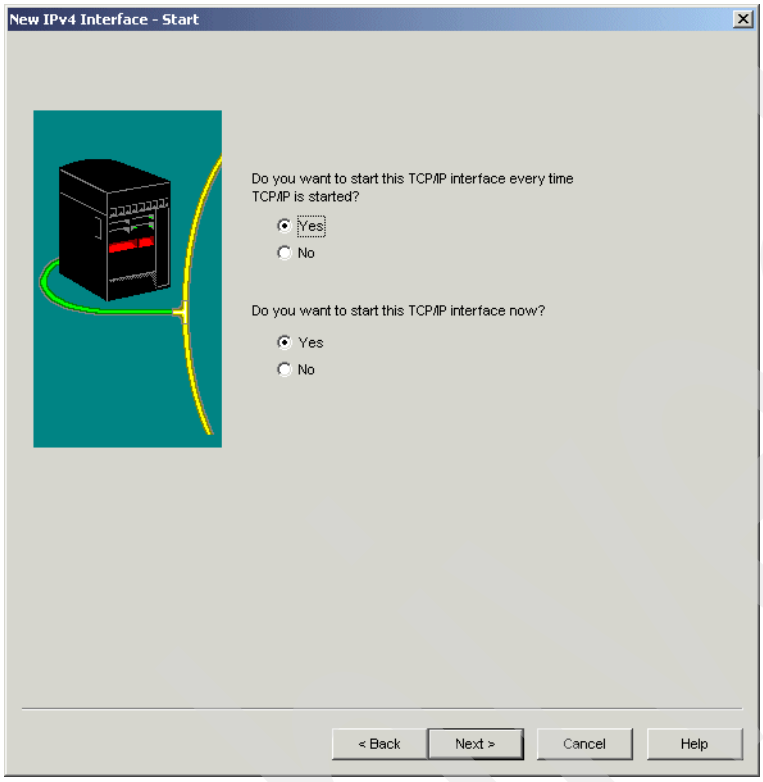


Figure 15-77 New IPv4 Interface wizard: Start window

10. The Summary window (Figure 15-78) presents all the options you specified. Click **Finish** to create the IP interface.

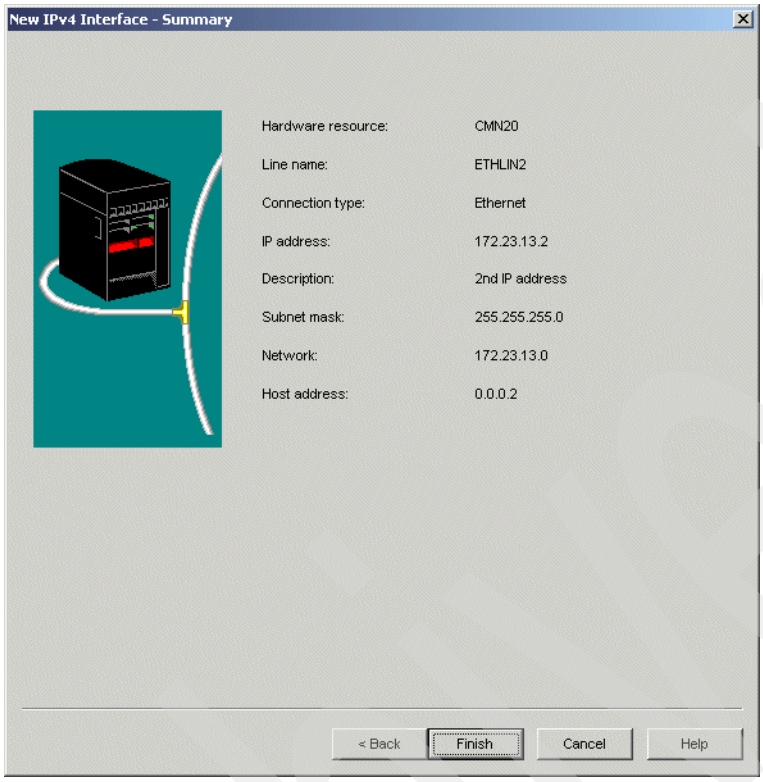


Figure 15-78 New IPv4 Interface wizard: Summary window

11. Because we choose to start the IP interface now, the Test TCP/IP interface is opened. Click **Test now**.

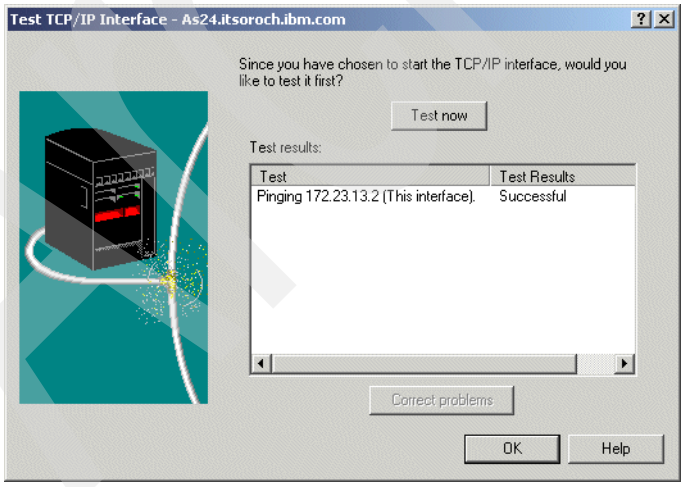


Figure 15-79 Test TCP/IP Interface

If the test result was successful, click **OK**. Otherwise, try to correct the problem by clicking **Correct problems** (Figure 15-79).

12. Figure 15-80 shows that the IP interface was added to the TCP/IP configuration and was activated.

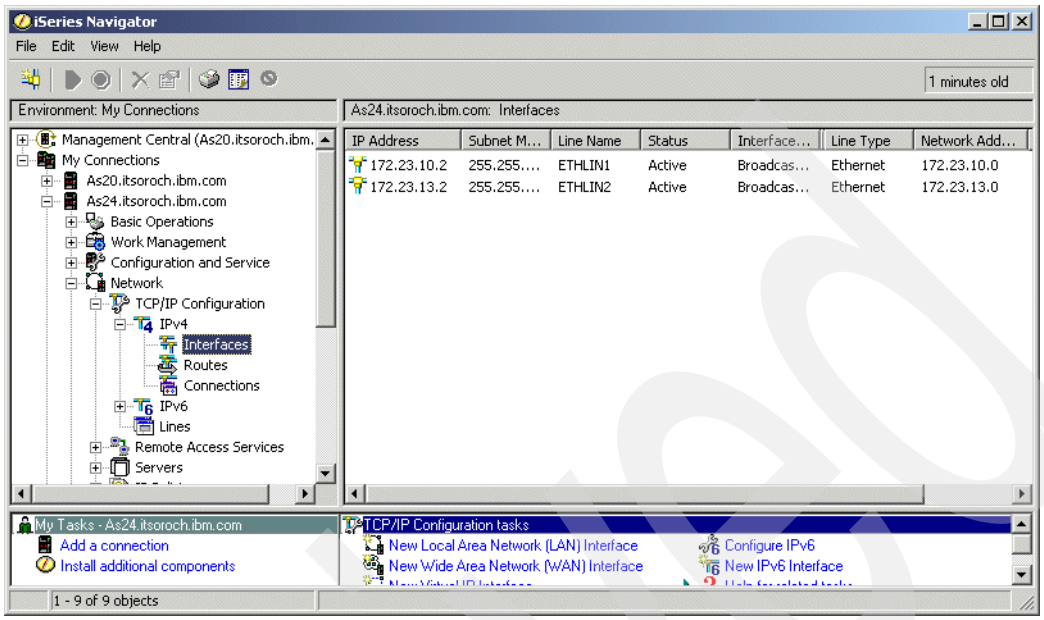


Figure 15-80 iSeries Navigator - TCP/IP interfaces

This ends the procedure.

Step 2: Activate IP forwarding on System i AS24

To activate the IP forwarding on System i AS24, perform the following steps:

1. Start iSeries Navigator.
2. Expand your System i → **Network** → **TCP/IP Configuration**.
3. Right-click **IPv4** and select **Properties** from the context menu (Figure 15-81).

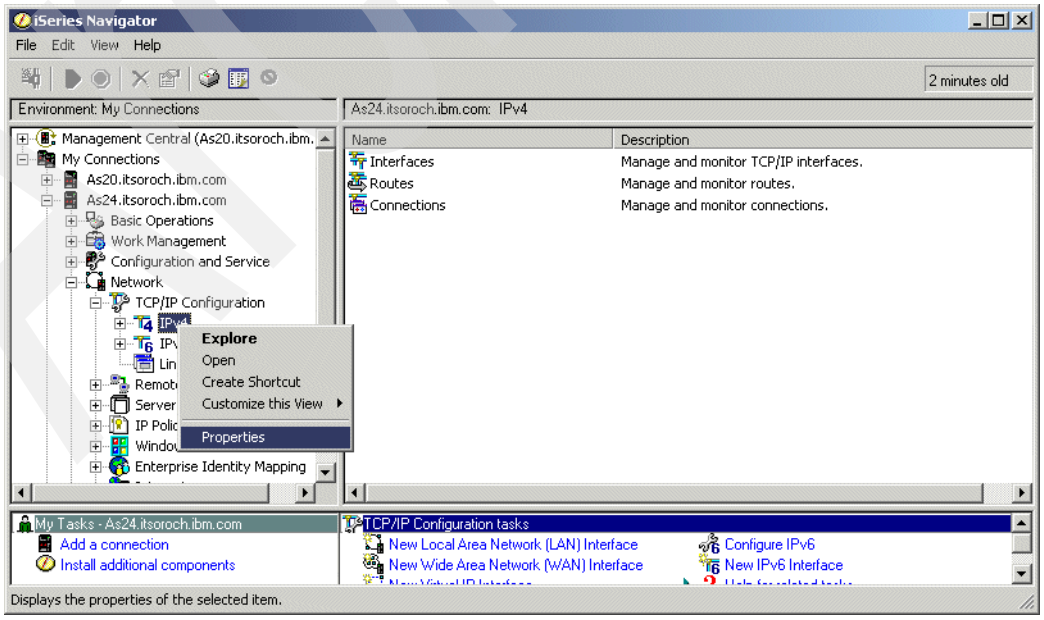


Figure 15-81 iSeries Navigator: open IPv4 properties

4. The IPv4 Properties window is displayed (Figure 15-82). Check the **IP datagram forwarding** check box. Click **OK** to save the modification. The modification takes place immediately.

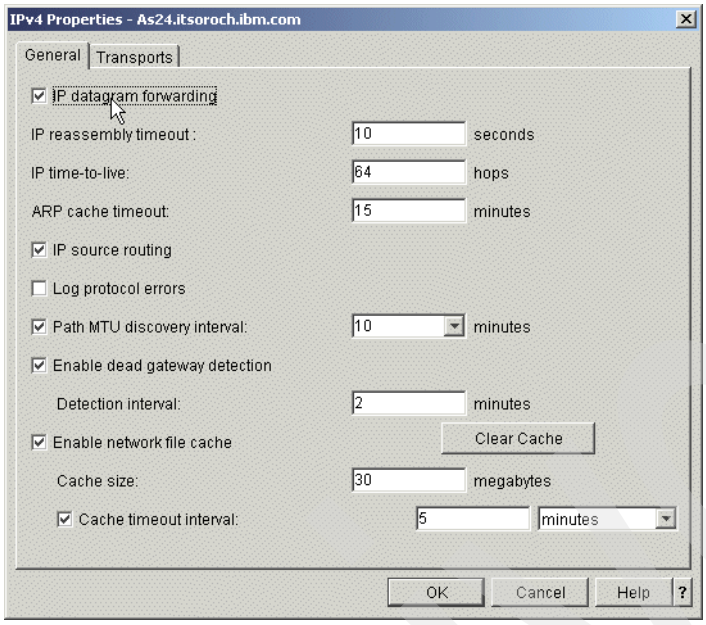


Figure 15-82 IPv4 Properties

This ends the procedure.

Step 3: Plan the new subnet configuration

We add a new subnet in the AS24 DHCP configuration to enable the DHCP server to serve the DHCP requests received on its new IP interface 172.23.13.2.

Table 15-10 shows the properties of this subnet and the place in iSeries Navigator where these properties can be set.

Table 15-10 Planning the DHCP server AS24: properties for subnet 172.23.13.0

Property	Value	Configuration reference
Subnet name	172.23.13.0	Subnet Properties → General
Subnet description	ITSO subnet 4	Subnet Properties → General
Subnet address	172.23.13.0	Subnet Properties → Address Pool → Subnet address
Subnet mask	255.255.255.0	Subnet Properties → Address Pool → Subnet mask
Address range for leasing	172.23.13.1 to 172.23.13.254	Subnet Properties → Address Pool → Range to assign
Lease time	Inherit from server (1 day)	Subnet Properties → Leases → Inherit lease time
IP addresses excluded from the address pool	Range 172.23.13.1 to 172.23.13.10	Subnet Properties → Address Pool → IP addresses excluded

Property	Value	Configuration reference
Options offered to DHCP clients 01 - Subnet mask 03 - Router 06 - DNS IP address 15 - Domain name	 255.255.255.0 172.23.13.2 172.23.13.1 itsoroch.ibm.com	Subnet Properties → Options

Step 4: Add the new subnet to the AS24 DHCP configuration

To add the new subnet to the AS24 DHCP configuration, perform the following steps using the values specified in Table 15-10 on page 332:

1. Start the iSeries Navigator.
2. Expand your System i → **Network** → **Servers** then click **TCP/IP**.
3. Double-click **DHCP**. This starts the DHCP server configuration window.
4. Right-click **Global** and select **New Subnet - Advanced** from the context menu.
5. The New Subnet Properties window opens. In the General tab, specify the Name and the Description of the subnet (Figure 15-83).

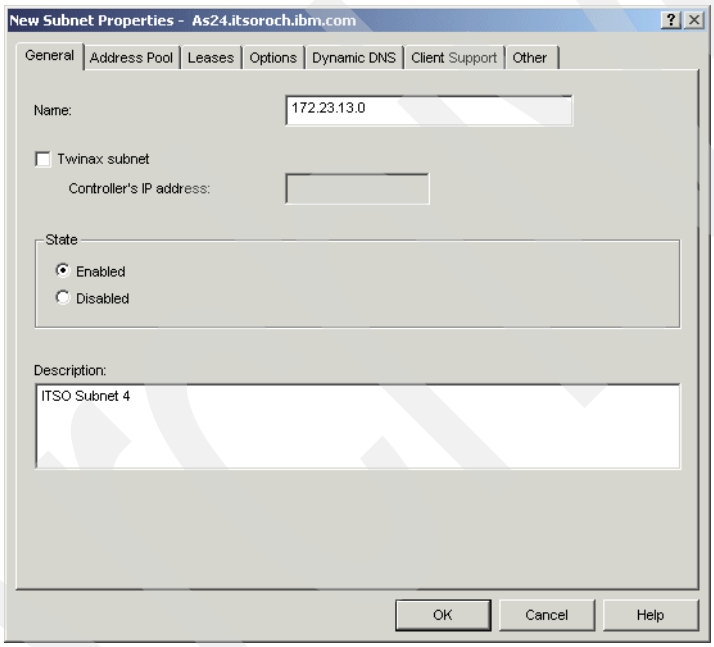


Figure 15-83 Configure subnet 172.23.13.0: General tab

6. Select the **Address Pool** tab. Specify the Subnet address and the Subnet mask. Specify the Address range of IP addresses excluded from the pool (Figure 15-84). Click **Add**.

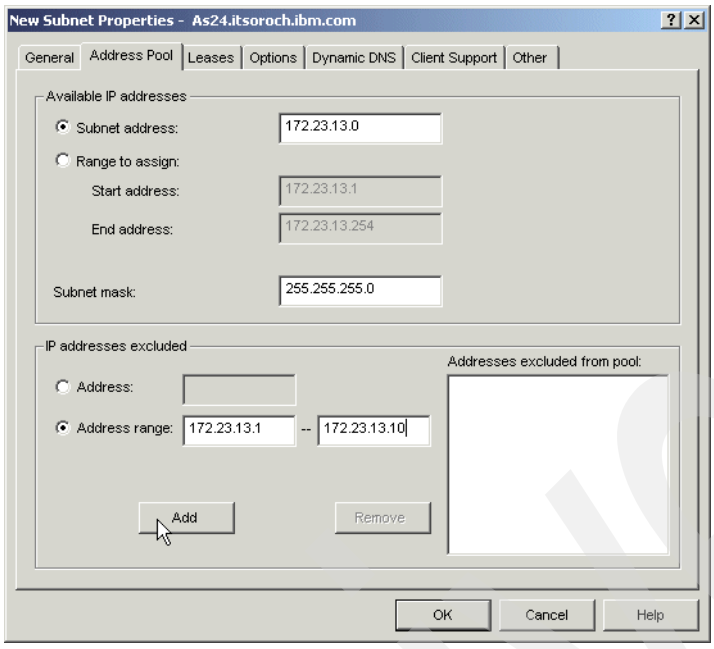


Figure 15-84 Configure subnet 172.23.13.0: Address Pool tab

7. Select the **Leases** tab. Select **Inherit lease time** (Figure 15-85). The lease time specified at the global level will be used for this subnet.

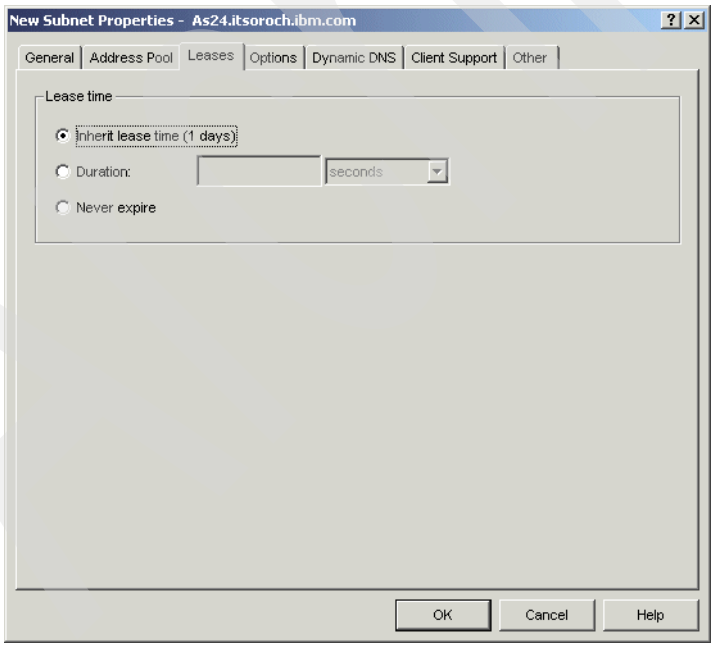


Figure 15-85 Configure subnet 172.23.13.0: Leases tab

8. Select the **Options** tab. From the Available options list, select option **1 - Subnet mask** and click **Add**. In the lower pane, specify the value of this option (Figure 15-86).

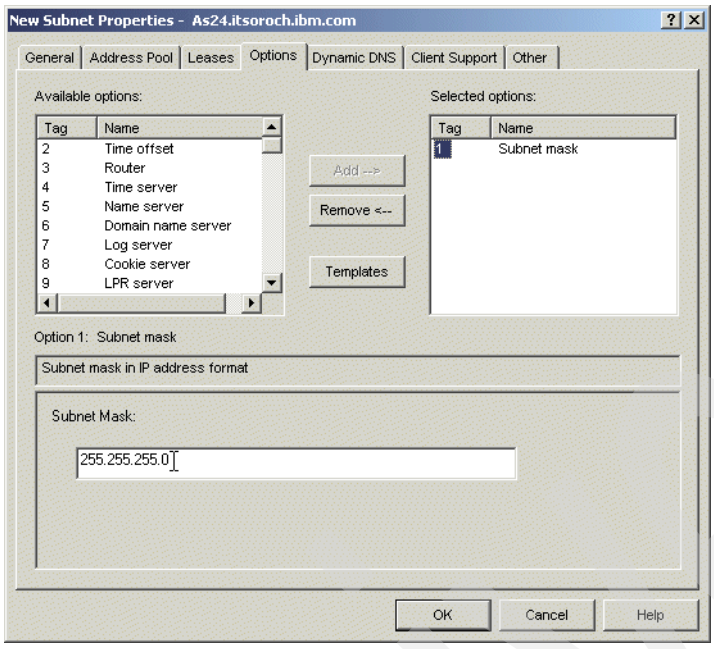


Figure 15-86 Configure subnet 172.23.13.0: Options tab (subnet mask)

9. From the Available options list, select option **3 - Router** and click **Add**. In the lower pane, click **Add**, specify the IP address of the router, and press Enter (Figure 15-87).

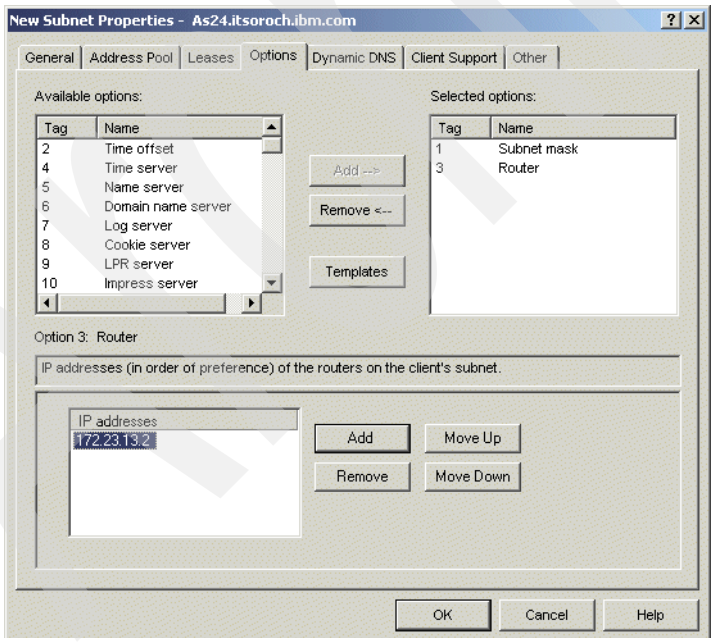


Figure 15-87 Configure subnet 172.23.13.0: Options tab (router)

10. From the Available options list, select option **6 - Domain name server** and click **Add**. In the lower pane, click **Add**, specify the IP address of the domain name server and press Enter (Figure 15-88).

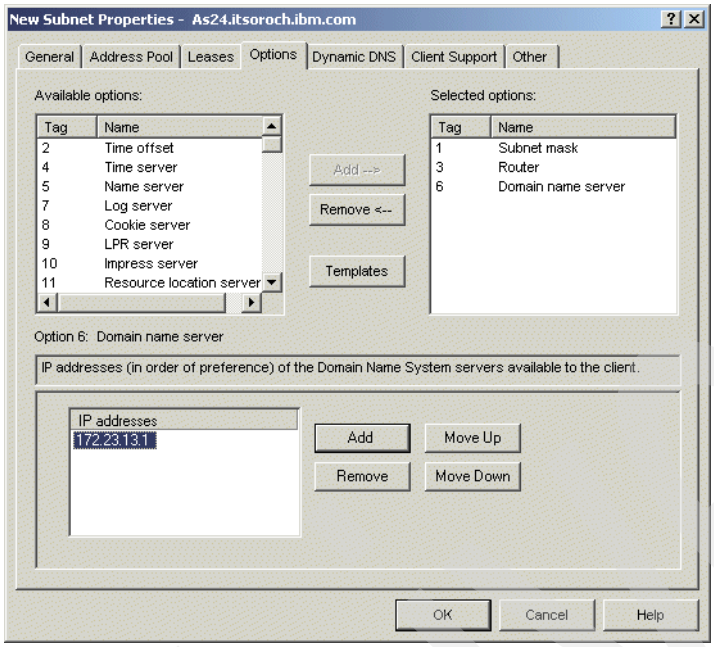


Figure 15-88 Configure subnet 172.23.13.0: Options tab (domain name server)

11. From the Available options list, select option **15 - Domain name** and click **Add**. In the lower pane, specify the domain name (Figure 15-89).

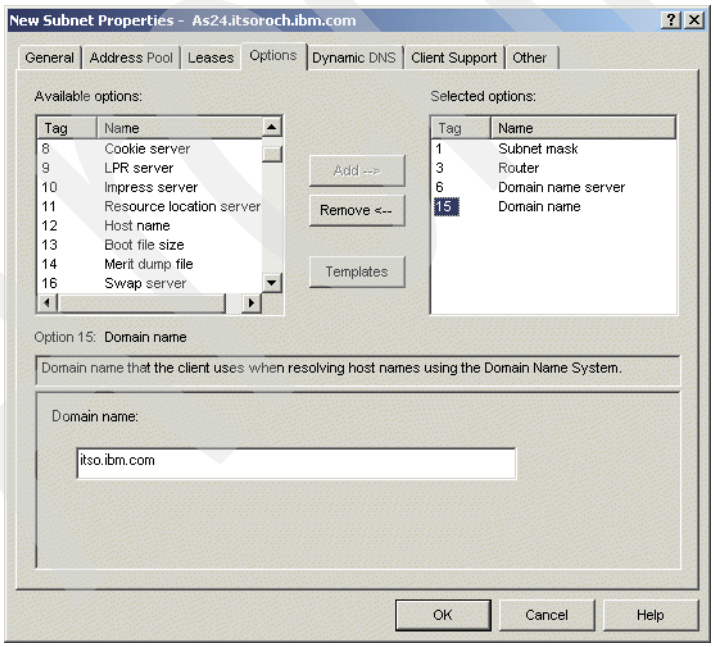


Figure 15-89 Configure subnet 172.23.13.0: Options tab (domain name)

12. Select the **Dynamic DNS** tab. Select **Inherited** (Updates not performed) under Update client records, and **Inherited** under Append domain name to host name (Figure 15-90).

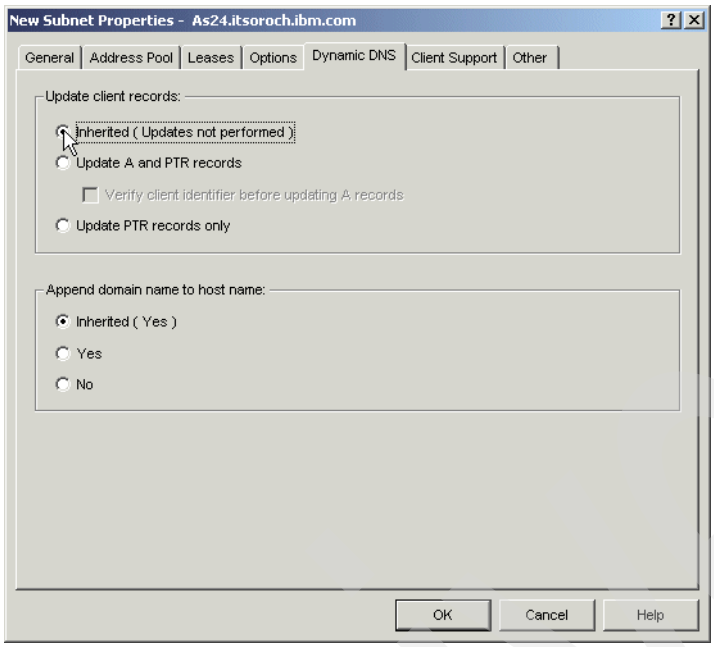


Figure 15-90 Configure subnet 172.23.13.0: Dynamic DNS tab

13. Select the **Client Support** tab. Select **Inherited** (Figure 15-91).

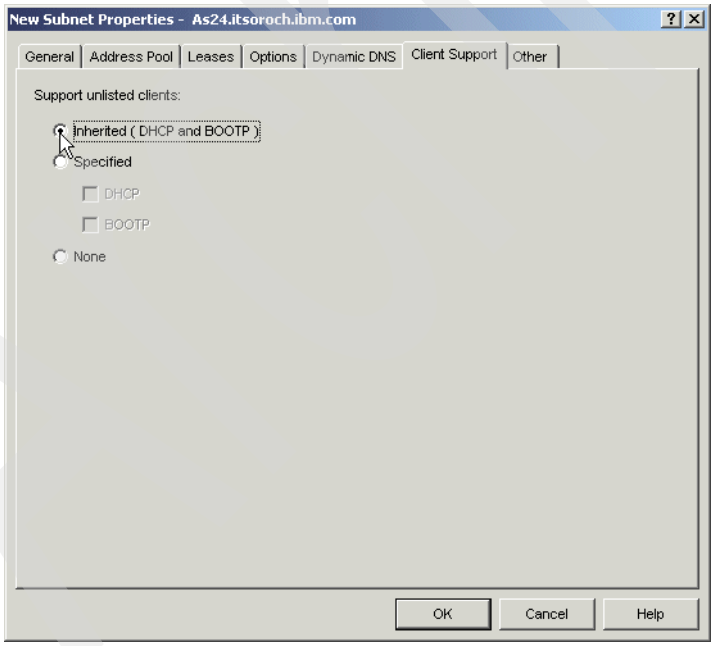


Figure 15-91 Configure subnet 172.23.13.0: Client Support tab

14. Select the **Other** tab. Select **Inherited** for the Bootstrap server (Figure 15-92).

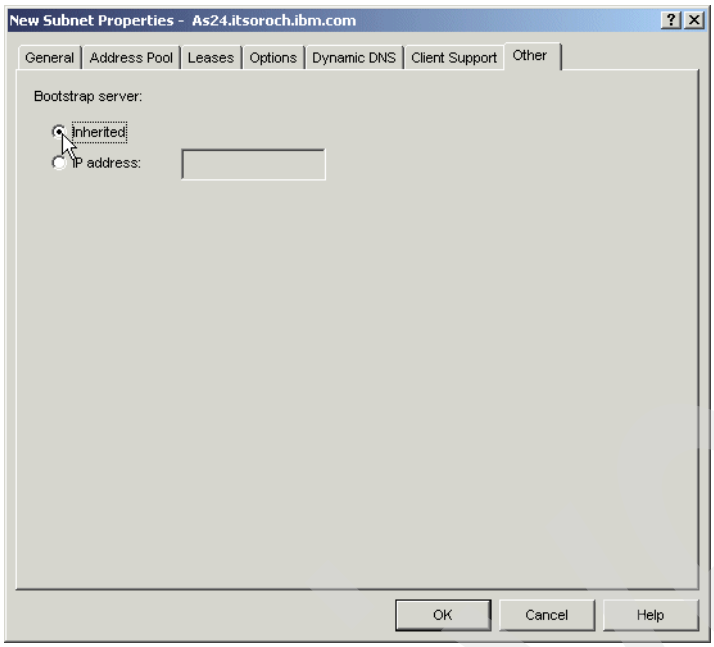


Figure 15-92 Configure subnet 172.23.13.0: Other tab

15. Click **OK** to create the new subnet.

As you can see in Figure 15-93, the subnet 172.23.13.0 has been added to the DHCP configuration.

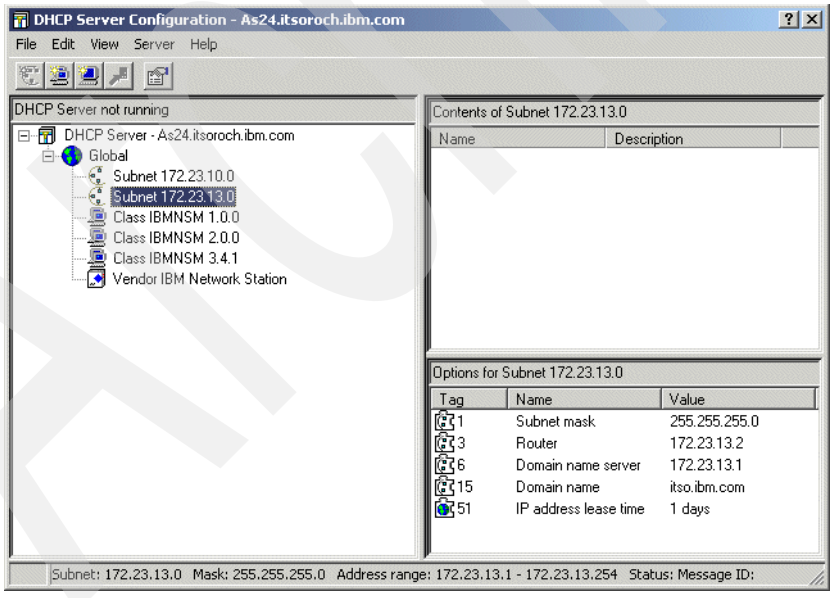


Figure 15-93 DHCP Configuration: the new created subnet

This ends the configuration of the new subnet.

Step 5: Add the new option to existing subnet on DHCP server AS24

The clients in subnet A must be able to communicate with the clients in subnet B. To do that, the DHCP clients in subnet A must be configured with a default gateway that points to the

AS24 IP interface 172.23.10.2. Therefore, option 3 - Router must be added to the Subnet 172.23.10.0 configuration on the DHCP servers AS24 and AS20.

To add option 3 - Router to subnet 172.23.10.0 in the AS24 DHCP configuration, perform the following steps:

1. Start the iSeries Navigator.
2. Expand **as24.itsoroch.ibm.com** → **Network** → **Servers** then click **TCP/IP**.
3. Double-click **DHCP**. This starts the DHCP server configuration window.
4. Right-click subnet **172.23.13.0** and select the **Properties** option from the context menu (Figure 15-94).

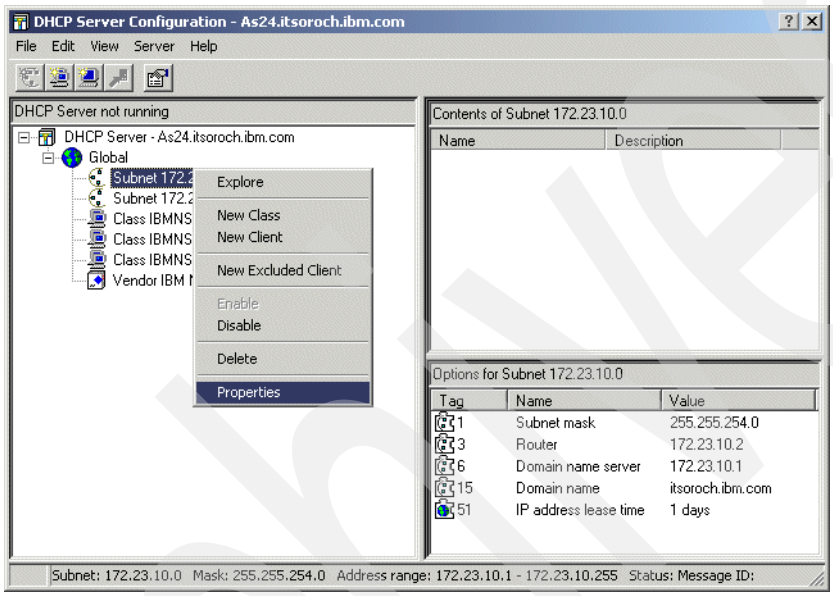


Figure 15-94 DHCP Server Configuration: selecting subnet properties

5. The Subnet Properties window is displayed. Select the **Options** Tab. From the Available options list, select option **3 - Router** and click **Add**. In the lower pane, click **Add**, specify the IP address of the router - `172.23.10.2`, and press Enter (Figure 15-95).

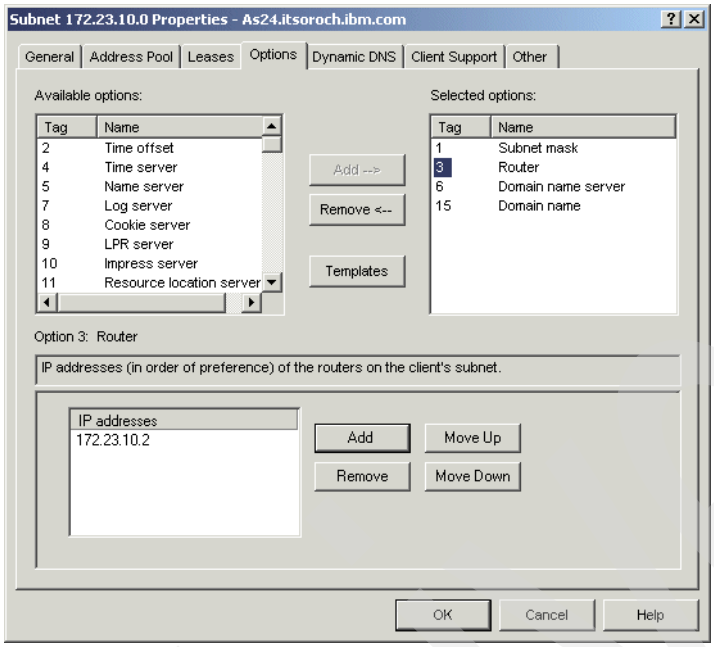


Figure 15-95 Configure subnet 172.23.10.0: Options tab (router)

6. Click **OK** to save the modifications.

This ends the procedure.

Step 6: Add the new option to existing subnet on DHCP server AS20

To add the option 3 - Router to subnet 172.23.10.0 in the DHCP configuration on AS20, perform the steps from 1 to 7 from "Step 5: Add the new option to existing subnet on DHCP server AS24" on page 338 on AS20.

Step 7: Configure the DHCP clients

To configure the DHCP clients, refer to "Step 5: Configure your Windows 2000 DHCP client" on page 288.

Step 8: Start the DHCP server

To start the DHCP server AS24, refer to "Step 4: Start the DHCP server" on page 287.

Step 9: Test the configuration

In this scenario, we use two DHCP clients to test the configuration. The DHCP client Client1 is located in subnet A, and Client2 is located in subnet B (Figure 15-70 on page 322). To test the configuration implemented in this scenario, perform the following steps:

1. Power on Client1. During the boot of this PC the DHCP client will request an IP address from the System i DHCP server.
2. After the workstation starts, open a command prompt panel and run this command:

```
C:\> ipconfig /all
```

3. The IP configuration of Client1 is displayed on the command prompt panel. The configuration in Figure 15-96 shows that the client obtained the IP address from the subnet 172.23.10.0 configured on the DHCP server AS24. Also, you can see that the default gateway was configured on the client. The value of the default gateway was taken from the DHCP option 3 - router.

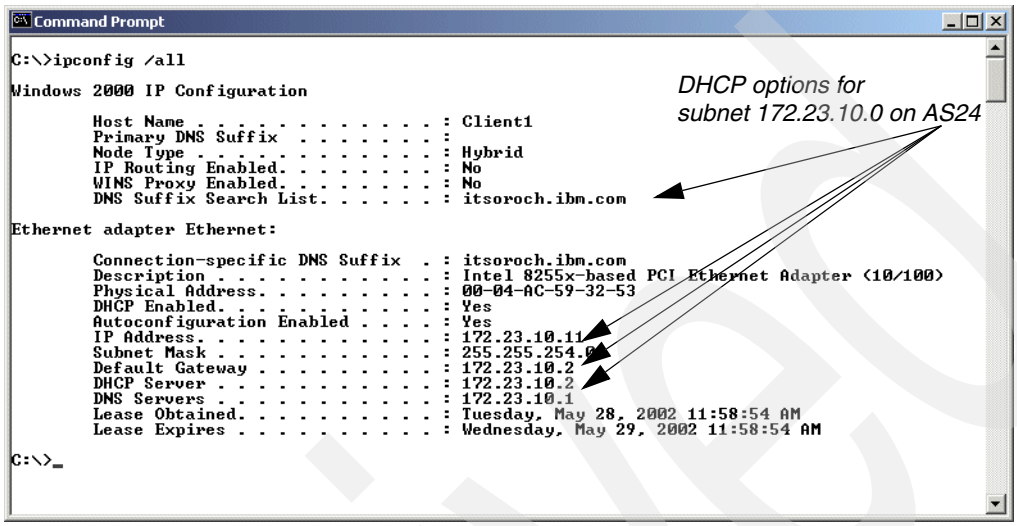


Figure 15-96 Client1 IP configuration after IP address allocation

4. Power on the DHCP client Client2. During the boot of this PC the DHCP client will request an IP address from the System i DHCP server.
5. After the workstation starts, open a command prompt panel and run this command:

```
C:\> ipconfig /all
```

6. The IP configuration of Client2 is displayed on the command prompt panel. The configuration in Figure 15-97 shows that the client obtained the IP address from the subnet 172.23.13.0 configured on the DHCP server AS24.

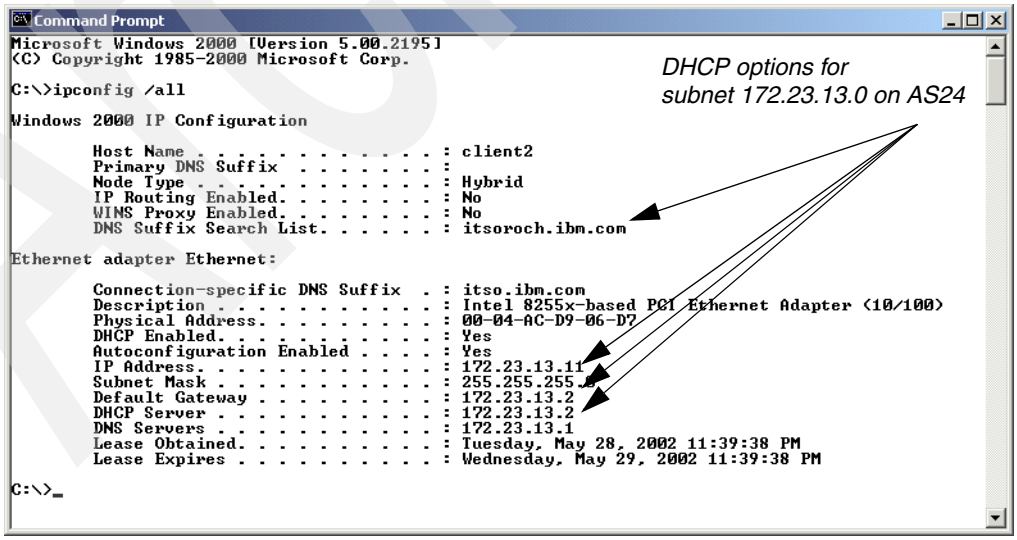


Figure 15-97 Client2 IP configuration after IP address allocation

On System i AS24:

1. Start iSeries Navigator.
2. Expand **as24.itsoroch.ibm.com** → **Network** → **Servers** then click **TCP/IP**.
3. Right-click the DHCP and select **Monitor** from the context menu. The DHCP Monitor window is displayed.
4. Expand **Subnets**. The two subnets that were configured on the DHCP server AS24 are displayed. Select subnet **172.23.10.0**. The right pane displays all of the IP addresses of this subnet. Figure 15-98 shows that the IP address 172.23.10.11 was leased to Client1.

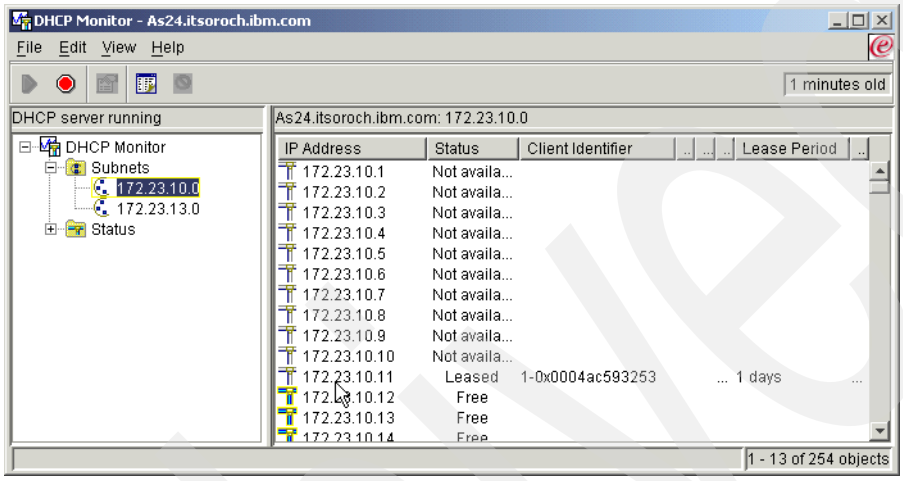


Figure 15-98 DHCP Monitor: subnet 172.23.10.0

5. Select subnet **172.23.13.0**. As you can see in Figure 15-99, the IP address 172.23.13.11 was leased to Client2.

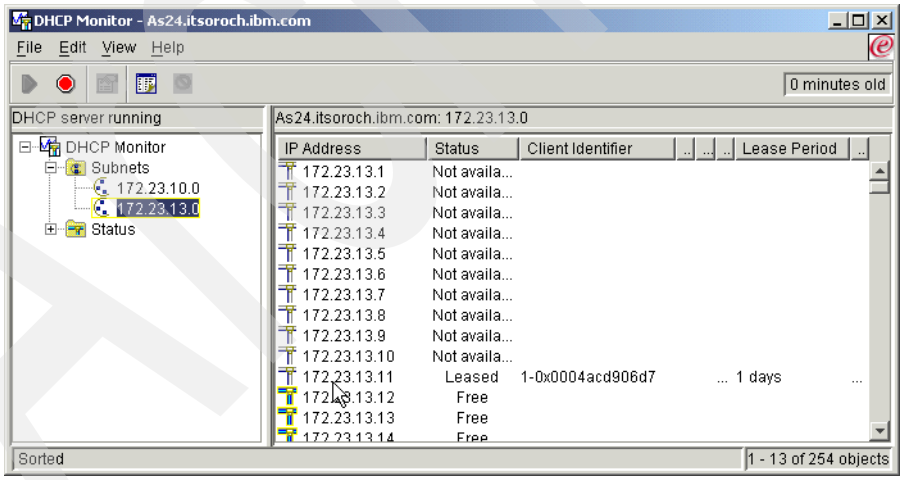


Figure 15-99 DHCP Monitor: subnet 172.23.13.0

This ends the DHCP configuration test.

Review, conclusions and references

In this scenario, we configured a multi-homed DHCP server on System i. The DHCP server listens for DHCP messages from clients on all of its interfaces, but it responds to DHCP requests only on the interfaces for which it has an address pool configured.

To enable a DHCP server to respond to DHCP messages on a certain IP interface, you must add to the DHCP configuration a subnet that has the same network address as the IP interface.

Finally, we showed how to test the DHCP configuration.

15.5 DHCP: multiple physical, logical networks, and DHCP servers using Relay Agents

In this scenario we configure the BOOTP/DHCP Relay Agent on System i and show how it works with the System i DHCP server. Also, we show how to configure the DHCP Relay Agent on a Windows 2000 system.

Problem definition

This scenario is based on the scenario in 15.4, “DHCP: multiple physical networks, logical networks, and DHCP servers” on page 322, in which two System i’s were configured as DHCP servers to serve clients from two subnets: subnet A and subnet B. A new subnet is added to the configuration. Figure 15-100 shows that the new subnet, C, is connected to subnet A through a router. To keep the IP configuration centralized, the clients in network C should be able to request an IP address from server AS24.

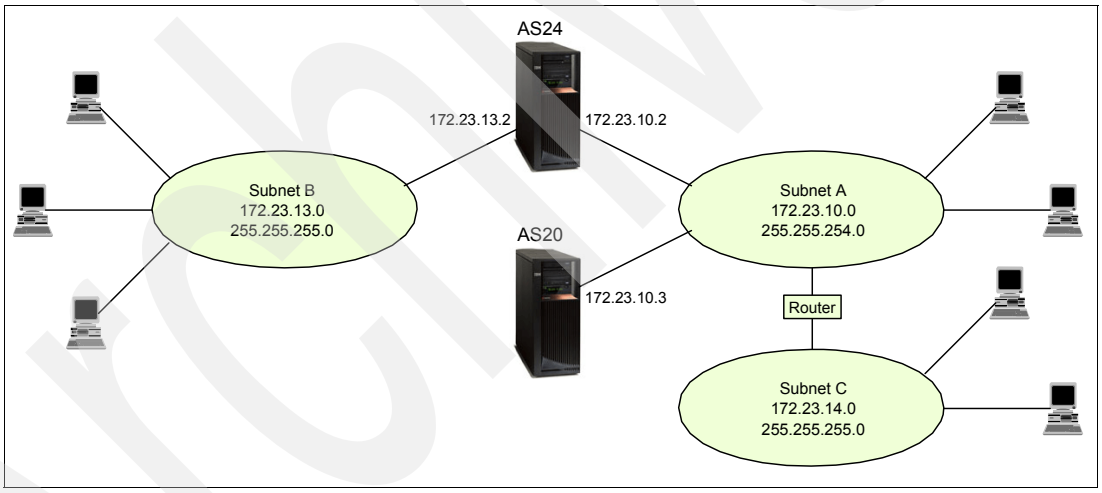


Figure 15-100 Multiple physical networks, multiple logical networks, multiple DHCP servers

When a DHCP client requests an IP address, it sends a DHCPDISCOVER packet (a broadcast packet). The TCP/IP architecture does not allow the broadcast messages to leave their own subnet. That means that the DHCPDISCOVER messages that are sent by the DHCP clients from subnet C do not reach the DHCP server AS24, because the router will discard this packet.

Solution definition

To solve this problem, a BOOTP/DHCP Relay Agent must be implemented in subnet C.

The BOOTP/DHCP Relay Agent intercepts the DHCPDISCOVER packets sent by the DHCP clients and unicast the packets to a destination specified in its configuration (Figure 15-101).

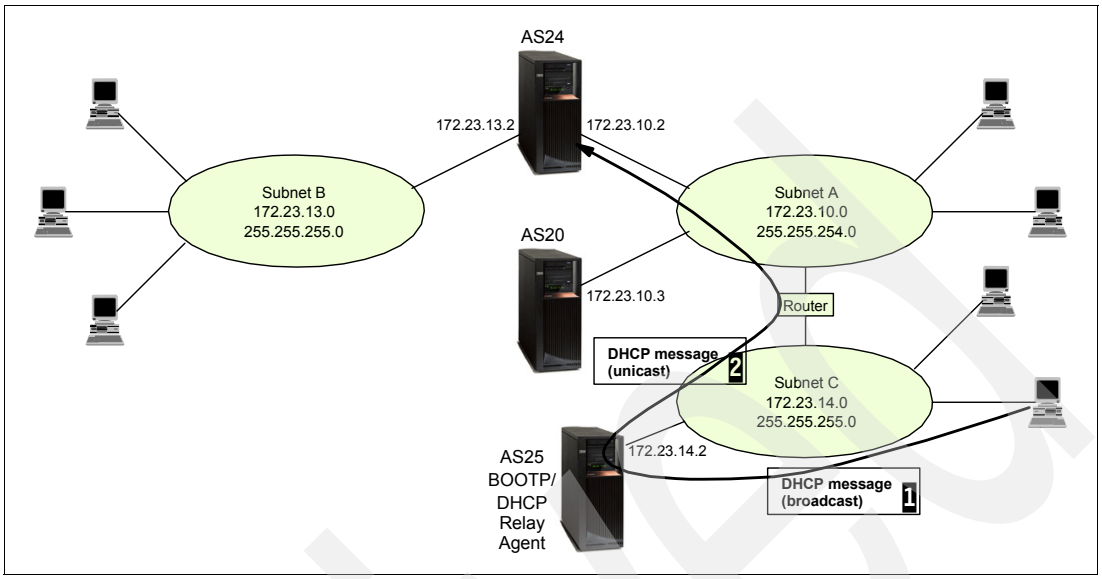


Figure 15-101 Multiple physical networks, multiple logical networks, multiple DHCP servers, BOOTP/DHCP relay agents

When the BOOTP/DHCP Relay Agent receives a DHCPDISCOVER packet, first it verifies whether the field RELAY AGENT in the DHCPDISCOVER packet is zero. If the field is zero, the BOOTP/DHCP Relay Agent writes its IP address in this field and then forwards the package to a destination specified in its configuration. The destination can be a DHCP server or another DHCP relay agent.

When the DHCP server receives the DHCPDISCOVER packet, it verifies the RELAY AGENT field. If the field is not zero, the server uses this value to choose the address pool from which to offer an IP address to the client. Then it responds to the BOOTP/DHCP Relay Agent with a DHCPOFFER packet, and the BOOTP/DHCP Relay Agent broadcasts the packet in the subnet so that the DHCP client can receive the packet.

The BOOTP/DHCP Relay Agents clients can be configured on different types of systems. In this scenario we show how to configure the BOOTP/DHCP Relay Agents on System i and on Windows 2000 server.

In this scenario the DHCP server on AS24 is configured to serve the DHCP clients in subnet C. The DHCP server holds all of the IP addresses from 172.23.14.1 to 172.23.14.254. However, the address range 172.23.14.1 to 172.23.14.10 is reserved for future uses and is excluded from the addressing pool. The server will provide the clients with the following options related to IP:

IP address	As assigned from available IP address in the pool
Network mask	255.255.255.0
IP address of the DNS	172.23.14.1
Router	172.23.14.10
Domain name	itsoroch.ibm.com

Also, a BOOTP/DHCP Relay Agent is configured in subnet C to enable DHCP clients to request IP addresses from server AS24. The IP interface of the BOOTP/DHCP Relay Agent is 172.23.14.2.

To enable the System i to communicate with the BOOTP/DHCP Relay Agent in subnet C, a new route must be defined on the server.

Assumptions

The network presented in Figure 15-101 on page 344 has the following characteristics:

- There are three physical subnets in the network and two System i's. The DHCP servers on AS24 and AS20 are already configured to serve the clients in subnet A and subnet B.
- Subnet A and subnet B are connected through System i AS24. AS24 routes the IP packets between these two subnets.
- Subnet C is connected to subnet A through a router. The router routes the IP packets between subnet A and subnet C. The network address of subnet C is 172.23.14.0. The network mask is 255.255.255.0, which allows up to 254 clients in the subnet.
- DHCP servers AS24 and AS20 are already configured to serve subnet A and subnet B.

How-to

Because this scenario is based on the configuration in the scenario in 15.4, "DHCP: multiple physical networks, logical networks, and DHCP servers" on page 322, we plan and perform only the modifications that occur in the network configuration.

To implement this configuration, perform the following steps:

- Step 1: Add a route on server AS24.
- Step 2: Plan the new subnet.
- Step 3: Add the new subnet to the AS24 DHCP server configuration.
- Step 4: Configure the BOOTP/DHCP Relay Agent:
  - Option 1: Configure the System i BOOTP/DHCP Relay Agent.
  - Option 2: Configure the BOOTP/DHCP Relay Agent on Windows 2000 Server.
- Step 5: Configure the DHCP clients in subnet C.
- Step 6: Start the DHCP server AS24.
- Step 7: Start the BOOTP/DHCP Relay Agent:
  - Option 1: Start the BOOTP/DHCP Relay Agent on System i.
  - Option 2: Start the DHCP Relay Agent on Windows 2000 Server.
- Step 8: Test the configuration.

Step 1: Add a route on server AS24

To add a route on System i AS24 using iSeries Navigator, perform the following steps:

1. Start iSeries Navigator.
2. Expand your System i → **Network** → **TCP/IP Configuration** → **IPv4**.

3. Right-click **Routes** and select **New Route** from the context menu (Figure 15-102).

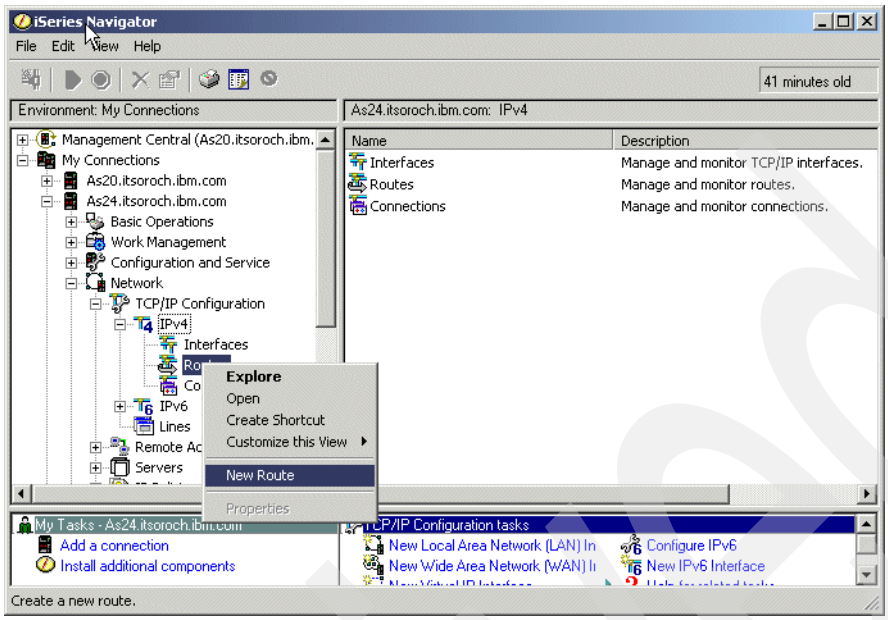


Figure 15-102 iSeries Navigator: adding a new IP route

4. The New IPv4 Route wizard is opened (Figure 15-103). In the Welcome window, click **Next**.

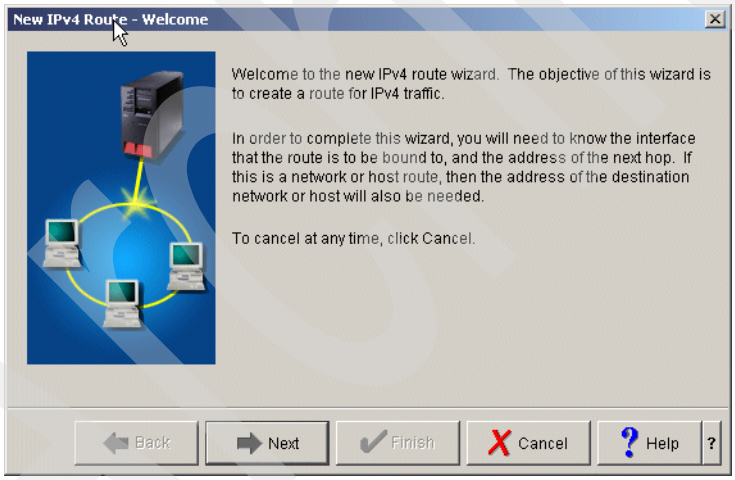


Figure 15-103 New IPv4 Route wizard: Welcome window

5. In the Binding Interface window, select **Yes**. From the Binding interface list, select the IP interface to which you want to bind this route (Figure 15-104). Click **Next**.

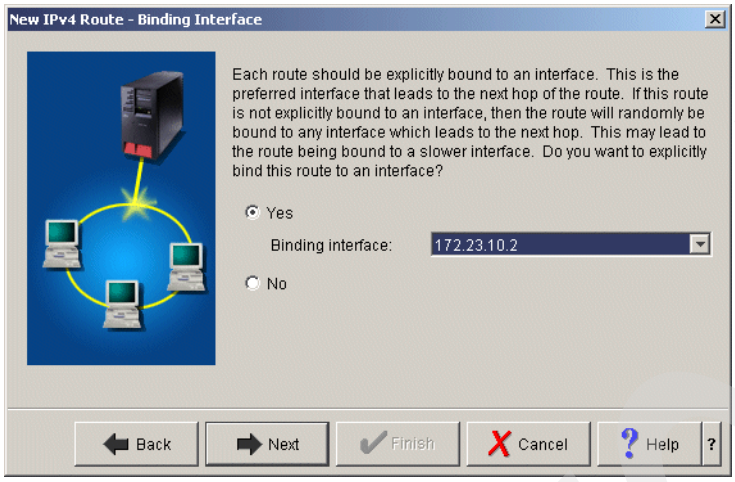


Figure 15-104 New IPv4 Route wizard: Binding Interface window

6. In the Attributes window, select **Network route**. Specify Destination address, Destination subnet mask, and Next hop address (Figure 15-105). Click **Next**.

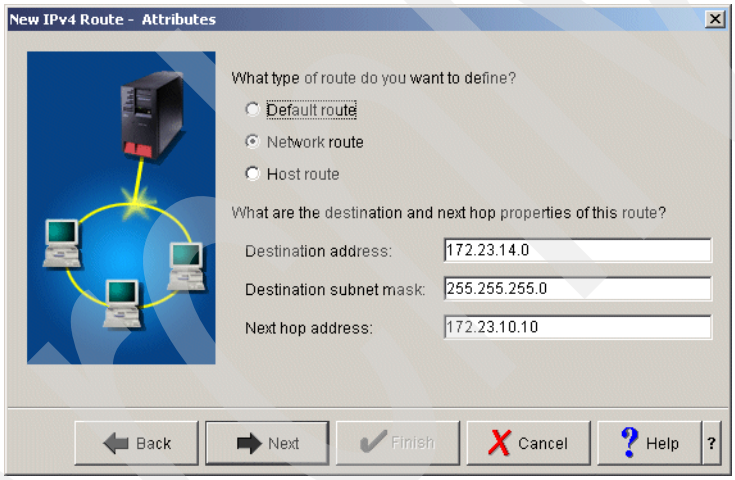


Figure 15-105 New IPv4 Route wizard: Attributes window

7. The Summary window presents the options you specified previously (Figure 15-106). Click **Finish** to create the route.

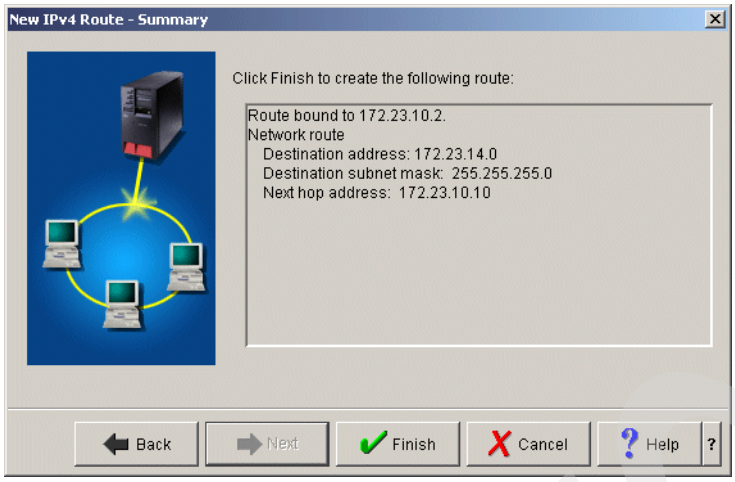


Figure 15-106 New IPv4 Route wizard: Summary window

This ends the procedure.

Step 2: Plan the new subnet

A new subnet must be added to the AS24 DHCP configuration to enable the DHCP server to service clients from subnet C.

The properties of the new subnet 172.23.14.0 and the place in iSeries Navigator where these properties can be set are presented in Table 15-11.

Table 15-11 Planning the DHCP server AS24: properties for subnet 172.23.14.0

Property	Value	Configuration reference
Subnet name	172.23.14.0	Subnet Properties → General
Subnet description	ITSO subnet 5	Subnet Properties → General
Subnet address	172.23.14.0	Subnet Properties → Address Pool → Subnet address
Subnet mask	255.255.255.0	Subnet Properties → Address Pool → Subnet mask
Address range for leasing	172.23.14.1 to 172.23.14.254	Subnet Properties → Address Pool → Range to assign
Lease time	Inherit from server (1 day)	Subnet Properties → Leases → Inherit lease time
IP addresses excluded from the address pool	Range 172.23.14.1 to 172.23.14.10	Subnet Properties → Address Pool → IP addresses excluded
Options offered to DHCP clients 01 - Subnet mask 03 - Router 06 - DNS IP address 15 - Domain name	 255.255.255.0 172.23.14.10 172.23.14.1 itsoroch.ibm.com	Subnet Properties → Options

Step 3: Add the new subnet to the AS24 DHCP server configuration

To add subnet 172.23.14.0 to AS24 DHCP configuration, perform the following steps using the values specified in Table 15-11 on page 348:

1. Start the iSeries Navigator.
2. Expand your System i → **Network** → **Servers** then click **TCP/IP**.
3. Double-click **DHCP**. This starts the DHCP server configuration window.
4. Right-click **Global** and select **New Subnet - Advanced** from the context menu.
5. The New Subnet Properties window is displayed. In the General tab, specify the name and the description of the subnet (Figure 15-107).

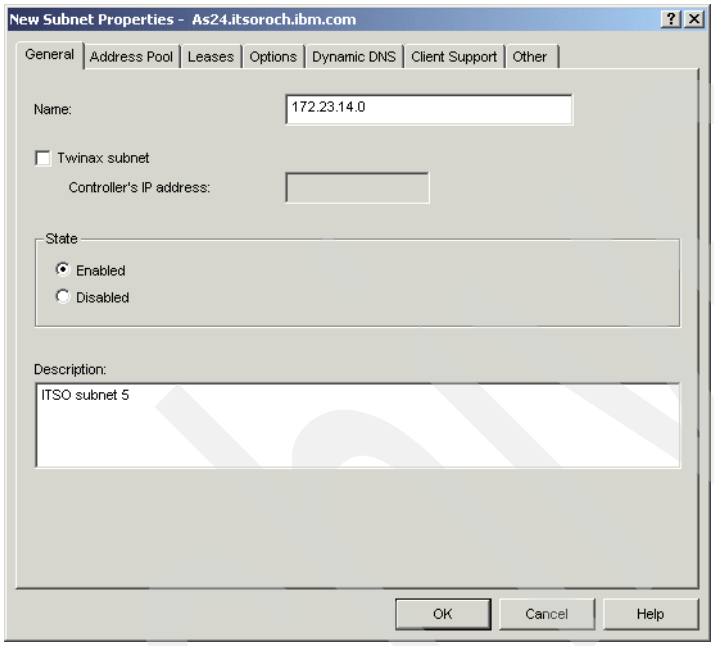


Figure 15-107 Configure subnet 172.23.14.0: General tab

6. Select the **Address Pool** tab. Specify the Subnet address and the Subnet mask. Specify the Address range of IP addresses excluded from the pool (Figure 15-108). Click **Add**.

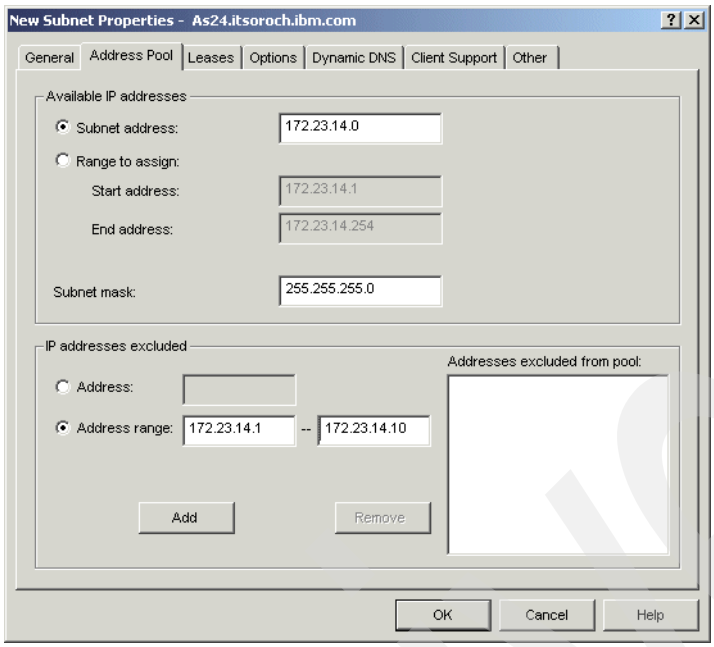


Figure 15-108 Configure subnet 172.23.14.0: Address Pool tab

7. Select the **Leases** tab. Select **Inherit lease time** (Figure 15-109). The lease time specified at the global level will be used for this subnet.

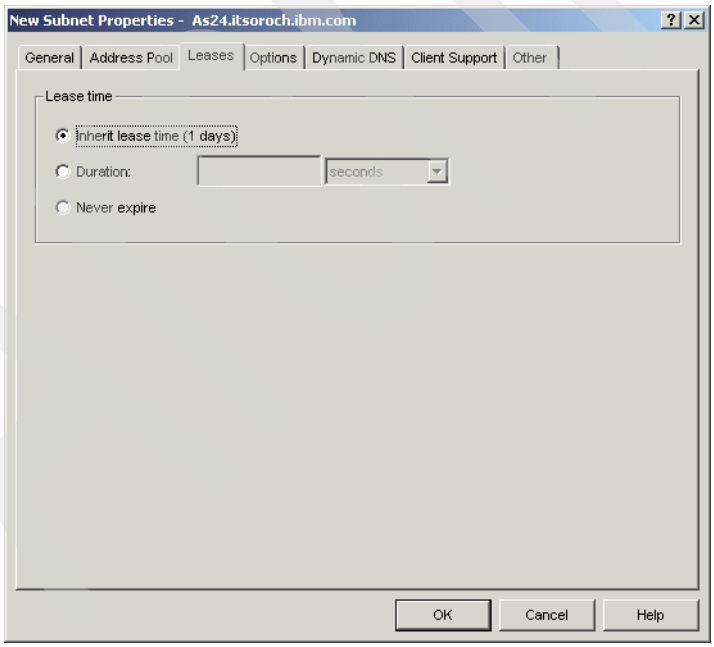


Figure 15-109 Configure subnet 172.23.14.0: Leases tab

8. Select the **Options** tab. From the Available options list, select option **1 - Subnet mask** and click **Add**. In the lower pane, specify the value of this option (Figure 15-110).

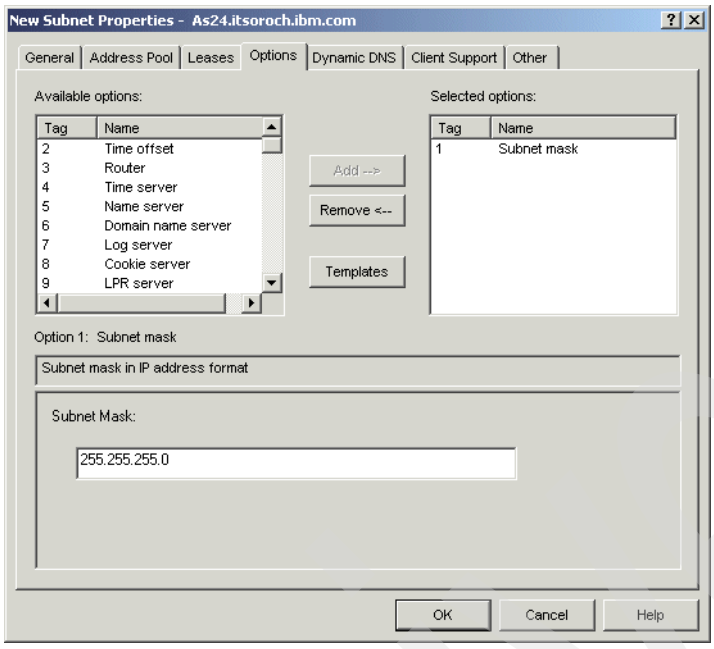


Figure 15-110 Configure subnet 172.23.14.0: Options tab (subnet mask)

9. From the Available options list, select option **3 - Router** and click **Add**. In the lower pane, click **Add**. Specify the IP address of the router and press Enter (Figure 15-111).

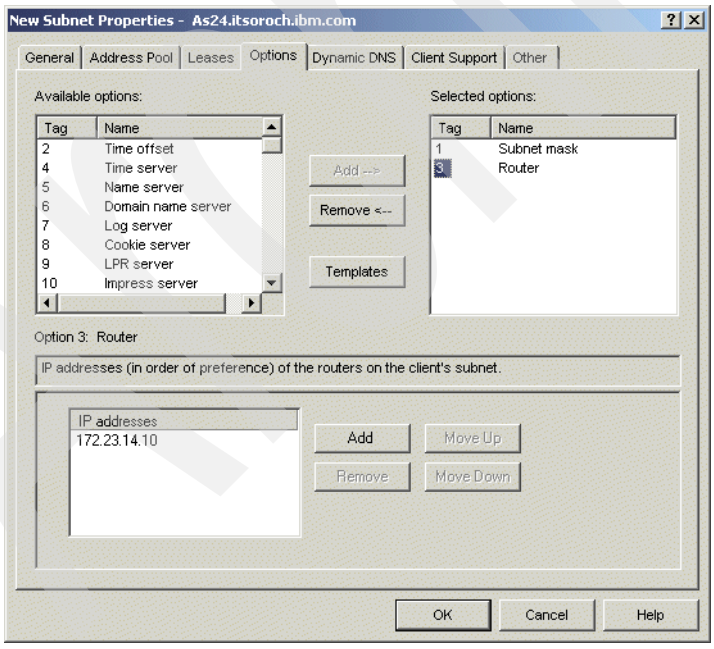


Figure 15-111 Configure subnet 172.23.14.0: Options tab (router)

10. From the Available options list, select option **6 - Domain name server** and click **Add**. In the lower pane, click **Add**. Specify the IP address of the domain name server and press Enter (Figure 15-112).

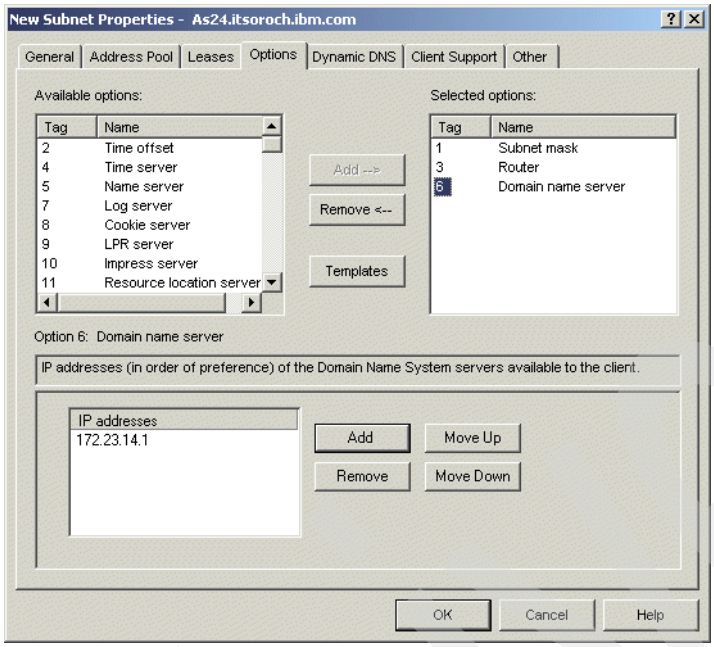


Figure 15-112 Configure subnet 172.23.14.0: Options tab (domain name server)

11. From the Available options list, select option **15 - Domain name** and click **Add**. In the lower pane, specify the domain name (Figure 15-113).

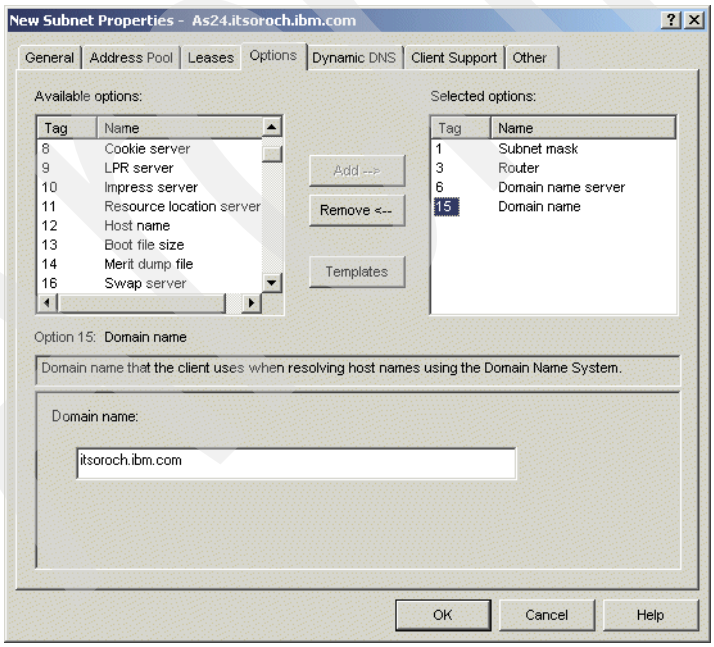


Figure 15-113 Configure subnet 172.23.14.0: Options tab (domain name)

12. Select the **Dynamic DNS** tab. Select **Inherited** (Updates not performed) under Update client records, and **Inherited** under Append domain name to host name (Figure 15-114).

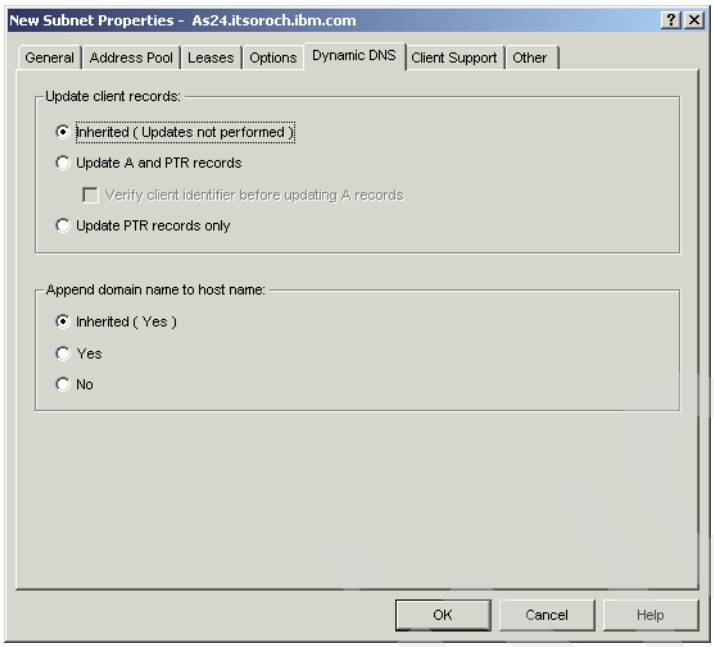


Figure 15-114 Configure subnet 172.23.14.0: Dynamic DNS tab

13. Select the **Client Support** tab. Select **Inherited** (Figure 15-115).

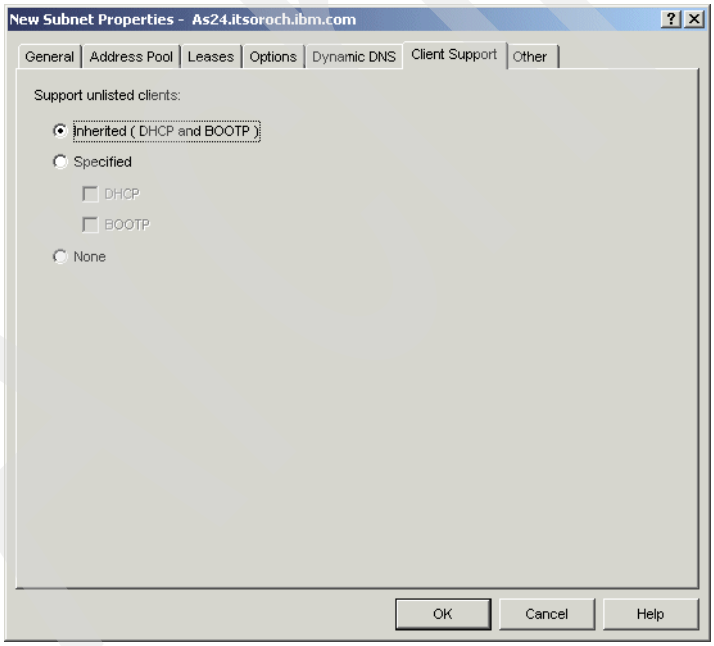


Figure 15-115 Configure subnet 172.23.14.0: Client Support tab

14. Select the **Other** tab. Select **Inherited** for the Bootstrap server (Figure 15-116).

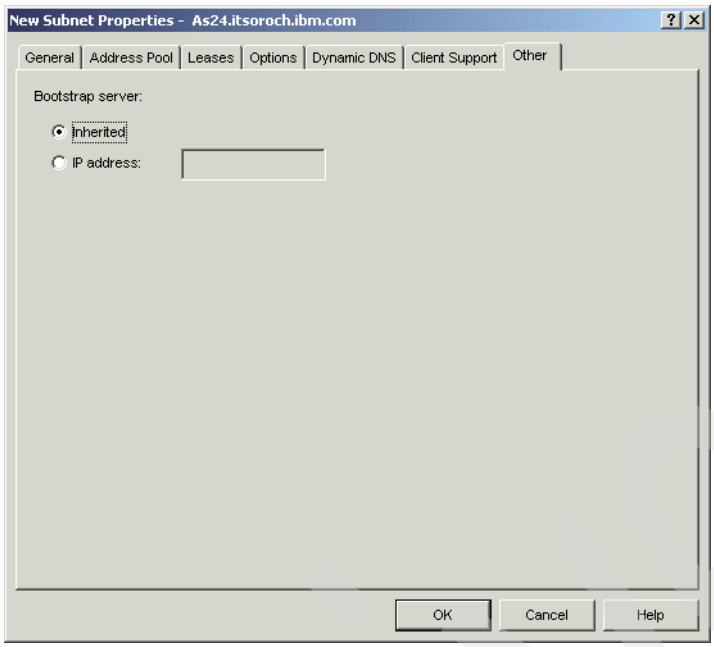


Figure 15-116 Configure subnet 172.23.14.0: Other tab

15. Click **OK** to create the new subnet.

    As shown in Figure 15-117, the subnet 172.23.14.0 has been added to the DHCP configuration.

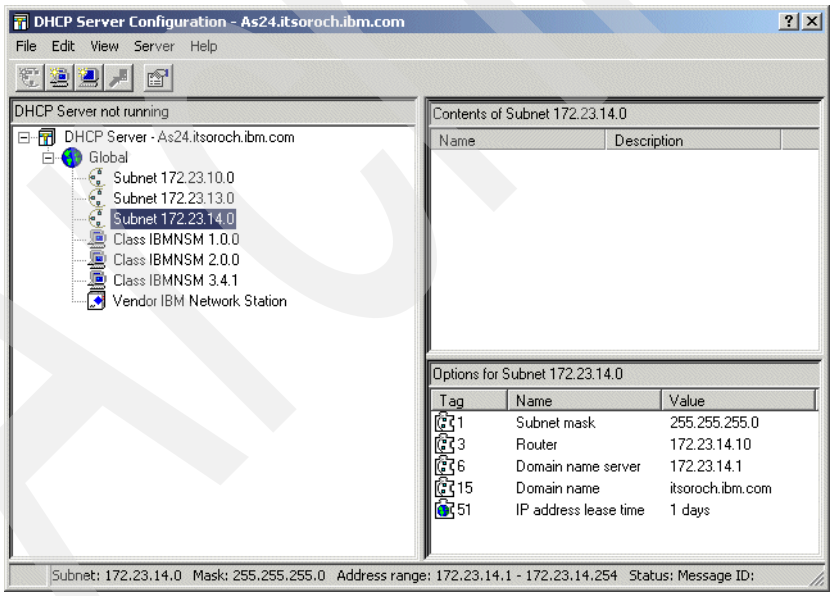


Figure 15-117 DHCP configuration: the subnet 172.23.14.0

This ends the configuration of the new subnet.

Step 4: Configure the BOOTP/DHCP Relay Agent

To enable clients in subnet C to obtain an IP address from the DHCP server AS24, located in subnet A, a BOOTP/DHCP Relay Agent must be configured in subnet C to intermediate the

dialog between DHCP clients and DHCP server. As you can see in Figure 15-101 on page 344, the BOOTP/DHCP Relay Agent is configured with the IP interface 172.23.14.2.

In the following, we show you how to configure two types of BOOTP/DHCP Relay Agents:

- Option 1: Configure the System i BOOTP/DHCP Relay Agent.
- Option 2: Configure the BOOTP/DHCP Relay Agent on Windows 2000 Server.

Option 1: Configure the System i BOOTP/DHCP Relay Agent

Remember that you cannot run the BOOTP/DHCP Relay Agent and the DHCP server on the same System i simultaneously.

In our scenario, we use the System i AS25, which is located in subnet C and has the IP interface 172.23.14.2 configured, as a BOOTP/DHCP Relay Agent to forward DHCP messages to server AS24 directly and without delay.

To configure the System i BOOTP/DHCP Relay Agent on AS25, perform the following steps:

1. Start iSeries Navigator.
2. Expand your System i → **Network** → **Servers** then click **TCP/IP**.
3. Right-click **DHCP/Relay Agent** in right pane, and select **Configuration** from the context menu (Figure 15-118).

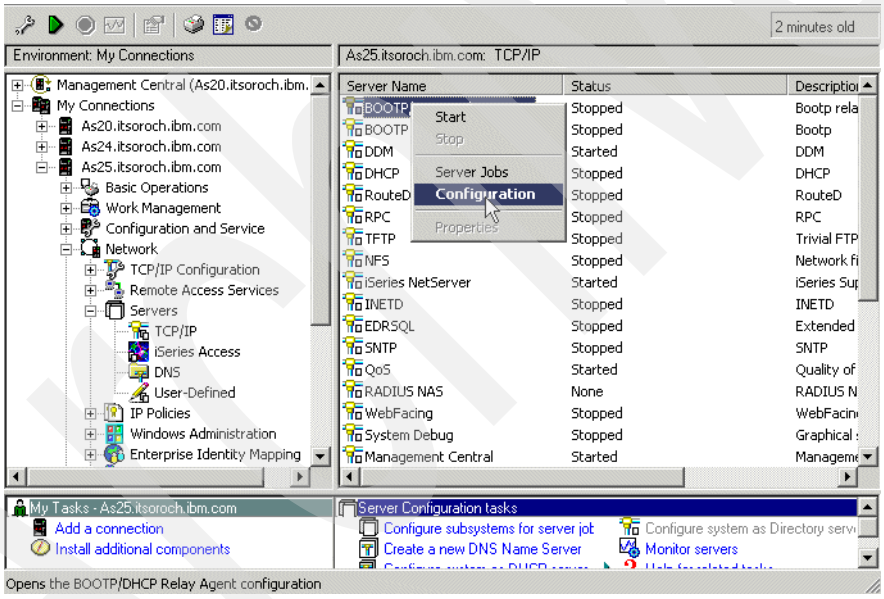


Figure 15-118 iSeries Navigator: starting the BOOTP/DHCP Relay Agent configuration

4. The BOOTP/DHCP Relay Agent Properties window is displayed (Figure 15-119). Select the **Start when TCP/IP is started** check box. Click **Add**.

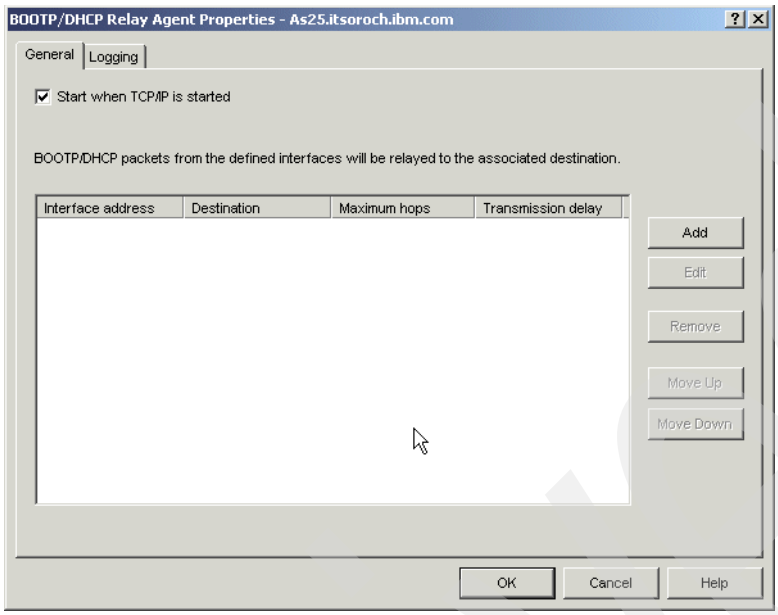


Figure 15-119 BOOTP/DHCP Relay Agent Properties

5. The New Relay Definition window is displayed (Figure 15-120).
   a. From the Interface address list, select the IP interface of the server that can accept packets from BOOTP/DHCP relay agent. In our case, the IP interface is 172.23.14.2.
   b. Under Relay packets to, select **Server IP address** and specify the IP address of the DHCP server to which the BOOTP/DHCP Relay Agent will send the packets. In our case, the Server IP address is the AS24 IP address 172.23.10.2.
   c. Leave the default value (4) for Maximum hops.
   d. Leave the default value (0) in Packet transmission delay.

      Note: Using the parameter Packet transmission delay, we can delay the DHCP packets sent by a DHCP server. Consequently, you can make the DHCP clients prefer a DHCP server by delaying all other DHCP servers.

   e. Click **OK** to add the definition to the BOOTP/DHCP Relay Agent configuration.

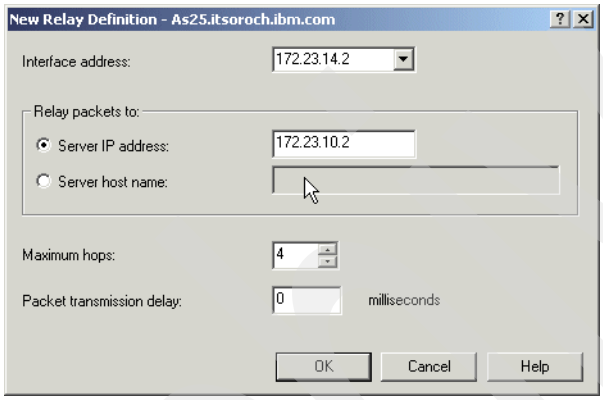


Figure 15-120 New Relay Definition window

6. The definition has been added to the BOOTP/DHCP Relay Agent configuration (Figure 15-121). Click **OK** to save the configuration.

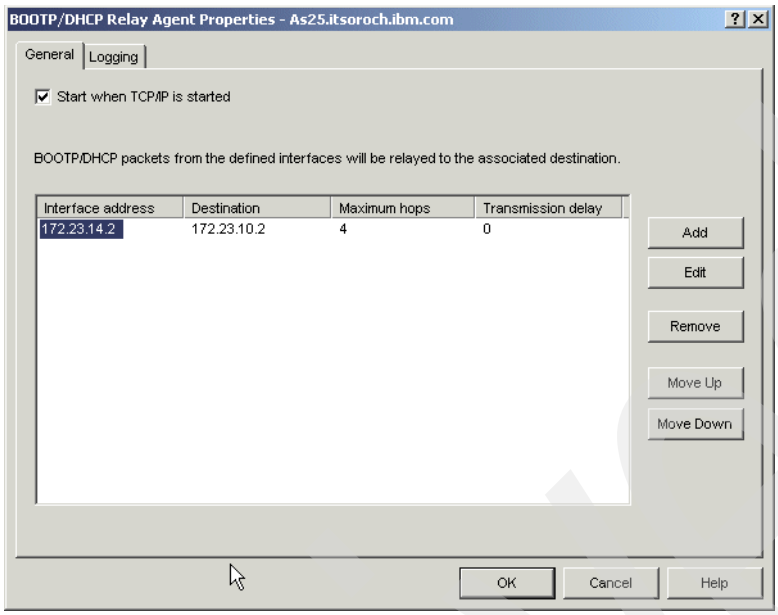


Figure 15-121 BOOTP/DHCP Relay Agent Properties: the new configured definition

This ends the configuration of BOOTP/DHCP Relay Agent on System i.

Option 2: Configure the BOOTP/DHCP Relay Agent on Windows 2000 Server

To configure the BOOTP/DHCP Relay Agent on a Windows 2000 Server system, perform the following steps:

1. Select **Start** → **Programs** → **Administrative Tools** → **Routing and Remote Access**. The Routing and Remote Access window is displayed.
2. Expand the server name. In our scenario, the server name is ITSOW2000.
3. Expand IP Routing.

4. If the DHCP Relay Agent is not in the list, you must install it. To install the DHCP Relay Agent, right-click **General** and select **New Routing Protocol** from the context menu (Figure 15-122).

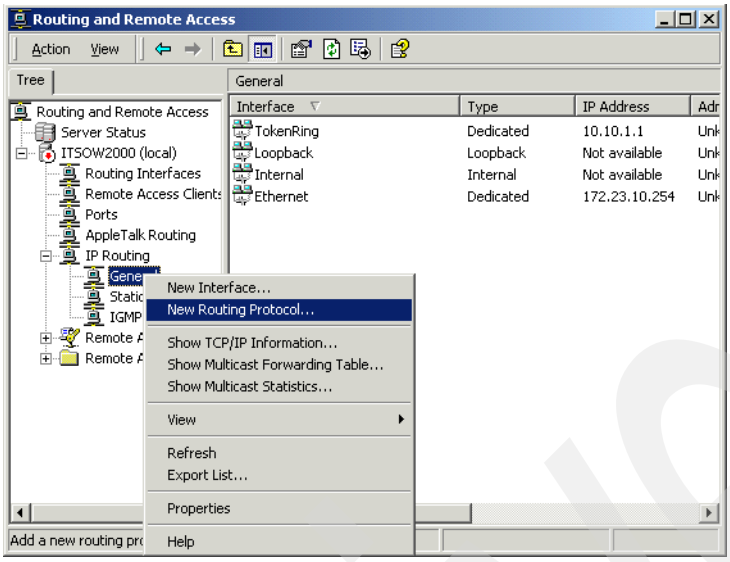


Figure 15-122 Routing and Remote Access configuration window

5. The New Routing Protocol window is displayed. Select **DHCP Relay Agent** and click **OK** to install it (Figure 15-123).

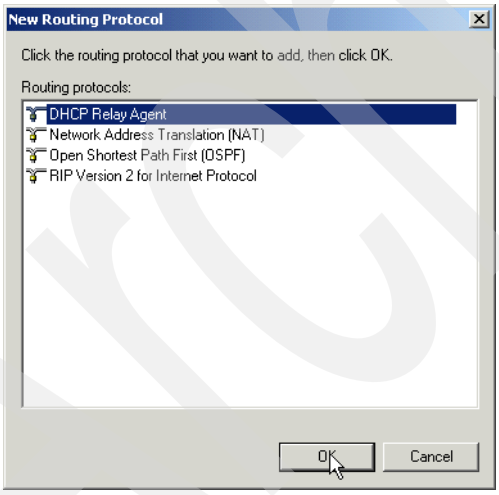


Figure 15-123 Installation of DHCP Relay Agent

6. The DHCP Relay Agent has been added to the Routing and Remote Access configuration (Figure 15-124).

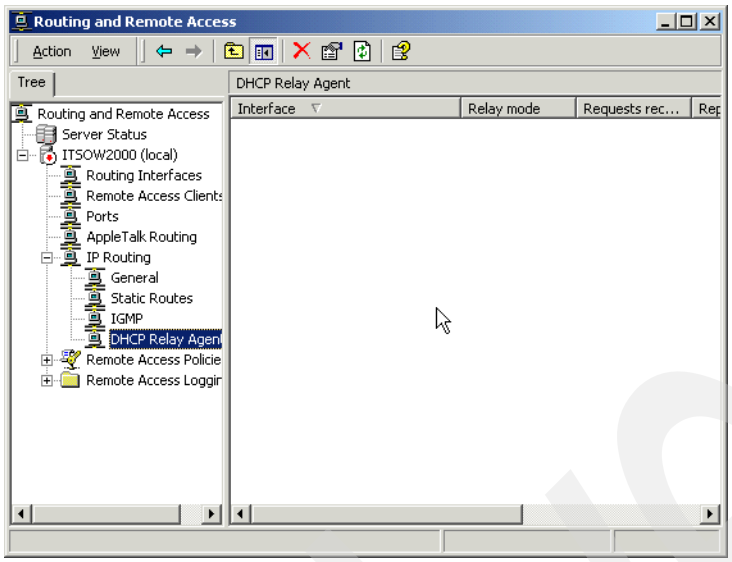


Figure 15-124 Routing and Remote Access configuration window: DHCP Relay Agent

7. Right-click **DHCP Relay Agent** and select **Properties** from the context menu (Figure 15-125).

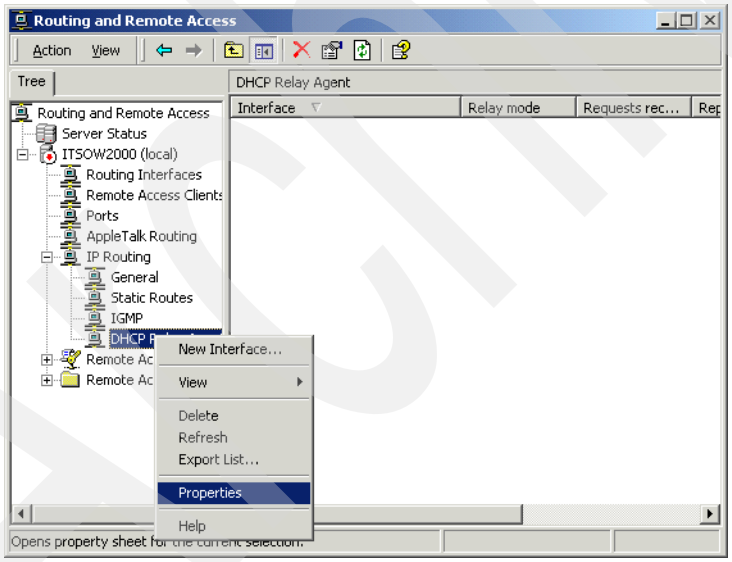


Figure 15-125 Routing and Remote Access configuration window: properties for DHCP Relay Agent

8. The DHCP Relay Agent Properties window opens. Under Server address, specify the IP address of the DHCP server AS24 (`172.23.10.2`) and click **Add**. The IP address is added to the list (Figure 15-126). Click **OK**.

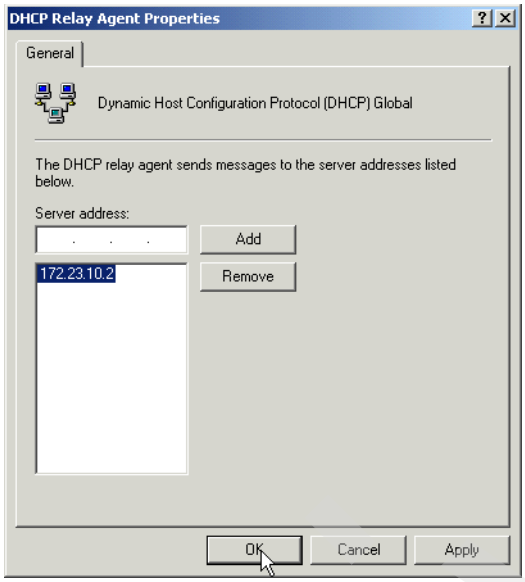


Figure 15-126 DHCP Relay Agent properties

9. Right-click **DHCP Relay Agent** and select **New Interface** from the context menu (Figure 15-127).

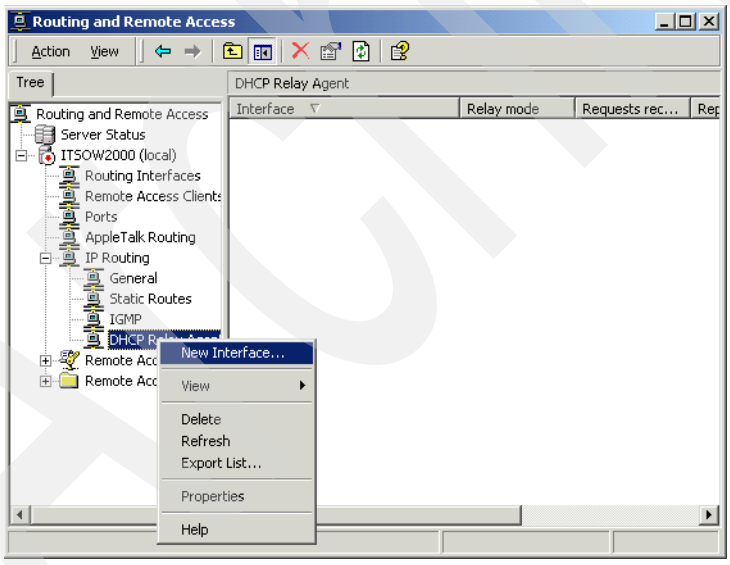


Figure 15-127 Routing and Remote Access: adding a new interface to DHCP Relay Agent

10. The New Interface for DHCP Relay Agent window opens. The Interface list displays all configured interfaces on the system. In our scenario, interface Ethernet is connected in subnet C. Select **Ethernet** and click **OK** (Figure 15-128).

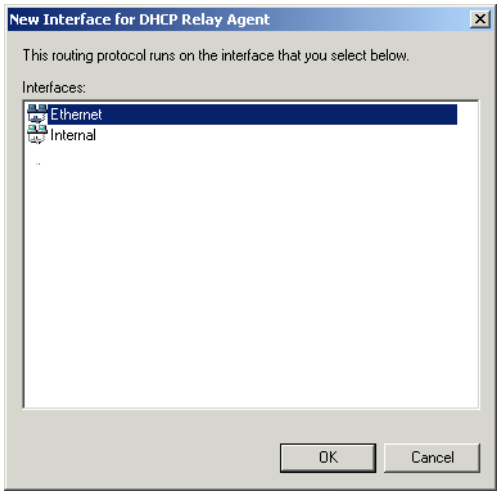


Figure 15-128 New Interface for DHCP Relay Agent window

11. The DHCP Relay Properties - Ethernet Properties window opens. Check the **Relay DHCP packets** check box. Leave the default value for Hop count threshold. Specify value `0` in Boot threshold (Figure 15-129). Click **OK**.

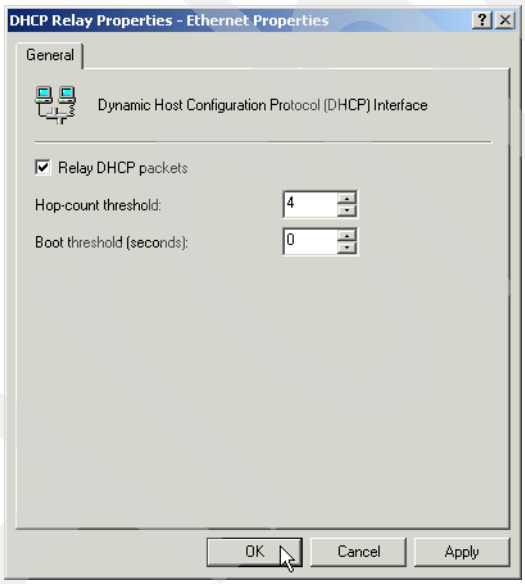


Figure 15-129 DHCP Relay Properties: Ethernet Properties

12. The interface Ethernet has been added to the DHCP Relay Agent configuration (Figure 15-130). Now, this interface listens for DHCP messages. Close the Routing and Remote Access window.

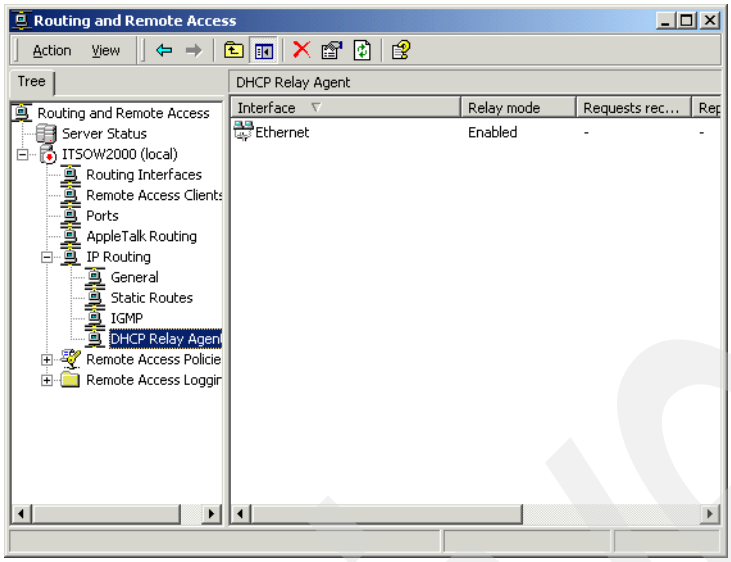


Figure 15-130 Routing and Remote Access: DHCP Relay Agent configuration

This ends the DHCP Relay Agent configuration on Windows 2000 Server.

Step 5: Configure the DHCP clients in subnet C

To configure the clients in subnet C, refer to “Step 5: Configure your Windows 2000 DHCP client” on page 288.

Step 6: Start the DHCP server AS24

To start the DHCP server, refer to “Step 4: Start the DHCP server” on page 287.

Step 7: Start the BOOTP/DHCP Relay Agent

These options show how to start each type of BOOTP/DHCP relay agent:

- Option 1: Start the BOOTP/DHCP Relay Agent on System i.
- Option 2: Start the DHCP Relay Agent on Windows 2000 Server.

Option 1: Start the BOOTP/DHCP Relay Agent on System i

To start the BOOTP/DHCP Relay Agent on System i AS25, perform the following steps:

1. Start iSeries Navigator.
2. Expand your System i → **Network** → **Servers** then click **TCP/IP**.

3. Right-click **DHCP/Relay Agent** in the right pane and select **Start** from the context menu (Figure 15-131).

This ends the procedure.

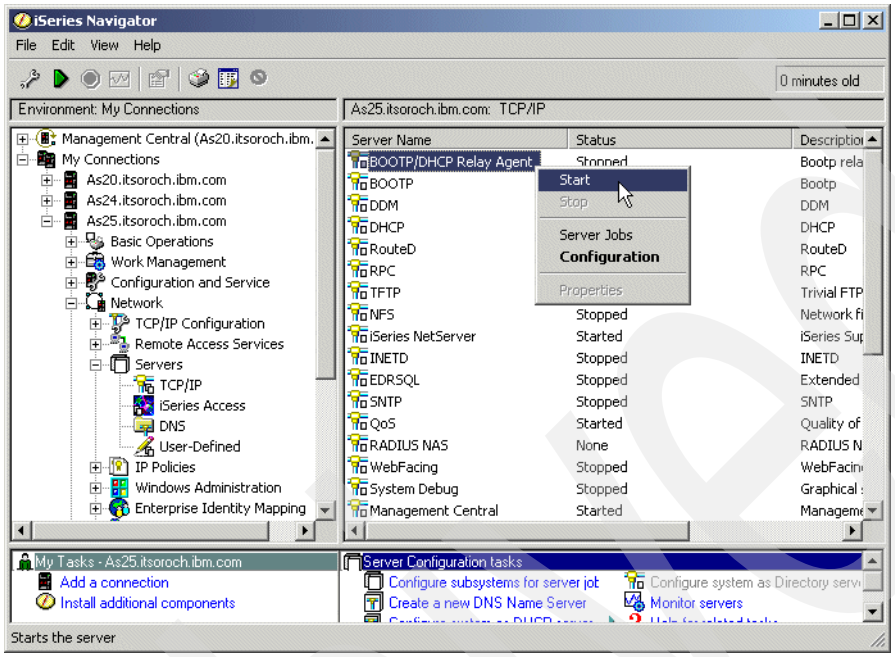


Figure 15-131 iSeries Navigator: Start the BOOTP/DHCP Relay Agent

Option 2: Start the DHCP Relay Agent on Windows 2000 Server

To start the DHCP Relay Agent on Windows 2000 server, perform the following steps:

1. Click **Start** → **Programs** → **Administrative Tools** → **Services** to open the Services window.
2. From the services list, right-click **Routing and Remote Access** and select **Start** from the context menu. The Routing and Remote Access service is started.

This ends the procedure.

Step 8: Test the configuration

In this scenario, we use one DHCP client (Client1) to test the configuration. Client1 is located in subnet C. To test the configuration, perform the following steps:

1. Power on the DHCP client Client1.
2. At boot, the DHCP client sends a DHCPDISCOVER message to obtain an IP address. The DHCP packet is intercepted by the BOOTP/DHCP Relay Agent located in subnet C and passed to the DHCP server AS24 in subnet A. All dialog between DHCP server and the DHCP client is realized through the relay agent.
3. After the workstation starts, open a command prompt panel.
4. Run the following command:

```
C:\> ipconfig /all
```

5. The IP configuration of Client1 is displayed on the command prompt panel. The configuration in Figure 15-132 shows that the DHCP client obtained the IP address from subnet 172.23.14.0 configured on the DHCP server AS24.

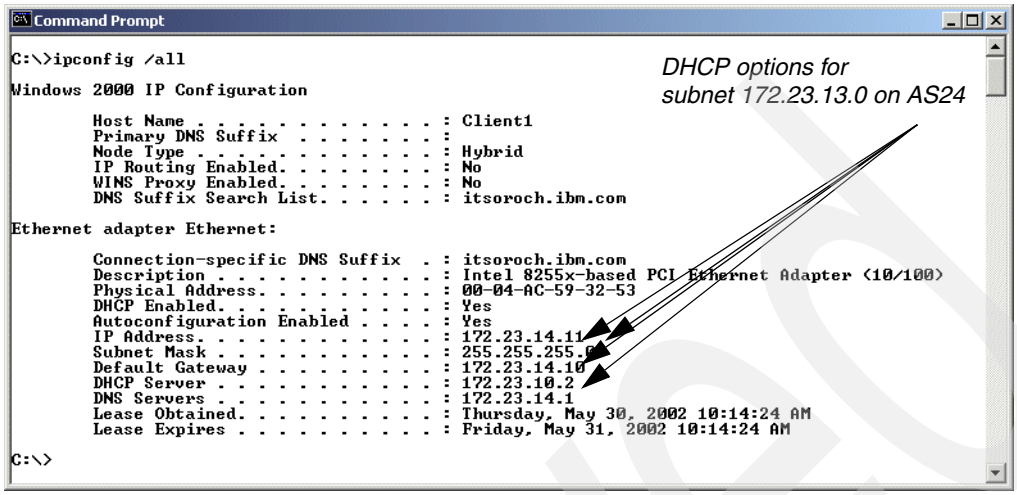


Figure 15-132 Client1 IP configuration after IP address allocation

On the System i side, perform the following steps:

1. Start iSeries Navigator.
2. Expand your System i → **Network** → **Servers** then click **TCP/IP**.
3. Right-click **DHCP** and select **Monitor** from the context menu. The DHCP Monitor window is displayed.
4. Expand **Subnets.** The two subnets that were configured on the DHCP server AS24 are displayed. Select subnet **172.23.14.0**. In the right pane, all IP addresses of this subnet are displayed. Figure 15-133 shows that the IP address 172.23.14.11 was leased to Client1.

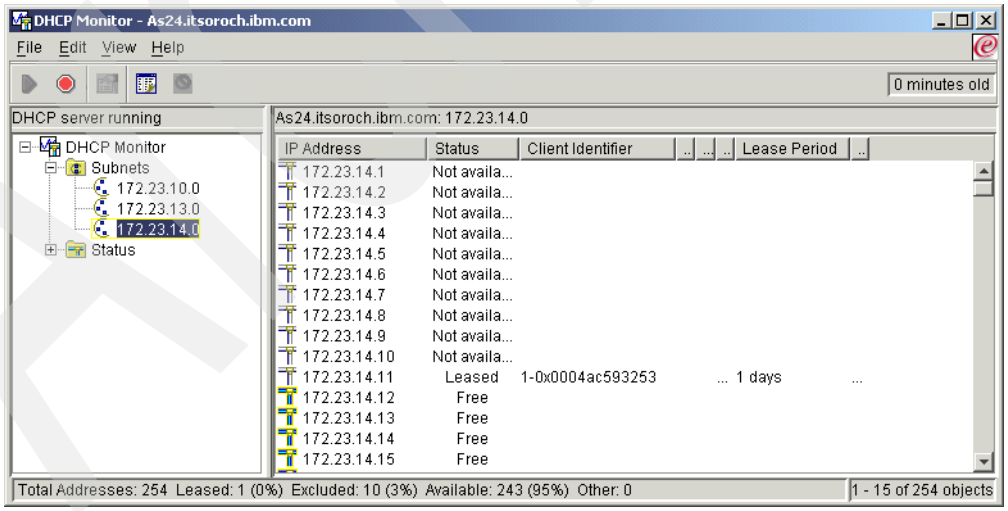


Figure 15-133 DHCP Monitor for server AS24: subnet 172.23.14.0

This ends the DHCP configuration test.

Review, conclusions, and references

In this scenario we explained how to use a DHCP Relay Agent.

We configured a network with multiple subnets and multiple DHCP servers. The clients in one physical subnet were not able to directly reach the DHCP server with DHCP messages. Therefore, we configured in this subnet a BOOTP/DHCP Relay agent that intercepts the DHCP messages sent by the clients and forwards them to the DHCP server. Another utilization of the DHCP relay agent is to determine the DHCP clients to prefer a certain DHCP server by delaying all DHCP messages to all other DHCP servers.

We configured the DHCP server to service the clients in the new added network and configured the BOOTP/DHCP Relay Agent on System i to send DHCP messages from clients to a DHCP server. Also, we showed how to configure the DHCP Relay Agent on a Windows 2000 system.

Finally, we demonstrated how to test the DHCP configuration.

16

Dynamic DNS scenarios

This chapter contains sample Dynamic DNS (DDNS) scenarios referring to the cases explained in Chapter 8, “Dynamic Domain Name System (Dynamic DNS)” on page 133. Each sample scenario consists of four sections:

- Scenario overview: Includes conditions to choose the scenario and sample network configuration.
- Planning worksheet: Prepares required parameters to configure the sample configuration. All questions are prepared; just answer the questions for your own configurations.
- Configuring the sample configuration.
- Testing the sample configuration.

This chapter contains the following sample configurations:

- 16.1, “Single DDNS and DHCP server on the same server” on page 368
- 16.2, “Single DDNS and DHCP servers without secured updates” on page 402
- 16.3, “Single DDNS and DHCP servers with secured updates” on page 438
- 16.4, “Primary DDNS and DHCP servers on one server, secondary server as backup” on page 447
- 16.5, “Primary DDNS and DHCP servers, secondary DNS server Red Hat Linux 7.2” on page 460
- 16.6, “Split DNS: Private and Public DNS with masquerade NAT” on page 466

16.1 Single DDNS and DHCP server on the same server

This procedure configures the single DDNS server on your system. In this scenario, the DHCP server and DDNS server are configured on the same server, and the DHCP server updates the A and PTR records to the DDNS server dynamically right after the DHCP server assigns an IP address to the client. We show how to configure the single DDNS server step-by-step.

This scenario provides the steps to configure a single DDNS server on your network. In this scenario, the DHCP server and DDNS server are configured on the same System i. Using this method sets up the DHCP server to update the A and PTR records on the DNS server dynamically when the DHCP server assigns an IP address to the client.

16.1.1 Scenario overview

You might choose this scenario if these conditions apply:

- If there is one System i available for configuring both the DDNS server and the DHCP server
- If you do not need the secondary DNS server as a fault-tolerant backup
- If this DNS server is used in the private network and no security consideration to isolate the network from the public network is required

Sample network configuration

Figure 16-1 shows the sample network configuration of this scenario.

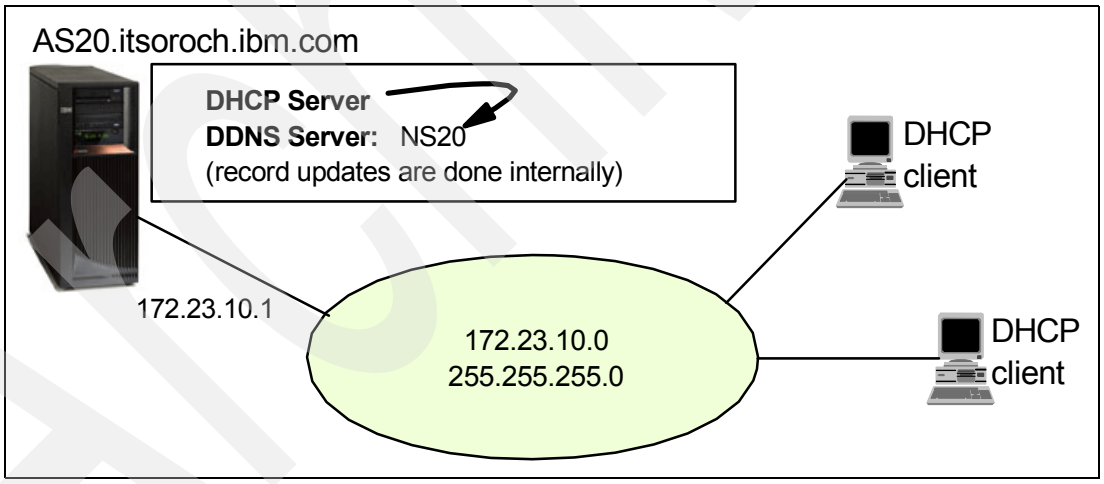


Figure 16-1 Sample network configuration: single DDNS server and DHCP server on the same server

16.1.2 Planning worksheet: single DDNS and DHCP servers on one server

Table 16-1 shows the planning worksheet that we used to prepare for the configuration of DDNS and DHCP on the System i. Our response to each of the questions is shown in the Scenario answers column.

Table 16-1 Planning worksheet for the single DDNS server and DHCP server scenario

No.	Questions to create the single DDNS server and DHCP server on the same server	Scenario answers
1	What is the TCP/IP domain information seen in CFGTCP option 12 panel? - Host name - Domain name - Domain search list - Domain name server IP address	 AS20 itsoroch.ibm.com itsoroch.ibm.com 172.23.10.1
2	What is the DDNS server instance name?	NS20
3	What TCP interface is used for listening the query from the clients? (Query is the request to resolve the host name or IP address)	172.23.10.1
4	Do you want the DDNS server to start when TCP/IP starts?	Yes
5	What is the Cache Time Interval (NS TTL) for the DNS server? (Cache Time Interval means the time-out value of each record on the DNS cache. When the A or PTR record is queried by a client, the record retains on the cache. After the NS TTL time, the A or PTR record on the DNS cache is discarded.	1 day
6	Do you want the DNS server to perform Dynamic Updates?	Yes
7	What is the Start of Authority cache time interval (SOA TTL) value? Start of Authority is the main definition of DNS.	1 day
8	What is the subnet of your network? What subnet mask is assigned for your TCP Interface?	Subnet 172.23.10.x Netmask 255.255.255.0 Now the domain name for the reverse lookup is: 10.23.172.in-addr.arpa.

16.1.3 Configuration: single DDNS and DHCP servers on one server

In this scenario, we create a single DDNS configuration in the following steps:

- Step 1: Confirm the TCP domain information.
- Step 2: Confirm the DHCP configuration for dynamic update.
- Step 3: Creating a single DDNS configuration using iSeries Navigator:
  - Step 3a: Creating the new DNS instance NS20.
  - Step 3b: Creating a new Primary Zone in the Forward Lookup Zone.
  - Step 3c: Creating new Primary Zone in a Reverse Lookup Zone.
- Step 4: Test the configuration.

Step 1: Confirm the TCP domain information

It is important to confirm the correct domain name setup to configure the DNS server:

1. In the iSeries Navigator, expand **Network**.
2. Right-click **TCP/IP Configuration** and choose **Properties** as shown in Figure 16-2 to open the TCP/IP Configuration Properties window.

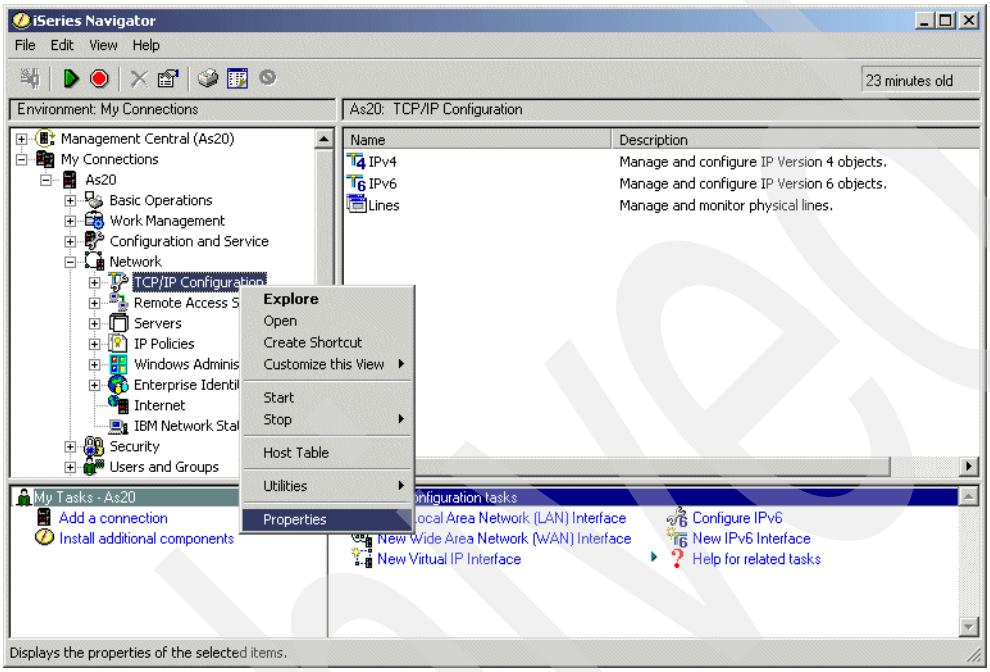


Figure 16-2 iSeries Navigator: TCP/IP Configuration Properties

3. Click the **Host Domain information** tab (Figure 16-3).

> **Tip:** You can get the same information via 5250 command entry with Configure TCP/IP (CFGTCP) option 12=Change TCP/IP domain information.

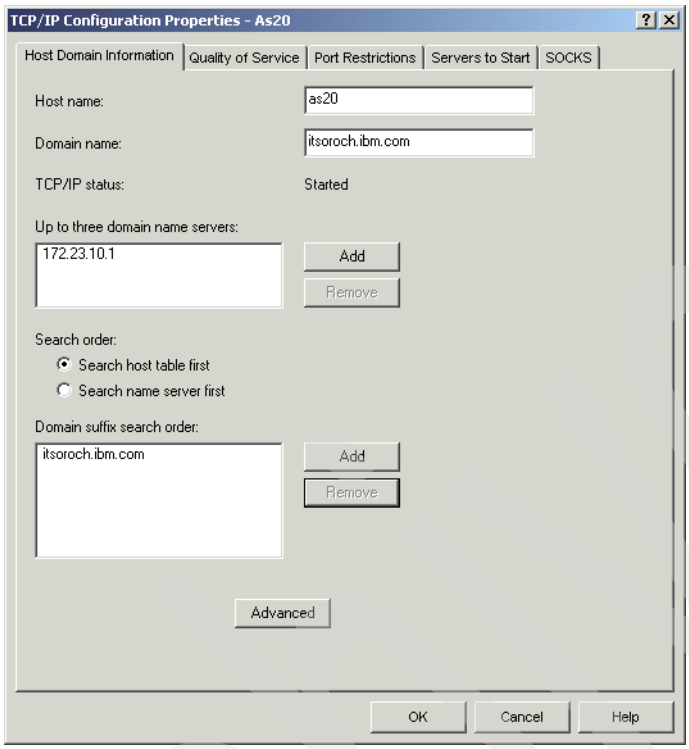


Figure 16-3 TCP/IP Configuration Properties window

The fields shown in Figure 16-4 on page 372 are:

- Host name, domain name

  The combination of the host name and the domain name makes the fully qualified domain name, which is used to present the host name of the DNS server that you are going to create. In this example, the fully qualified domain name is as20.itsoroch.ibm.com.

  This fully qualified domain name is case sensitive. In some servers, uppercase letters are used for the host name or the domain name, though generally lowercase letters are used. If either the host name or domain name includes uppercase letters, consider changing the host name or domain name to lowercase letters only.

- Up to three domain name servers

  This should include your DNS server's IP address. In this example, 172.23.10.1 must be included. This scenario does not call for a secondary or other DNS.

- Domain suffix search order

  This should include your domain name. In this example, it includes itsoroch.ibm.com.

Step 2: Confirm the DHCP configuration for dynamic update

If you want the DHCP server to update the A and PTR records dynamically, you must configure your DHCP server to perform dynamic updates. Check the DHCP server configuration using this procedure:

1. In the iSeries Navigator, expand **Network** → **Servers**.
2. Click **TCP/IP**. In the right pane, right-click **DHCP** and choose **Configuration** from the context menu as shown in Figure 16-4.

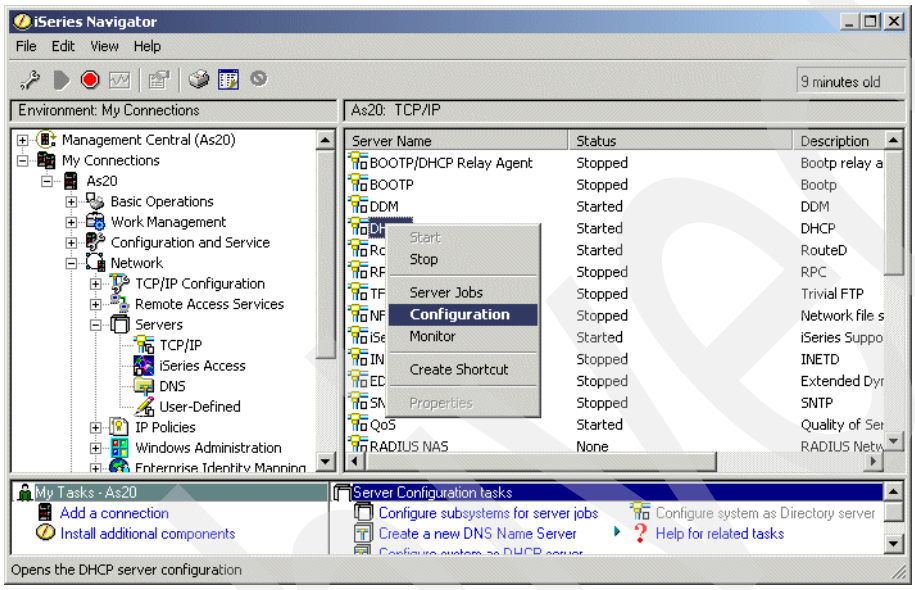


Figure 16-4 iSeries Navigator window

3. In the DHCP Server Configuration - As20 window, right-click **Global**. Select **Properties** from the context menu as shown in Figure 16-5.

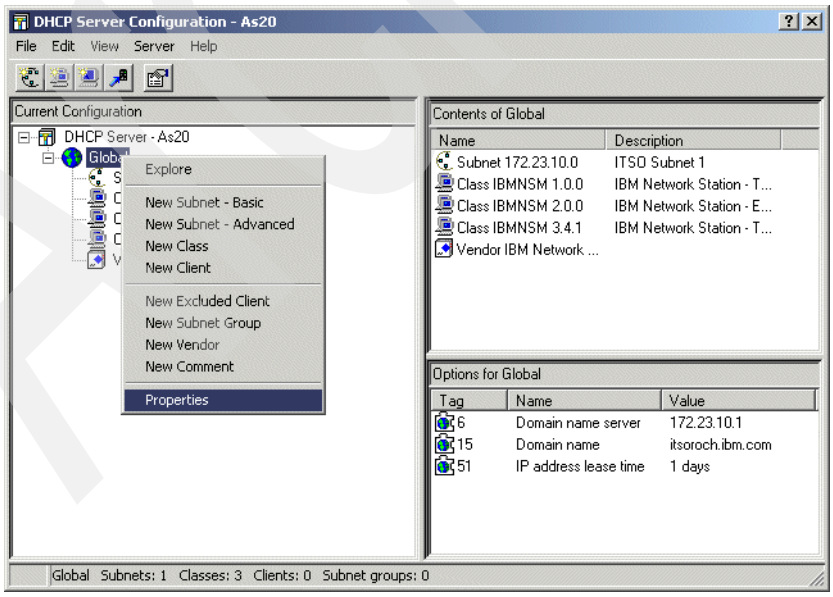


Figure 16-5 DHCP Server Configuration - As20 window

4. In the Global Properties- As20 window, click the **Dynamic DNS** tab. Ensure that **DHCP server updates both A and PTR records** is selected (Figure 16-6). This performs the dynamic A and PTR records update from DHCP server to DNS server.

   Tip: For more information about these parameters see Figure 6-13 on page 117 and the discussion that leads up to it.

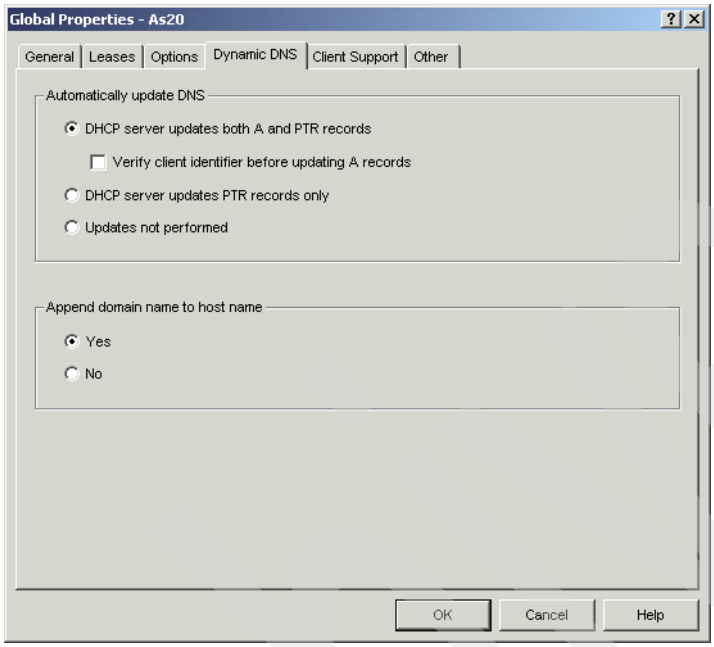


Figure 16-6 Global Properties - As20 window

5. In the Global Properties - As20 window, click the **Options** tab. The Selected options column should include **6 - Domain Name Server** and **15 - Domain Name**. This domain name server IP address and domain name will be sent to the client when the DHCP server is going to lease an IP address to the client. The domain name server and domain name information will remain on the client to recognize the DNS IP address and the domain name if the client is configured to obtain the DNS IP address and domain name from the DHCP server.

   Domain name should be the right one. In this example, the domain name should match 15 - Domain Name on the Options tab, and the domain name seen in Figure 16-3 on page 371. If 15 - Domain Name on the Options tab is different from Domain Name configured in Figure 16-3 on page 371, the DHCP server cannot update A and PTR records correctly to the DDNS server.

In the Global Properties - As20 window, click **6** under Selected options, as shown in Figure 16-7. In this example, confirm that the DNS IP address is 172.23.10.1.

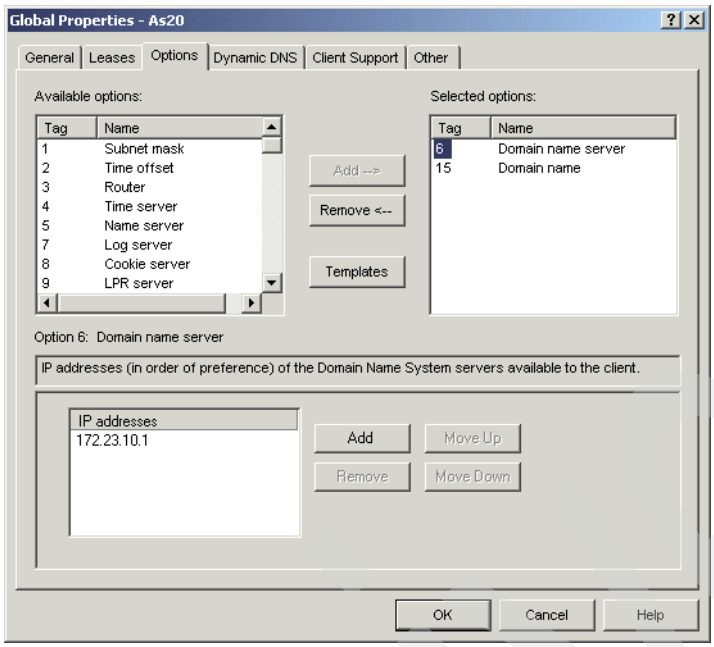


Figure 16-7 Global Properties - As20 window

6. Click **15** as shown in Figure 16-8. In this example, confirm that the Domain name is `itsoroch.ibm.com`. Click **Cancel**.

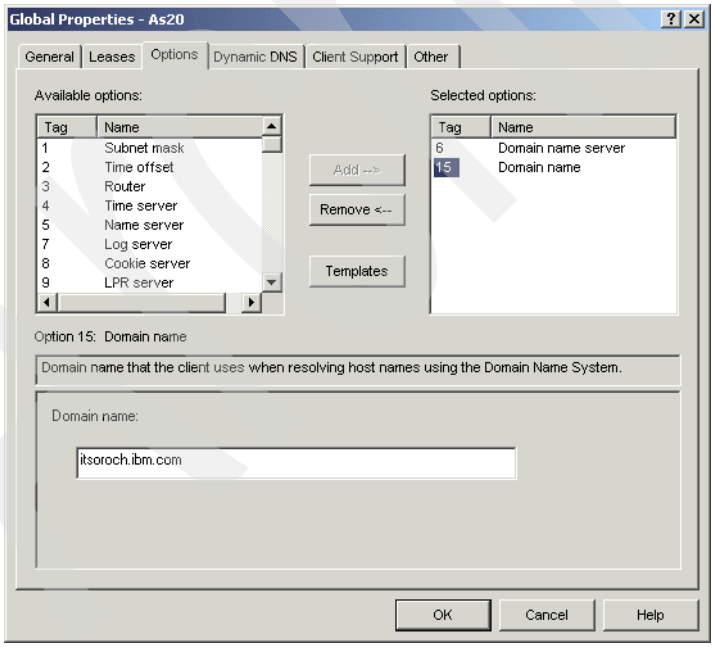


Figure 16-8 Global Properties - As20 window

7. In the DHCP Configuration - As20 window, choose **File** → **Close** on the task bar. This ends the procedure to check the configuration of the DHCP server.

Step 3: Creating a single DDNS configuration using iSeries Navigator

There are three substeps to creating a single DDNS configuration on your System i:

- Step 3a: Creating the new DNS instance NS20
- Step 3b: Creating a new Primary Zone in the Forward Lookup Zone
- Step 3c: Creating new Primary Zone in a Reverse Lookup Zone

Step 3a: Creating the new DNS instance NS20

The first step in creating a DDNS configuration is to create the new DNS instance. To create the new DNS instance NS20:

1. In the iSeries Navigator, expand **Network** → **Servers**.
2. Right-click **DNS** and select **New Name Server** as shown in Figure 16-9.

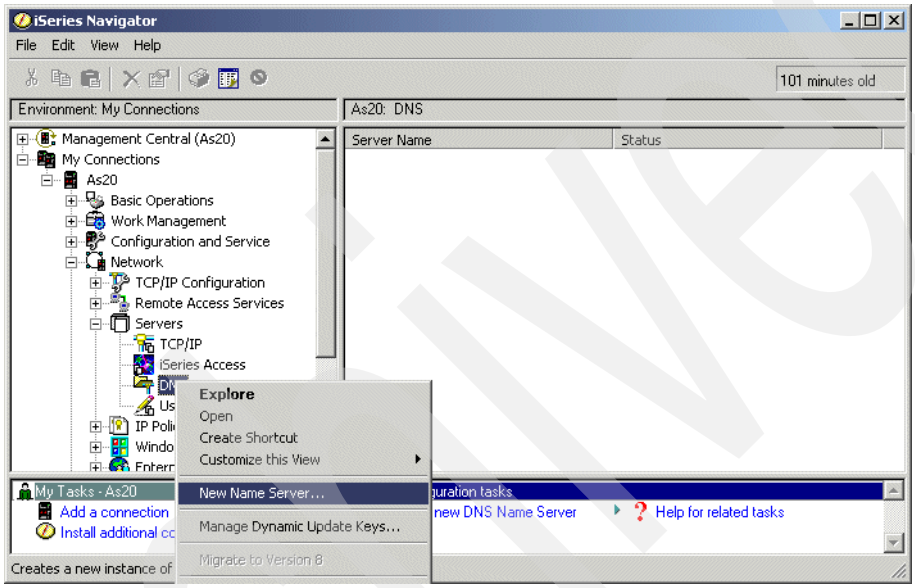


Figure 16-9 Creating a new name server

3. In the New DNS Configurations window (Figure 16-10), click **Next**.

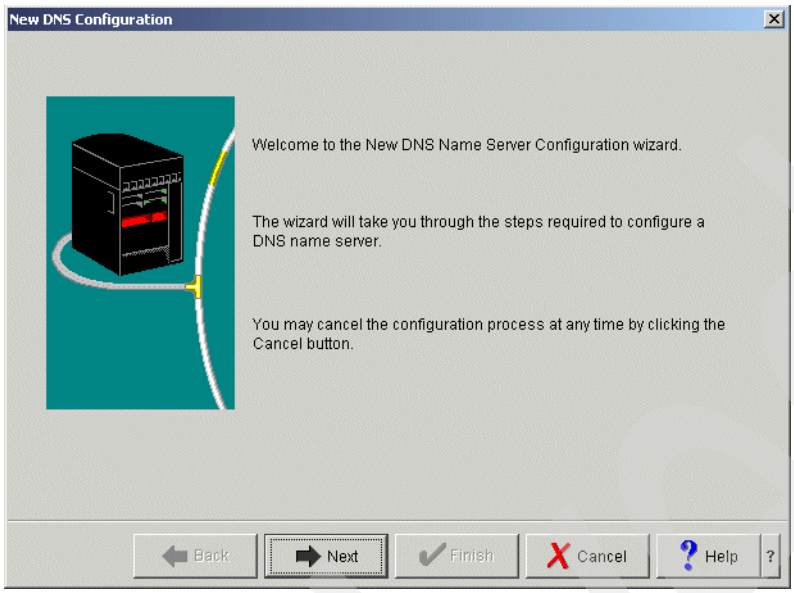


Figure 16-10 New DNS Configuration window

4. In the DNS Server Name window, type `NS20` as a DNS server instance name (answer 2 in Table 16-1 on page 369), as shown in Figure 16-11. Click **Next** to continue.

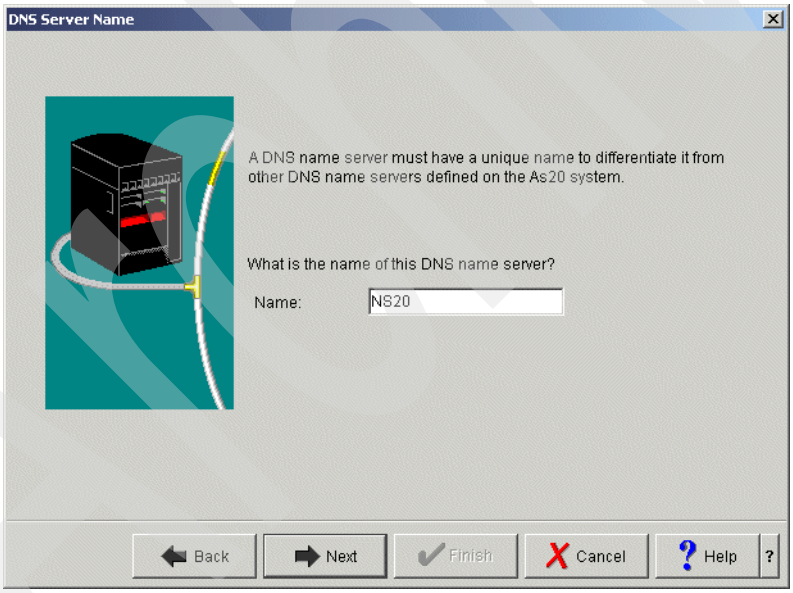


Figure 16-11 DNS Server Name window

5. In the Listen On IP Addresses window (Figure 16-12), select the IP Address **172.23.10.1** as a Query-listening TCP Interface (answer 3 in Table 16-1 on page 369). Click **Next** to continue.

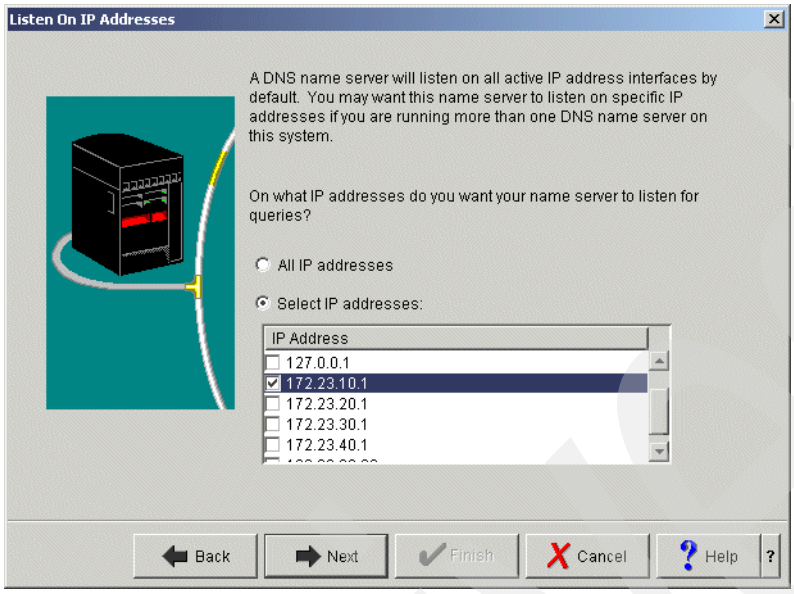


Figure 16-12 Listen On IP Addresses window

6. In the Root Servers window, if you need to add root servers, click **Add** and add the root server IP addresses. In this scenario, considering the small office use in the company, the root server entry is not required, as shown in Figure 16-13. Click **Next** to continue.

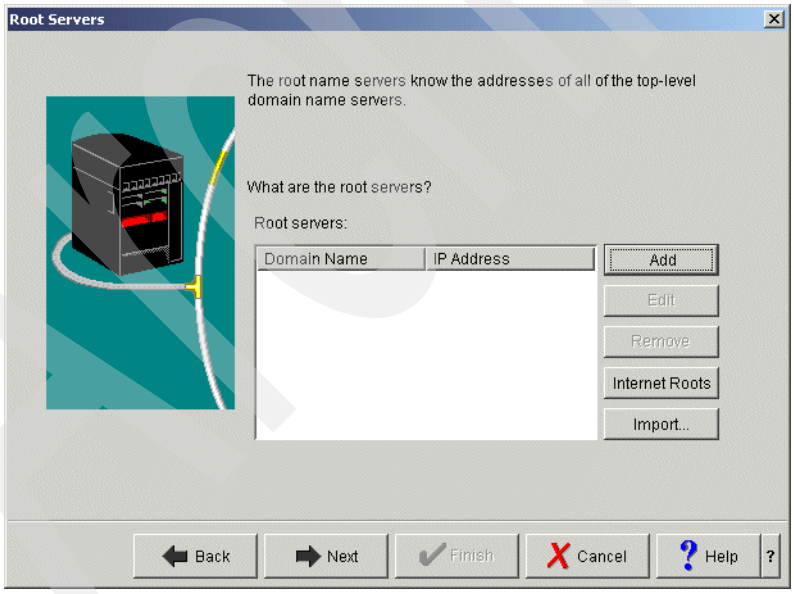


Figure 16-13 Root Servers window

7. In the Start DNS Server window, click **Yes** (answer 4 in Table 16-1 on page 369), as shown in Figure 16-14. Click **Next** to continue.

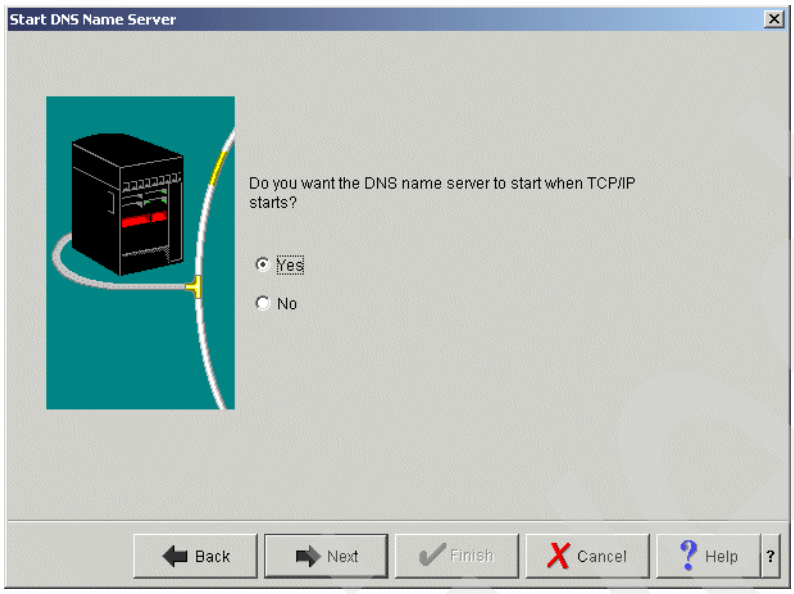


Figure 16-14 Start DNS Server window

8. In the Summary window (Figure 16-15), confirm your entry data. Click **Finish**.

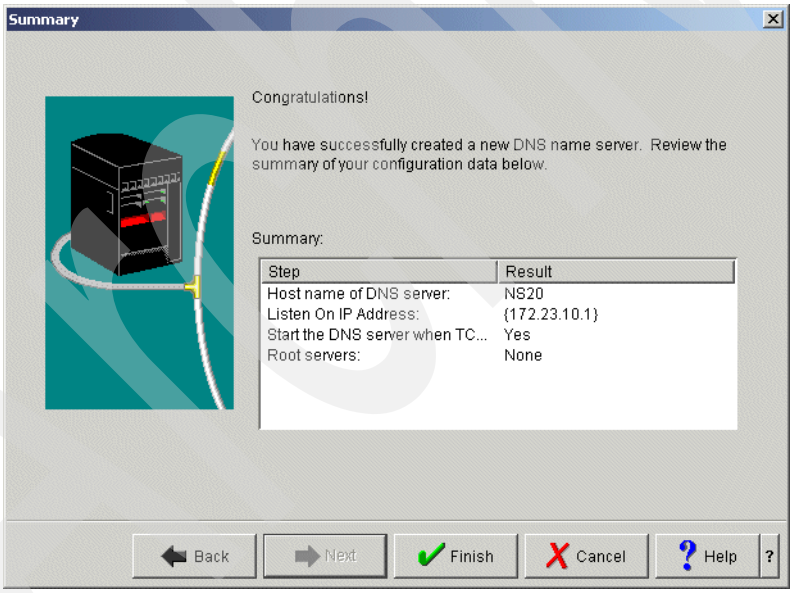


Figure 16-15 Summary window

Step 3b: Creating a new Primary Zone in the Forward Lookup Zone

After you created a new DNS instance NS20, configure two primary zones: a Forward Lookup Zone and a Reverse Lookup Zone. This step shows how to create a new Primary Zone in the Forward Lookup Zone:

1. In the iSeries Navigator window, right-click the newly created DNS instance **NS20**. From the pull-down menu, select **Configuration**, as shown in Figure 16-16.

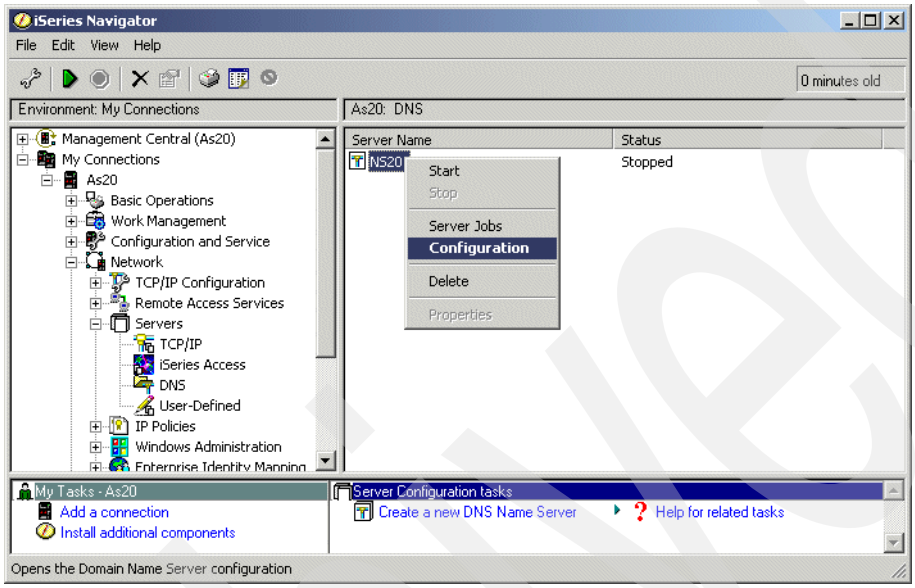


Figure 16-16 iSeries Navigator window

2. In the DNS Configuration - NS20 window, right-click **Forward Lookup Zones**. In the context menu, select **New Primary Zone** as shown in Figure 16-17.

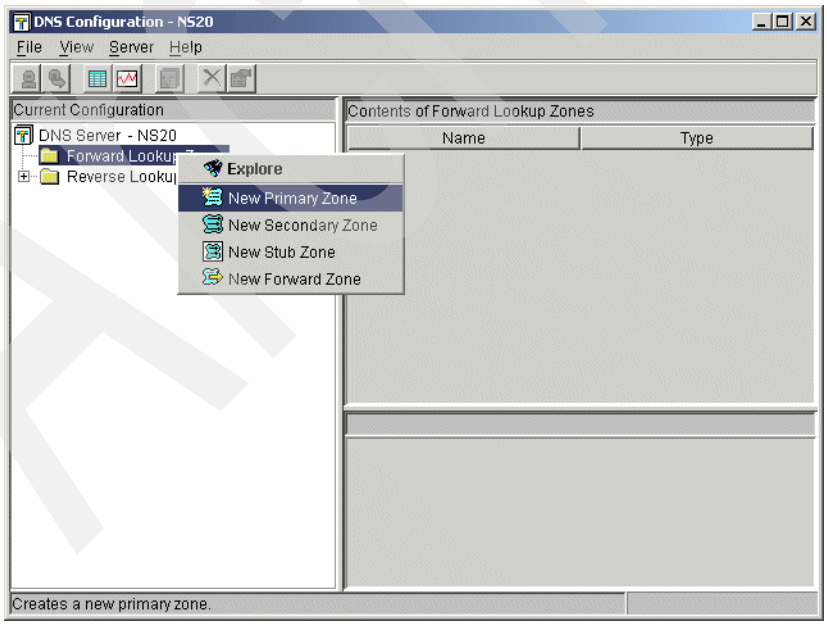


Figure 16-17 DNS Configuration: NS20 window

3. In the Zone Domain name window, type the domain name `itsoroch.ibm.com.` (answer 1 in Table 16-1 on page 369), as shown in Figure 16-18. Click **Next** to continue.

> **Tip:** Do not forget to type the period at the end of the fully qualified domain name.

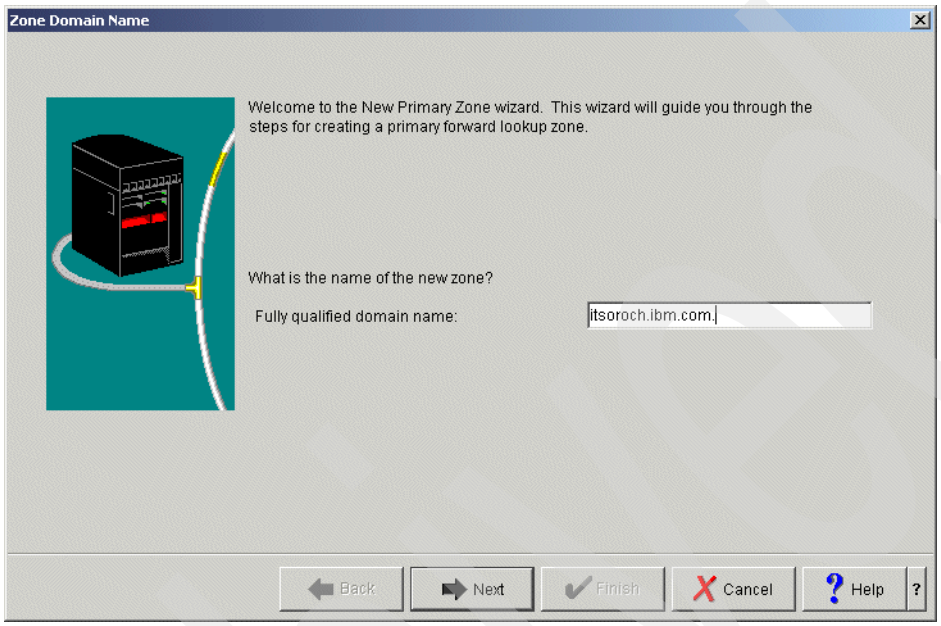


Figure 16-18 Zone Domain Name window

4. In the Name Servers window shown in Figure 16-19, select the name server **as20.itsoroch.ibm.com.** and click **Edit**.

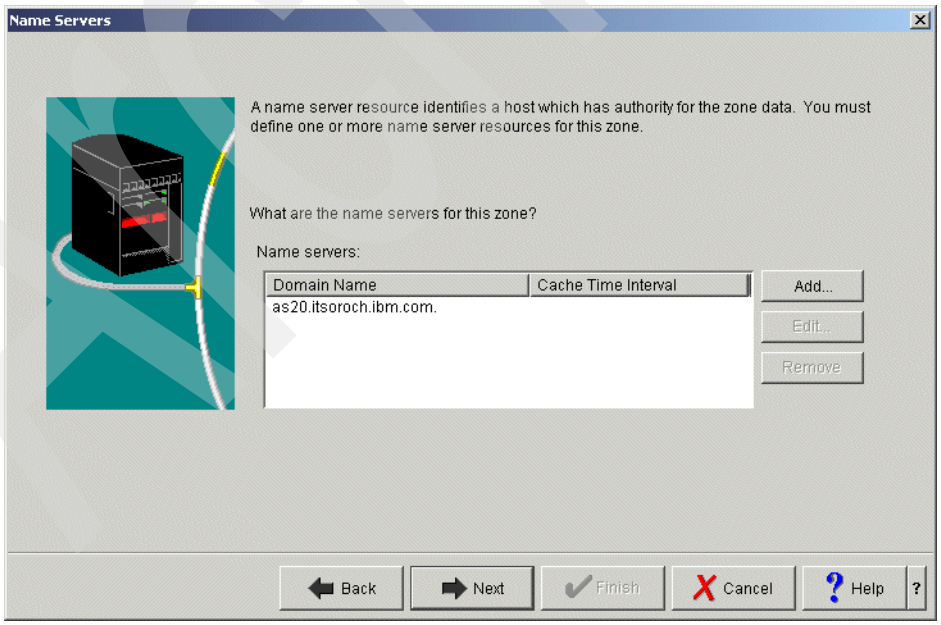


Figure 16-19 Name Servers window

5. In the Edit Name Server (NS) window, check **Cache Time Interval (NS TTL)** and type a 1 in the column, then choose **days** (answer 5 in Table 16-1 on page 369), as shown in Figure 16-20. Click **OK** to continue.

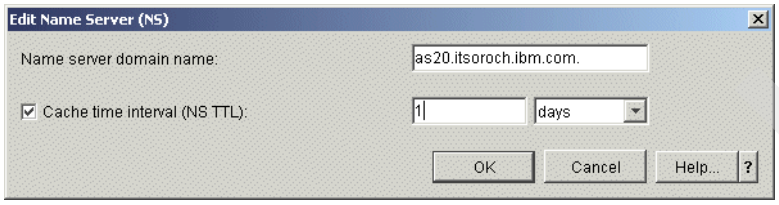


Figure 16-20 Edit Name Server (NS) window

6. In the Name Server IP Addresses window, click **Add**. Type 172.23.10.1 as the IP address for as20.itsoroch.ibm.com. (answer 1 in Table 16-1 on page 369), as shown in Figure 16-21. Click **OK** to continue.

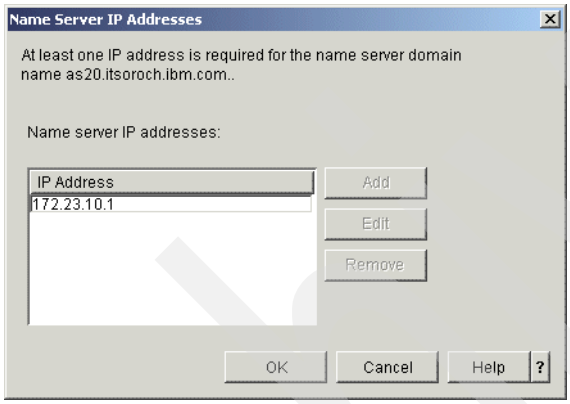


Figure 16-21 Name Server IP addresses window

In the Name servers window, click **Next**.

7. In the Static or Dynamic Zone window, choose **Perform dynamic updates** (answer 6 in Table 16-1 on page 369), as shown in Figure 16-22. Click **Next** to continue.

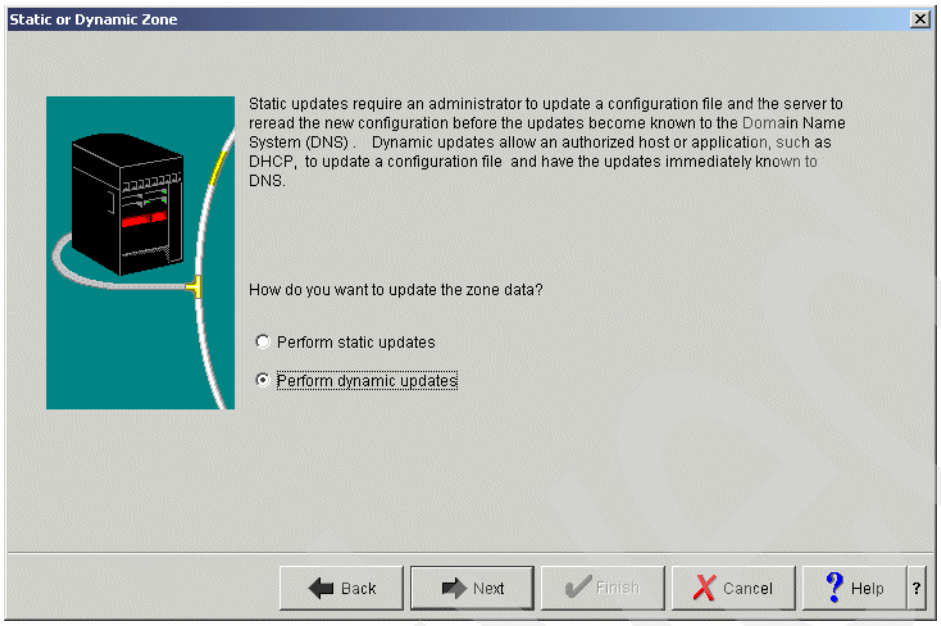


Figure 16-22 Static or Dynamic Zone window

8. In the Allow Update window, click **Add**, as shown in Figure 16-23.

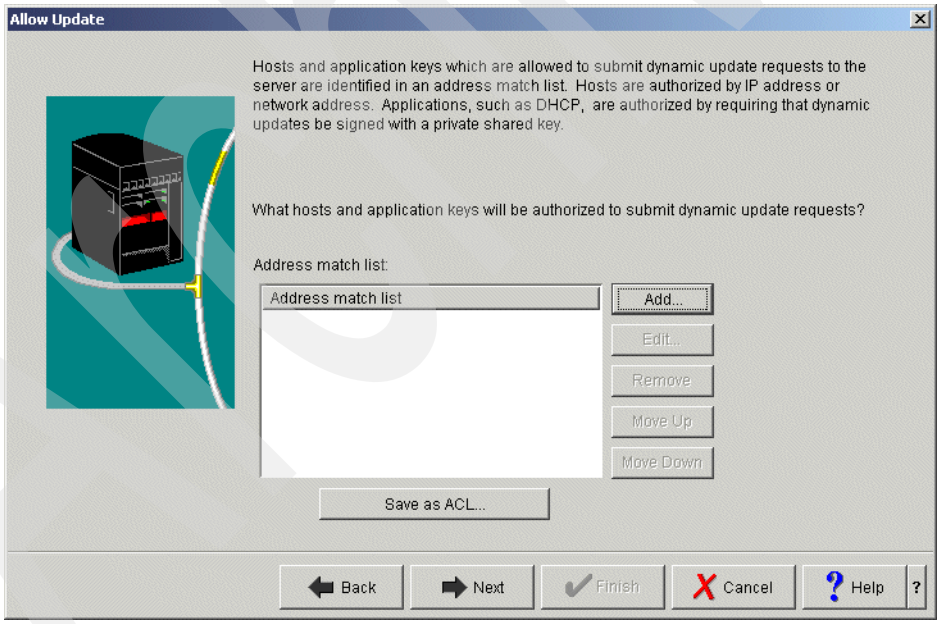


Figure 16-23 Allow Update window

9. In the Address Match List Element window, make sure that **IP Address** is selected as an Address match list element type. Type `172.23.10.1` (answer 1 in Table 16-1 on page 369) for IP address. Select **Allow access to this IP address** as shown in Figure 16-24. Click **OK** to continue.

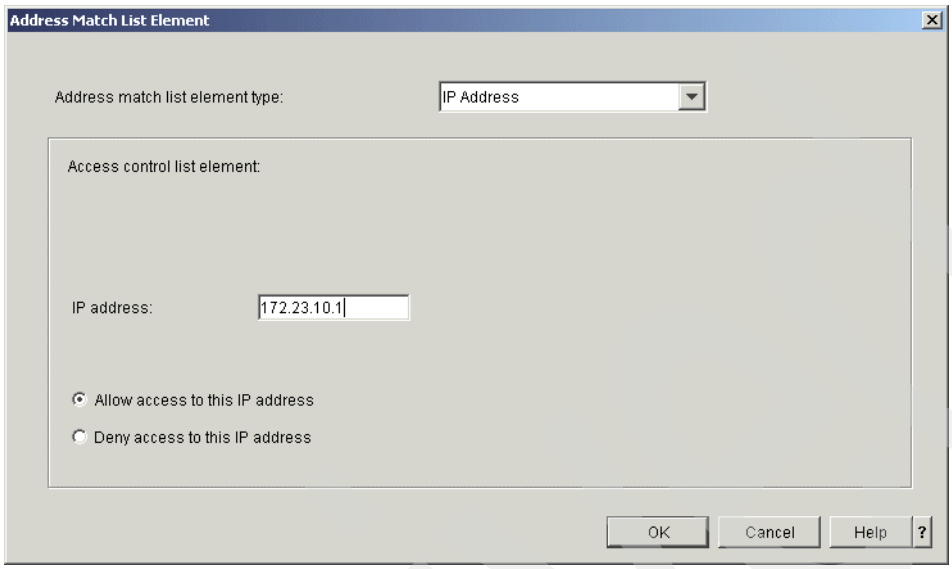


Figure 16-24 Address Match List Element window

In the Allow Update window, click **Next** to continue.

10. In the Summary window (Figure 16-25), click **Finish** to continue.

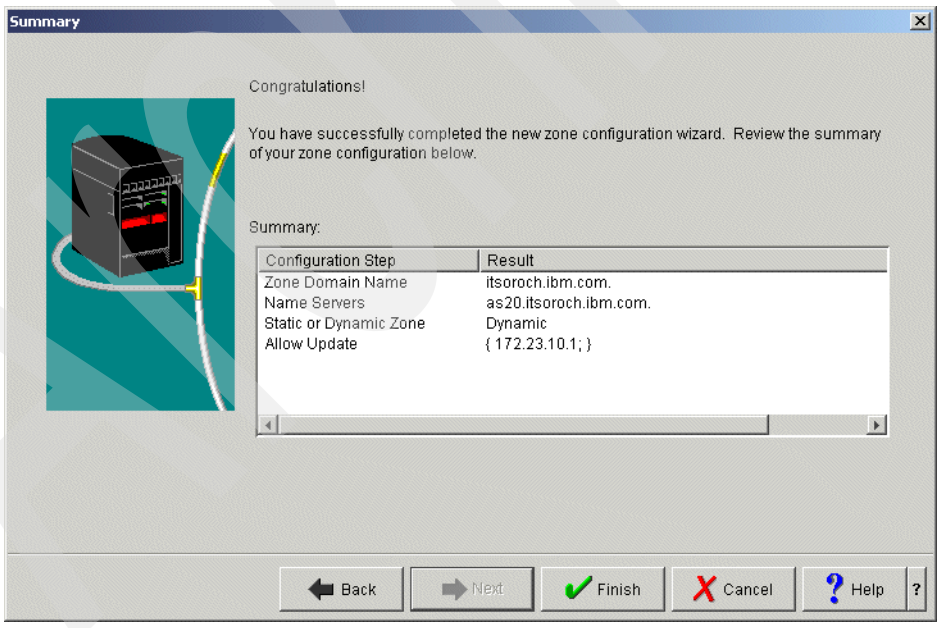


Figure 16-25 Summary window

11. In the DNS Configuration - NS20 window (Figure 16-26), expand **Forward Lookup Zones** and right-click **Primary Zone itsoroch.ibm.com.** On the context menu, click **Properties**.

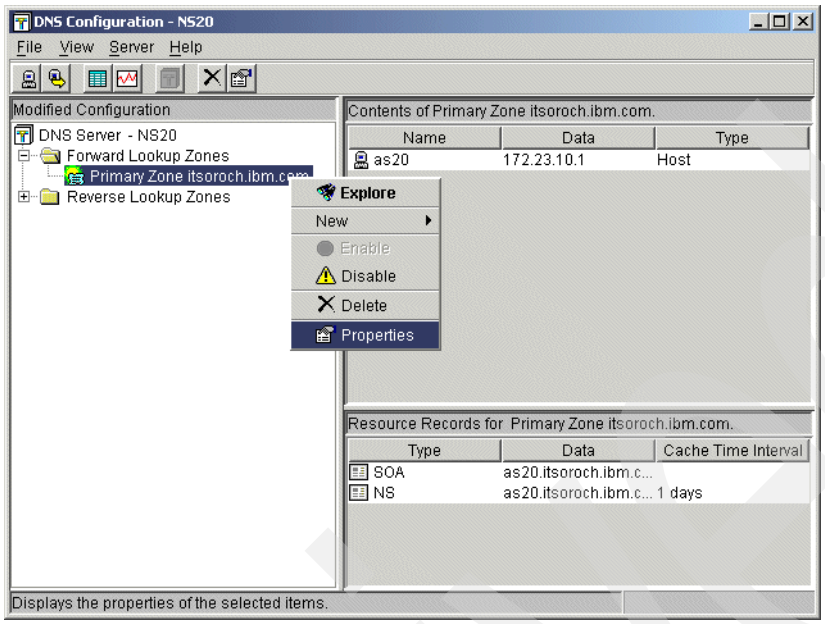


Figure 16-26 DNS Configuration: NS20 window

12. In the Primary Zone Properties - itsoroch.ibm.com. window, click the **Options** tab. Expand **Access Control**, as shown in Figure 16-27.

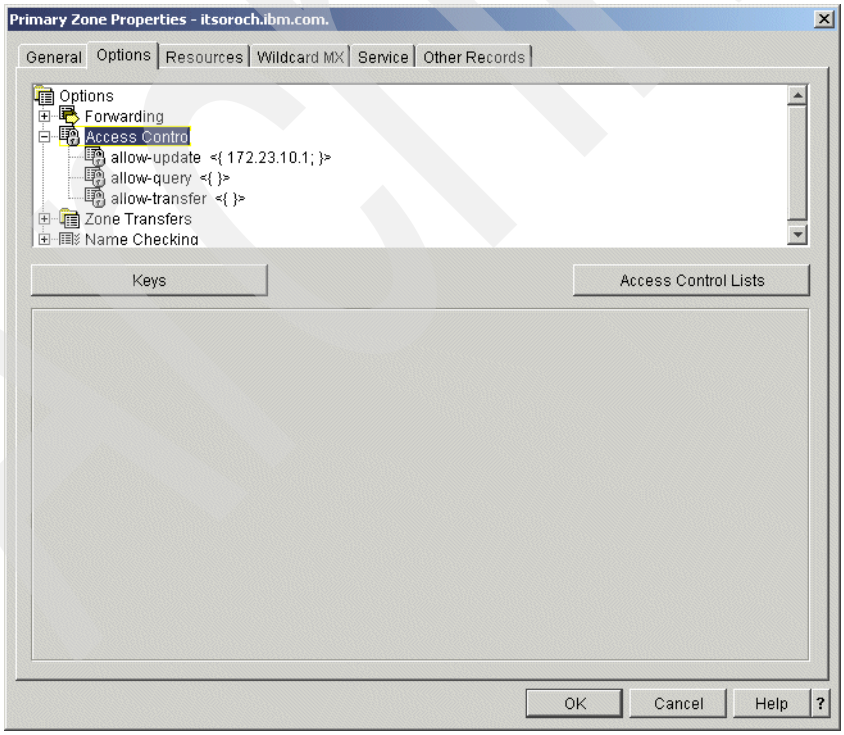


Figure 16-27 Primary Zone Properties: itsoroch.ibm.com window

13. Select **allow-query** and choose **Access Control List** as the Match list element type. Click **Add** as shown in Figure 16-28.

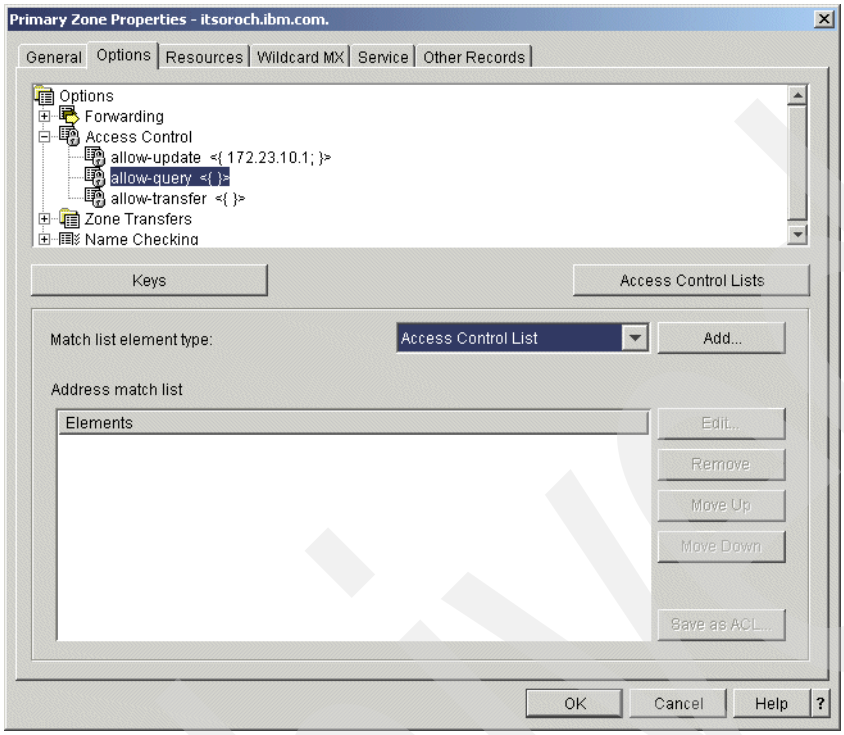


Figure 16-28 Primary Zone Properties: itsoroch.ibm.com. window

14. In the Access Control List window, select **any**. Choose **Allow access to this access control list** as shown in Figure 16-29. Click **OK**.

> **Tip:** This allows the Query request from any client to resolve IP address or host name.

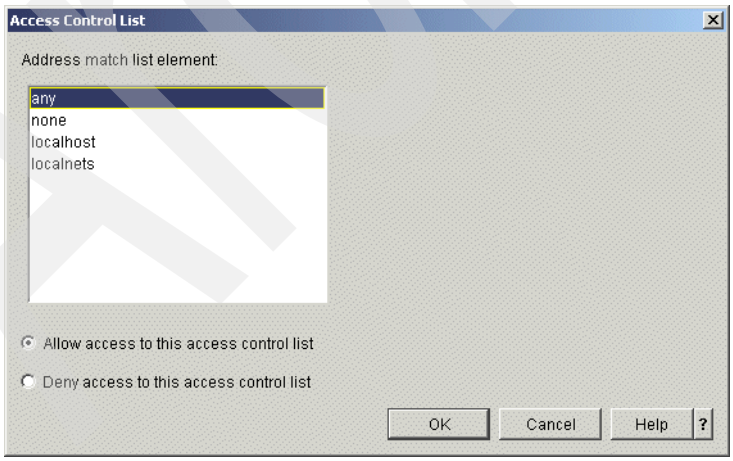


Figure 16-29 Access Control List window

15. In the Primary Zone Properties - itsoroch.ibm.com. window, click the **Resources** tab. Select **SOA** and click **Edit** as shown in Figure 16-30.

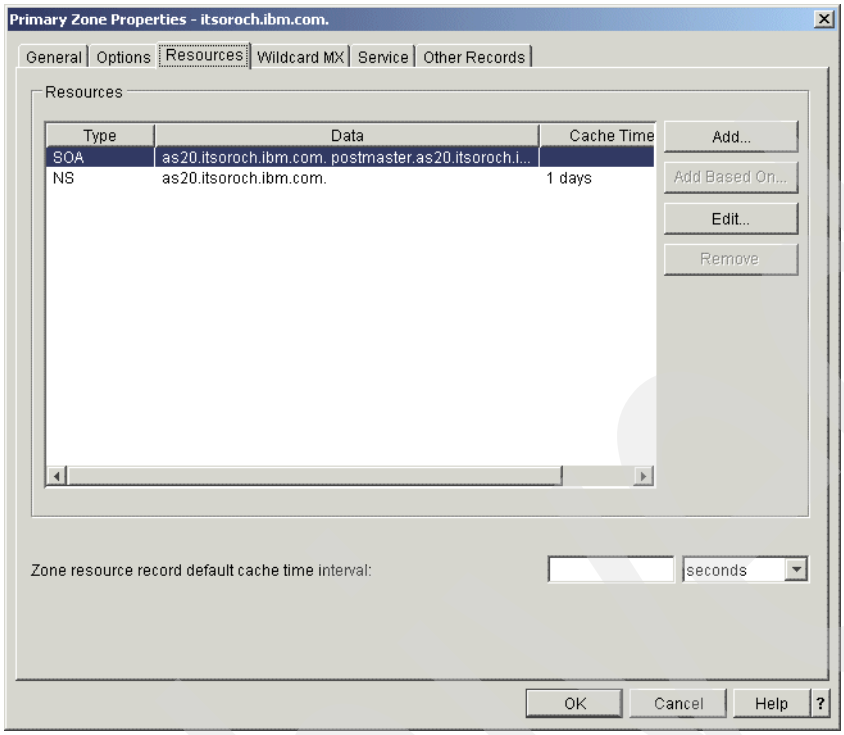


Figure 16-30 Primary Zone Properties - itsoroch.ibm.com window

16. In the Add/Edit Resource - itsoroch.ibm.com. window, check **Start of Authority cache time interval (SOA TTL)**. Type 1 and choose **days** (answer 7 in Table 16-1 on page 369) as a value of SOA TTL, as shown in Figure 16-31. Click **OK**.

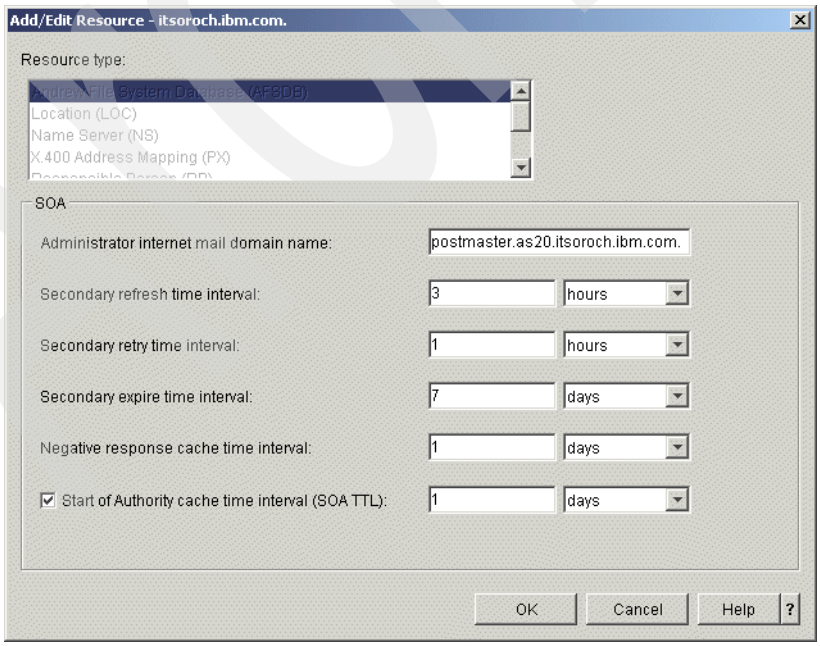


Figure 16-31 Add/Edit Resource: itsoroch.ibm.com.

Step 3c: Creating new Primary Zone in a Reverse Lookup Zone

Now we create a new Primary Zone in a Reverse Lookup Zone:

1. In the DNS Configuration - NS20 window, right-click **Reverse Lookup Zones** and choose **New Primary Zone**, as shown in Figure 16-32.

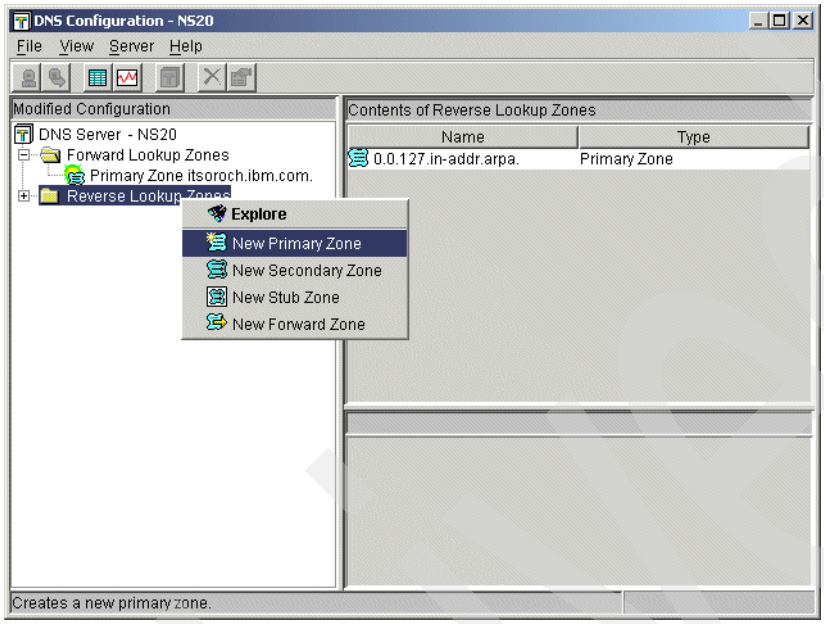


Figure 16-32 DNS Configuration: NS20 window

2. In the Zone Domain Name window, choose **Fully qualified domain name** and type `10.23.172.in-addr-arpa` (answer 8 in Table 16-1 on page 369), as shown in Figure 16-33. Click **Next** to continue.

> **Tip:** Do not forget to type the period at the end of the fully qualified domain name.

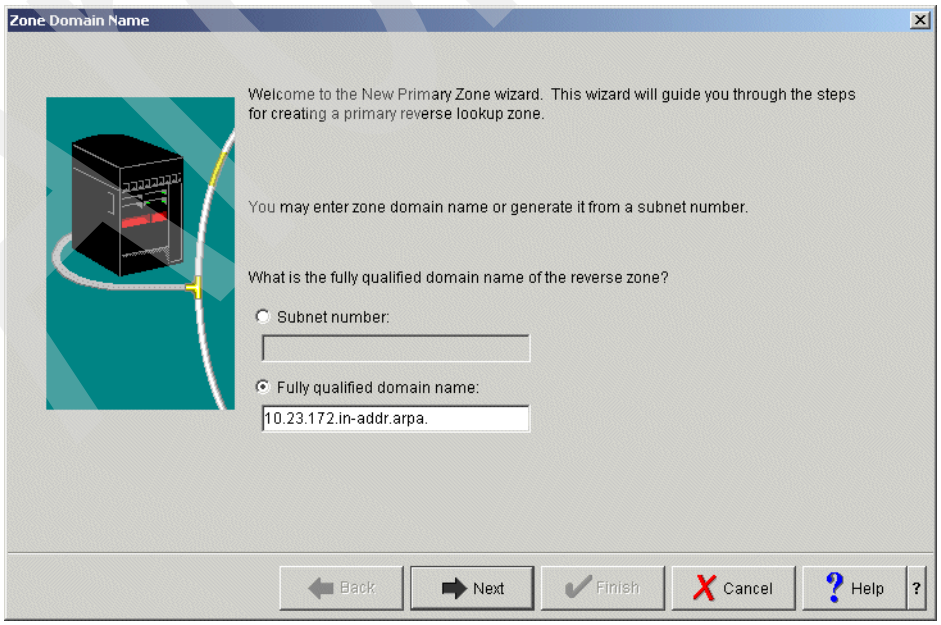


Figure 16-33 Zone Domain Name window

3. In the Name Servers window, select **as20.itsoroch.ibm.com** and click **Edit**, as shown in Figure 16-34. Click **Next** to continue.

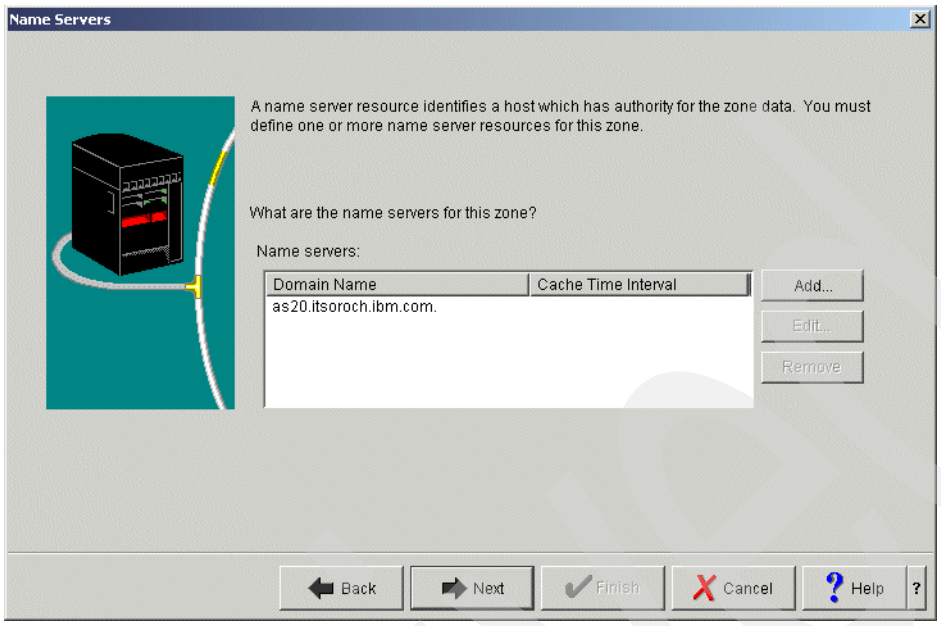


Figure 16-34 Name Servers window

4. In the Edit Name Server (NS) window, select **Cache Time Interval (NS TTL)**. Type 1 and choose **days** (answer 5 in Table 16-1 on page 369), as shown in Figure 16-35. Click **OK**.

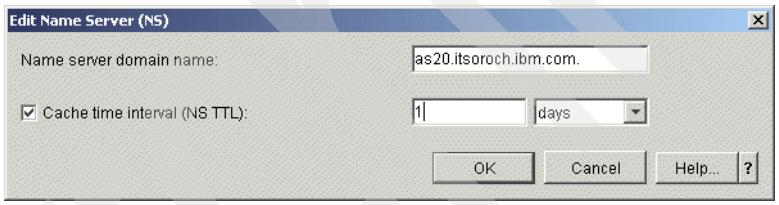


Figure 16-35 Edit Name Server (NS) window

In the Name Servers window, click **Next** to continue.

5. In the Static or Dynamic Zone window, choose **Perform Dynamic Updates** (answer 6 in Table 16-1 on page 369), as shown in Figure 16-36. Click **Next** to continue.

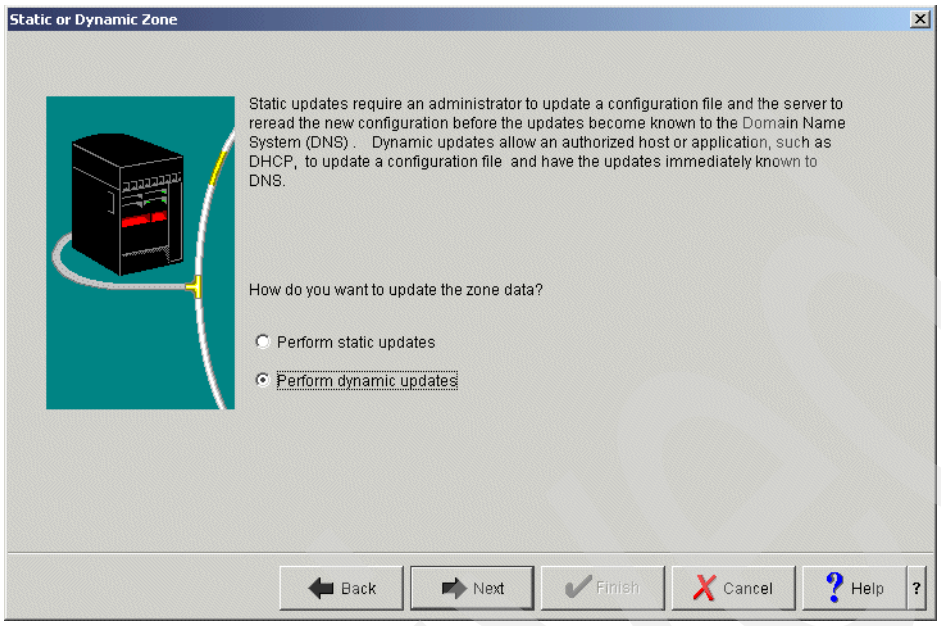


Figure 16-36 Static or Dynamic Zone window

6. In the Allow Update window, click **Add**, as shown in Figure 16-37.

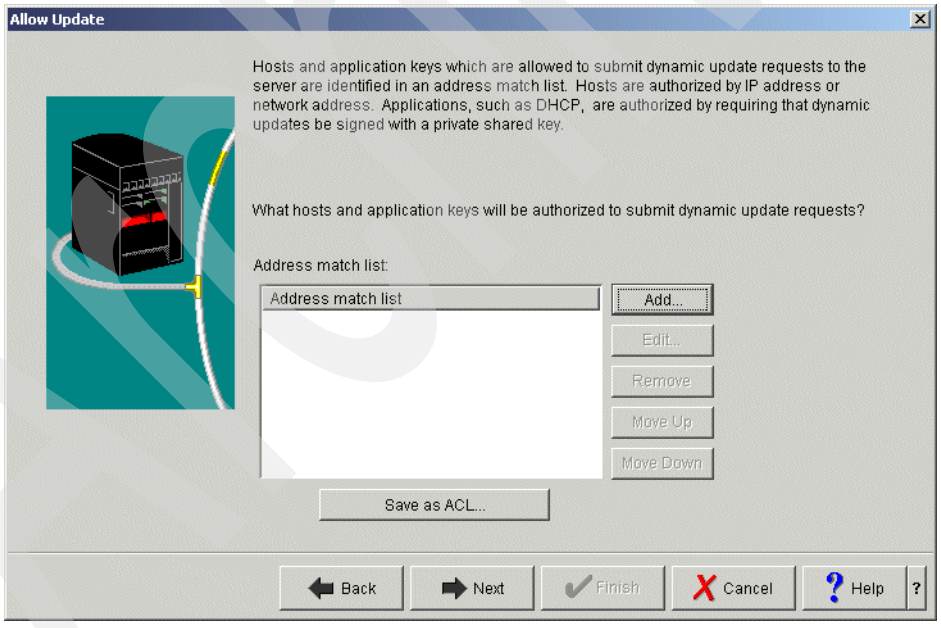


Figure 16-37 Allow Update window

7. In the Address Match List Element window, make sure that **IP Address** is selected as the Address match list element type. Type `172.23.10.1` (answer 1 in Table 16-1 on page 369) for the IP address. Choose **Allow access to this IP address** as shown in Figure 16-38. Click **OK** to continue. In the Allow Update window, click **Next** to continue.

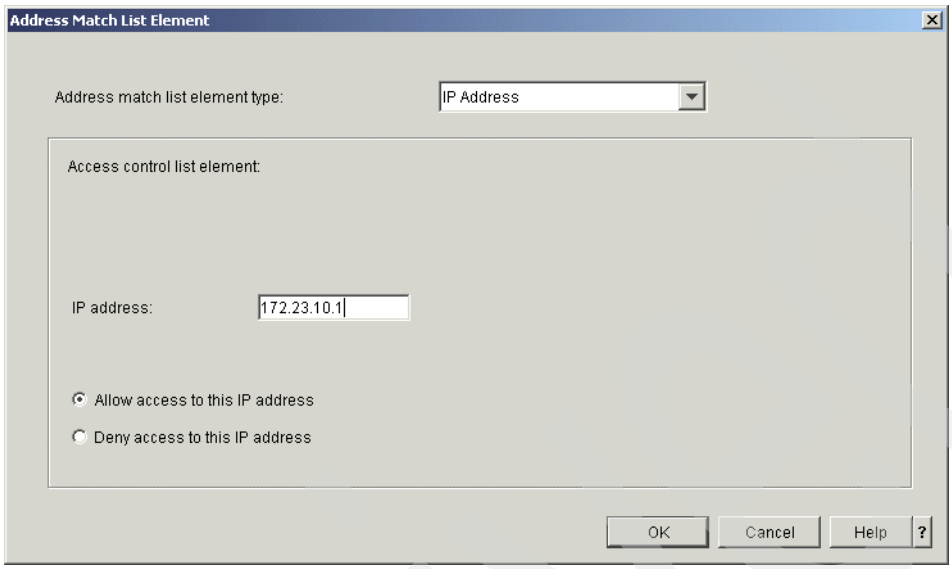


Figure 16-38 Address Match List Element window

8. If all of your entries are correct in the Summary window (Figure 16-39), click **Finish** to continue.

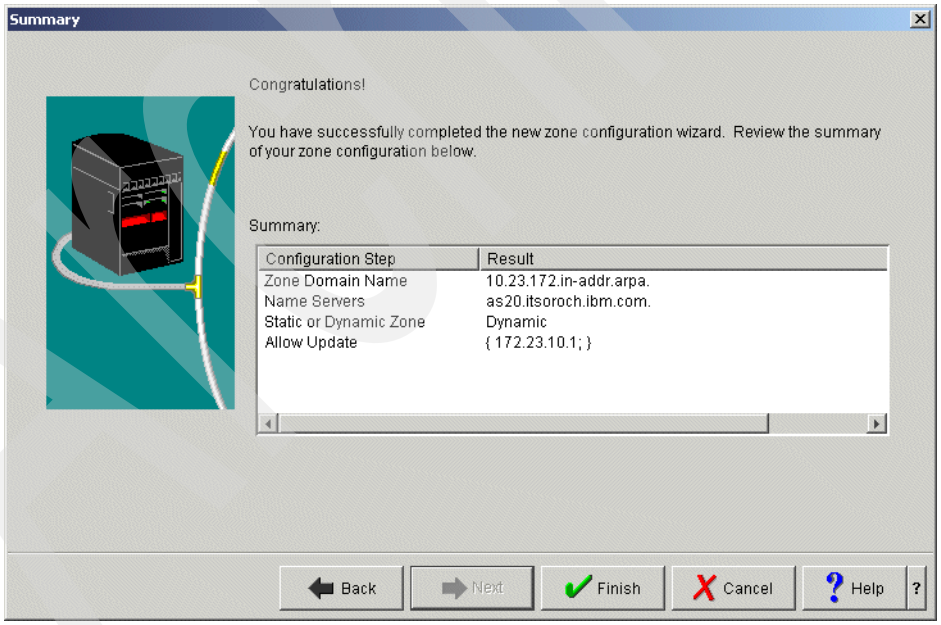


Figure 16-39 Summary window

9. In the DNS Configuration - NS20 window (Figure 16-40), right-click **Primary Zone 10.23.172-in-addr.arpa.** and choose **Properties**.

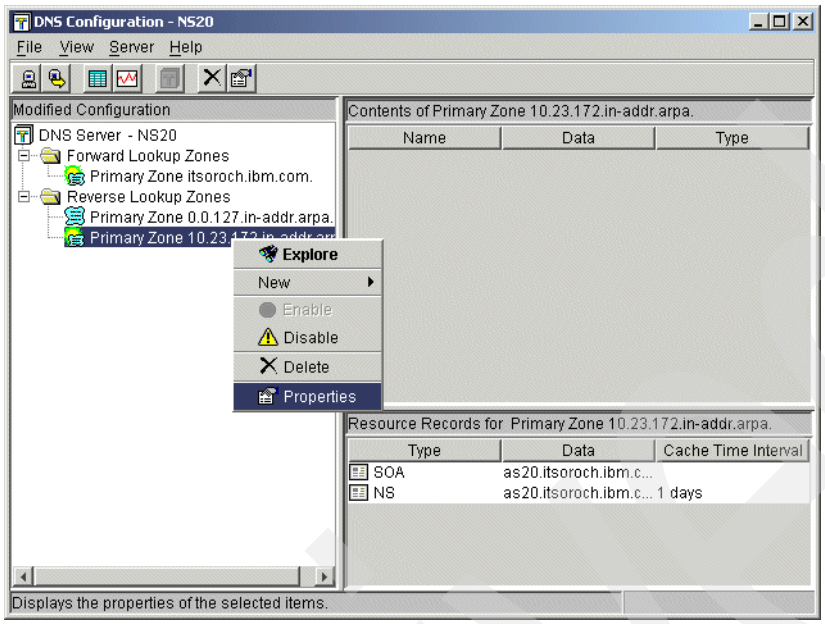


Figure 16-40 DNS Configuration - NS20 window

10. In the Primary Zone Properties - 10.23.172-in.addr.arpa. window, click the **Options** tab. Expand **Access Control**, as shown in Figure 16-41.

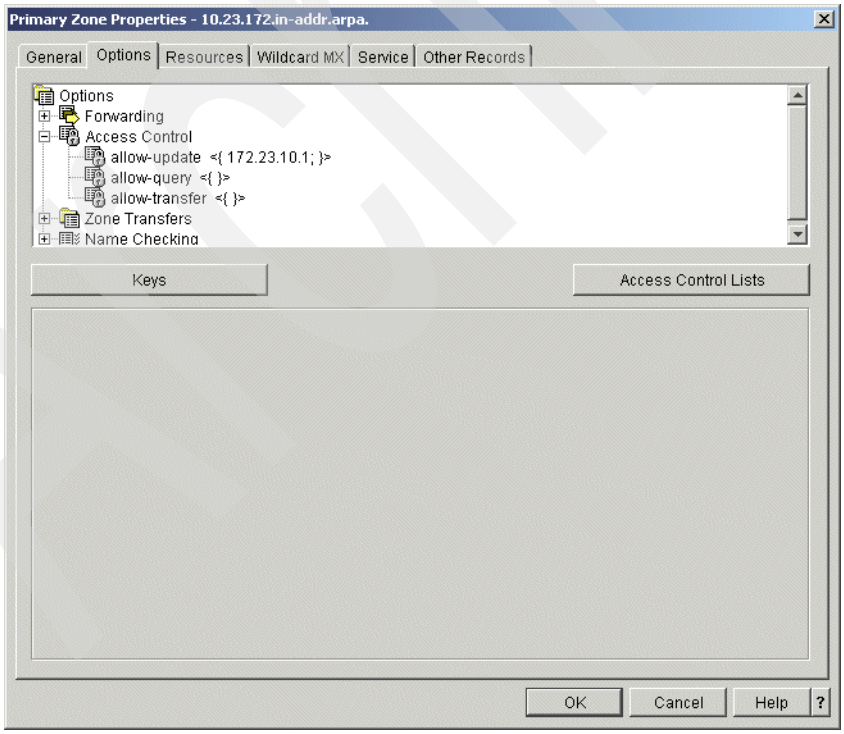


Figure 16-41 Primary Zone Properties - 10.23.172-in-addr.arpa. window

11. Click **allow-query**. Choose **Access Control List** for Match list element type. Click **Add** as shown in Figure 16-42.

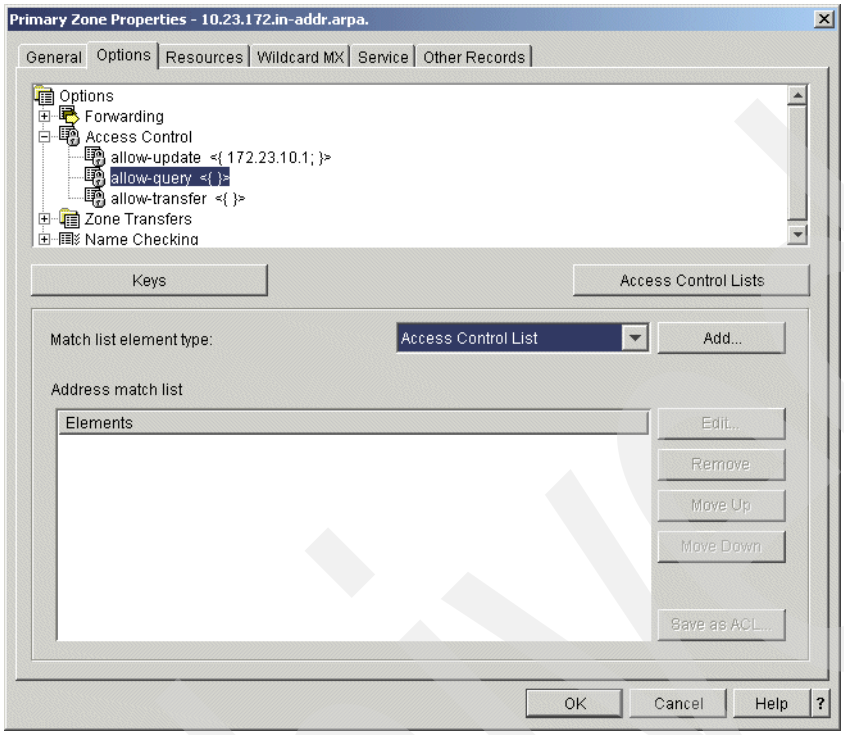


Figure 16-42 Primary Zone Properties - 10.23.172-in-addr.arpa. window

12. In the Access Control List window, choose **any**. Select **Allow access to this access control list** as shown in Figure 16-43. This allows the Query request from any client to resolve IP address or host name. Click **OK**.

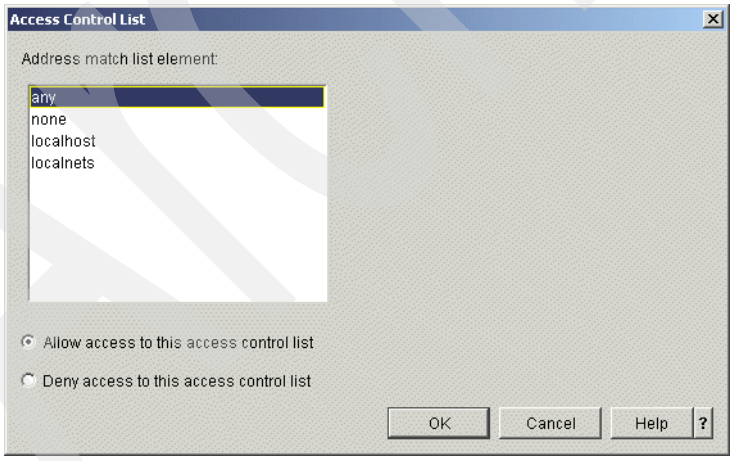


Figure 16-43 Access Control List window

13. In the Primary Zone Properties - 10.23.172-in.addr.arpa. window, click the **Resources** tab. Choose **SOA**, and click **Edit** as shown in Figure 16-44.

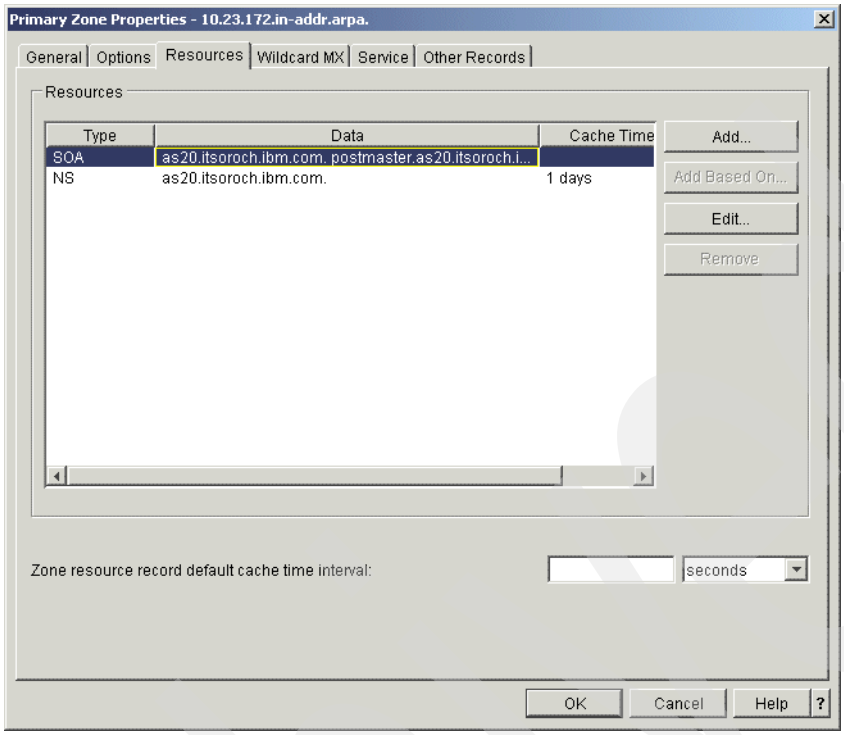


Figure 16-44 Primary Zone Properties - 10.23.172-in.addr.arpa. window

14. In the Add/Edit Resource - 10.23.172.in.addr.arpa. window, check **Start of Authority cache time interval (SOA TTL)**. Type a 1 and choose **days** (answer 7 in Table 16-1 on page 369) as a value of SOA TTL, as shown in Figure 16-45. Click **OK**.

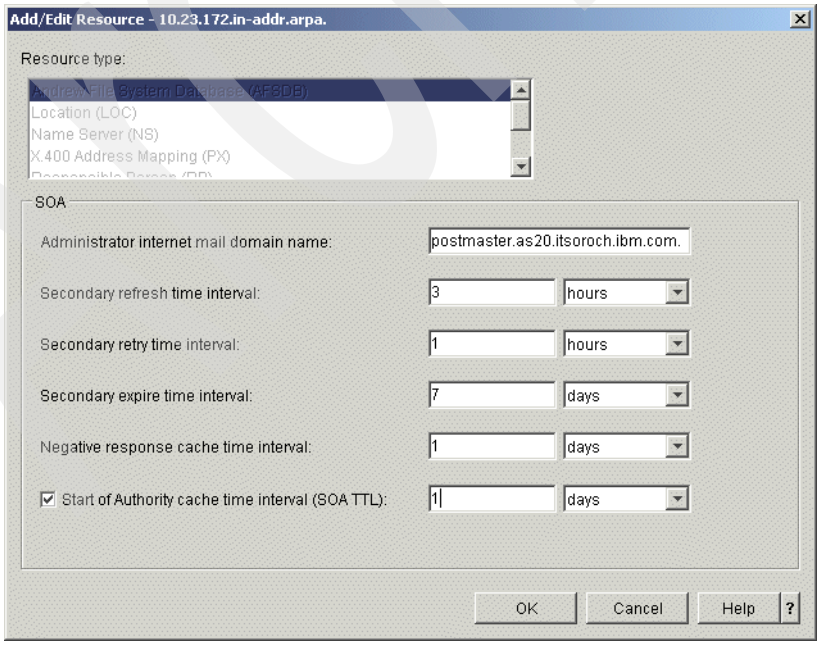


Figure 16-45 Add/Edit Resource - 10.23.172-in.addr.arpa. window

15. In the DNS Configuration - NS20 window, right-click **Primary Zone 10.23.172-in-addr-arpa.** Click **New** → **Host** as shown in Figure 16-46.

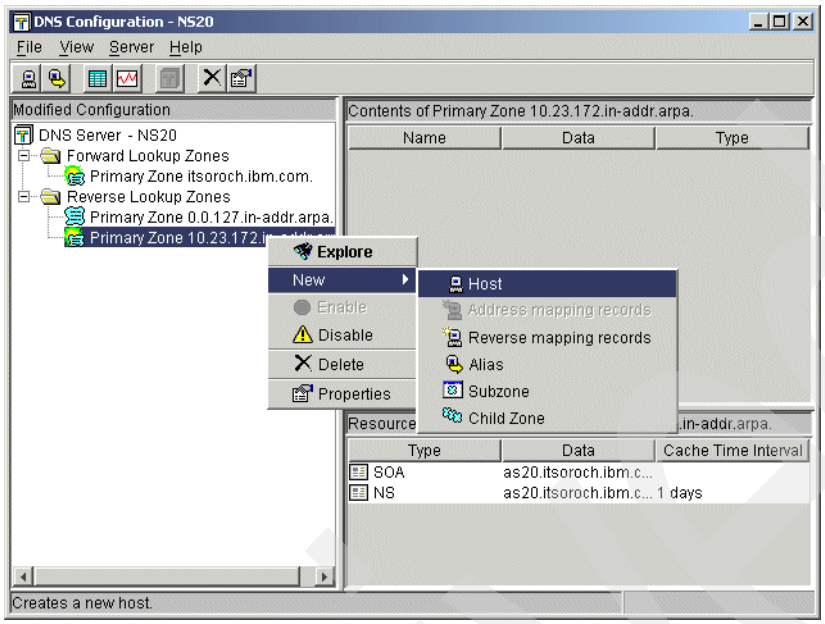


Figure 16-46 DNS Configuration -NS20 window

16. In the New Host window, choose **IP address** and type `172.23.10.1` (answer 1 in Table 16-1 on page 369), as shown in Figure 16-47. Click **Next** to continue.

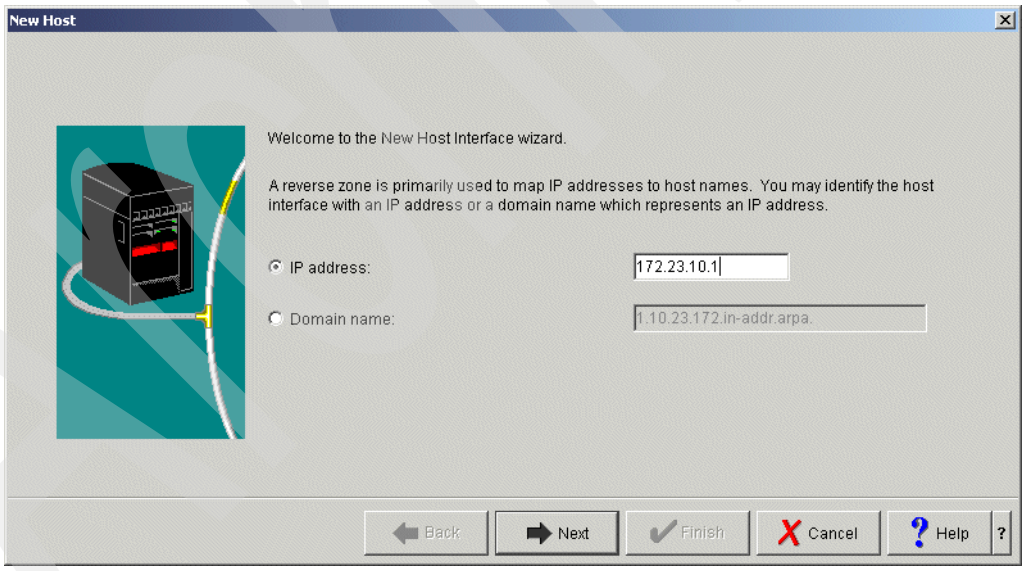


Figure 16-47 New Host window

17. In the New Host Resources window, click **Add** as shown in Figure 16-48.

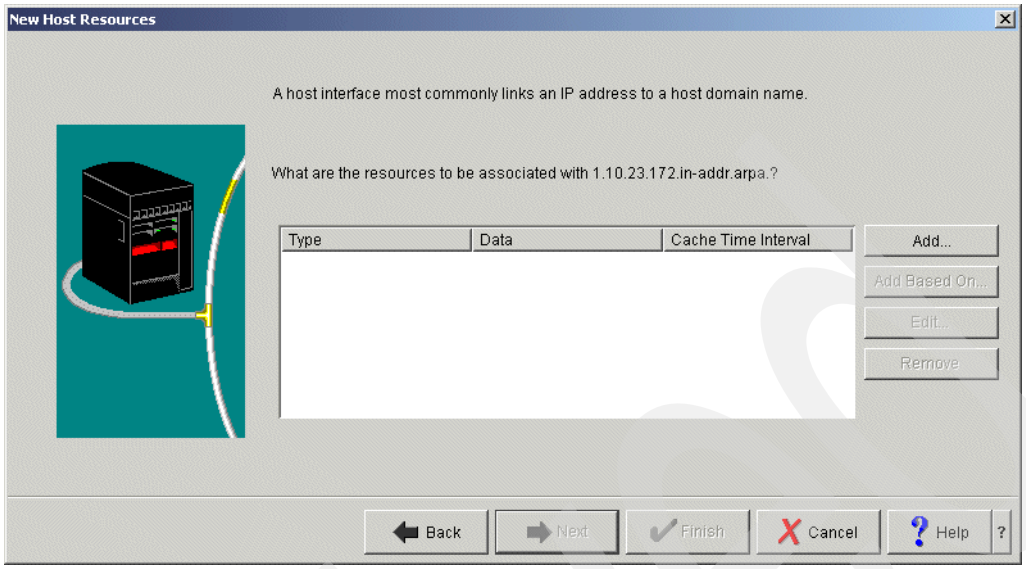


Figure 16-48 New Host Resources window

18. In the Add/Edit Resource - 1.10.23.172.in-addr.arpa window, choose **Reverse Mapping (PTR)**. Type `as20.itsoroch.ibm.com.` for the fully qualified host domain name. Check **Cache time interval (PTR TTL)**, then type 1 and choose **days** (answer 5 in Table 16-1 on page 369), as shown in Figure 16-49. Click **OK** to continue.

> **Tip:** Do not forget to type the period at the end of the fully qualified domain name.

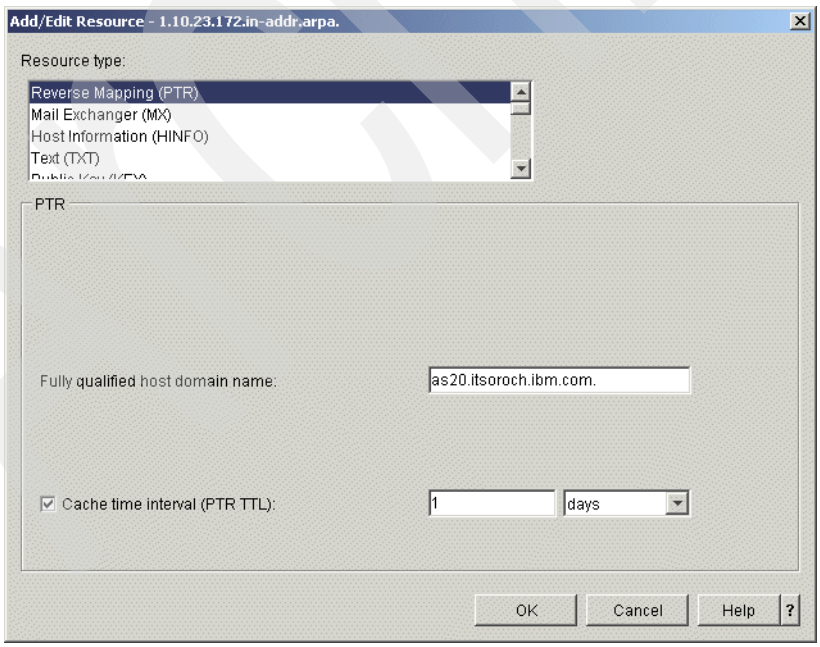


Figure 16-49 Add/Edit Resource: 1.10.23.172.in-addr.arpa window

In the New Host Resources window, click **Finish**.

19. In the DNS Configuration - NS20 window, click **File** → **Save Configuration** to save this configuration, as shown in Figure 16-50. Click **Close**.

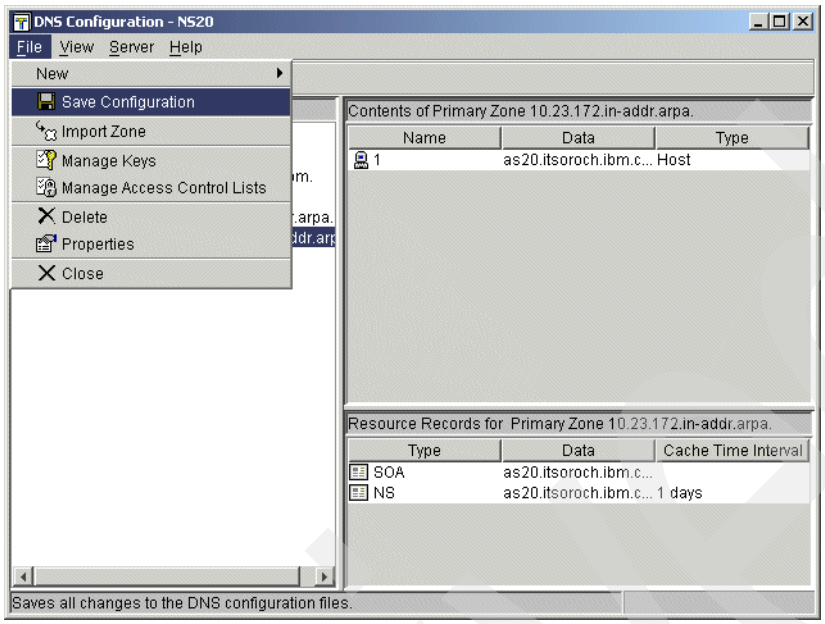


Figure 16-50 DNS Configuration - NS20 window

Step 4: Test the configuration

Now you are ready to start the DNS server. This section demonstrates the procedures to determine whether the configuration has been created correctly:

1. In the iSeries Navigator, right-click the DNS server instance that you have created through this procedure. In this example, right-click the name server instance **NS20,** as shown in Figure 16-51. Click **Start**. Wait until the DNS server comes up.

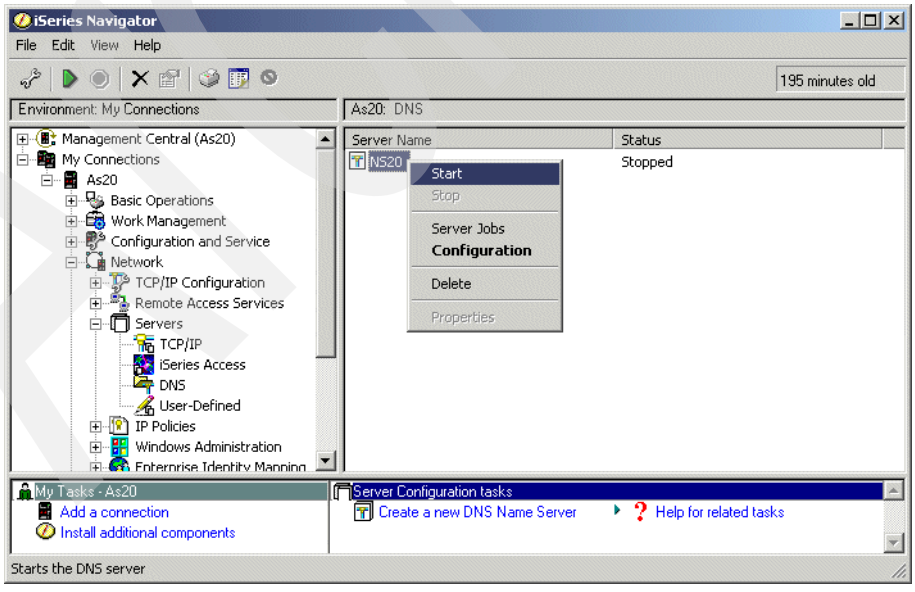


Figure 16-51 iSeries Navigator window

2. If the DNS server came up successfully, there is no serious mistake in the server definition, and you may skip to step 3 on page 398.

   If the DNS server did not come up, try to isolate the problem by observing the job log.

   a. Right-click the DNS server instance name. In this example, right-click **NS20** and choose **Server Jobs**, as shown in Figure 16-52.

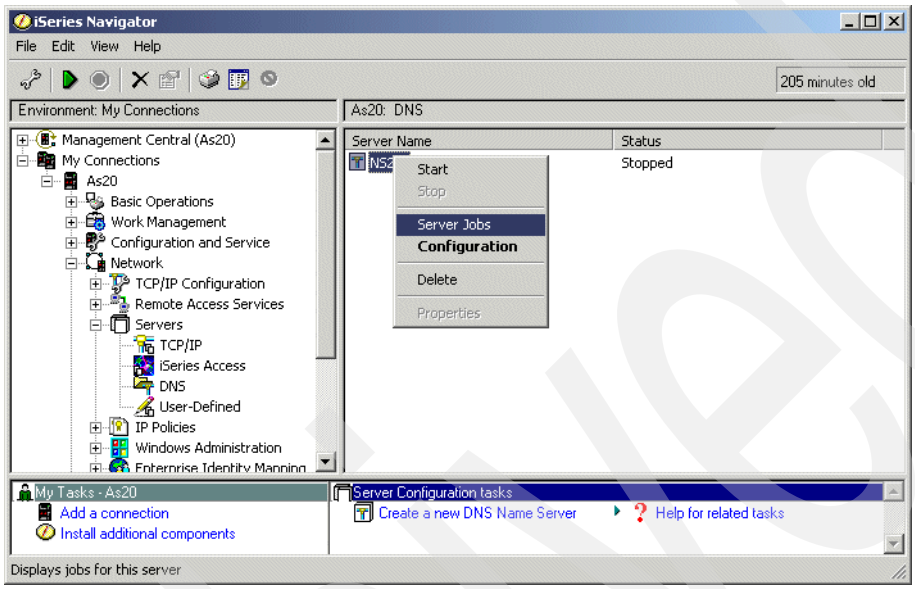


Figure 16-52 iSeries Navigator window

   b. In the Jobs - As20 window (Figure 16-53), right-click the DNS server job name (in this example, right-click **Qtobdns20**) and click **Job Log**.

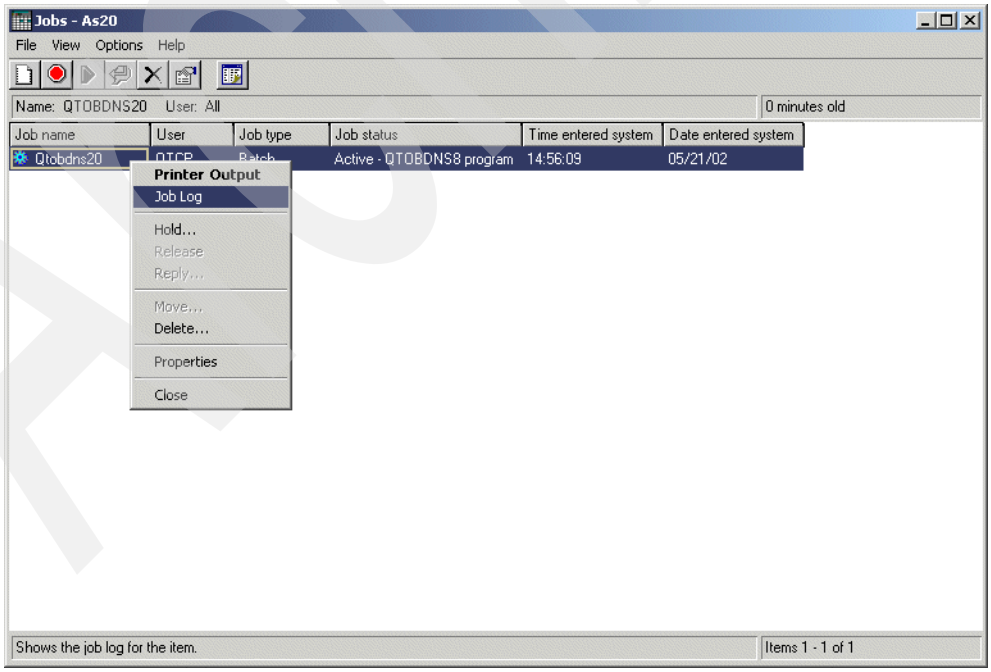


Figure 16-53 Jobs - As20 window

c. In the Job Log window, try to find any errors on the log. Figure 16-54 is an example of the NS20 DNS server starting without errors. If you find errors in the Job Log, ask IBM Support for further assistance.

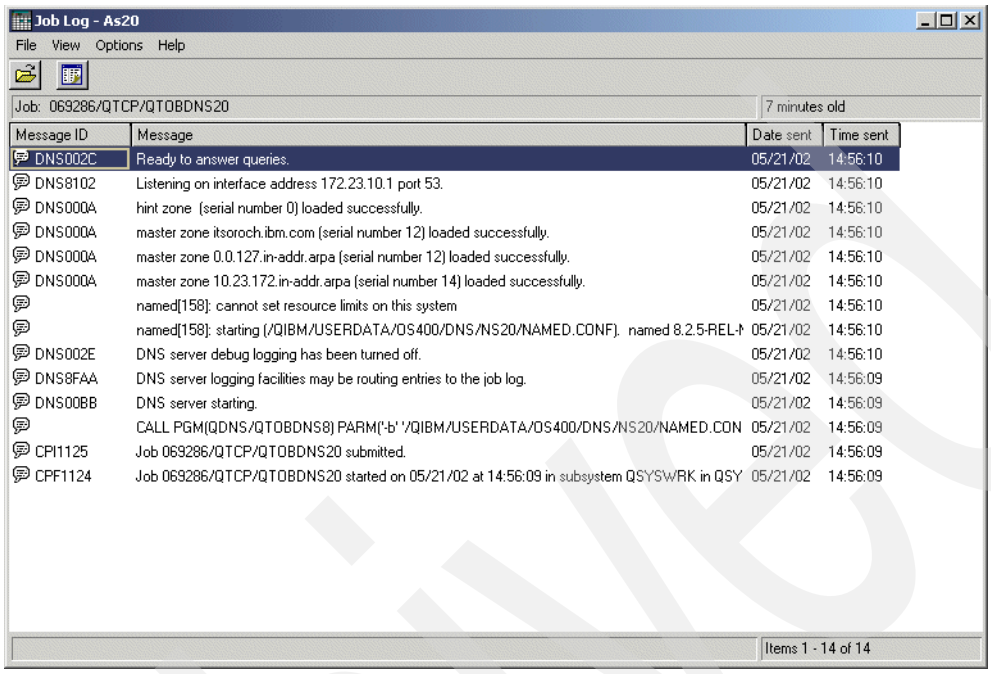


Figure 16-54 Job Log - As20 window

3. In the iSeries Navigator window, expand **Network** → **Servers**. The status of the DHCP server should be Started. If not, right-click **DHCP** and select **Start** from the context menu.
4. In this example, we assume that Windows 2000 is used for the client. We will now confirm that the System i DHCP server updates both A and PTR records dynamically in the System i DNS.

   At a Windows 2000 command prompt, type `ipconfig /release` then type `ipconfig /renew`. This recycles the leased IP address on the Windows 2000 client.

After you receive a response of ipconfig /renew, type `ipconfig /all` to show detailed information. Figure 16-55 shows the output of this example.

Note: Windows 95 and Windows 98 require you to reboot the system to effectively release and renew the leased IP address from the System i DHCP server.

```
ipconfig /all

Windows 2000 IP Configuration
   Host Name . . . . . . . . . . . . : mk5
   Primary DNS Suffix  . . . . . . . :
   Node Type . . . . . . . . . . . . : Hybrid
   IP Routing Enabled. . . . . . . . : No
   WINS Proxy Enabled. . . . . . . . : No
   DNS Suffix Search List. . . . . . : itsoroch.ibm.com

Ethernet adapter:

   Connection-specific DNS Suffix  . : itsoroch.ibm.com
   Description . . . . . . . . . . . : IBM Ethernet Credit Card Adapter II
   Physical Address. . . . . . . . . : 00-06-29-14-C9-CB
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   IP Address. . . . . . . . . . . . : 172.23.10.11
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :
   DHCP Server . . . . . . . . . . . : 172.23.10.1
   DNS Servers . . . . . . . . . . . : 172.23.10.1
   Lease Obtained. . . . . . . . . . : Tuesday, May21, 2002 3:02:53 PM
   Lease Expires . . . . . . . . . . : Wednesday, May22, 2002 3:02:53 PM
```

Figure 16-55 Windows 2000 command prompt: ipconfig /all window

If both the DHCP server and DNS server are configured correctly, they should have the same IP address (172.23.10.1).

The Domain name itsoroch.ibm.com is assigned from the DHCP server at the time of DHCP lease. It is seen in `DNS Suffix Search List and Connection-specific DNS Suffix.`

In the next step, we check your DNS server using the **`nslookup`** command to get the query response from DNS server. If your DNS server is working properly, it will respond with query answers, which includes an answer for A or PTR queries.

Tip: It is possible to check A or PTR records by observing Host records using iSeries Navigator if the DNS server is stopped. If the DNS server is up and running, it keeps changes to the A and PTR records cached in memory, and these records are invisible while the DNS server is running. That is why we use the **`nslookup`** command to check it.

5. On the i5/OS 5250 Command Entry panel enter Start DNS Query (NSLOOKUP) as shown in Figure 16-56.

```
                      Command Entry                          AS20
                                                          Request level:   4
Previous commands and messages:

  (No previous commands or messages)




                                                                        Bottom
Type command, press Enter.
===> nslookup
```

Figure 16-56 i5/OS Command Entry panel: nslookup

6. This opens the NSLOOKUP initial panel shown in Figure 16-57.

   Confirm that the Default Server is set to 172.23.10.1. The System i will default to the IP address specified in the DNS search order (refer to Figure 16-3 on page 371).

> **Tip:** At any time, you can type `help` in NSLOOKUP to see a list of valid commands and their syntax.

```
Default Server:  as20.itsoroch.ibm.com
  Address:  172.23.10.1

  >




===>
```

Figure 16-57 NSLOOKUP initial panel

7. On the NSLOOKUP panel, type the IP address that was leased by DHCP server. In this example, type `172.23.10.11` to ensure that the PTR record was updated dynamically by the DHCP server. If the dynamic update was performed, you will receive the response from the DNS server. Figure 16-58 shows the sample output in this configuration.

```
Default Server:  as20.itsoroch.ibm.com
   Address:  172.23.10.1

   >
 > 172.23.10.11
   Server:  as20.itsoroch.ibm.com
   Address:  172.23.10.1

   Name:    mk5.itsoroch.ibm.com
   Address:  172.23.10.11

   >

 ===> 172.23.10.11
```

Figure 16-58 NSLOOKUP panel: confirm A record written to DDNS

8. On the NSLOOKUP panel, type the host name of the client. In this example, type `mk5.itsoroch.ibm.com` to confirm that the A record was updated dynamically by the DHCP server. If the dynamic update was performed, you will receive the response from DNS server. Figure 16-59 shows the sample output in this configuration.

```
 >
 > mk5.itsoroch.ibm.com
   Server:  as20.itsoroch.ibm.com
   Address:  172.23.10.1

   Name:    mk5.itsoroch.ibm.com
   Address:  172.23.10.11

   >

 ===> mk5.itsoroch.ibm.com
```

Figure 16-59 NSLOOKUP panel: confirm PTR record written to DDNS

9. Now we test whether DHCP server deletes A and PTR records dynamically right after the client releases the IP address. On the Windows 2000 command prompt, type `ipconfig /release` to release the leased IP address.

10. On the i5/OS NSLOOKUP panel, type `172.23.10.11` to confirm that the PTR record was deleted dynamically by the DHCP server. If the dynamic update was performed, you will receive the response from the DNS server indicating that the record could not be found. Figure 16-60 shows the sample output in this configuration.

```
>
> 172.23.10.11
  Server:  as20.itsoroch.ibm.com
  Address:  172.23.10.1

  Server as20.itsoroch.ibm.com can not find host 172.23.10.11.
  Host or domain does not exist.
  >

===> 172.23.10.11
```

Figure 16-60 NSLOOKUP panel: confirm delete of A record from DDNS

11. On the NSLOOKUP panel, type `mk5.itsoroch.ibm.com` to confirm that the A record was deleted dynamically by the DHCP server. If the dynamic update was performed, you will receive a response from the DNS server indicating that the record could not be found. Figure 16-61 shows the sample output in this configuration.

```
>
> mk5.itsoroch.ibm.com
  Server:  as20.itsoroch.ibm.com
  Address:  172.23.10.1

  Server as20.itsoroch.ibm.com can not find host mk5.itsoroch.ibm.com.
  Server failed.
  >

===> mk5.itsoroch.ibm.com
```

Figure 16-61 NSLOOKUP panel: confirm deletion of PTR record from DDNS

We have confirmed that the DHCP server updates A and PTR records dynamically. This ends the procedure to test the configuration.

16.2 Single DDNS and DHCP servers without secured updates

This procedure configures a single DDNS server on your system. In this scenario, the DHCP server and DDNS server are configured on different servers, and the DHCP server updates A and PTR records to the DDNS server dynamically right after the DHCP server assigns an IP address to the client. We show how to configure the single DDNS server step-by-step.

16.2.1 Scenario overview

You might choose this scenario if these conditions apply:

- If you want to configure the DDNS server and DHCP server on two different servers.
- If you do not need the secondary DNS server as a fault-tolerant backup.
- If this DNS server is used in the private network and no security consideration is required to isolate the network from the public network.

Sample network configuration

Figure 16-62 shows the sample network configuration of this scenario.

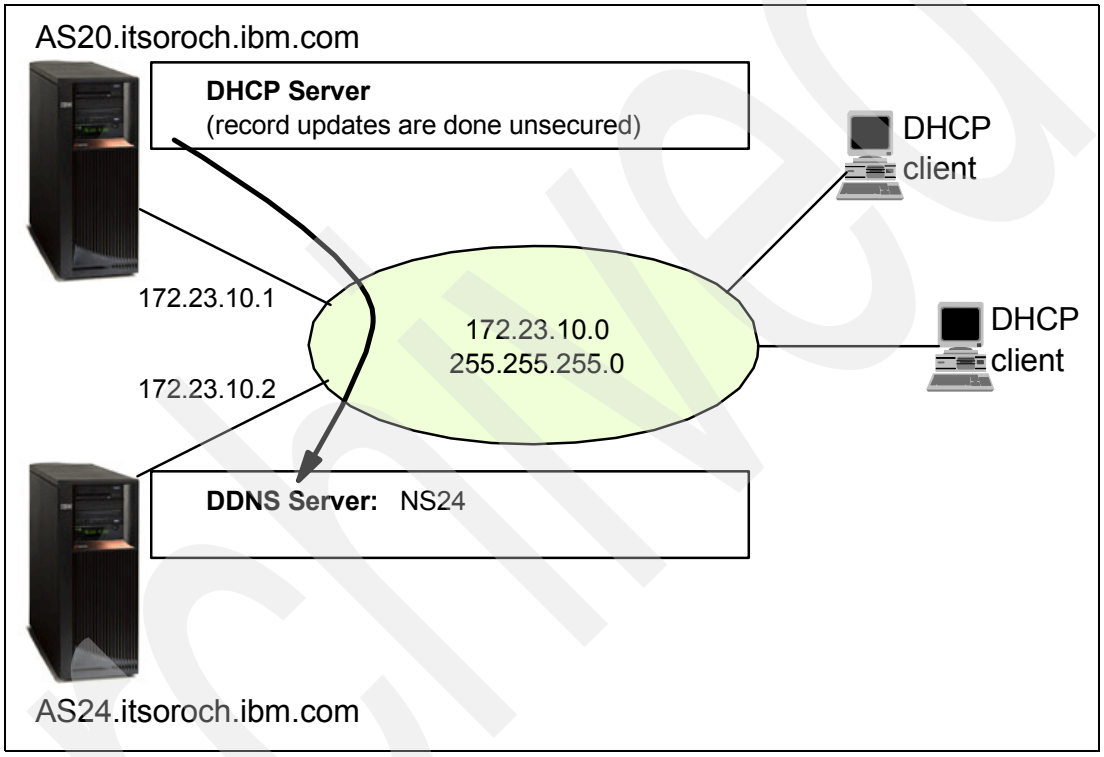


Figure 16-62 Single DDNS server and DHCP server on different servers with unsecured record updates

16.2.2 Planning: single DDNS and DHCP servers without secured updates

Table 16-2 shows the planning worksheet for preparing the required parameters to configure the single DDNS server and DHCP server on different servers without secured records updates. We have filled in our answers for each question in the adjacent Scenario answers column.

Table 16-2 Planning worksheet for the single DDNS server and DHCP server on different servers

No.	Questions for creating the single DDNS server and DHCP server on different servers without secured records update scenario	Scenario answers
1	What is the TCP/IP Domain Information seen in CFGTCP option 12 panel on AS24 (DNS server) side? - Host name - Domain name - Domain search list - Domain name server IP address	 as24 itsoroch.ibm.com itsoroch.ibm.com 172.23.10.2
2	What is the DDNS server instance name?	NS24
3	What TCP interface is used for listening for the query from the clients? (Query is the request to resolve the host name or IP address)	172.23.10.2
4	Do you want the DDNS server to start when TCP/IP starts?	Yes
5	What is the Cache Time Interval (NS TTL) for the DNS server? Cache Time Interval means the time-out value of each record on the DNS cache. When the A or PTR record is queried by a client, the record retains on the cache. After the NS TTL time, the A or PTR record on the DNS cache is discarded.	1 day
6	Do you want the DNS server to perform Dynamic Updates?	Yes
7	What is the Start of Authority cache time interval (SOA TTL) value? Start of Authority is the main definition of DNS.	1 day
8	What is the subnet of your network? What subnet mask is assigned for your TCP Interface?	Subnet 172.23.10.x Netmask 255.255.255.0 Now the domain name for the Reverse lookup is: 10.23.172.in-addr.arpa.
9	What is the TCP/IP Domain Information seen in CFGTCP option 12 panel on AS20 (DHCP server) side: - Host name - Domain name - Domain search list - Domain name server IP address	 AS20 itsoroch.ibm.com itsoroch.ibm.com 172.23.10.2

16.2.3 Configuration: single DDNS and DHCP servers without secured updates

In this scenario, we create a single DDNS configuration in the following steps:

- Step 1: Confirm the TCP domain information:
  - Step 1a: Confirm the information on AS20 (DHCP server) side.
  - Step 1b: Confirm the information on AS24 (DNS server) side.
- Step 2: Confirm DHCP configuration for dynamic updating on AS20.
- Step 3: Create the single DDNS server and DHCP server on the different server without secured records update configuration.
  - Step 3a: Creating the new DNS instance NS24.
  - Step 3b: Creating new Primary Zone in a Forward Lookup Zone.
  - Step 3c: Creating new Primary Zone on Reverse Lookup Zone.
- Step 4: Test the configuration.

Step 1: Confirm the TCP domain information

It is important to confirm the correct domain name setup for configuring the DNS server.

Step 1a: Confirm the information on AS20 (DHCP server) side

To do this:

1. In the iSeries Navigator window, expand **Network**.
2. Right-click **TCP/IP Configuration** and choose **Properties** as shown in Figure 16-63.

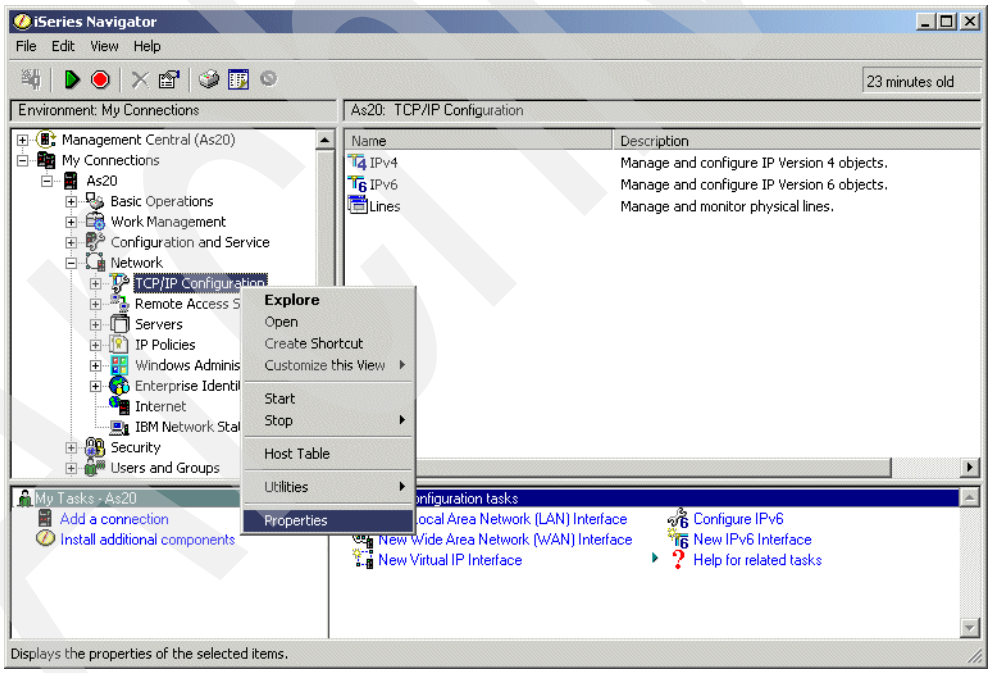


Figure 16-63 iSeries Navigator window

3. In the TCP/IP Configuration Properties window, click the **Host Domain Information** tab as shown in Figure 16-64.

> **Tip:** You can get the same information via 5250 command entry with Configure TCP/IP (CFGTCP) option 12=Change TCP/IP domain information.

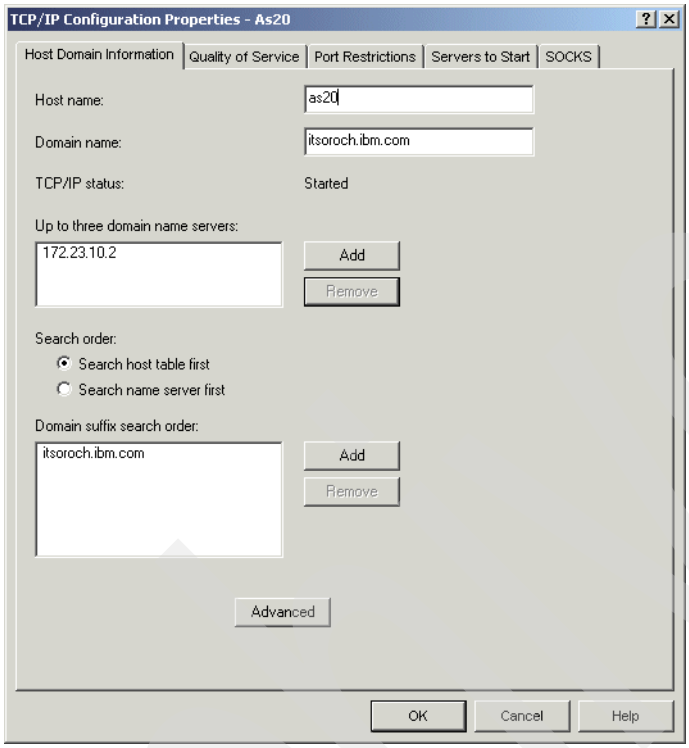


Figure 16-64 TCP/IP Configuration Properties window

The fields are:

- Host name, domain name

  The combination of the host name and the domain name is called a fully qualified domain name, and it is used to present the host name of the DNS server that you are going to create. In this example, the fully qualified domain name is as20.itsoroch.ibm.com.

  This fully qualified domain name is case sensitive. In some servers, uppercase letters are used to present the host name or the domain name, but lowercase letters are usually used. If either the host name or domain name includes uppercase letters, consider changing the host name or the domain name to use only lowercase letters.

- Up to three domain name servers

  This should include your DNS server's IP address. (In this example, 172.23.10.2 must be included. This scenario does not call for a secondary or other DNS.)

- Domain suffix search order

  This should include your domain name (in this example, itsoroch.ibm.com).

Step 1b: Confirm the information on AS24 (DNS server) side

To do this:

1. In the iSeries Navigator window, expand **Network**.

2. Right-click **TCP/IP Configuration** and choose **Properties**, as shown in Figure 16-65.

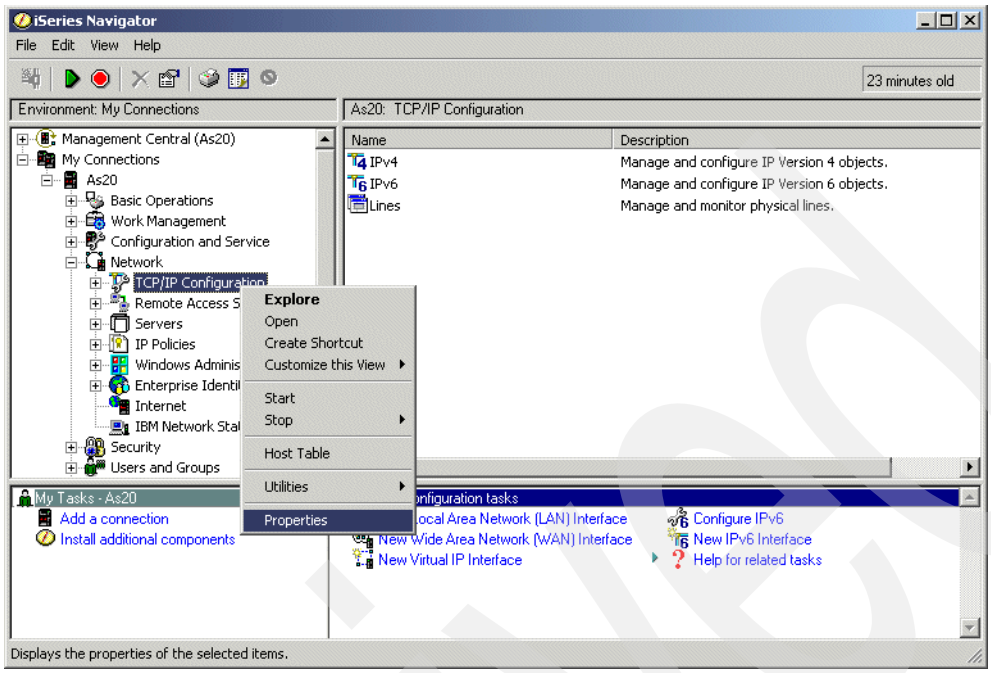


Figure 16-65 iSeries Navigator window

3. In the TCP/IP Configuration Properties window, click the **Host Domain Information** tab, as shown in Figure 16-66.

> **Tip:** You can get the same information via 5250 command entry with Configure TCP/IP (CFGTCP) option 12=Change TCP/IP domain information.

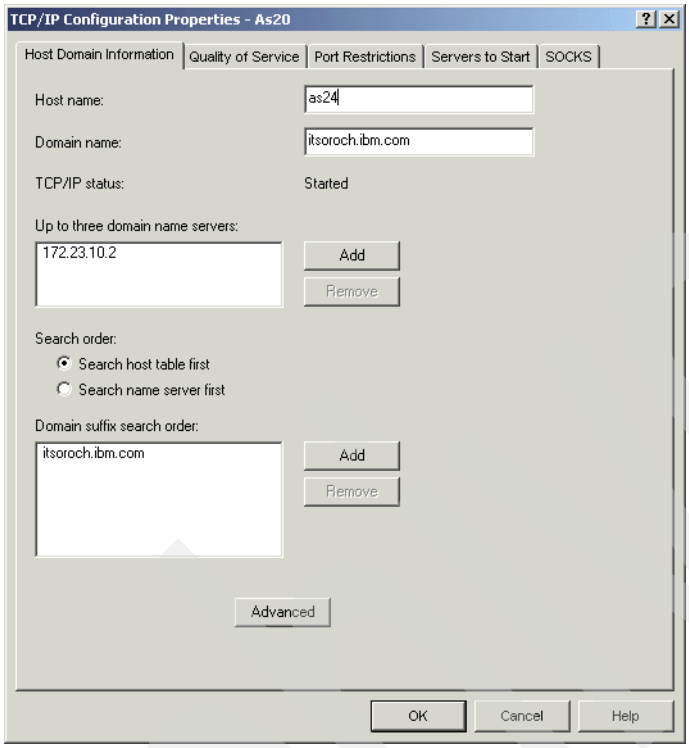


Figure 16-66 TCP/IP Configuration Properties window

The fields are:

- Host name, domain name

  The combination of the host name and the domain name is called a fully qualified domain name, and it is used to present the host name of the DNS server that you are going to create. In this example, the fully qualified domain name is as24.itsoroch.ibm.com.

  This fully qualified domain name is case sensitive. Although uppercase letters are used to present the host name or the domain name in some servers, but lowercase letters are usually used. If either the host name or domain name includes uppercase letters, consider changing the host name or the domain name to use only lowercase letters.

- Up to three domain name servers

  This should include your DNS server's IP address. (In this example, 172.23.10.2 must be included. This scenario does not call for a secondary or other DNS.)

- Domain suffix search order

  This should include your domain name (in this example, itsoroch.ibm.com).

Step 2: Confirm DHCP configuration for dynamic updating on AS20

If you want your DHCP server to update A and PTR records dynamically, you must configure it to perform dynamic update. Check the DHCP server configuration on your AS20 system using the procedure below:

1. In the iSeries Navigator window, expand **Network**.
2. Expand **Servers**.
3. Click **TCP/IP**. In the right-side pane, right-click **DHCP** and choose **Configuration**, as shown in Figure 16-67.

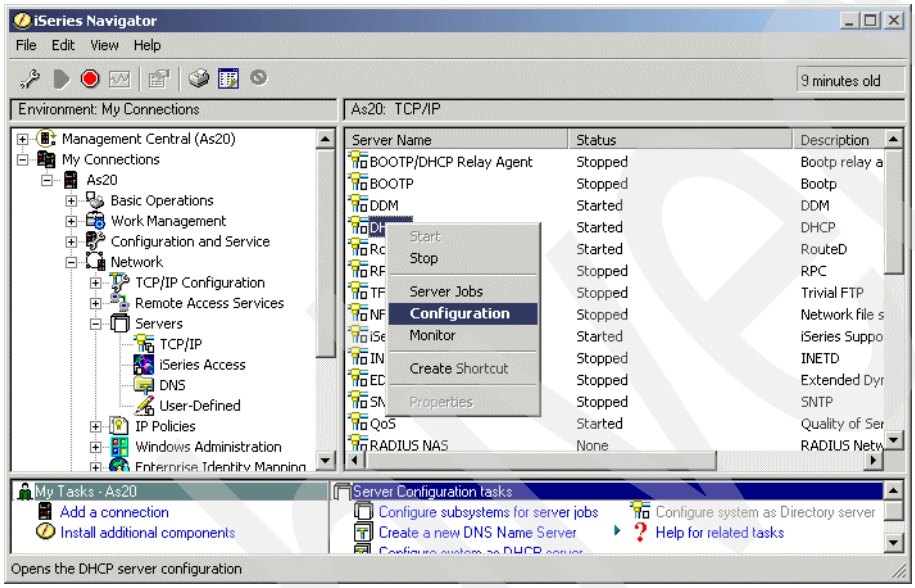


Figure 16-67 iSeries Navigator window

4. In the DHCP Server Configuration - As20 window, right-click **Global**. Choose **Properties** from the context menu, as shown in Figure 16-68.

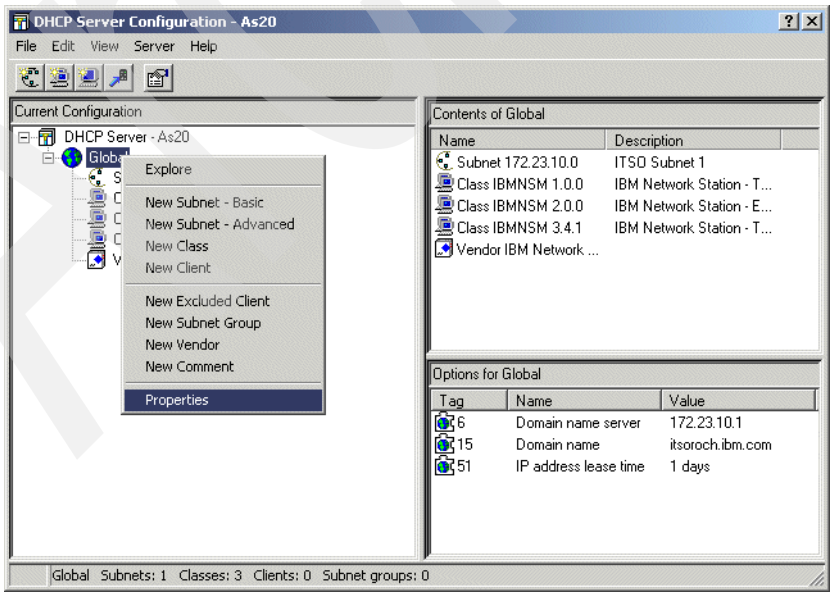


Figure 16-68 DHCP Server Configuration - As20 window

5. In the Global Properties - As20 window, click the **Dynamic DNS** tab. Make sure that **DHCP server updates both A and PTR records** is selected, as shown in Figure 16-69. This performs the Dynamic A and PTR records update from DHCP server to DNS server.

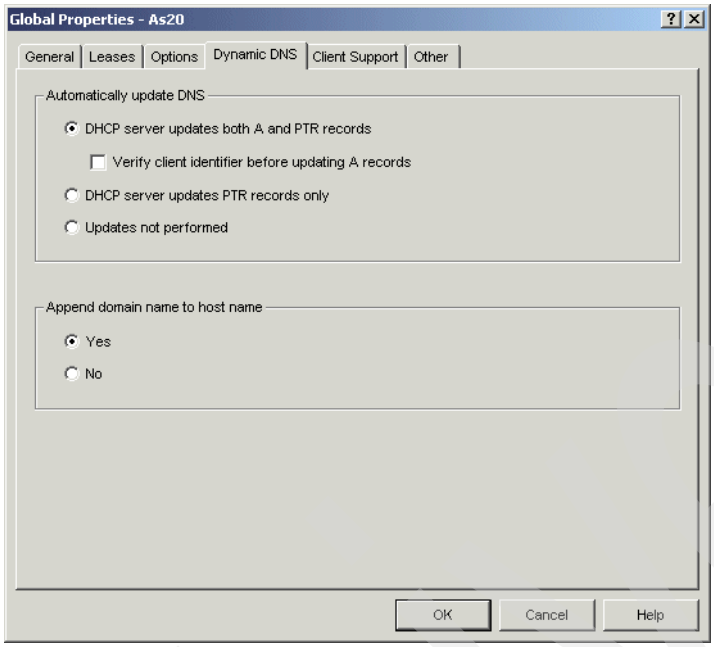


Figure 16-69 Global Properties - As20 window

6. In the Global Properties - As20 window, click the **Options** tab. Make sure that options **6 - Domain Name Server** and **15 - Domain Name** are in the Selected options column. This domain name server IP address and domain name will be sent to the client when the DHCP server is going to lease an IP address to the client. The domain name server and domain name information will remain on the client to recognize the DNS IP address and the domain name, if the client is configured to obtain DNS IP address and domain name from DHCP server.

   In this example, the domain name associated with tag 15 in the Options window should match the Host Domain Information shown in Figure 16-66 on page 408.

Click option **6** under the Options tab (Figure 16-70). In this example, confirm that the DNS IP address 172.23.10.2 is defined in the lower pane.

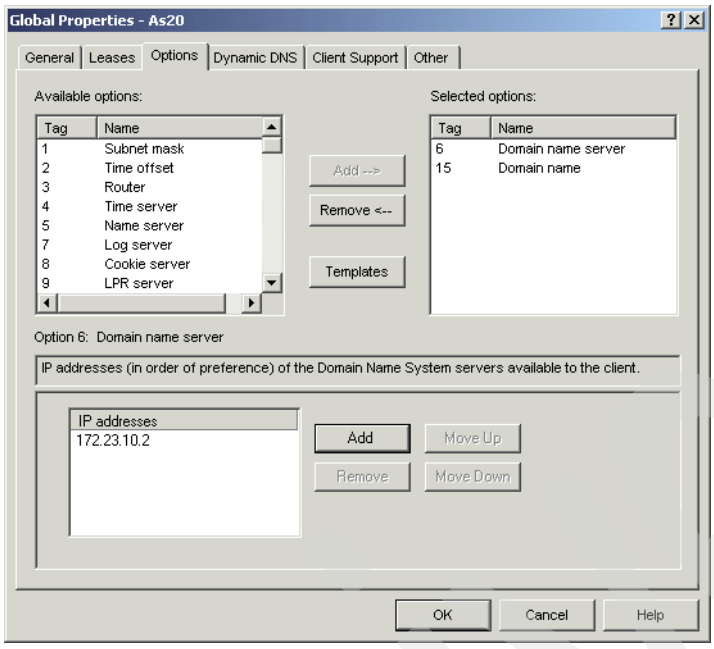


Figure 16-70 Global Properties - As20 window

7. Click option **15** as shown in Figure 16-71. In this example, confirm that domain name itsoroch.ibm.com is defined.

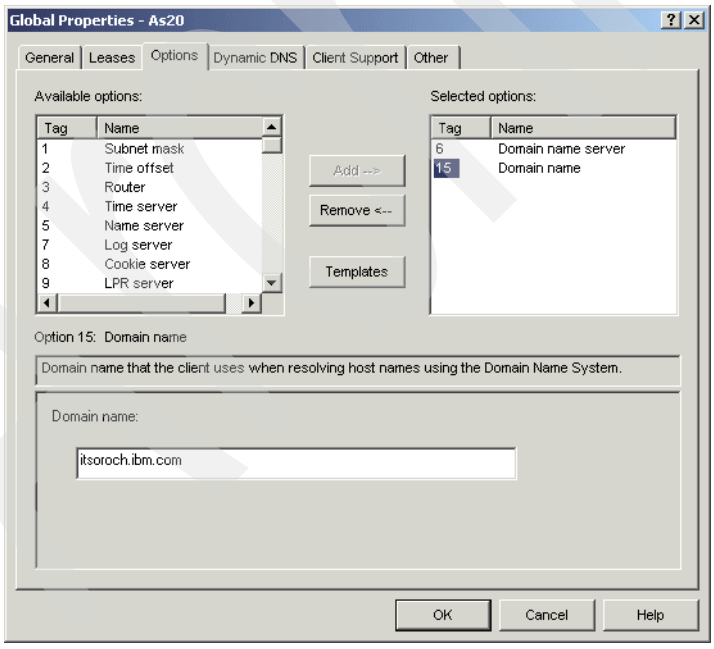


Figure 16-71 Global Properties - As20 window

8. Click **Cancel**. In the DHCP Configuration - As20 window, choose File - Close on the task bar. This ends the procedure to check the configuration of the DHCP server.

Step 3: Create the single DDNS server and DHCP server on the different server without secured records update configuration

This is done in three steps:

- Step 3a: Creating the new DNS instance NS24
- Step 3b: Creating new Primary Zone in a Forward Lookup Zone
- Step 3c: Creating new Primary Zone on Reverse Lookup Zone

Step 3a: Creating the new DNS instance NS24

To do this:

1. In the iSeries Navigator window, expand **Network**, then expand **Servers**.
2. Right-click **DNS** and choose **New Name Server** (Figure 16-72).

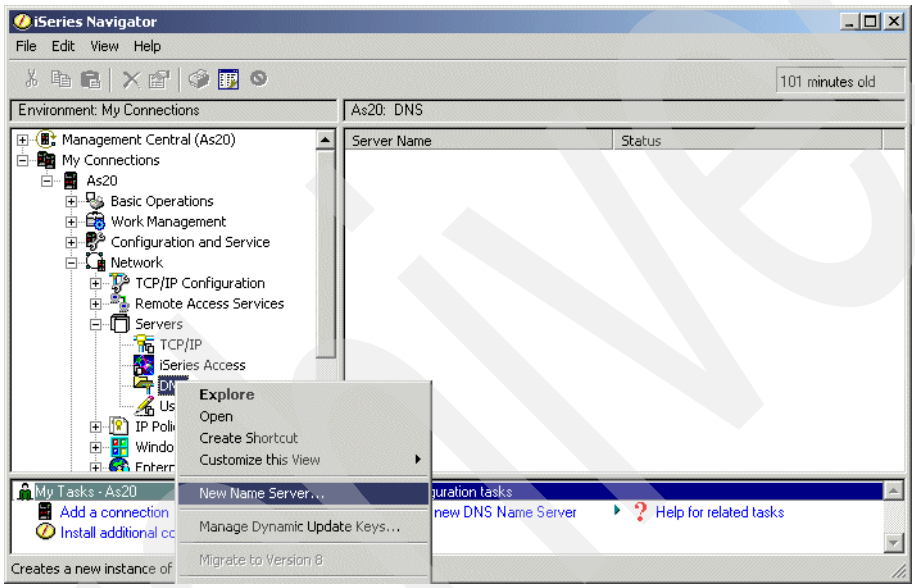


Figure 16-72 New Name Server window

3. In the New DNS Configuration window (Figure 16-73), click **Next**.

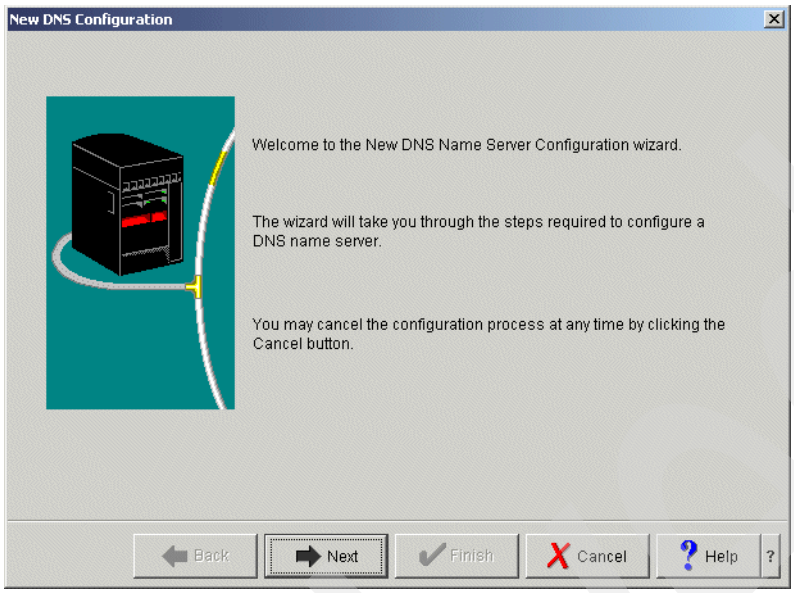


Figure 16-73 New DNS Configuration window

4. In the DNS Server Name window, type `NS24` as a DNS server instance name (answer 2 in Table 16-2 on page 404), as shown in Figure 16-74. Click **Next** to continue.

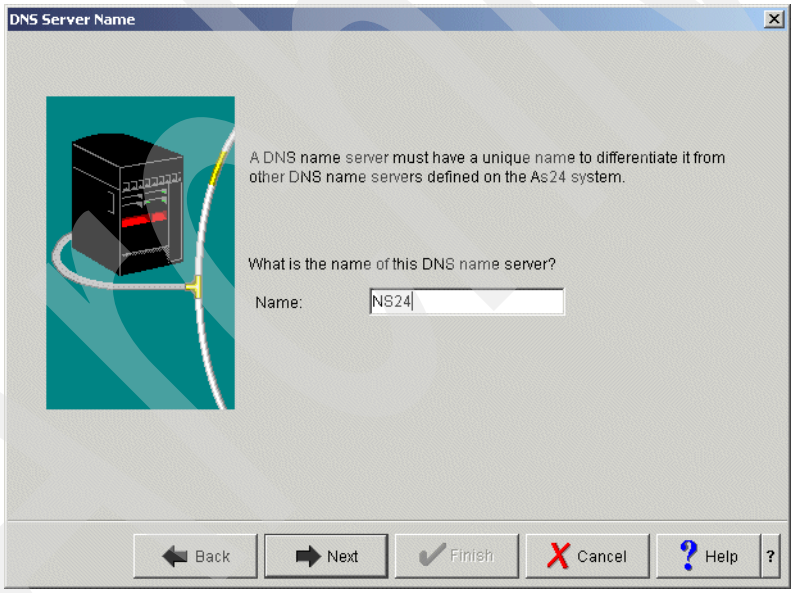


Figure 16-74 DNS Server Name window

5. In the Listen On IP Addresses window (Figure 16-75), select the IP Address **172.23.10.2** as a Query-listening TCP Interface (see answer 3 in Table 16-2 on page 404). Click **Next**.

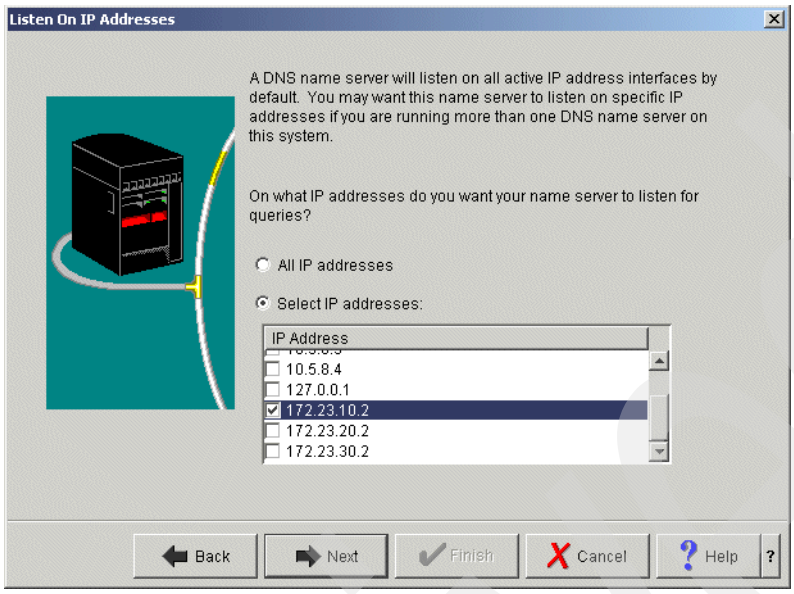


Figure 16-75 Listen On IP Addresses window

6. In the Root Servers window (Figure 16-76), if you need to add Root servers, click **Add** and add the root server IP addresses. In this scenario, considering the small office use in the company, the root server entry is not required. Click **Next** to continue.

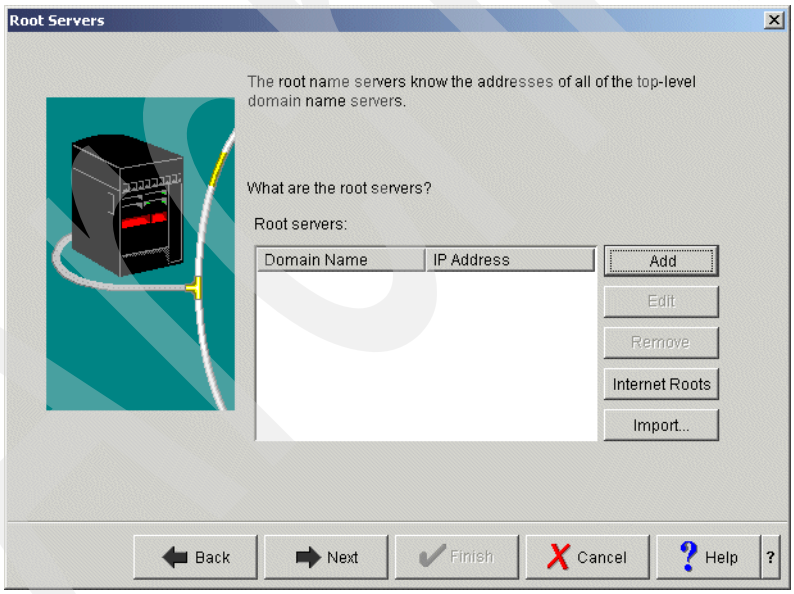


Figure 16-76 Root Servers window

7. In the Start DNS Server window, click **Yes** (answer 4 in Table 16-2 on page 404), as shown in Figure 16-77. Click **Next** to continue.

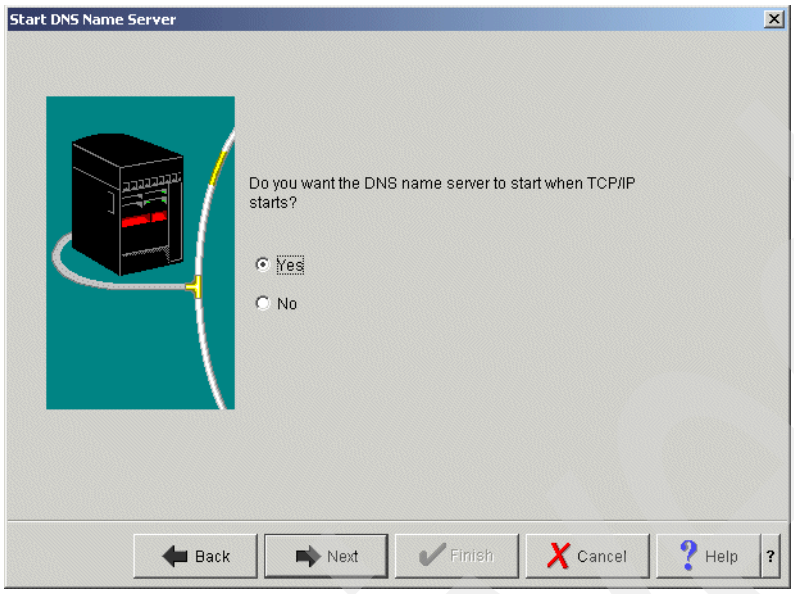


Figure 16-77 Start DNS Server window

8. In the Summary window, confirm your data as shown in Figure 16-78. Click **Finish** to continue.

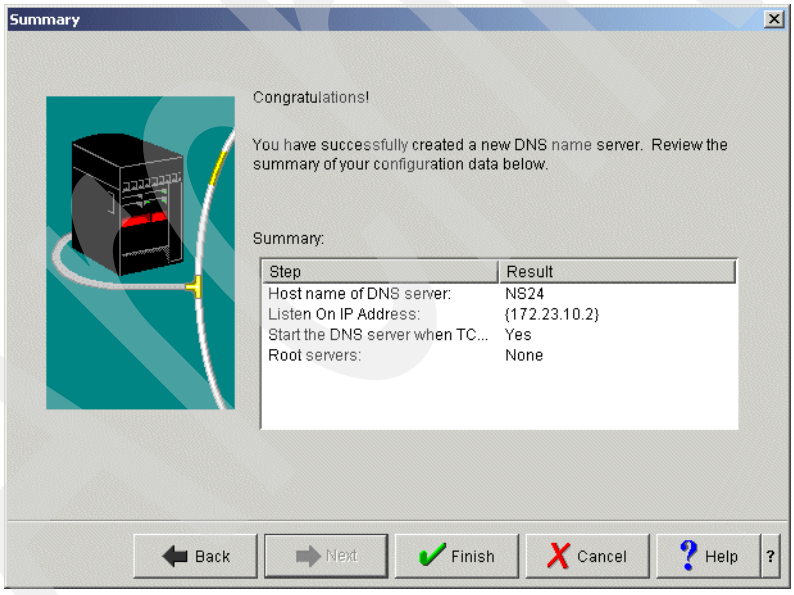


Figure 16-78 Summary window

Step 3b: Creating new Primary Zone in a Forward Lookup Zone

To do this:

1. In the iSeries Navigator window, right-click the newly created DNS instance **NS24** and select **Configuration**, as shown in Figure 16-79.

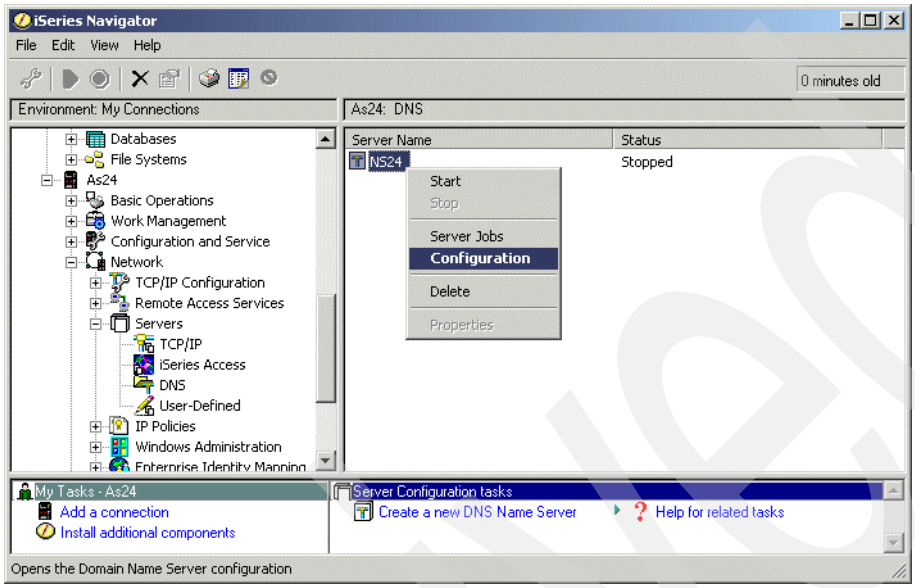


Figure 16-79 iSeries Navigator window

2. In the DNS Configuration: NS24 window, right-click **Forward Lookup Zones** and select **New Primary Zone** from the context menu, as shown in Figure 16-80.

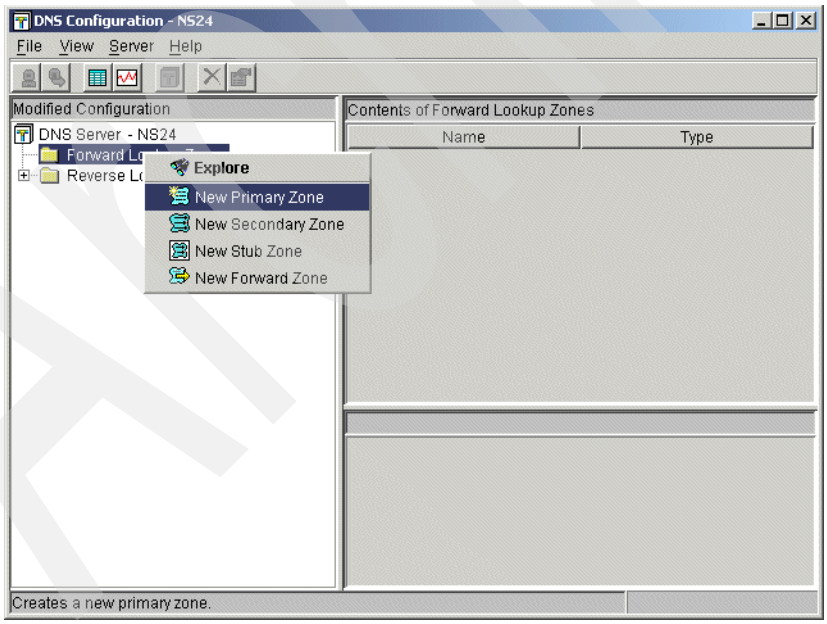


Figure 16-80 DNS Configuration: NS24 window

3. In the Zone Domain Name window, type the domain name `itsoroch.ibm.com.` (answer 1 in Table 16-2 on page 404), as shown in Figure 16-81.

> **Tip:** Do not forget to type the period at the end of the fully qualified domain name.

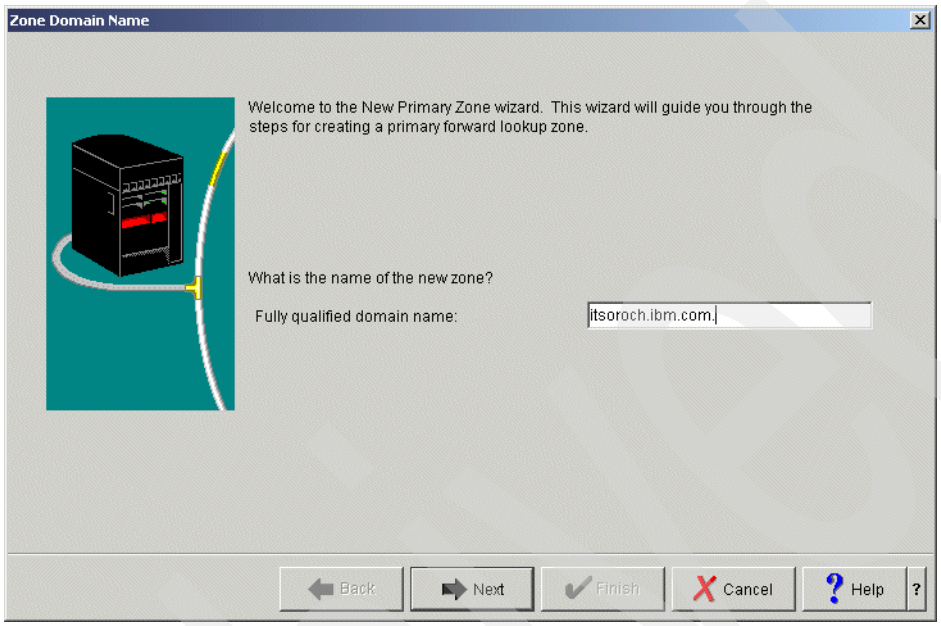


Figure 16-81 Zone Domain name window

4. In the Name Servers window (Figure 16-82), select **as24.itsoroch.ibm.com.**, and click **Edit**.

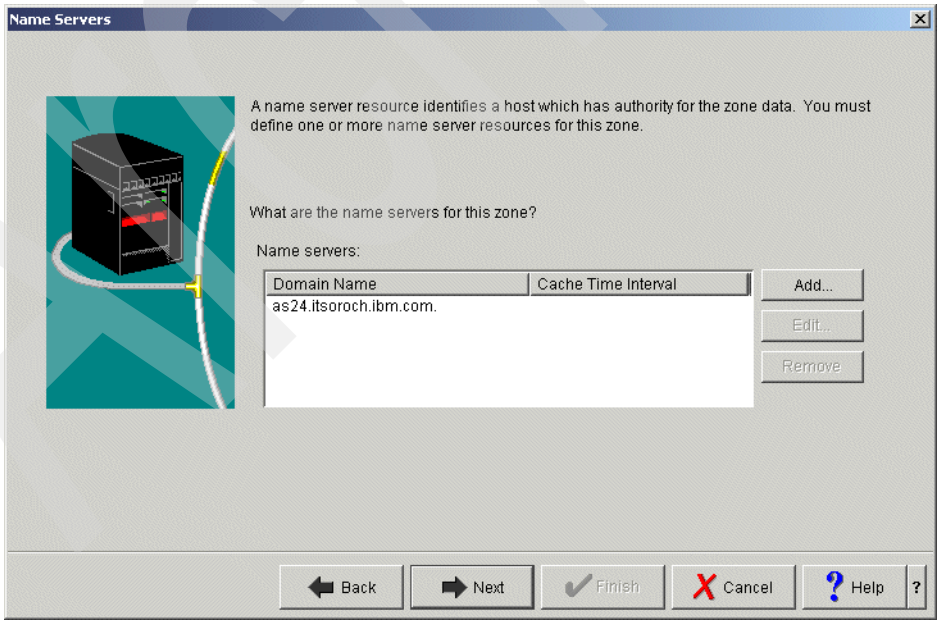


Figure 16-82 Name Servers window

5. In the Edit Name Server (NS) window, select **Cache time interval (NS TTL)**. Type 1 and choose **days** (answer 5 in Table 16-2 on page 404), as shown in Figure 16-83. Click **OK**.

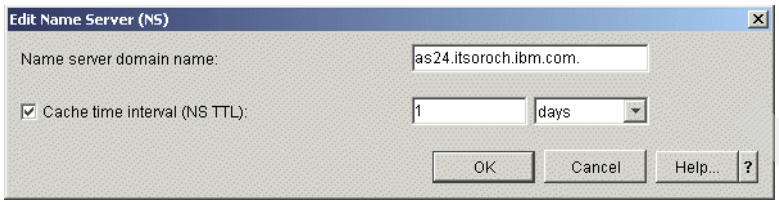


Figure 16-83 Edit Name Server (NS) window

6. In the Name Server IP Addresses window, click **Add**. Type the IP address of the as24.itsoroch.ibm.com, `172.23.10.2` (answer 1 in Table 16-2 on page 404), in the IP Address column, as shown in Figure 16-84. Click **OK** to continue.

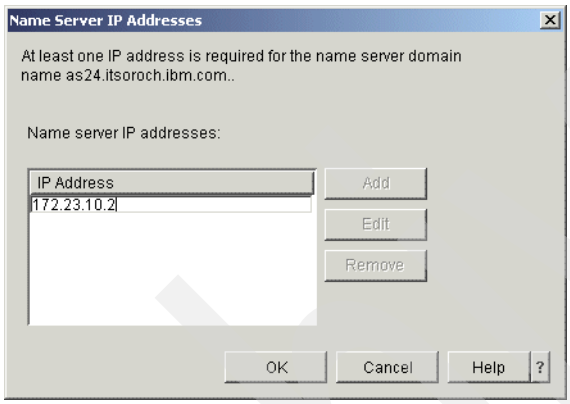


Figure 16-84 Name Server IP addresses window

In the Name servers window, click **Next**.

7. In the Static or Dynamic Zone window, choose **Perform dynamic updates** (answer 6 in Table 16-2 on page 404), as shown in Figure 16-85. Click **Next** to continue.

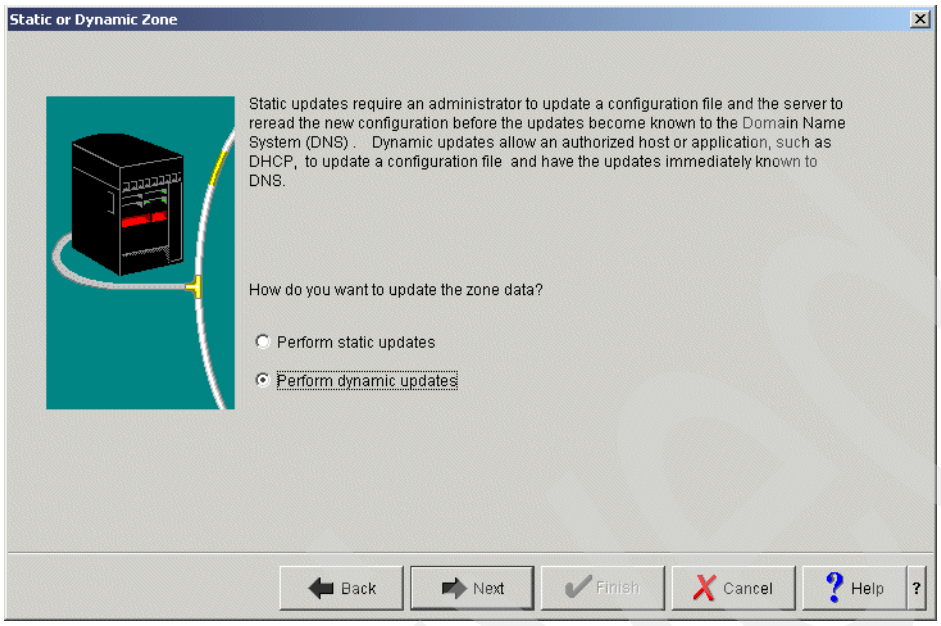


Figure 16-85 Static or Dynamic Zone window

8. In the Allow Update window (Figure 16-86), click **Add**.

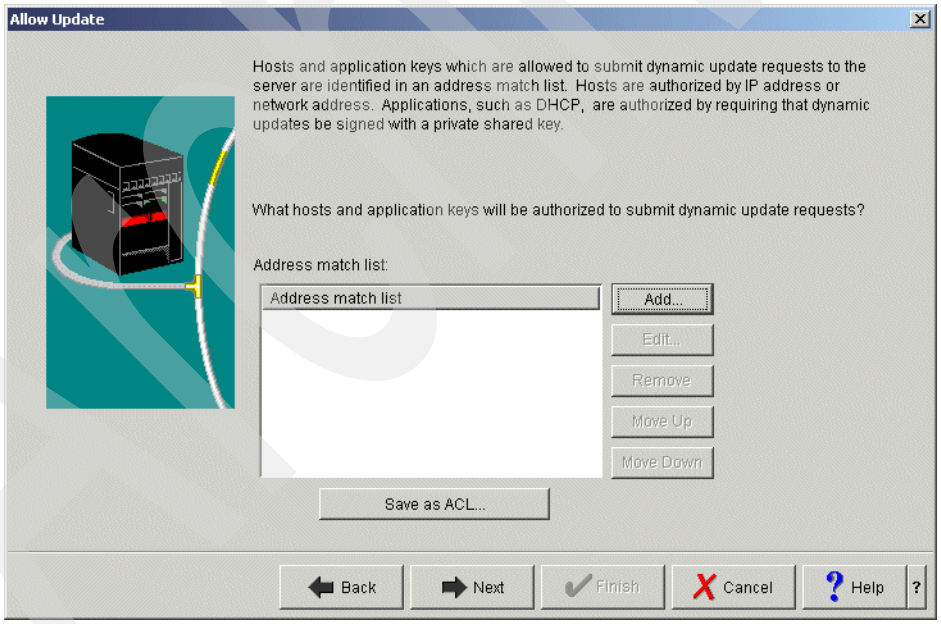


Figure 16-86 Allow Update window

9. In the Address Match List Element window, make sure that **IP Address** is selected as the type. Type `172.23.10.2` (answer 1 in Table 16-2 on page 404) for the IP address. Choose **Allow access to this IP address**, as shown in Figure 16-87. Click **OK** to return to the Allow Update window, and click **Next** to continue.

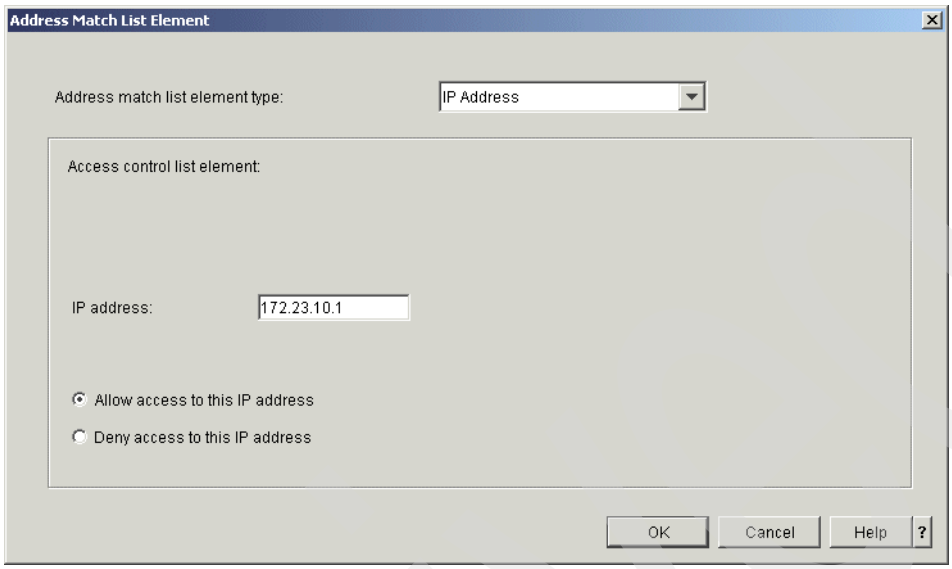


Figure 16-87 Address Match List Element window

10. In the Summary window (Figure 16-88), click **Finish**.

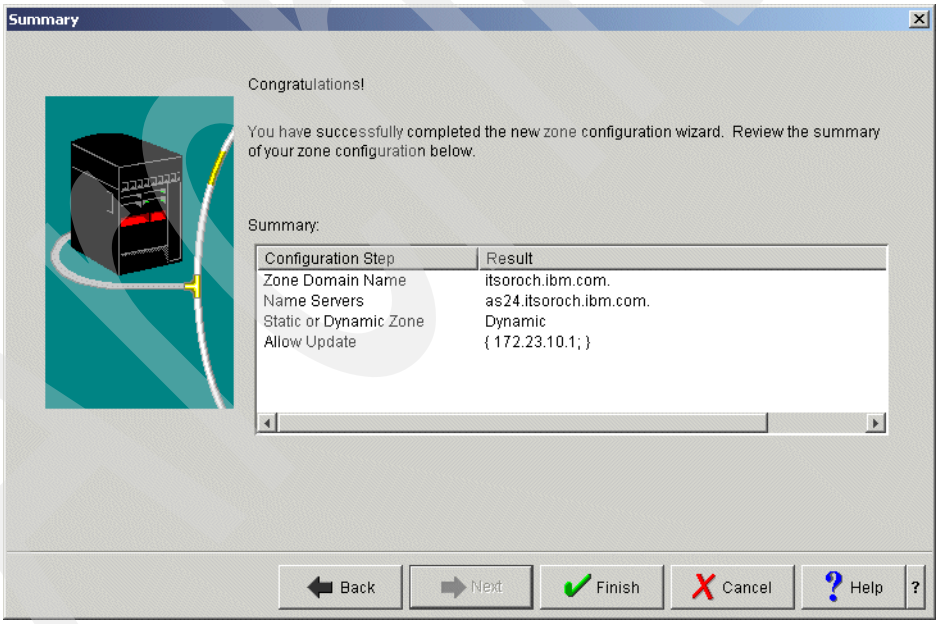


Figure 16-88 Summary window

11. In the DNS Configuration - NS24 window, expand **Forward Lookup Zones**. Right-click **Primary Zone itsoroch.ibm.com.** and select **Properties** from the context menu, as shown in Figure 16-89.

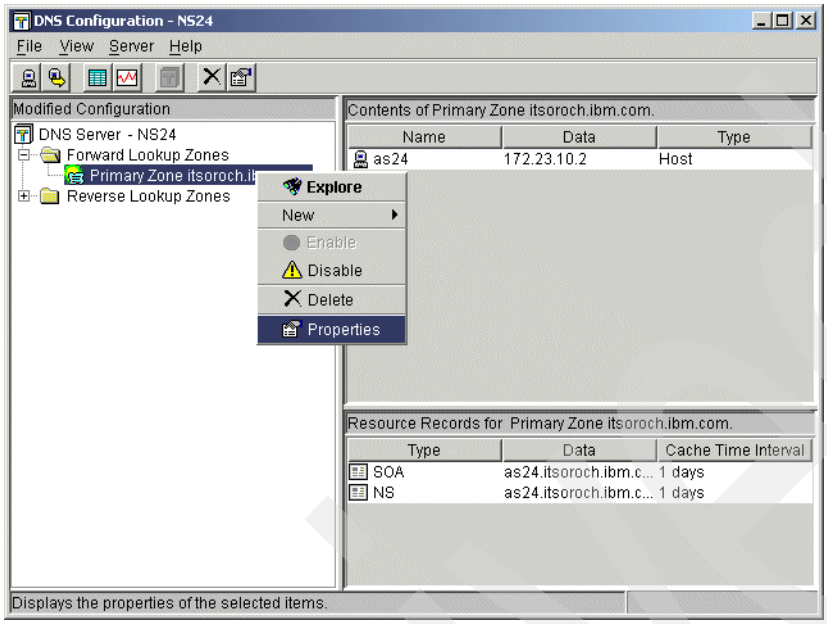


Figure 16-89 DNS Configuration - NS20 window

12. In the Primary Zone Properties - itsoroch.ibm.com. window, click the **Option** tab. Expand **Access Control** as shown in Figure 16-90.

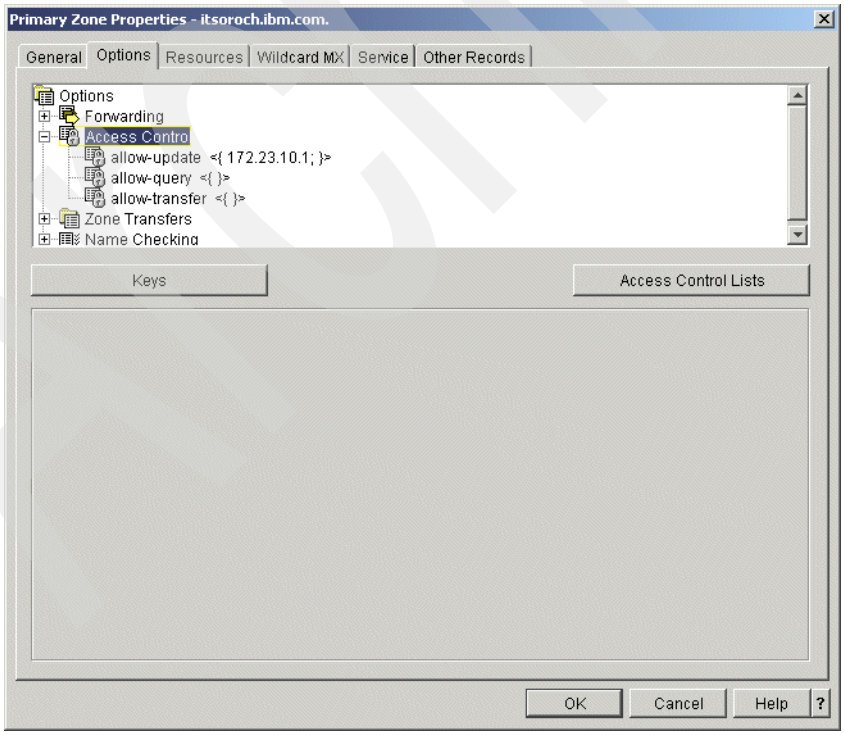


Figure 16-90 Primary Zone Properties: itsoroch.ibm.com. window

13. Click **allow-query**. Choose **Access Control List** for the Match List Element Type. Click **Add**, as shown in Figure 16-91.

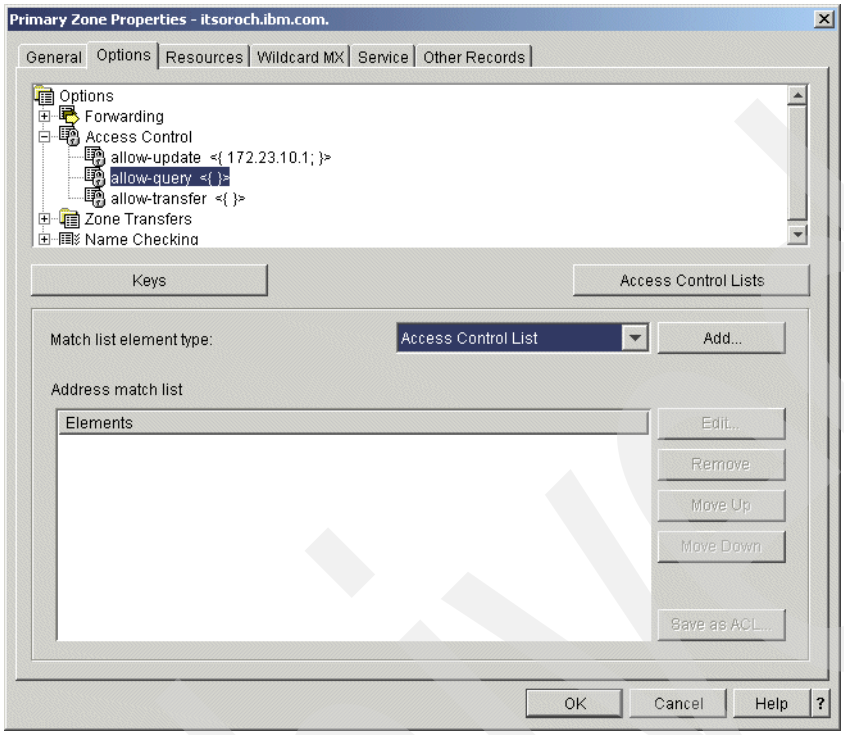


Figure 16-91 Primary Zone Properties - itsoroch.ibm.com. window

14. In the Access Control List window, select **any**. Choose **Allow access to this access control list** as shown in Figure 16-92. Click **OK**.

> **Tip:** This allows the query request from any clients to resolve IP address or host name.

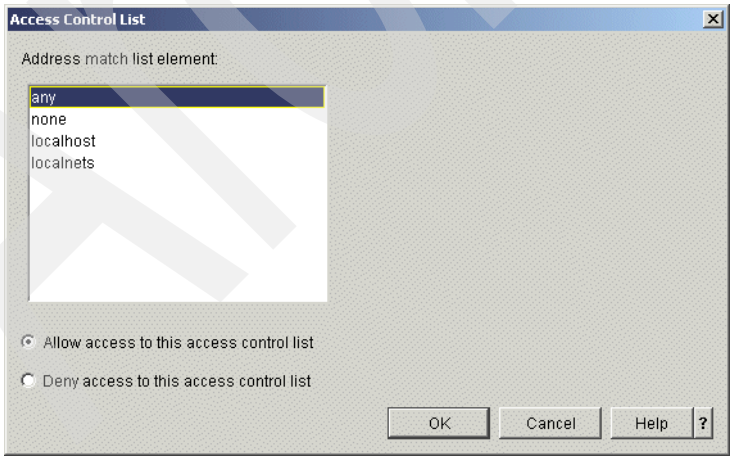


Figure 16-92 Access Control List window

15. In the Primary Zone Properties - itsoroch.ibm.com window, click the **Resources** tab. Select **SOA,** and click **Edit**, as shown in Figure 16-93.

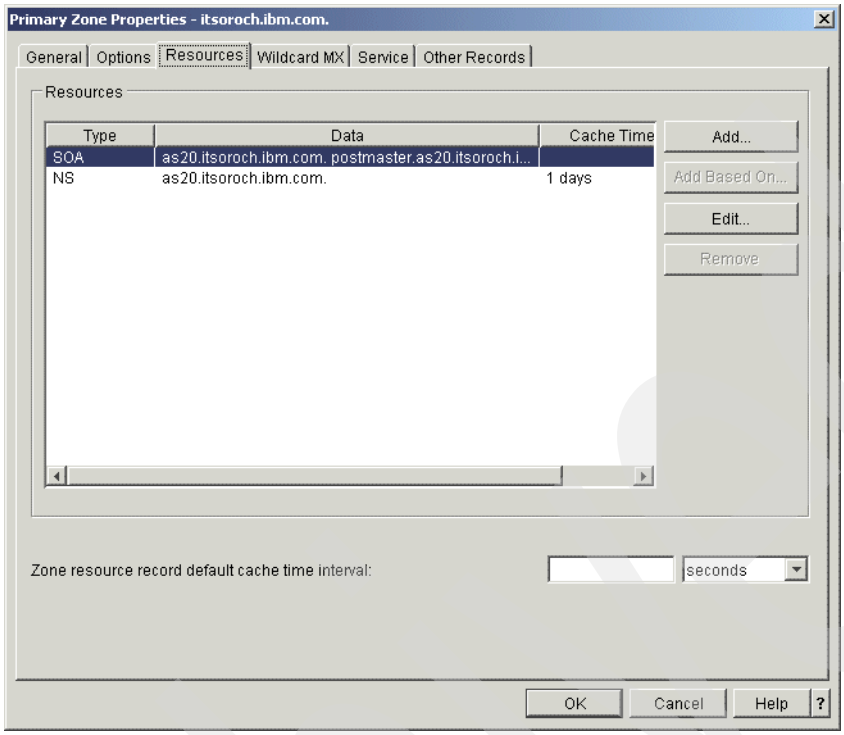


Figure 16-93 Primary Zone Properties - itsoroch.ibm.com. window

16. In the Add/Edit Resource - itsoroch.ibm.com. window, check **Start of Authority cache time interval (SOA TTL)**. Type 1 and choose **days** (answer 7 in Table 16-2 on page 404) as a value of SOA TTL, as shown in Figure 16-94.

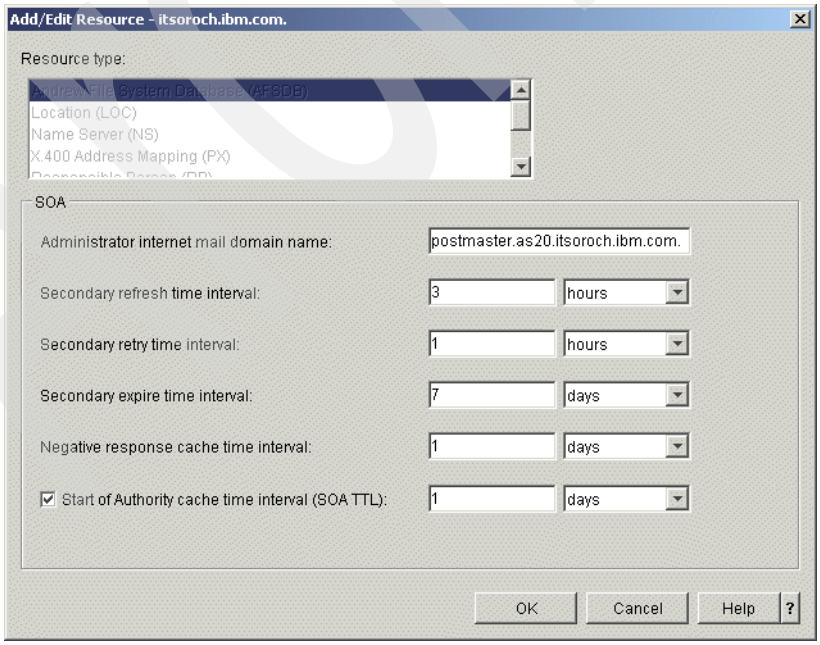


Figure 16-94 Add/Edit Resource - itsoroch.ibm.com.

17. In the Primary Zone Properties - itsoroch.ibm.com. window, right-click **Primary Zone itsoroch.ibm.com.** and choose **New**. On the next menu, select **Host** as shown in Figure 16-95.

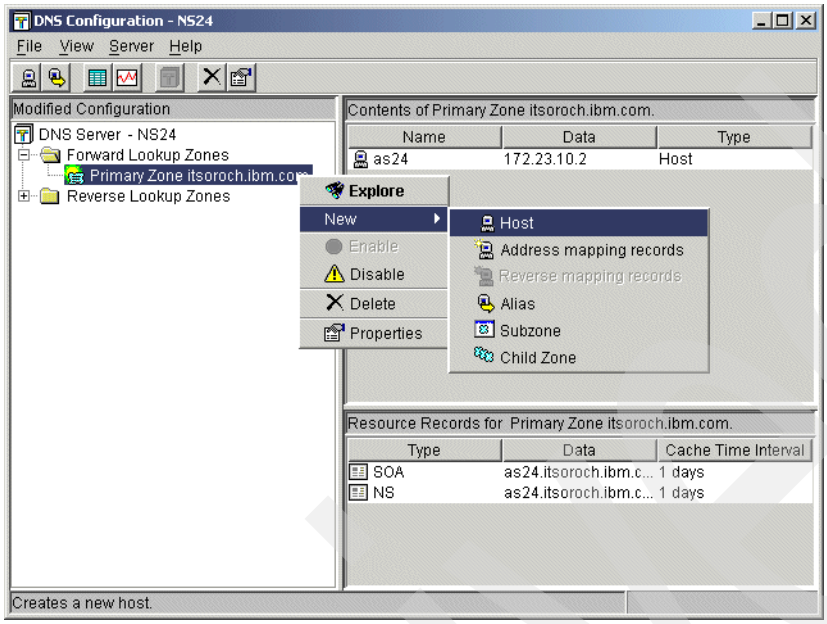


Figure 16-95 DNS Configuration - NS24 window

18. In the New Host window, type `as20.itsoroch.ibm.com.` for the Host domain name, as shown in Figure 16-96. Click **Next** to continue.

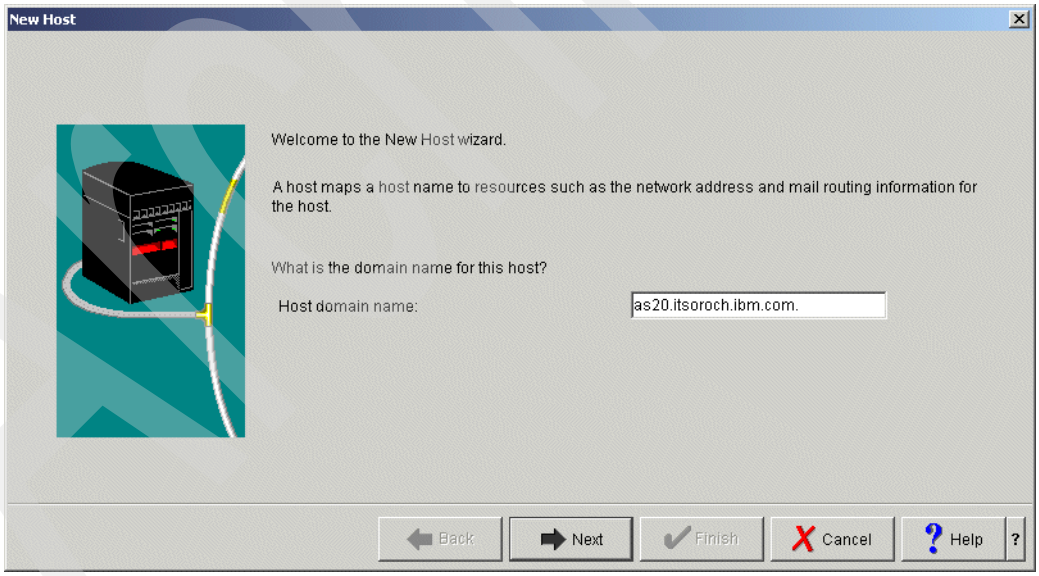


Figure 16-96 New Host window

19. In the New Host Resources window, click **Add**, as shown in Figure 16-97.

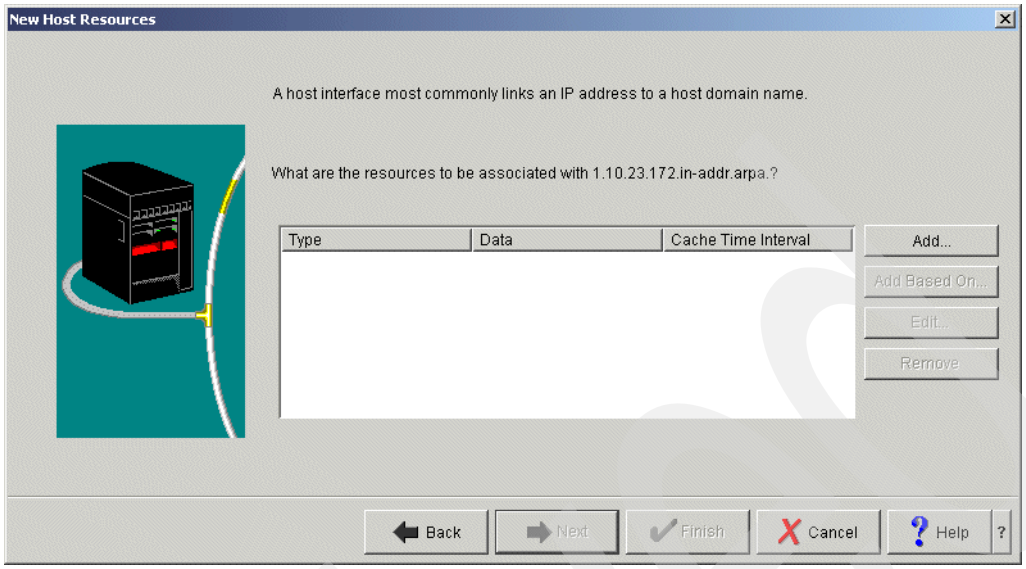


Figure 16-97 New Host Resources window

20. In the Add/Edit Resource - as20.itsoroch.ibm.com. window, choose **Address(A)**. Type `172.23.10.1` for the IP address. Select **Cache time interval** then type 1 and choose **days** (answer 5 in Table 16-2 on page 404), as shown in Figure 16-98. Click **OK** to continue.

    In the New Host Resources window, click **Finish**.

> **Tip:** Do not forget to type the period at the end of the fully qualified domain name.

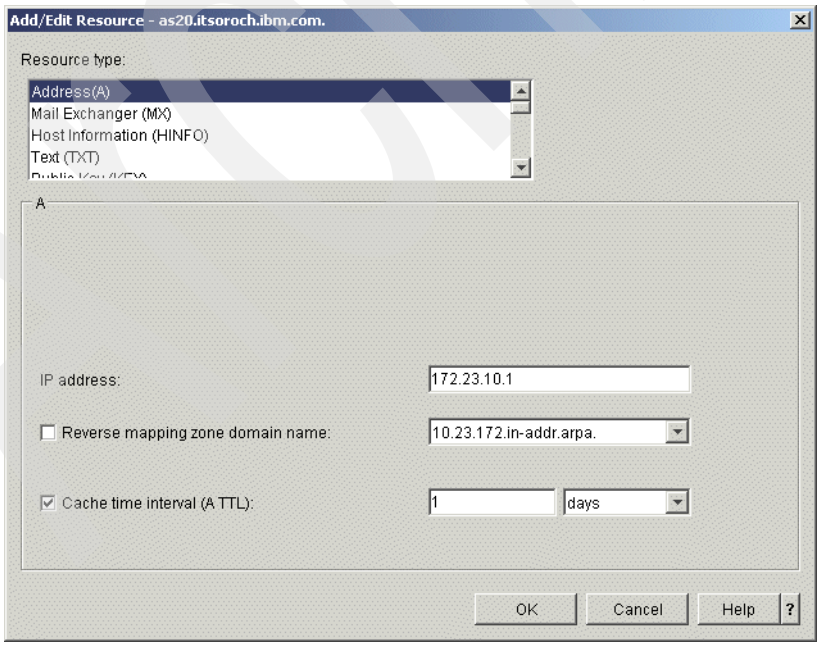


Figure 16-98 Add/Edit Resource - as20.itsoroch.ibm.com. window

Step 3c: Creating new Primary Zone on Reverse Lookup Zone

To do this:

1. In the DNS Configuration - NS24 window, right-click **Reverse Lookup Zones** and choose **New Primary Zone**, as shown in Figure 16-99.

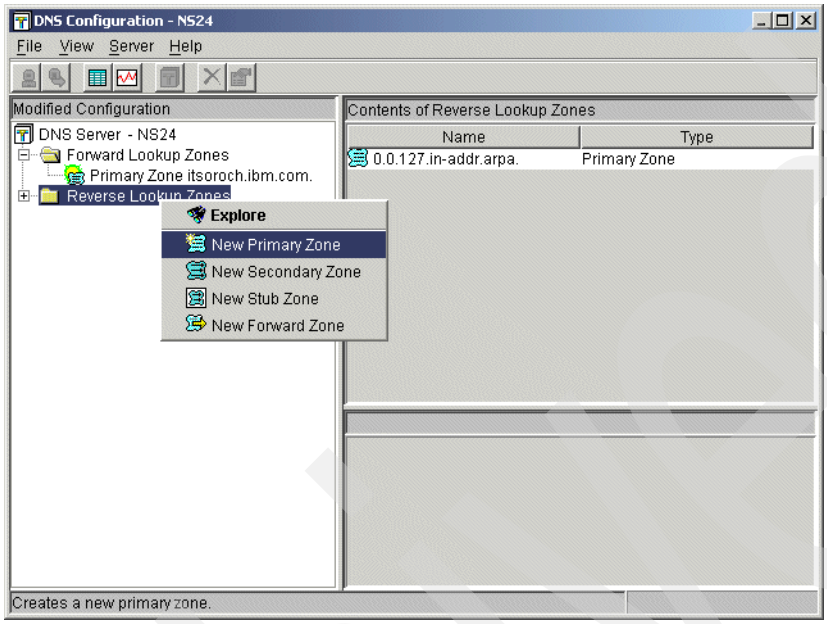


Figure 16-99 DNS Configuration - NS24 window

2. In the Zone Domain Name window, choose **Fully qualified domain name** and type `10.23.172.in-addr-arpa.` (answer 8 in Table 16-2 on page 404), as shown in Figure 16-100. Click **Next** to continue.

> **Tip:** Do not forget to type the period at the end of the fully qualified domain name.

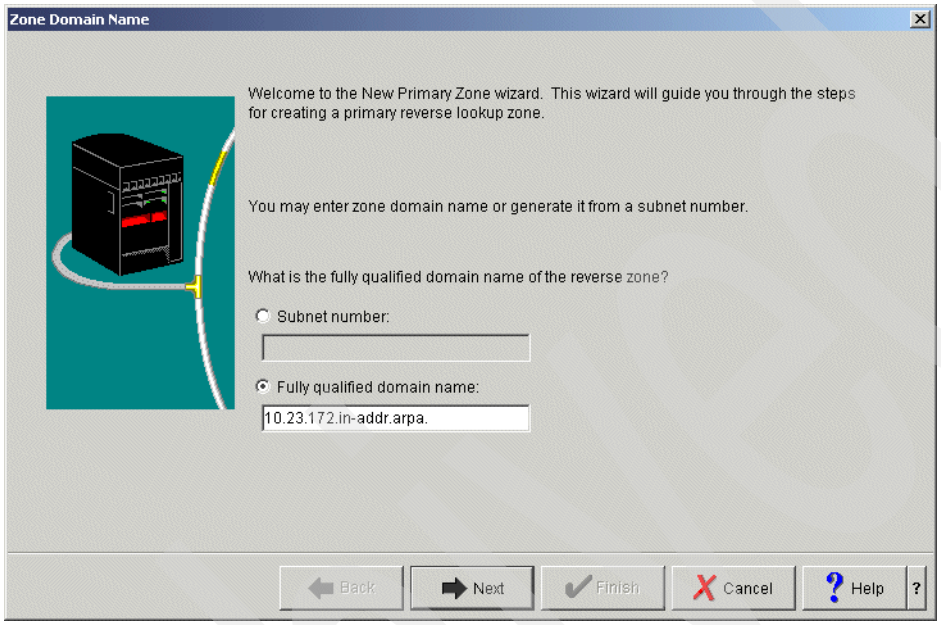


Figure 16-100 Zone Domain Name window

3. In the Name Servers window, click **as24.itsoroch.ibm.com.**, then click **Edit** as shown in Figure 16-101. Click **Next** to continue.

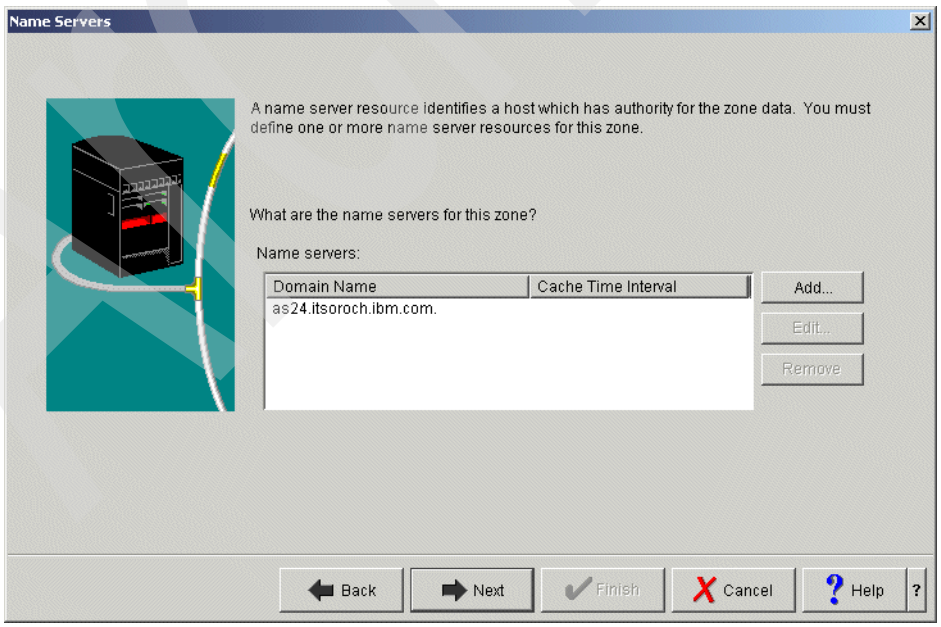


Figure 16-101 Name Servers window

4. In the Edit Name Server (NS) window, select **Cache Time Interval (NS TTL).** Type 1 in the column, then choose **days** (answer 5 in Table 16-2 on page 404), as shown in Figure 16-102. Click **OK**.

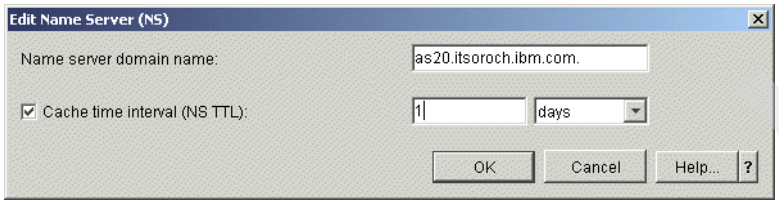


Figure 16-102 Edit Name Server (NS) window

5. In the Name Servers window, click **Next** to continue.
6. In the Static or Dynamic Zone window, choose **Perform dynamic updates** (answer 6 in Table 16-2 on page 404), as shown in Figure 16-103. Click **Next** to continue.

Figure 16-103 Static or Dynamic Zone window

7. In the Allow Update window, click **Add** as shown in Figure 16-104.

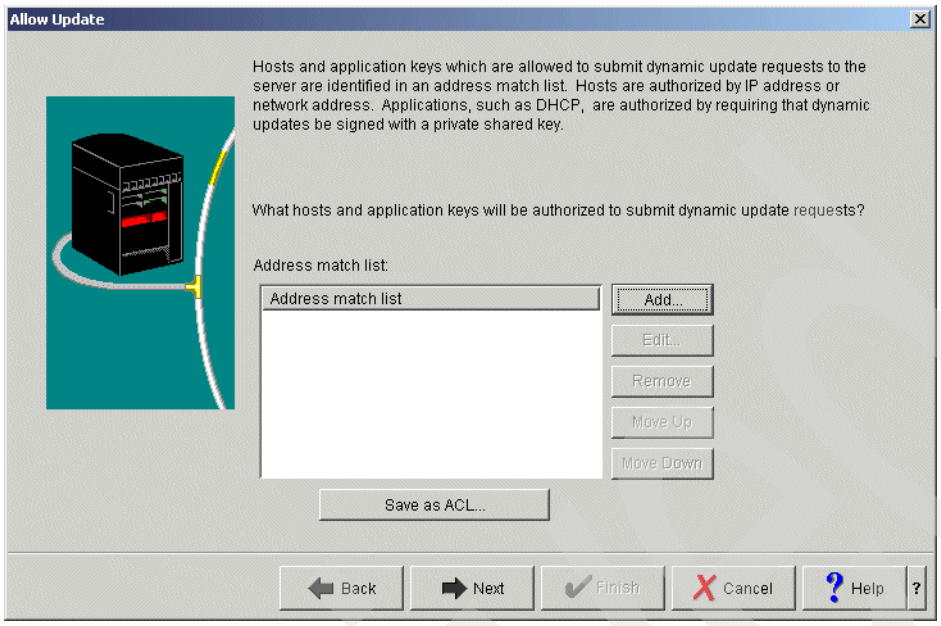


Figure 16-104 Allow Update window

8. In the Address Match List Element window, make sure that **IP Address** is selected as the Address match list element type. Type `172.23.10.1` (answer 1 in Table 16-2 on page 404) for the IP address. Choose **Allow access to this IP address**, as shown in Figure 16-105. Click **OK** to continue.

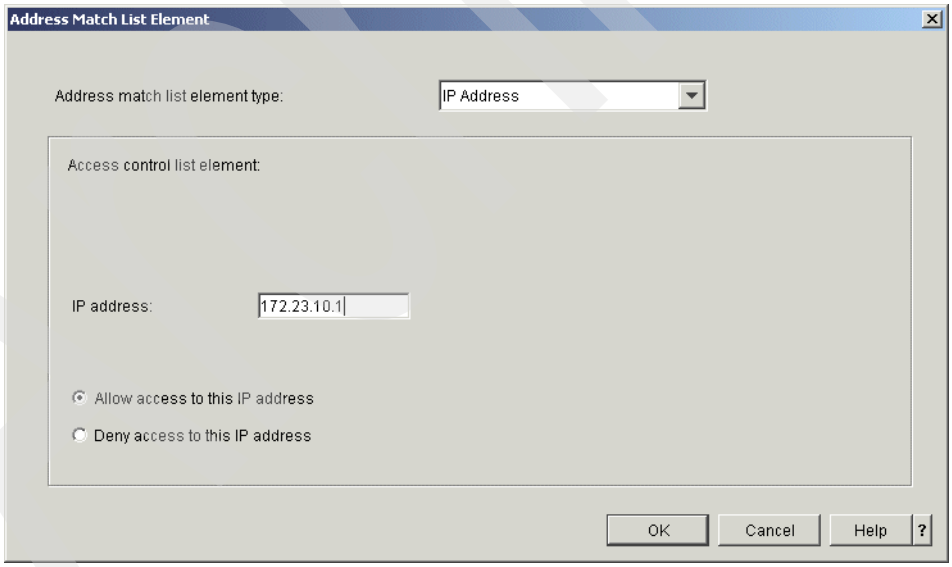


Figure 16-105 Address Match List Element window

9. In the Allow Update window, click **Next** to continue.

10. In the Summary window (Figure 16-106), click **Finish**.

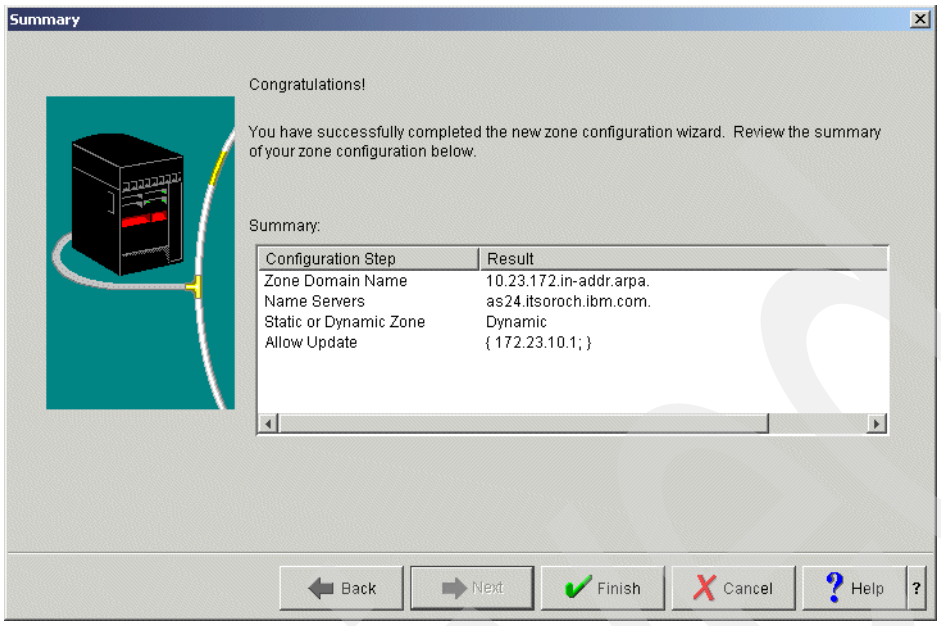


Figure 16-106 Summary window

11. In the DNS Configuration - NS24 window (Figure 16-107), right-click **Primary Zone 10.23.172-in-addr.arpa.** and click **Properties**.

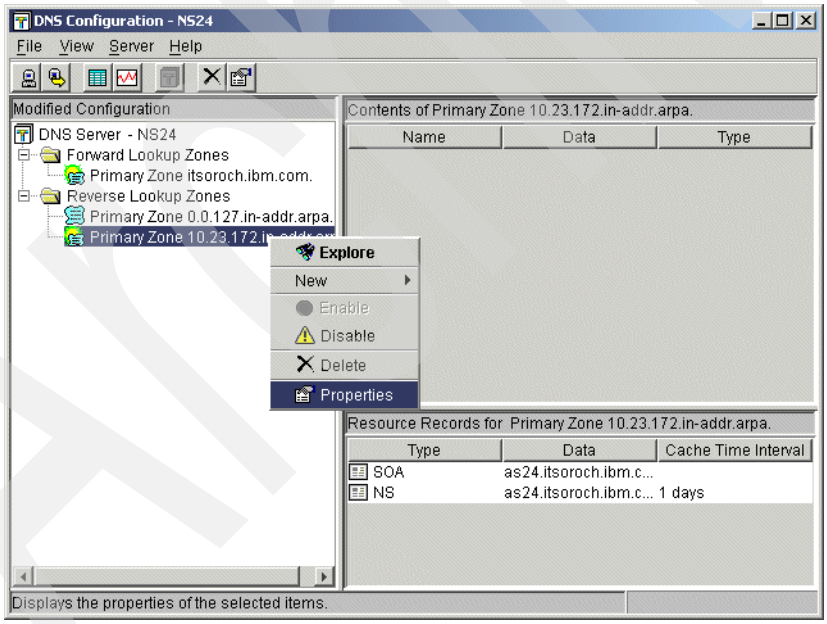


Figure 16-107 DNS Configuration - NS24 window

12. In the Primary Zone Properties - 10.23.172-in.addr.arpa window, click the **Options** tab. Expand **Access Control**, as shown in Figure 16-108.

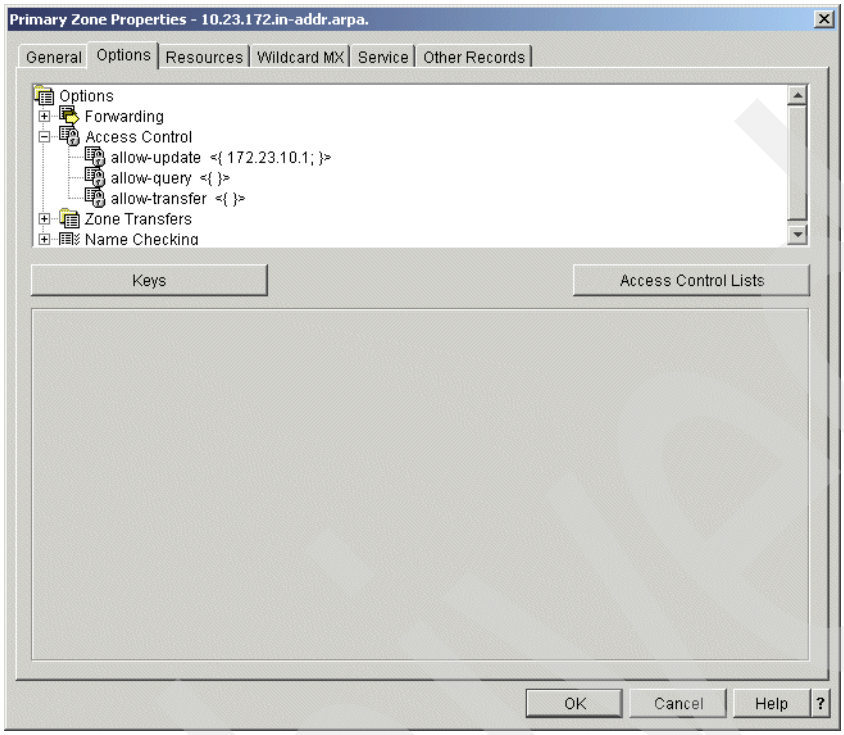


Figure 16-108 Primary Zone Properties - 10.23.172-in-addr.arpa window

13. Click **allow-query**. Choose **Access Control List** for the Match list element type. Click **Add** (Figure 16-109).

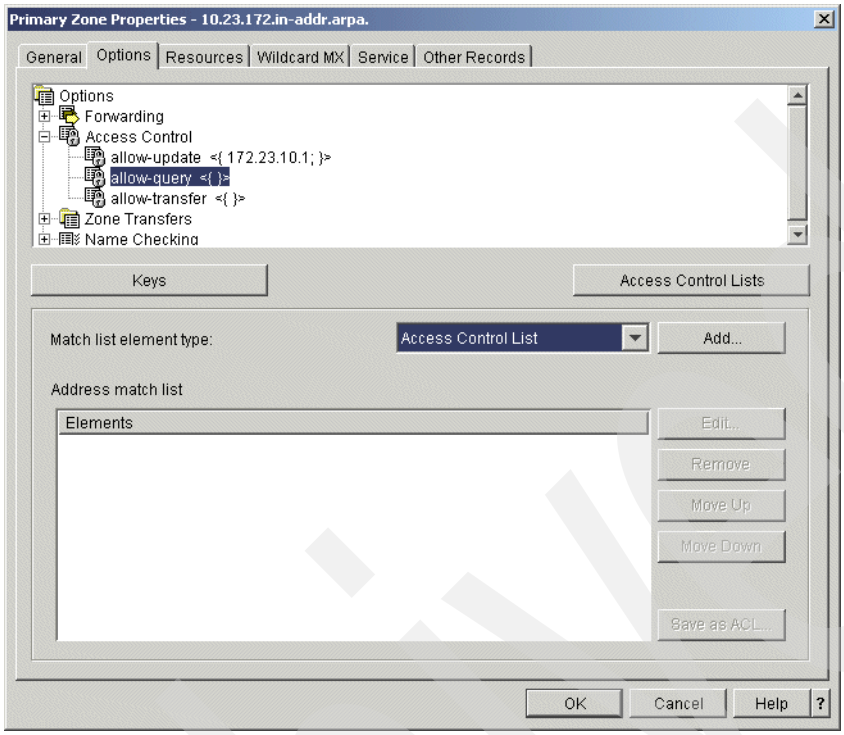


Figure 16-109 Primary Zone Properties - 10.23.172-in-addr.arpa window

14. In the Access Control List window, select **any**. Choose **Allow access to this access control list** as shown in Figure 16-110. Click **OK**.

> **Tip:** This allows the Query request from any clients to resolve IP address or Host name.

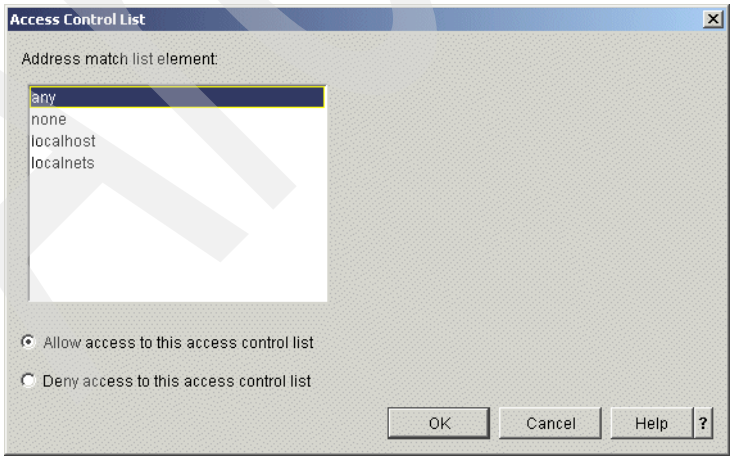


Figure 16-110 Access Control List window

15. In the Primary Zone Properties - 10.23.172-in.addr.arpa window, click the **Resources** tab. Select **SOA** and click **Edit**, as shown in Figure 16-111.

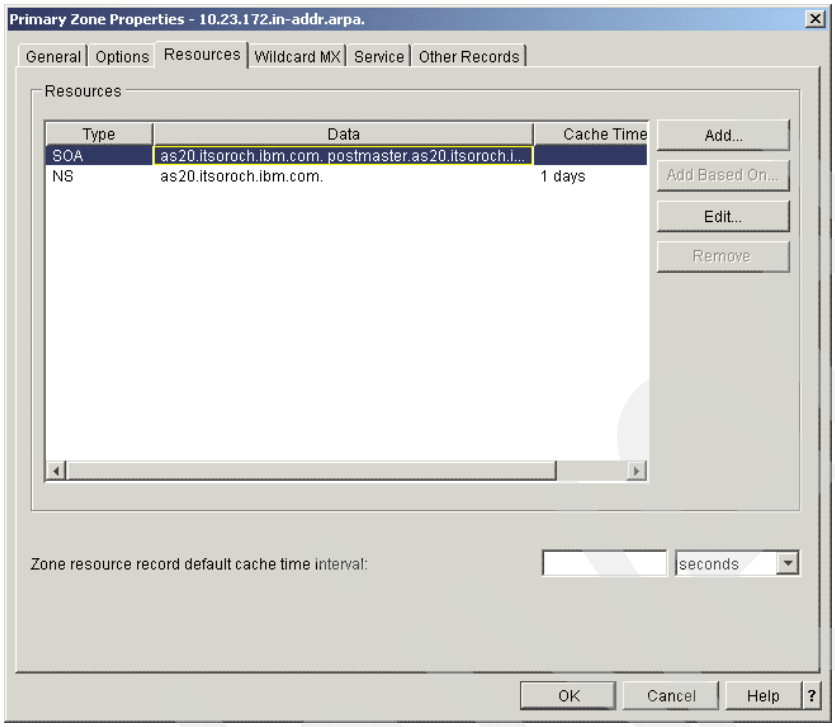


Figure 16-111 Primary Zone Properties - 10.23.172-in.addr.arpa window

16. In the Add/Edit Resource - 10.23.172.in-addr.arpa. window, check **Start of Authority cache time interval (SOA TTL)**. Type 1 and choose **days** (answer 7 in Table 16-2 on page 404) as a value of SOA TTL, as shown in Figure 16-112.

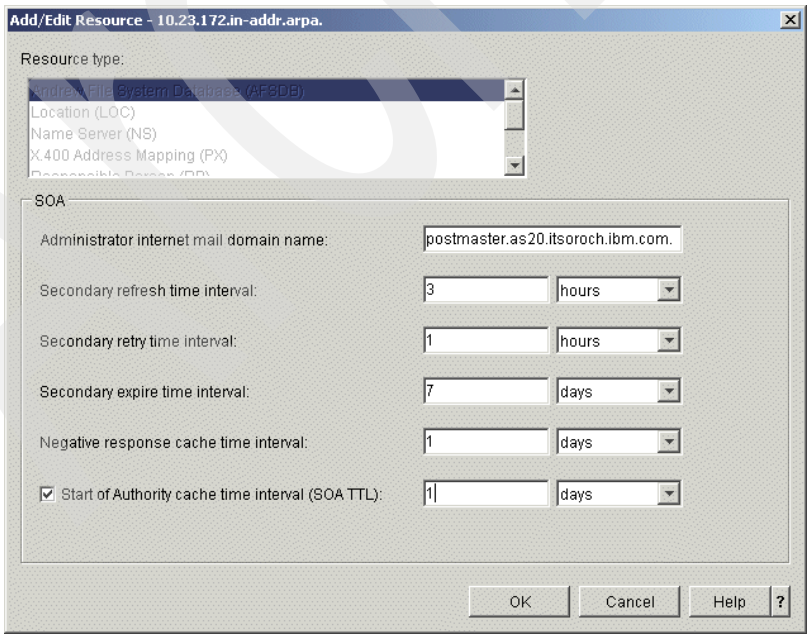


Figure 16-112 Add/Edit Resource - 10.23.172-in.addr.arpa window

In the Primary Zone Properties - 10.23.172-in.addr.arpa. window, click **OK**.

17. In the DNS Configuration - NS24 window (Figure 16-113), right-click **Primary Zone 10.23.172-in-addr-arpa.** and choose **New** → **Host**.

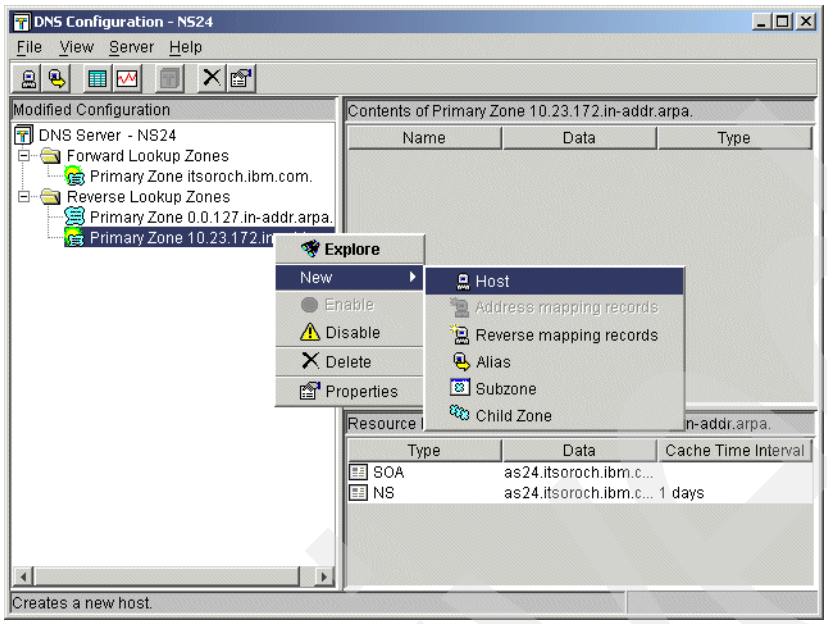


Figure 16-113 DNS Configuration - NS24 window

18. In the New Host window, type `172.23.10.1` (answer 1 in Table 16-2 on page 404) for the IP address, as shown in Figure 16-114. Click **Next** to continue.

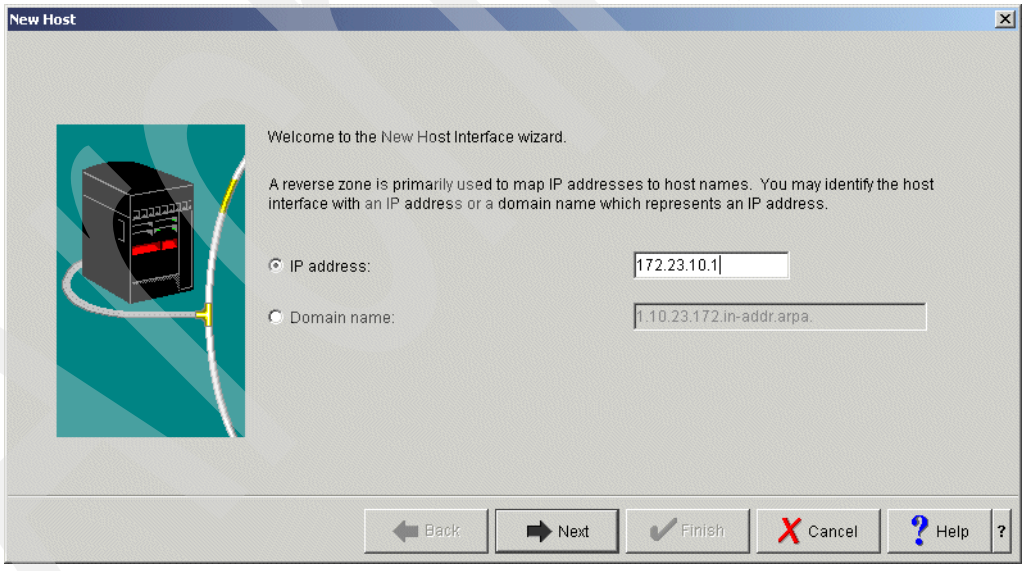


Figure 16-114 New Host window

19. In the New Host Resources window, click **Add**, as shown in Figure 16-115.

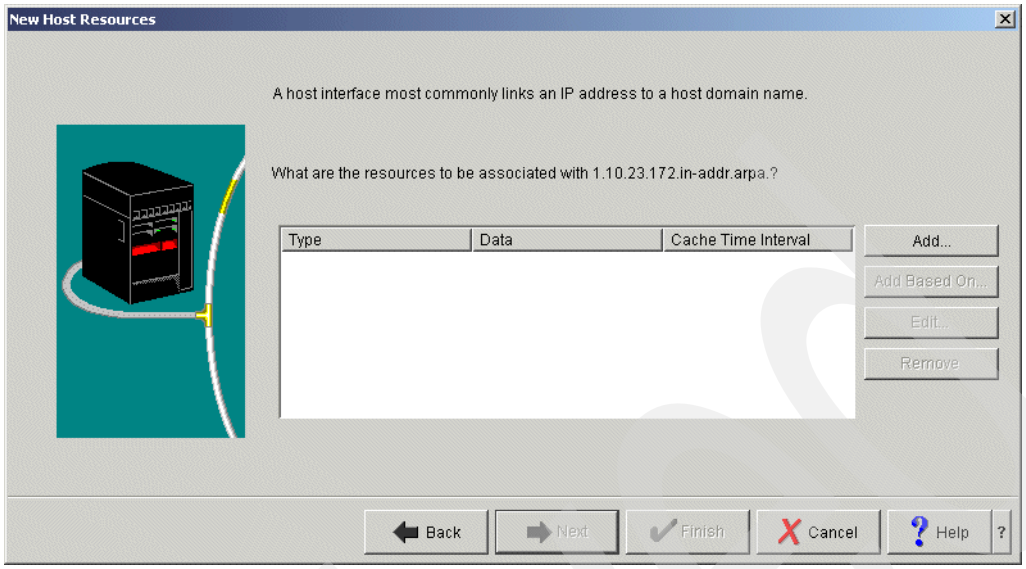


Figure 16-115 New Host Resources window

20. In the Add/Edit Resource - 1.10.23.172.in-addr.arpa window, choose **Reverse Mapping (PTR)**. Type `as20.itsoroch.ibm.com.` as the Fully qualified host domain name. Select **Cache time interval (PTR TTL)**, then type 1 and choose **days** (answer 5 in Table 16-2 on page 404), as shown in Figure 16-116. Click **OK** to continue.

> **Tip:** Do not forget to type the period at the end of the fully qualified domain name.

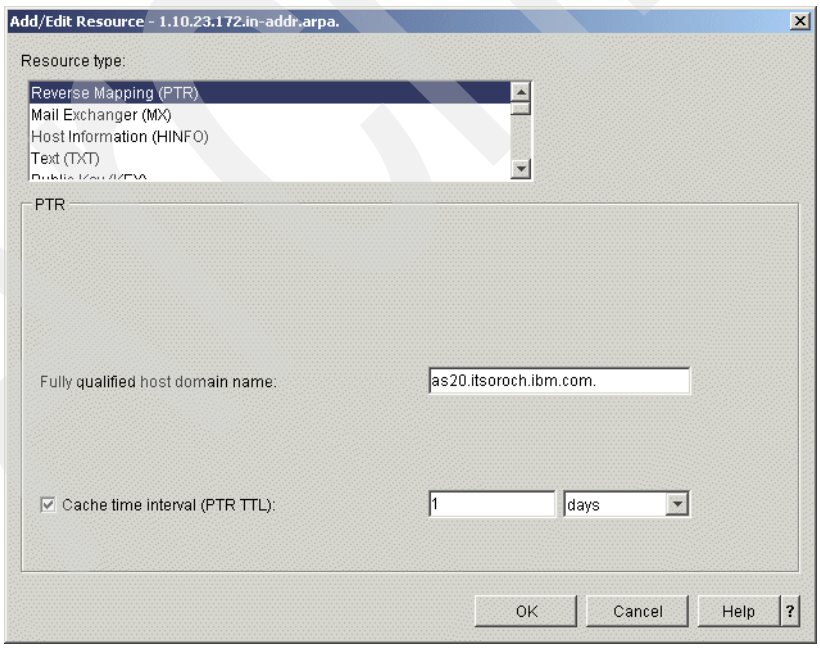


Figure 16-116 Add/Edit Resource - 1.10.23.172.in-addr.arpa window

21. In the New Host Resources window, click **Finish**.

22. In the DNS Configuration - NS24 window (Figure 16-117), right-click **Primary Zone 10.23.172-in-addr-arpa.** From the context menus, choose **New** → **Host**.

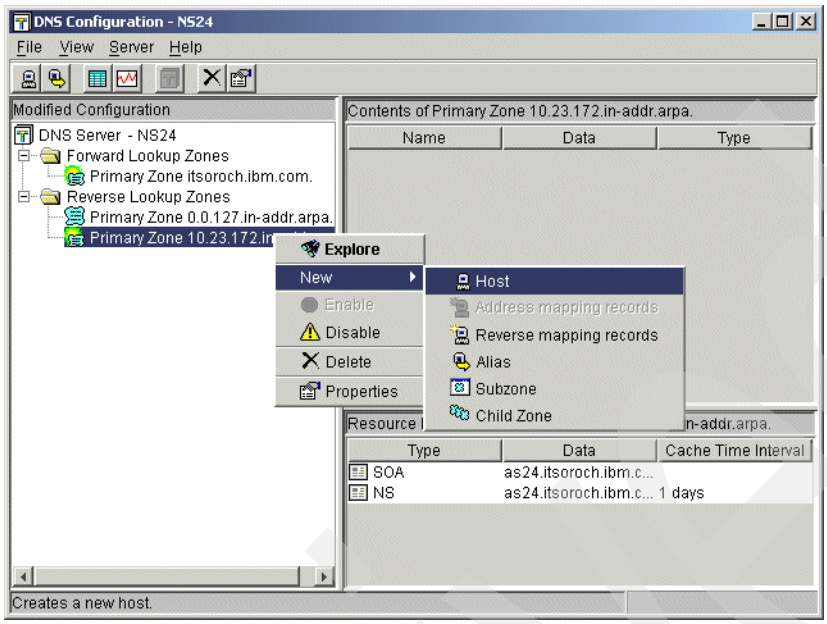


Figure 16-117 DNS Configuration - NS24 window

23. In the New Host window (Figure 16-118), type `172.23.10.2` (answer 1 in Table 16-2 on page 404) as the IP address. Click **Next** to continue.

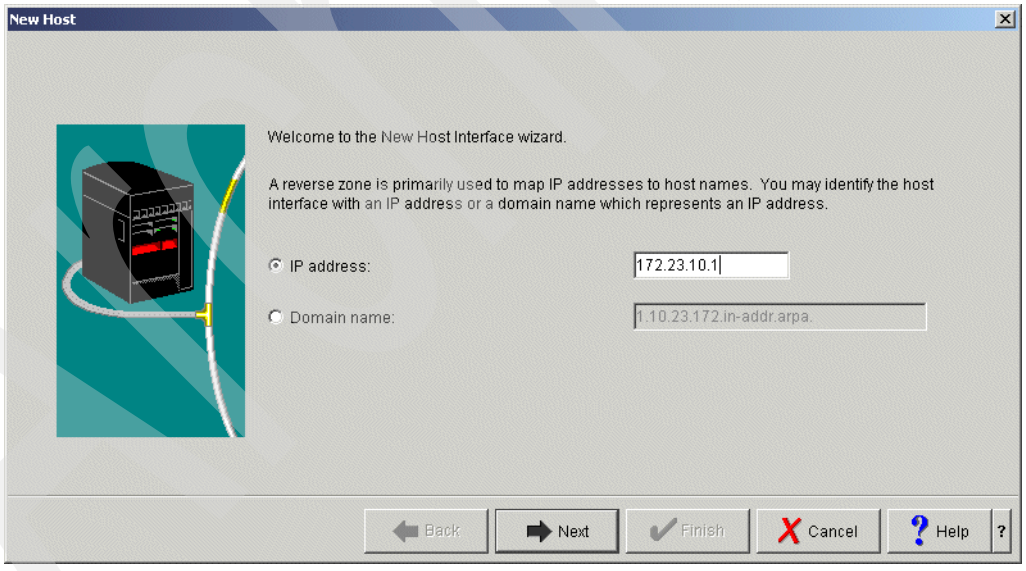


Figure 16-118 New Host window

24. In the New Host Resources window, click **Add**, as shown in Figure 16-119.

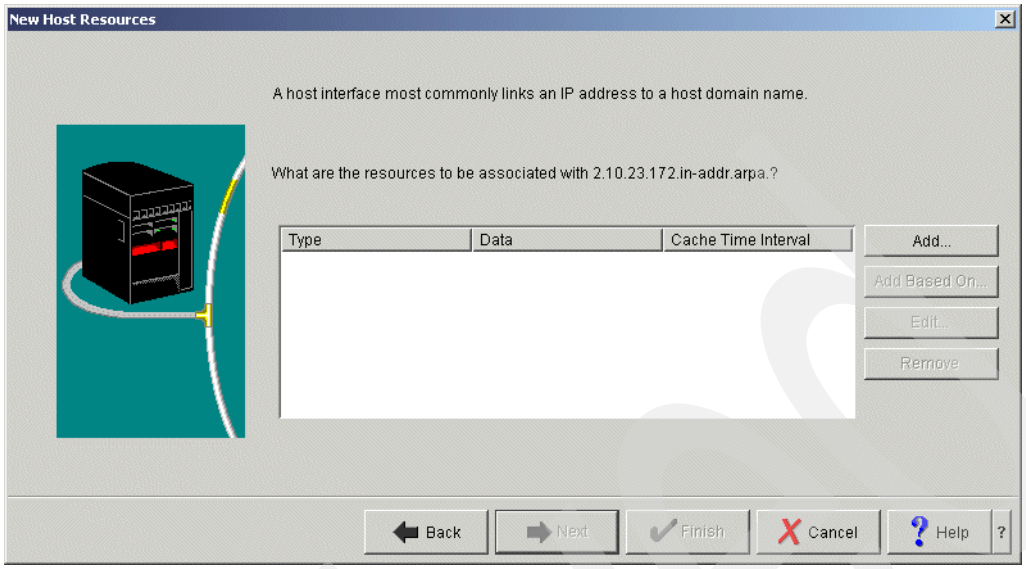


Figure 16-119 New Host Resources window

25. In the Add/Edit Resource - 2.10.23.172.in-addr.arpa window, choose **Reverse Mapping (PTR)**. Type `as24.itsoroch.ibm.com.` as the Fully qualified host domain name. Select **Cache time interval (PTR TTL)**, then type 1 and choose **days** (answer 5 in Table 16-2 on page 404), as shown in Figure 16-120. Click **OK** to continue.

> **Tip:** Do not forget to type the period at the end of the fully qualified domain name.

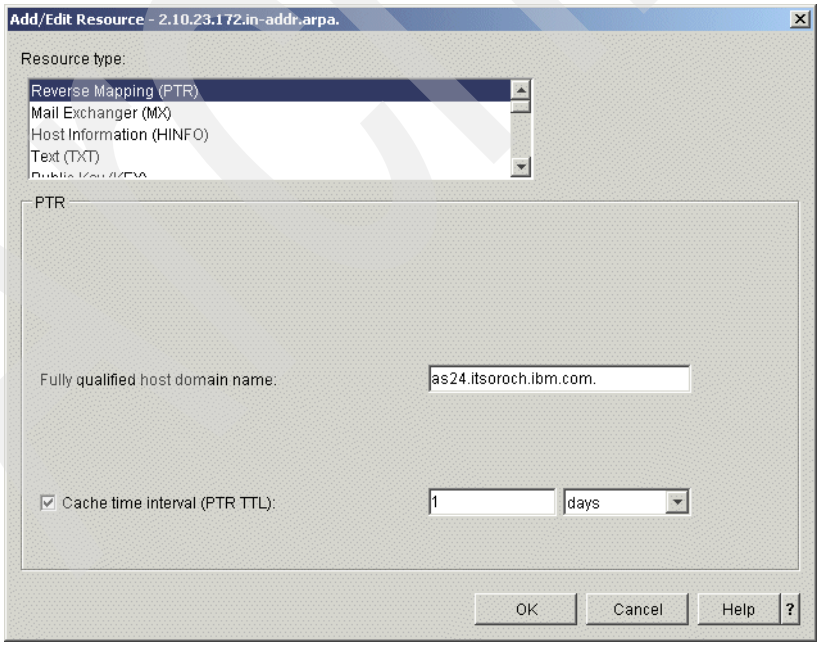


Figure 16-120 Add/Edit Resource - 2.10.23.172.in-addr.arpa window

26. In the New Host Resources window, click **Finish**.

27. In the DNS Configuration - NS24 window, click **File** → **Save Configuration** to save this configuration, as shown in Figure 16-121, and close the window.

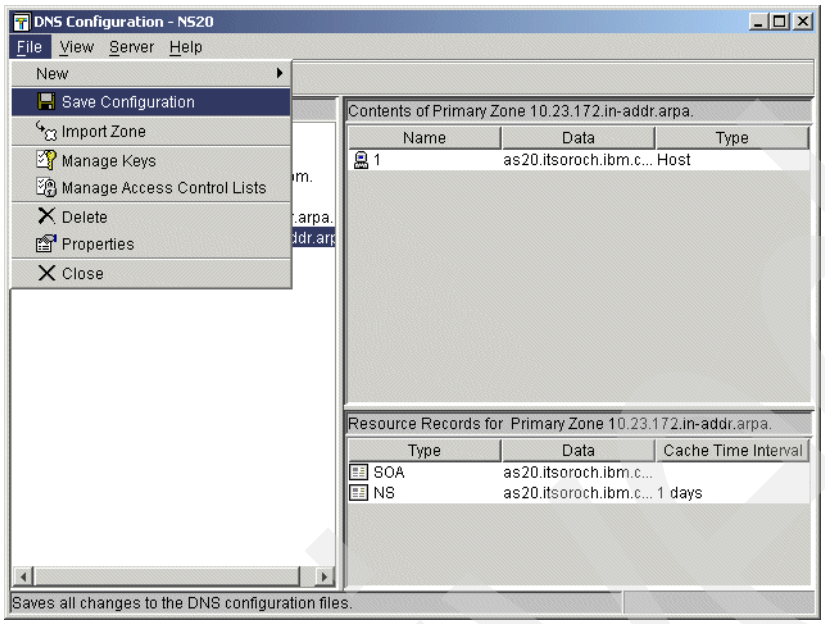


Figure 16-121 DNS Configuration - NS24 window

Step 4: Test the configuration

The procedure to test this scenario is very similar to 16.1, “Single DDNS and DHCP server on the same server” on page 368. Follow the steps outlined in “Step 4: Test the configuration” on page 396, but keep in mind that you are using two different System i’s. Use Table 16-2 on page 404 to help guide you.

16.3 Single DDNS and DHCP servers with secured updates

In this scenario, the DHCP server and DDNS server are configured on different servers, and the DHCP server updates A and PTR records to the DDNS server dynamically and securely with shared secrets right after the DHCP server assigns an IP address to the client.

The difference between 16.2, “Single DDNS and DHCP servers without secured updates” on page 402, and this section is that this section has an additional procedure for configuring the shared secret on the DHCP server and DDNS server.

Conditions to choose this scenario

You might choose this scenario under these conditions:

- If you want to configure the DDNS server and DHCP server individually
- If you want the DHCP server to update A and PTR records securely with shared secret
- If you do not need the secondary DNS server as a fault-tolerant backup
- If this DNS server is used in the private network and no security consideration is required to isolate the network from the public network

Sample network configuration

Figure 16-122 shows the sample network configuration of this scenario.

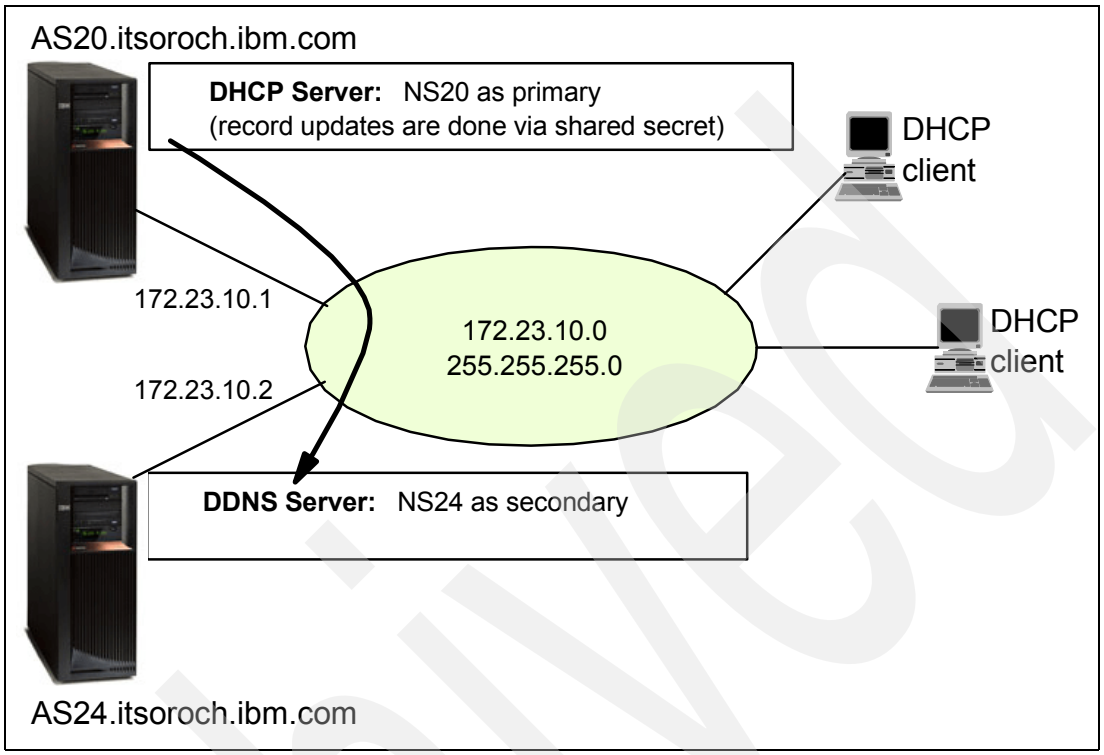


Figure 16-122 Sample network configuration: single DDNS server and DHCP server on different servers with secured records updates

16.3.1 Planning worksheet: single DDNS and DHCP servers with secured updates

Table 16-3 shows the planning worksheet for preparing the required parameters to configure the shared secret on DHCP server and DDNS server. Fill in the answers for each question in the adjacent Scenario answers column.

Table 16-3 Planning worksheet for the single DDNS server and DHCP server scenario

No.	Questions for creating single DDNS and DHCP servers on different servers with secured records updates	Scenario answers
1	What is the shared secret name? You can name it as you like.	shared
2	What is the seed value to create the shared secret? This seed will be used to calculate the shared secret for DHCP server and DDNS server.	makoto
3	What are the domain names to update A and PTR records? Domain name for the forward lookup and domain name for the reverse are required.	itsoroch.ibm.com. for forward lookup 10.23.172.in-addr.arpa. for reverse lookup

16.3.2 Configuration: single DDNS and DHCP servers with secured updates

In this scenario, you will create a shared secret on the DHCP server and DDNS server. This scenario assumes that you have already configured your System i's in a way similar to 16.2, "Single DDNS and DHCP servers without secured updates" on page 402. These extra steps must be taken:

- Step 1: Create the shared secret in the DHCP server on AS20.
- Step 2: Create the shared secret in the DDNS server on as24.
- Step 3: Test the configuration.

Step 1: Create the shared secret in the DHCP server on AS20

In this step, you will create a shared secret on DHCP server AS20 using the seed:

1. In the iSeries Navigator window, expand **Network** → **Servers**. Right-click **DNS** and choose **Manage Dynamic Update Keys** as shown in Figure 16-123.

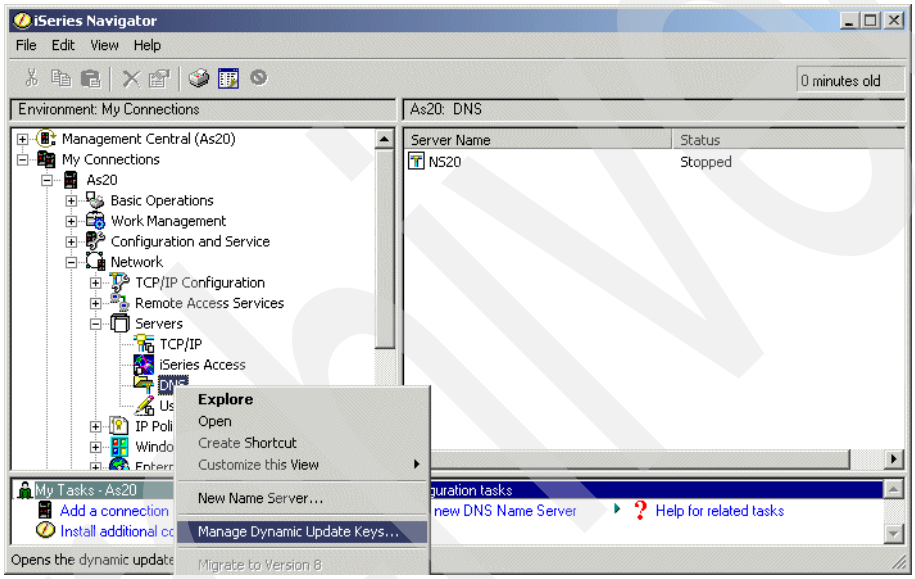


Figure 16-123 iSeries Navigator window

2. In the Manage Dynamic Update Keys window, click **Add** as shown in Figure 16-124.

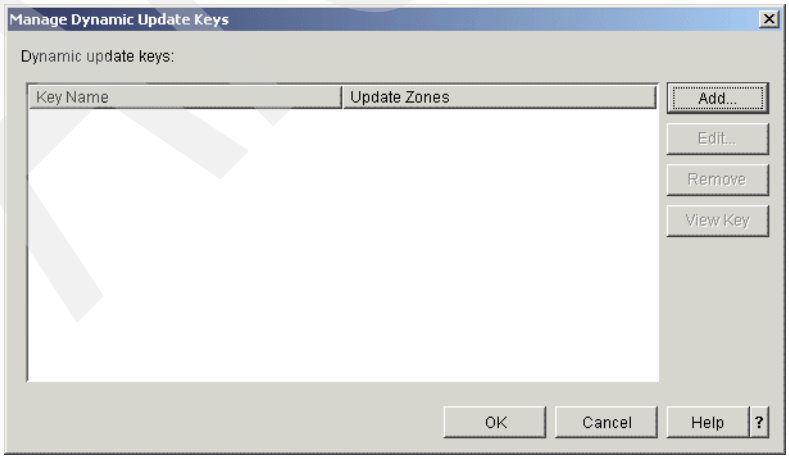


Figure 16-124 Manage Dynamic Update Keys window

3. In the Add Dynamic Update Keys window (Figure 16-125):
   a. Type `shared` for the Key name (answer 1 in Table 16-3 on page 439).
   b. Click **Add** and type `itsoroch.ibm.com`.
   c. Click **Add** and type `10.23.172.in-addr.arpa` (answer 2 in Table 16-3 on page 439).
   d. Make sure that the **Generate key using a secret as the seed value** radio button is selected.
   e. Type seed secret `makoto` in the Secret column (answer 3 in Table 16-3 on page 439).

   Click **OK** to continue.

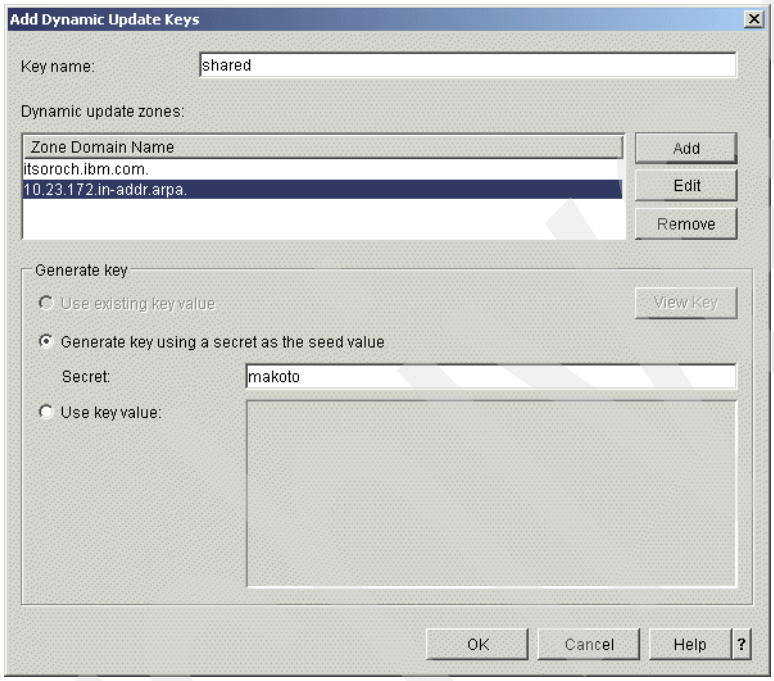


Figure 16-125 Add Dynamic Update Keys window

4. In the Manage Dynamic keys window, click **OK**.

Step 2: Create the shared secret in the DDNS server on as24

In this step, we create a shared secret on DDNS server AS20 using the seed:

1. In the iSeries Navigator window, expand **Network** → **Servers**. Right-click **NS24** and choose **Configuration**, as shown in Figure 16-126.

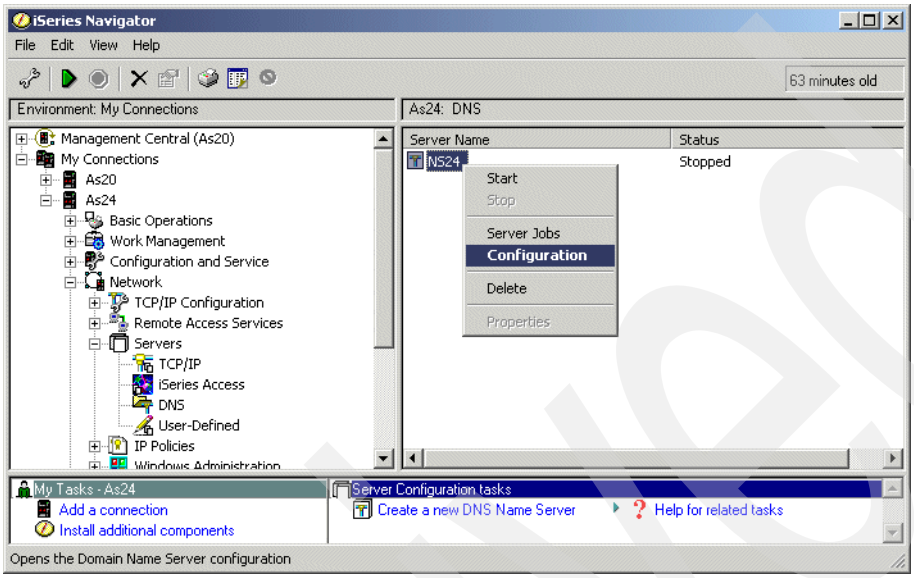


Figure 16-126 iSeries Navigator window

2. In the DNS Configuration - NS24 window (Figure 16-127), right-click **Primary lookup zone itsoroch.ibm.com**. and choose **Properties**.

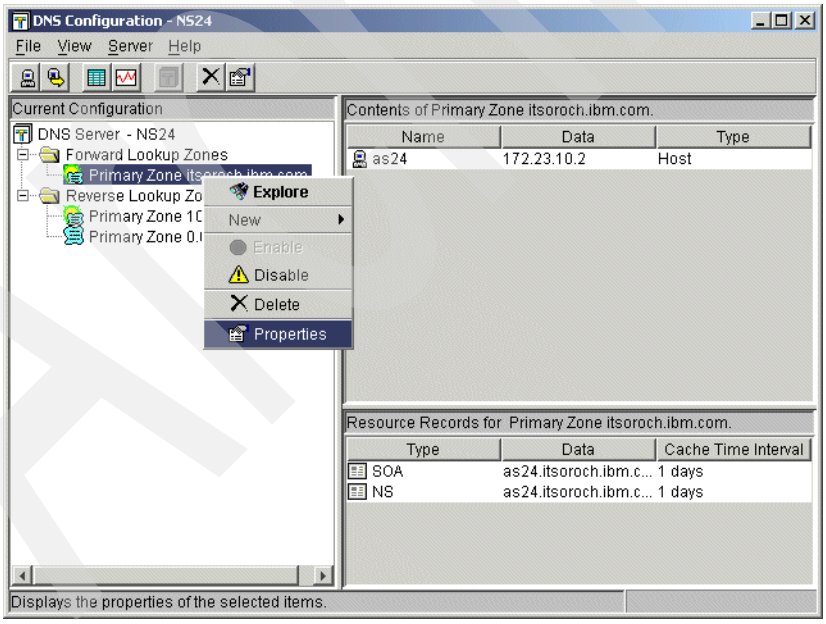


Figure 16-127 DNS Configuration for NS20: Edit properties of the Primary Lookup Zone

3. In the Primary Zone Properties itsoroch.ibm.com. window (Figure 16-128), click the **Options** tab. Expand **Access Control** and click **allow-update**. If there already is a configured element, select it and click **Remove**. (If you followed the scenario in 16.2, “Single DDNS and DHCP servers without secured updates” on page 402, the IP address 172.23.10.1 should be there. Select **172.23.10.1** and click **Remove**.)

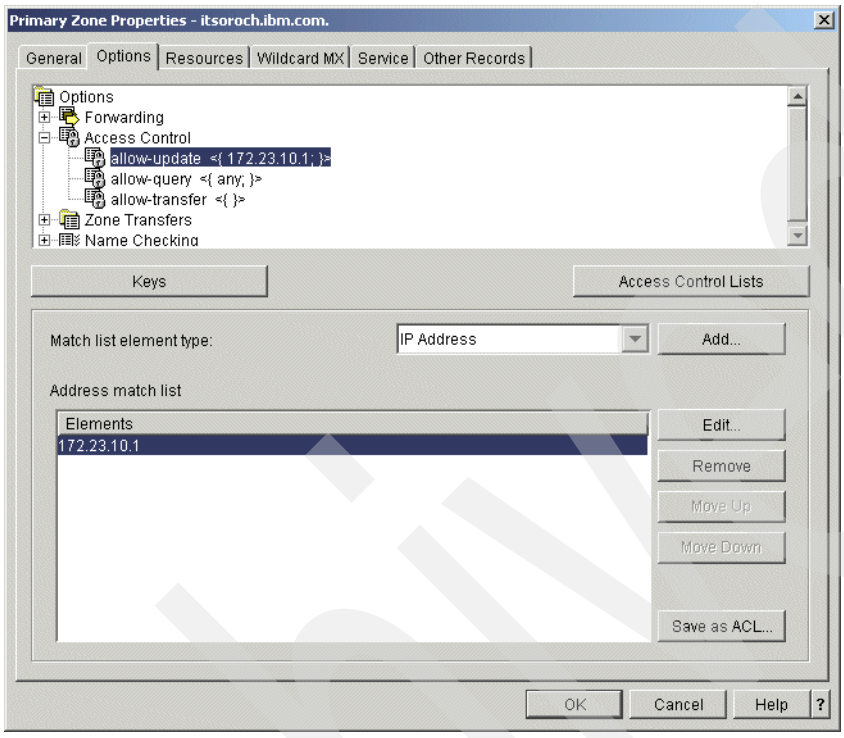


Figure 16-128 Primary Zone Properties itsoroch.ibm.com. window

4. In the Primary Zone Properties itsoroch.ibm.com. window, click **Keys**.
5. In the Manage Keys window, click **Add**.
6. In the Add Key window (Figure 16-129), type `shared`. Select **Generate key using a secret as the seed value** and type `makoto` as the Seed Secret. Click **OK**.

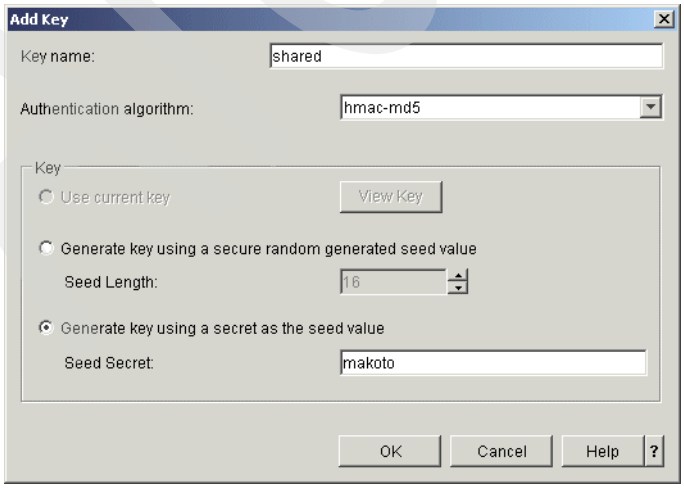


Figure 16-129 Add key window

7. In the Manage Keys window, click **OK** to return to the Primary Zone Properties itsoroch.ibm.com. window (Figure 16-130). Click **allow-update**. Choose **Key** for the Match list element type, and click **Add**.

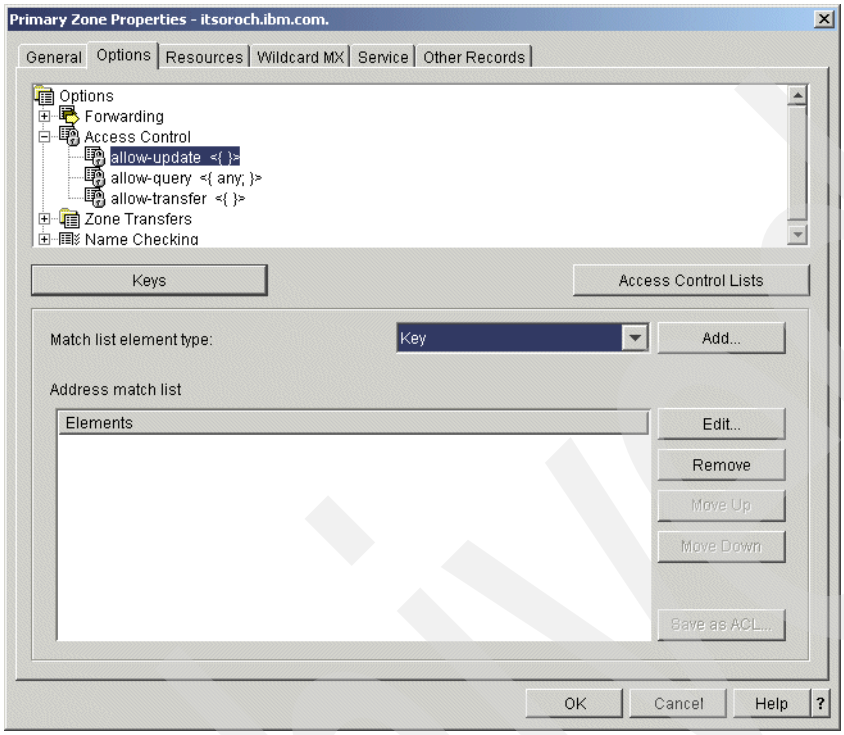


Figure 16-130 Primary Zone Properties itsoroch.ibm.com. window

8. In the Key window, **shared** must be in the list, as shown in Figure 16-131. Click **OK** to continue.

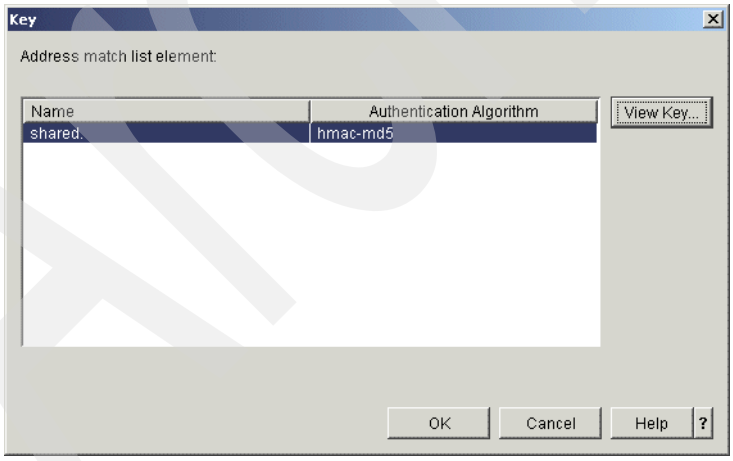


Figure 16-131 Key window

9. Back in the Primary Zone Properties itsoroch.ibm.com. window, make sure that the key `shared` appears for allow-update. Click **OK** to continue.

10. In the DNS Configuration - NS24 window (Figure 16-132), right-click **Primary Zone 10.23.172.in-addr.arpa.** and choose **Properties**.

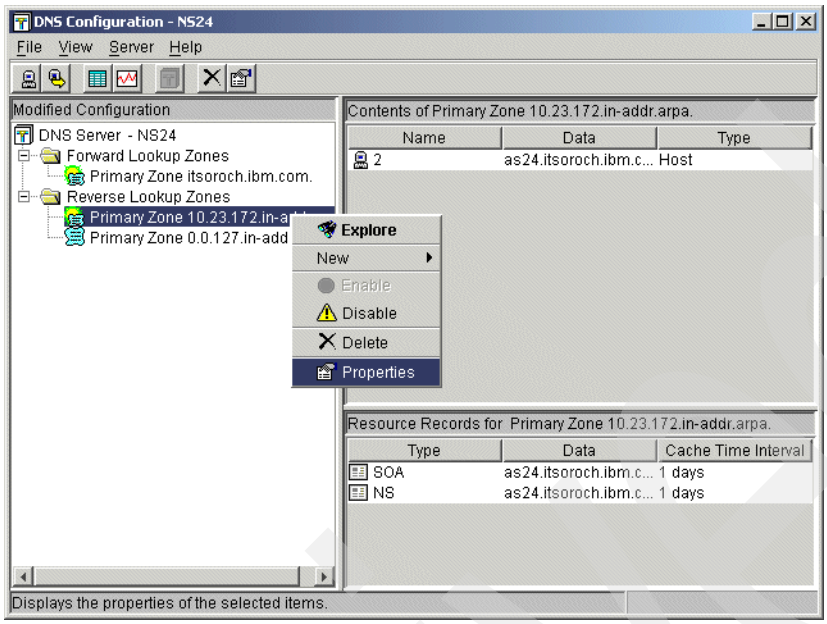


Figure 16-132 DNS Configuration - NS24 window

11. In the Primary Zone Properties 10.23.172.in-addr.arpa. window (Figure 16-133), click the **Options** tab. Expand **Access Control** and click **allow-update**. If there already is a configured element, select it and click **Remove**. If you followed the scenario in 16.2, “Single DDNS and DHCP servers without secured updates” on page 402, IP address 172.23.10.1 should be there. Select it and click **Remove**.

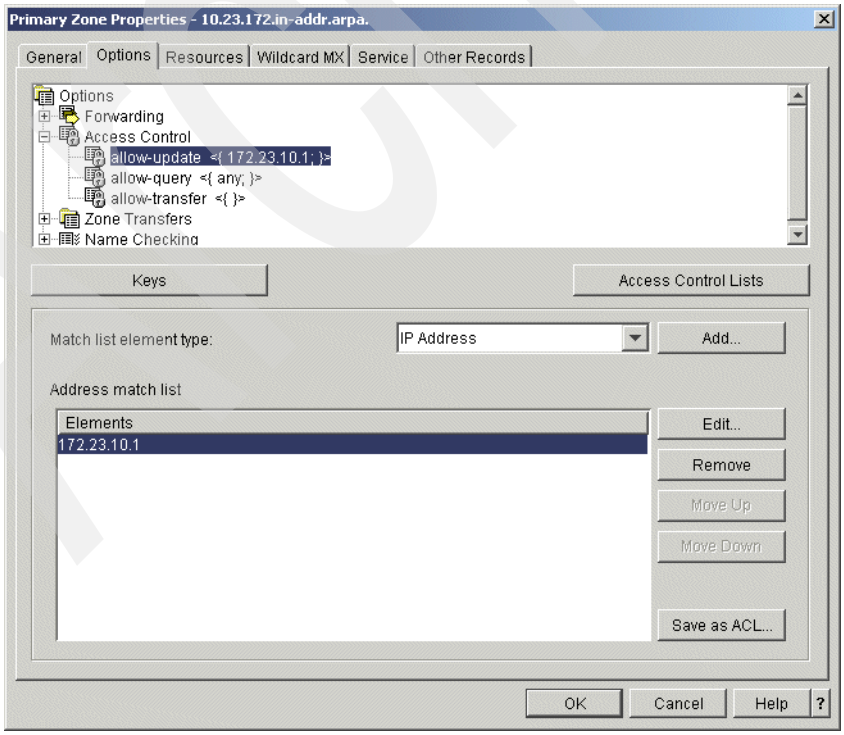


Figure 16-133 Primary Zone Properties 10.23.172.in-addr.arpa. window

12. In the Primary Zone Properties 10.23.172.in-addr.arpa. window, click **allow-update**. Choose **Key** as the Match list element type, and click **Add**, as shown in Figure 16-134.

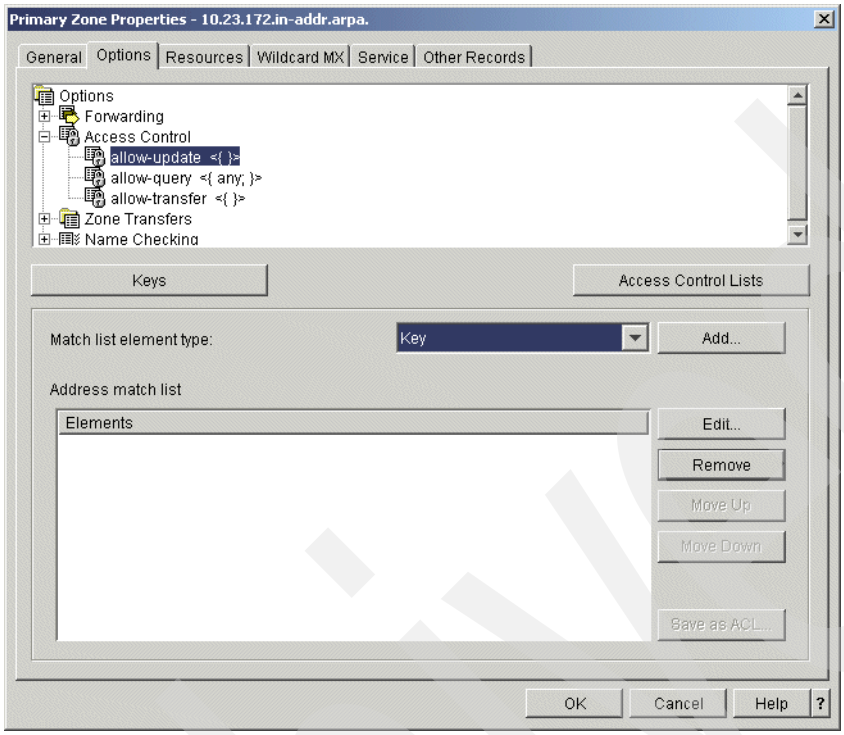


Figure 16-134 Primary Zone Properties 10.23.172.in-addr.arpa window

13. In the Key window, confirm that `shared` is in the list, as shown in Figure 16-135. Click **OK** to continue.

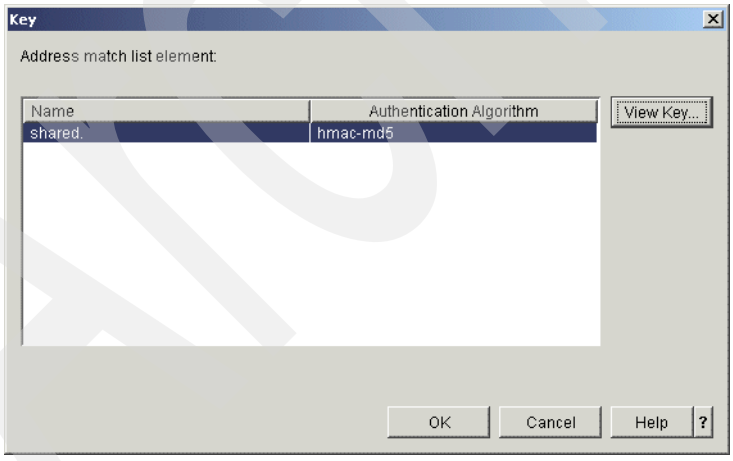


Figure 16-135 Key window

14. Back in the Primary Zone Properties 10.23.172.in-addr.arpa window, make sure that the key `shared` is on allow-update. Click **OK** to continue.

15. In the DNS Configuration - NS24 window (Figure 16-136), choose **File** → **Save Configuration** to save the configuration.

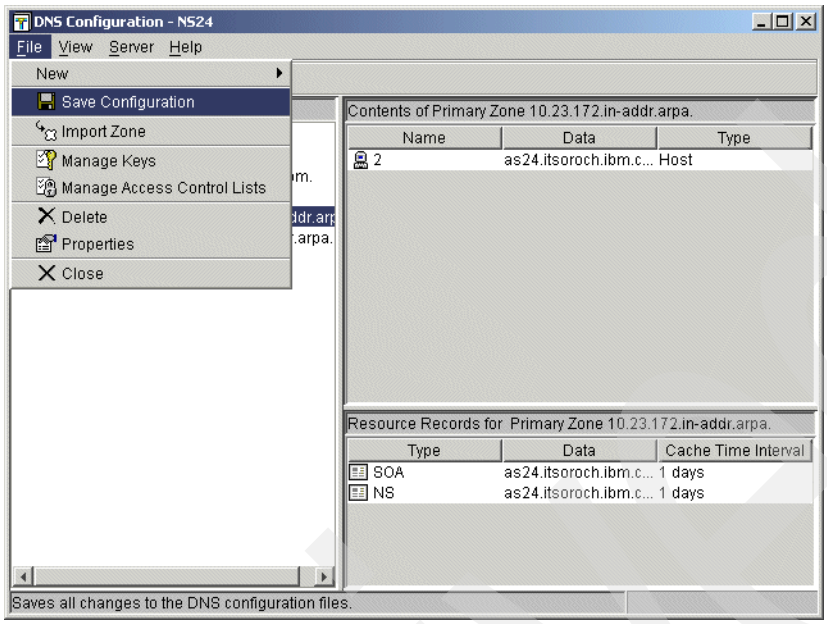


Figure 16-136 DNS Configuration - NS24 window: save the configuration

Step 3: Test the configuration

The procedure to test this scenario is very similar to 16.1, “Single DDNS and DHCP server on the same server” on page 368. Follow the steps outlined in “Step 4: Test the configuration” on page 396, but keep in mind that you are using two different System i’s.

16.4 Primary DDNS and DHCP servers on one server, secondary server as backup

This is the procedure to configure the primary DDNS server and DHCP server on the same server, plus a secondary server as a fault-tolerant backup. In this scenario, the DHCP server and primary DDNS server are configured on the same server, and the DHCP server updates A and PTR records to the DDNS server dynamically right after the DHCP server assigns an IP address to the client.

We will implement the procedure for configuring a secondary DNS for zone itsoroch.ibm.com. The secondary DDNS is updated by the primary DDNS every time the primary DDNS is dynamically updated by the DHCP server. This scenario is based on the scenario in 16.1, “Single DDNS and DHCP server on the same server” on page 368. In this scenario we demonstrate how to configure the secondary DDNS to receive updates from the primary DDNS using Incremental Zone Transfers (IXFR).

16.4.1 Scenario overview

You might choose this scenario if these conditions apply:

- If you have already configured DDNS and DHCP servers on AS20 using the procedure described in scenario number one (see “Single DDNS and DHCP server on the same server” on page 368).
- If you need the secondary DNS server as a fault-tolerant backup.

Sample network configuration

Figure 16-137 shows the sample network configuration of this scenario.

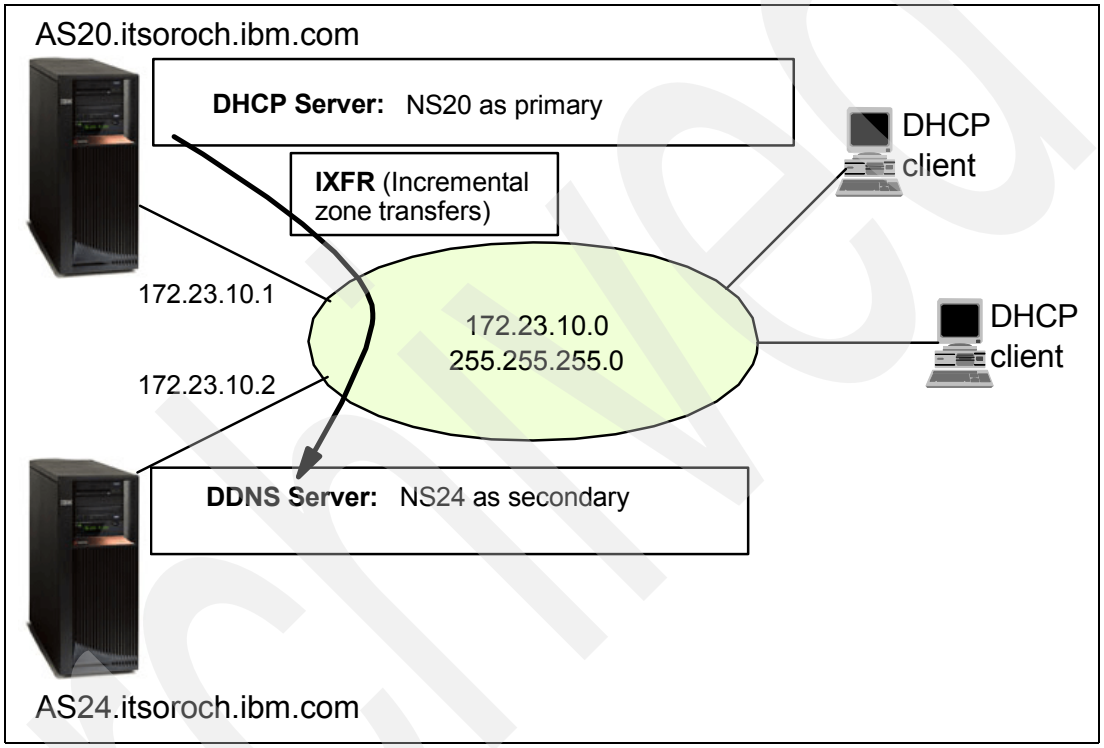


Figure 16-137 Primary DDNS server and DHCP server on the same server, plus secondary server as a fault-tolerant backup

16.4.2 Planning worksheet: Secondary DDNS

Table 16-4 shows the planning worksheet for preparing the required parameters to configure the secondary DDNS server. We have filled in our answers for each question in the adjacent Scenario answers column.

Table 16-4 Planning worksheet for the secondary DDNS

No	Questions to create the secondary DDNS server	Scenario answers
1	What is the TCP/IP Domain Information seen in CFGTCP Option 12 panel: - Host name - Domain name - Domain search list - Domain name server IP address	 AS24 itsoroch.ibm.com itsoroch.ibm.cm 172.23.10.2
2	What is the DDNS server instance name	NS24

No	Questions to create the secondary DDNS server	Scenario answers
3	What IP interface is used for listening to the query from the clients	172.23.10.2
4	Do you want the DDNS to start when TCP/IP starts	Yes
5	What is the name of the domain?	itsoroch.ibm.com
6	What is the address of the primary server for zone itsoroch.ibm.com	172.23.10.1

16.4.3 Configuration: Secondary DDNS

To implement the configuration in this scenario, perform the following steps:

- Step 1: Create the DNS NS24 on System i AS24.
- Step 2: Configure the secondary zone on NS24.
- Step 3: Update the DNS on AS20 to use AS24's DNS as secondary.

Step 1: Create the DNS NS24 on System i AS24

To create the new DNS NS24 on System i AS24, perform the following steps:

1. In the iSeries Navigator window, expand **Network** → **Servers**. Right-click **DNS** and select **New Name Server** as shown in Figure 16-138.

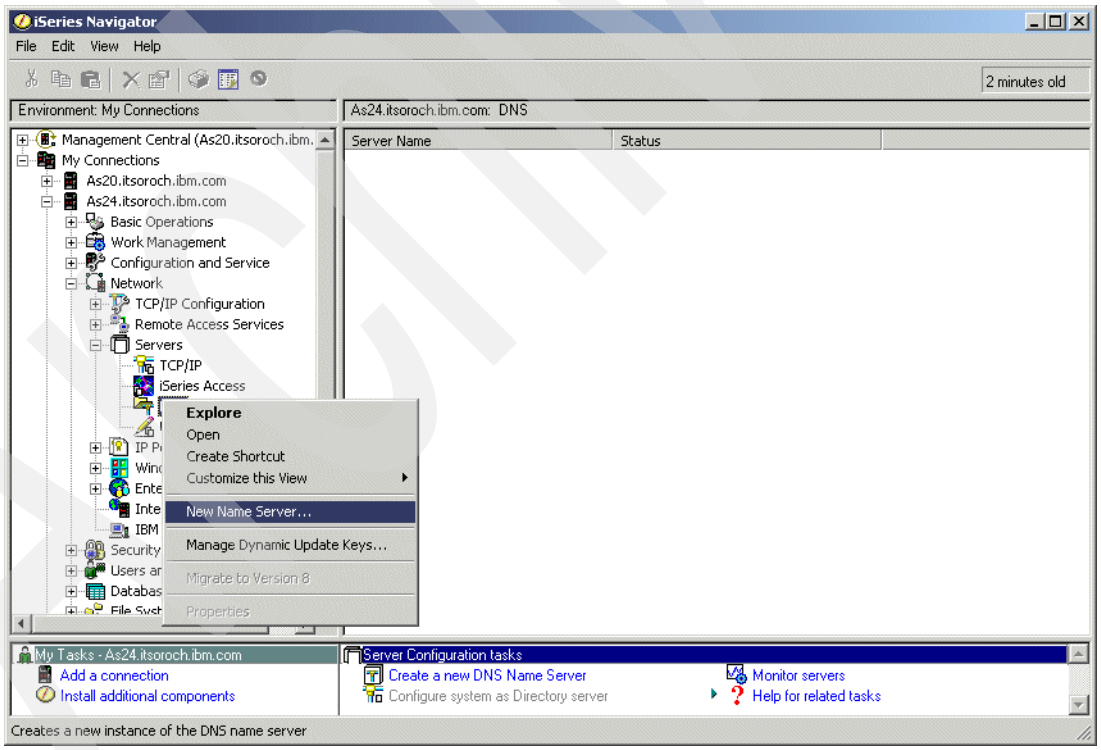


Figure 16-138 iSeries Navigator: configuring a new name server

2. This starts the New DNS Configuration wizard. Click **Next** in the New DNS Configuration window as shown in Figure 16-139.

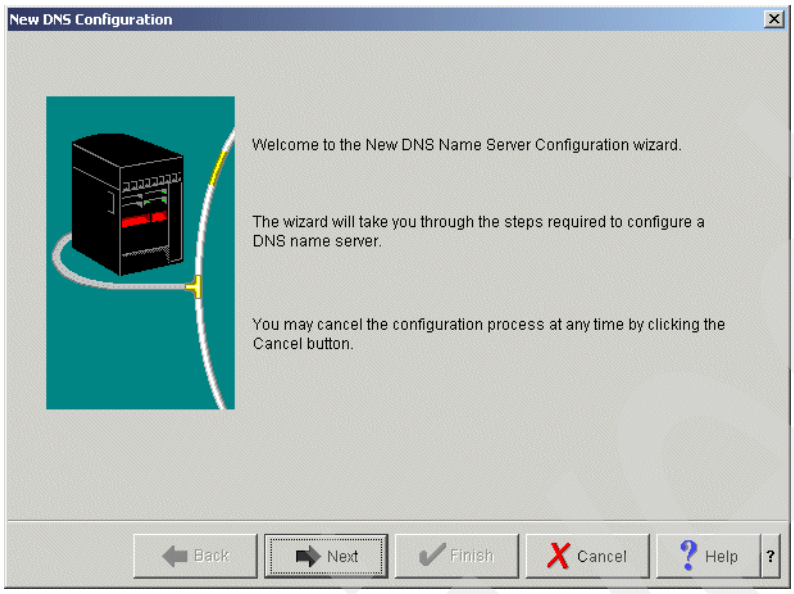


Figure 16-139 DNS Configuration wizard: New DNS Configuration

3. In the DNS Server Name window, type `NS24` (answer 2 in Table 16-6 on page 469) in the Name field, as shown in Figure 16-140. Click **Next**.

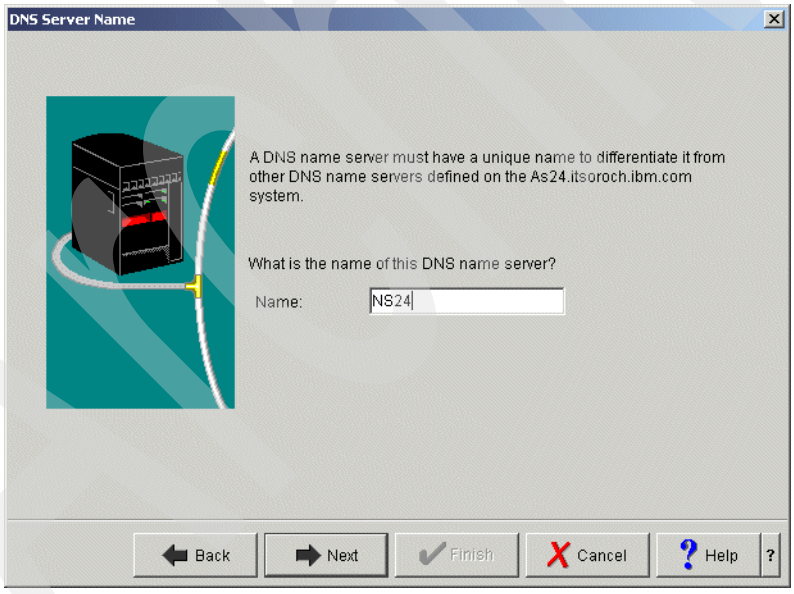


Figure 16-140 DNS Configuration wizard: DNS Server Name

4. In the Listen on IP Addresses window, choose **Select IP addresses** and select **172.23.10.2** (answer 3 in Table 16-6 on page 469) from the IP Address field, as shown in Figure 16-141. Click **Next**.

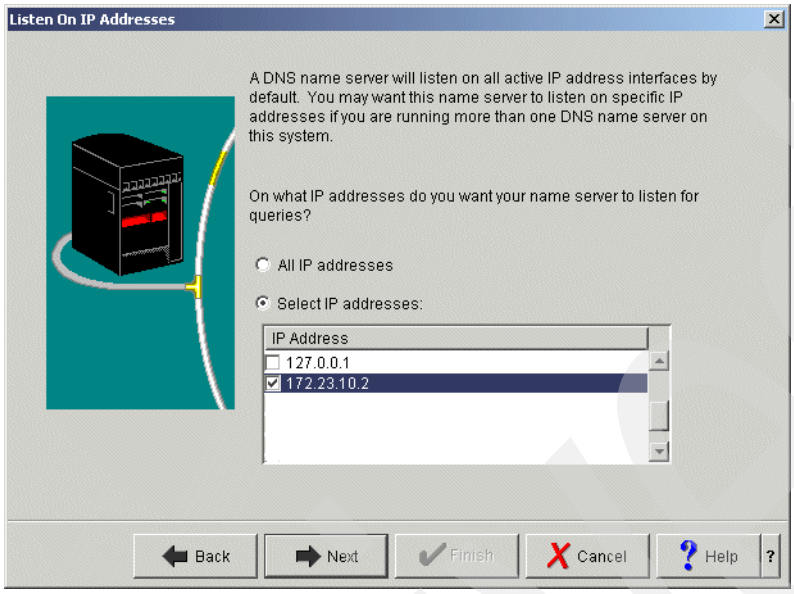


Figure 16-141 DNS Configuration wizard: Listen On IP Addresses

5. In the Root Servers window, click **Next**.
6. In the Start DNS Name Server window, select **Yes** (answer 4 in Table 16-6 on page 469), as shown in Figure 16-142. Click **Next** to continue.

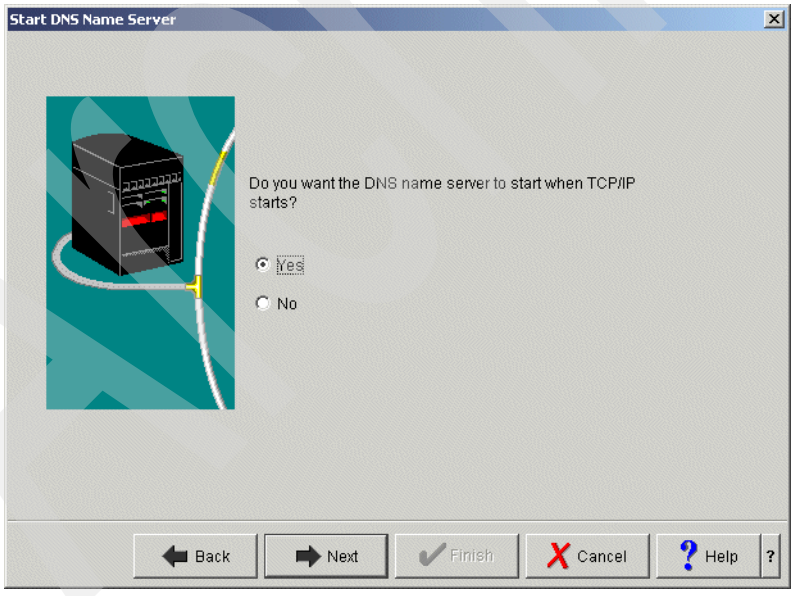


Figure 16-142 DNS Configuration wizard: Start DNS Name Server

7. In the Summary window, click **Finish**, as shown in Figure 16-143.

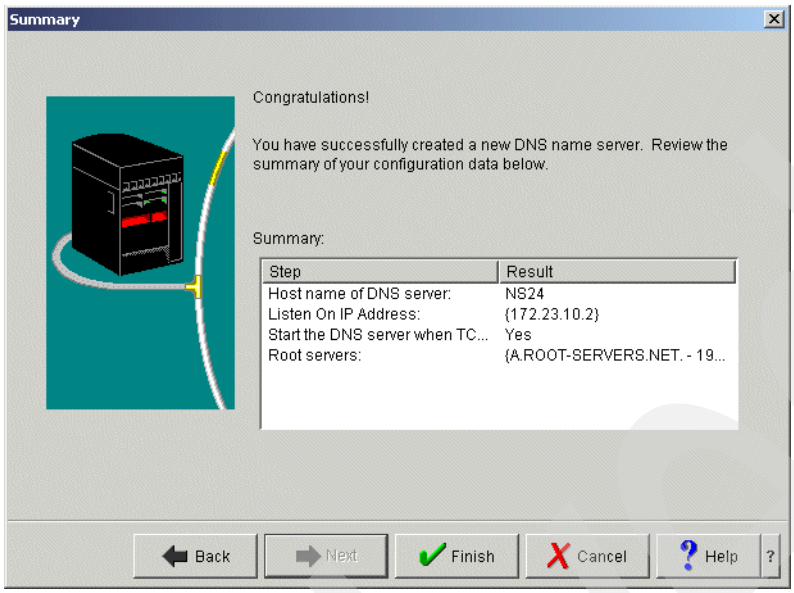


Figure 16-143 DNS Configuration wizard: Summary

Step 2: Configure the secondary zone on NS24

To configure the secondary zone on DNS server instance NS24:

1. In the iSeries Navigator window, right-click **NS24**. From the context menu select **Configuration**, as shown in Figure 16-144.

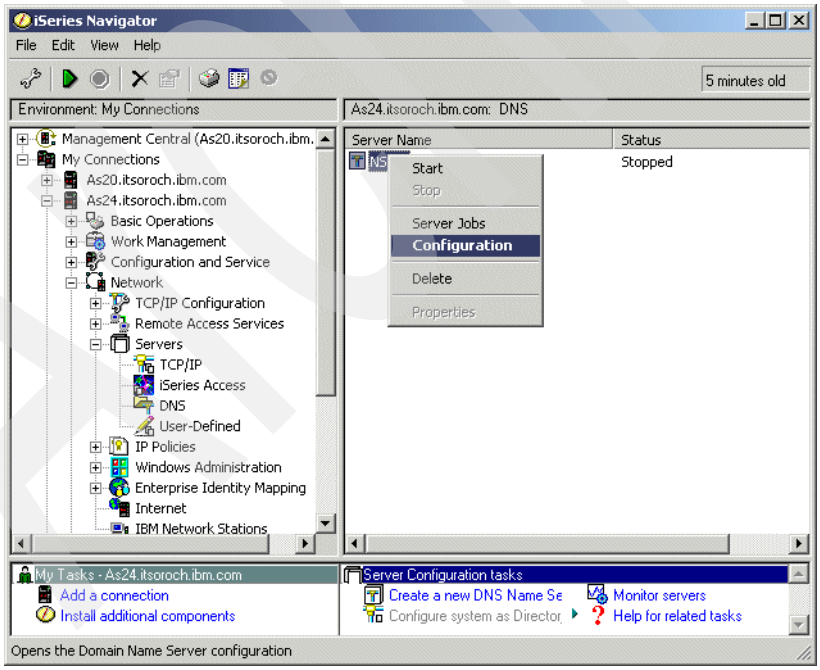


Figure 16-144 iSeries Navigator - starting the configuration of DNS server NS24

2. The DNS Configuration window opens. Right-click **Forward Lookup Zones**, and from the context menu, select **New Secondary Zone**, as shown in Figure 16-145.

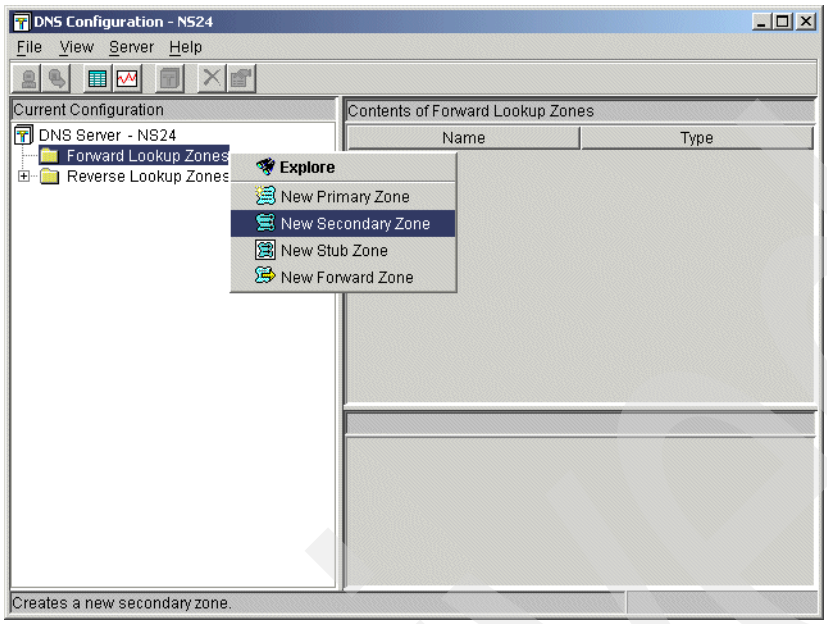


Figure 16-145 DNS Configuration: creating a new secondary forward lookup zone

3. In the Zone Domain Name window, type the name of the new zone, `itsoroch.ibm.com.`, as shown in Figure 16-146. Click **Next**.

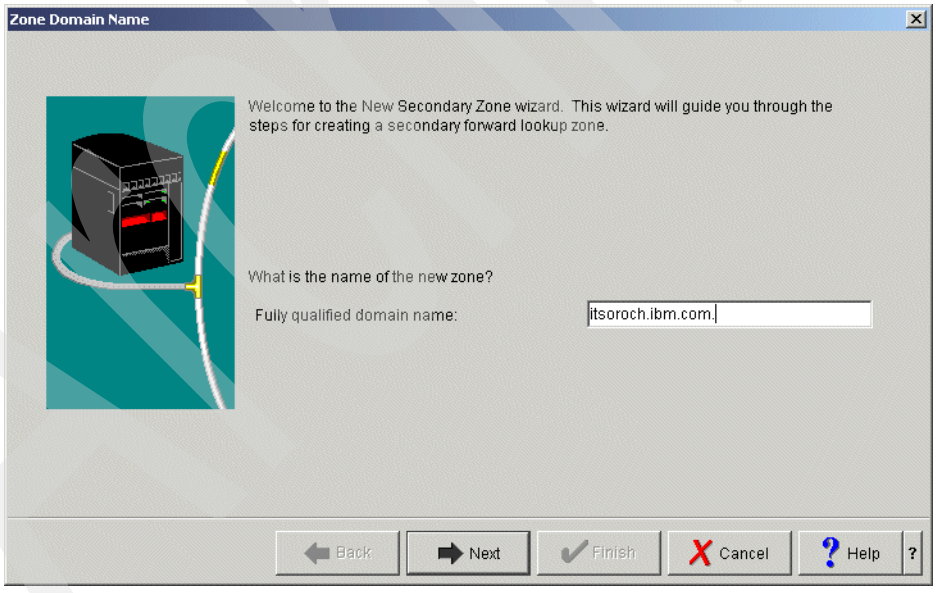


Figure 16-146 New Zone Domain: Zone Domain Name

4. In the Master Server window, click **Add** and type the IP address `172.23.10.1` (answer 6 in Table 16-6 on page 469), as shown in Figure 16-147. Click **Next**.

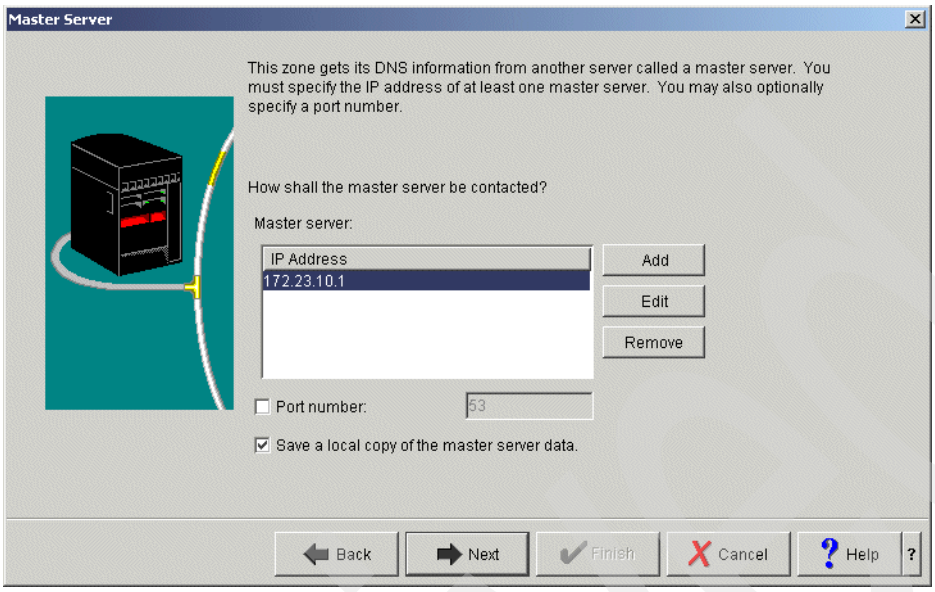


Figure 16-147 New Zone Domain: Master Server

5. In the Summary window, click **Finish,** as shown in Figure 16-148.

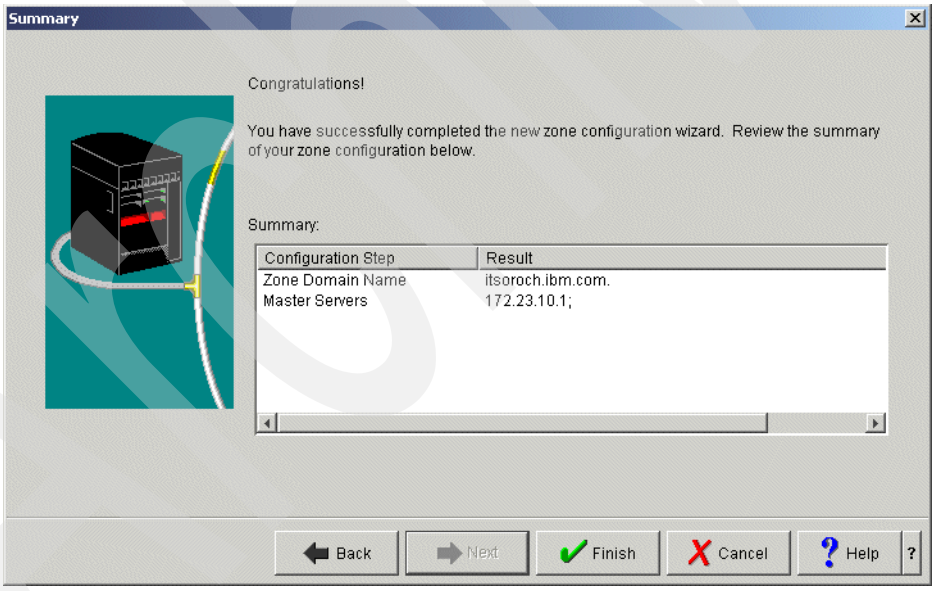


Figure 16-148 New Zone Domain: Summary

6. In the DNS Configuration window, right-click **Secondary Zone itsoroch.ibm.com** and select **Properties**, as shown in Figure 16-149.

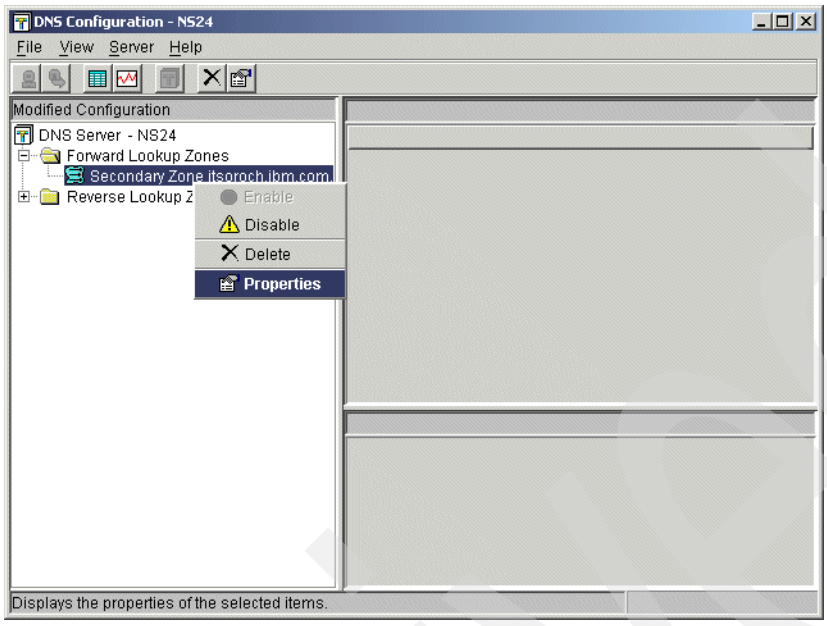


Figure 16-149 DNS Configuration: secondary zone properties

7. In the Secondary Zone Properties window, click the **Options** tab. Expand **Access Control**. For Match list element type, choose **Access Control List** from the pull-down menu, and click **Add**. In the Access Control List window, select **any** and click **OK**. Your Secondary Zone Properties window should look like Figure 16-150. Click **OK** to continue.

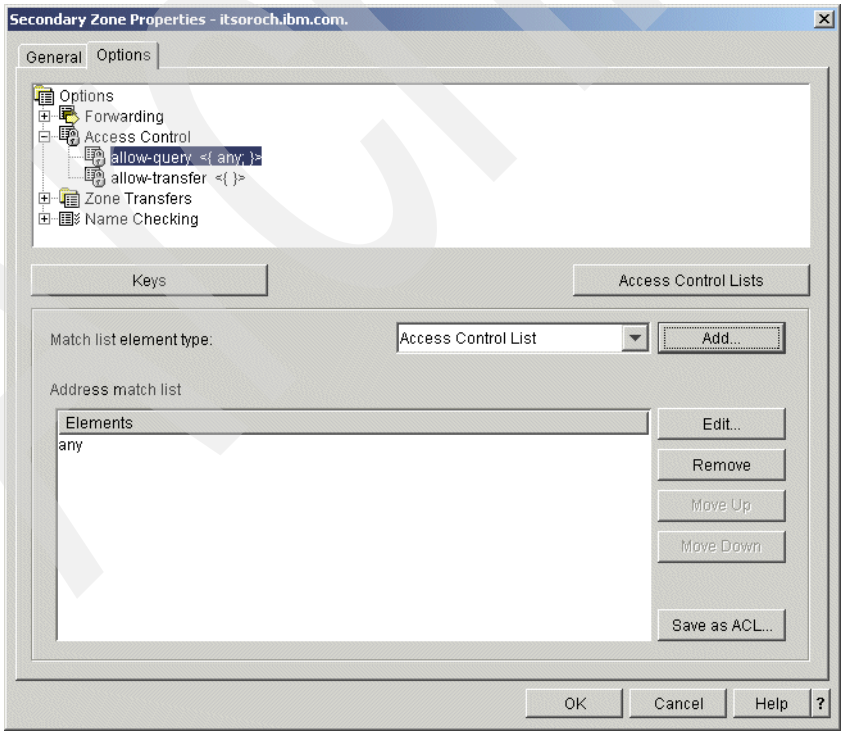


Figure 16-150 Secondary zone properties: Options tab

8. In the DNS Configuration window, right-click **Secondary Zone 10.23.172.in-addr.arpa.** and select **Properties**, as shown in Figure 16-151.

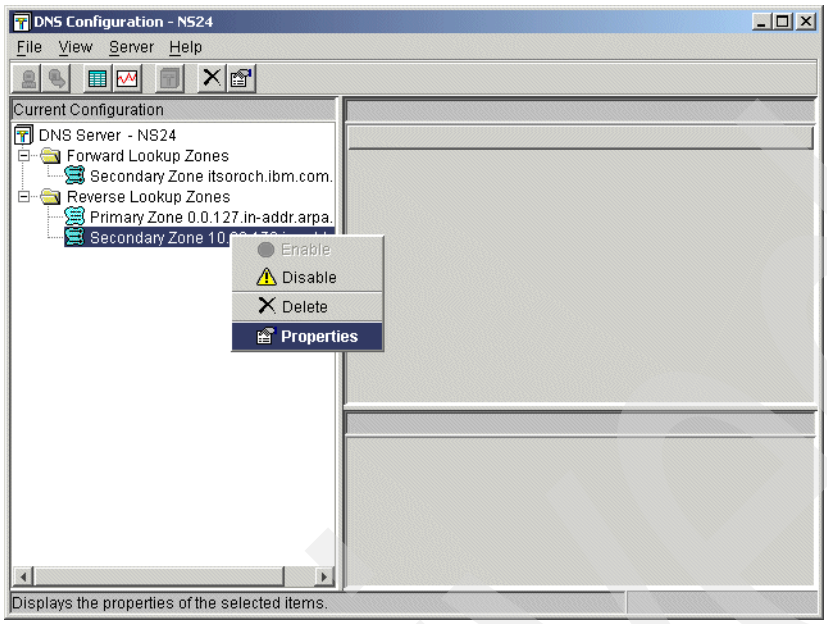


Figure 16-151 DNS Configuration window

9. In the Secondary Zone Properties window, click the **Options** tab. Expand **Access Control** and select **allow-query**. For Match list element type, choose **Access Control List** from the pull-down menu. Click **Add**.

10. In the Access Control List window, select **any** and click **OK**. This returns to the Secondary Zone Properties window, which should look like Figure 16-152. Click **OK** to continue.

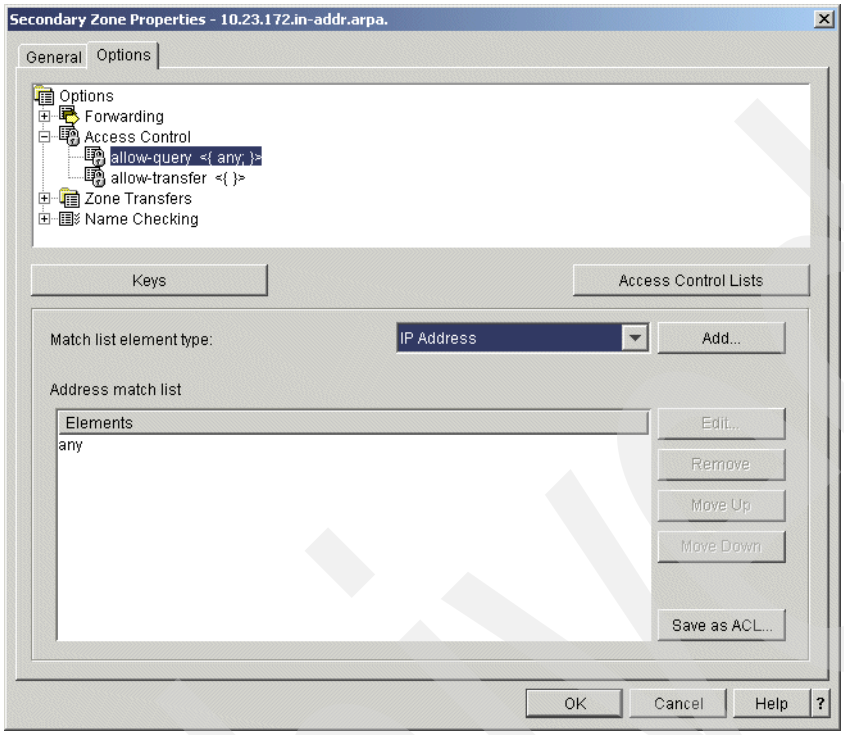


Figure 16-152 Secondary Zone Properties window

11. In the DNS Configuration window, right-click **DNS Server NS24** and select **Properties**, as shown in Figure 16-153.

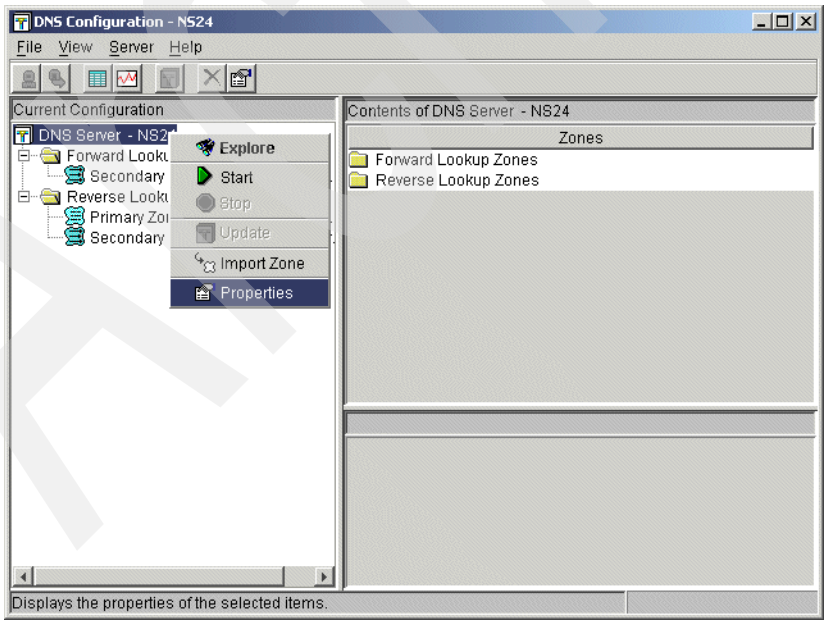


Figure 16-153 DNS Configuration window

12. In the Server Properties window, click the **Remote Name Servers** tab and click **Add**. Type `172.23.10.1` in the server IP address field. Click **support-ixfr** and select **Support incremental zone transfer**, as shown in Figure 16-154.

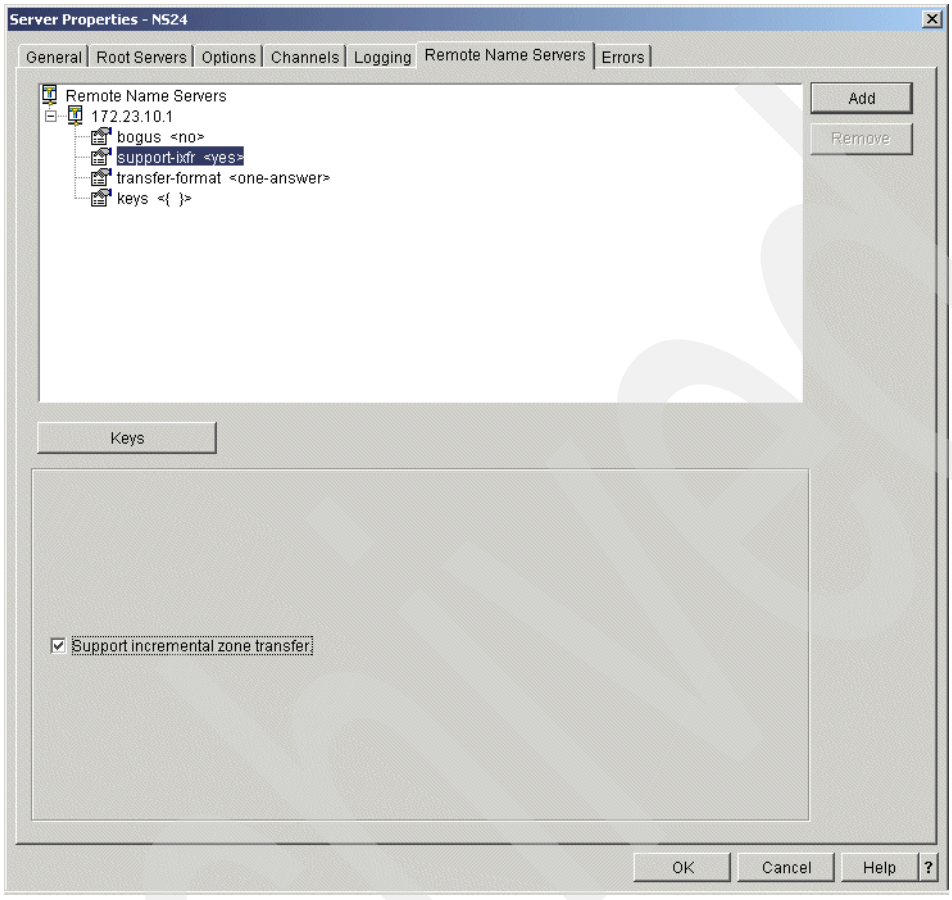


Figure 16-154 Server Properties window

13. Click **OK** to return to the DNS Configuration window. Select **File** → **Save Configuration** to save the configuration.

Step 3: Update the DNS on AS20 to use AS24's DNS as secondary

For this scenario, you must update the primary zone information in the DNS server on System i AS20 to automatically update the secondary DNS server on AS24.

1. On Primary Zone itsoroch.ibm.com. on DNS NS20, add a Host entry for as24.itsoroch.ibm.com with IP address 172.23.10.2. Refer to Figure 16-155:
   a. Right-click **Primary Zone itsoroch.ibm.com** and choose **New** → **Host** to enter the Host entry.
2. Add a Name Server (NS) resource on Primary Zone itsoroch.ibm.com for as24.itsoroch.ibm.com:
   a. Right-click **Primary Zone itsoroch.ibm.com** and choose **Properties**.
   b. In the Primary Zone Properties window, click the **Resources** tab.
   c. Click **Add** and choose **Name Server (NS)** from the list.
   d. Type `as24.itsoroch.ibm.com.` in the field. Type `1` and choose **days** for Cache Time Interval. Click **OK**. In the next window type `172.23.10.2` as a IP address for Name server as24.itsoroch.ibm.com. Click **OK**. Your DNS Configuration window should appear, as shown in Figure 16-155.

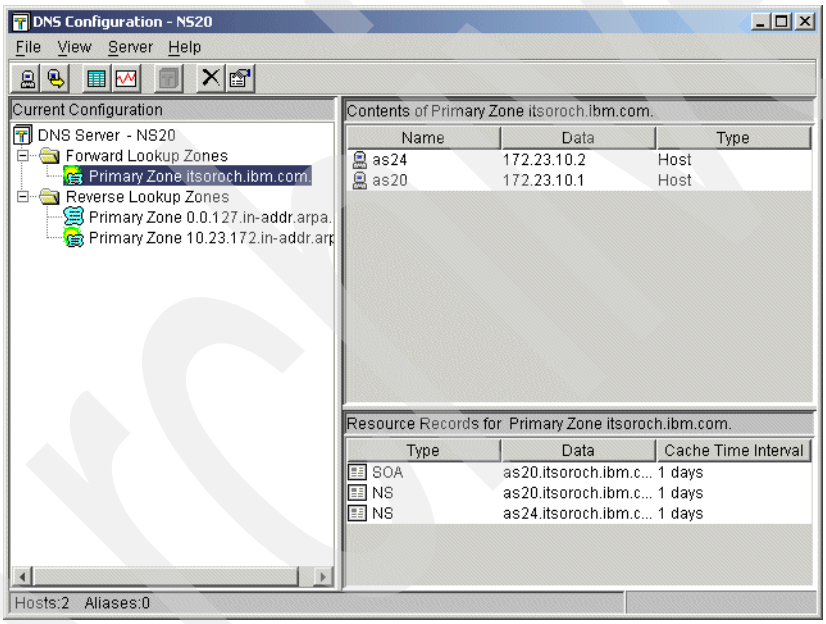


Figure 16-155 DNS Configuration window

3. On Primary Zone 10.23.172.in-addr-arpa on DNS NS20, add a Host entry for as24.itsoroch.ibm.com with IP address 172.23.10.2. Refer to Figure 16-156 on page 460:
   a. Right-click **Primary Zone 10.23.172.in-addr.arpa** and choose **New** → **Host** to enter the Host entry.
4. Add a Name Server (NS) resource on Primary Zone 10.23.172.in-addr.arpa:
   a. Right-click **Primary Zone 10.23.172.in-addr.arpa** and choose **Properties**.
   b. In the Primary Zone Properties window, click the **Resources** tab.
   c. Click **Add** and choose **Name Server (NS)** from the list. Type `as24.itsoroch.ibm.com.` in the field. Type `1` and choose **days** for Cache time interval. Click **OK**.

d. Click **OK** in the Primary Zone Properties window. You should see a window, as shown in Figure 16-156. To save the configuration select **File** → **Save Configuration**.

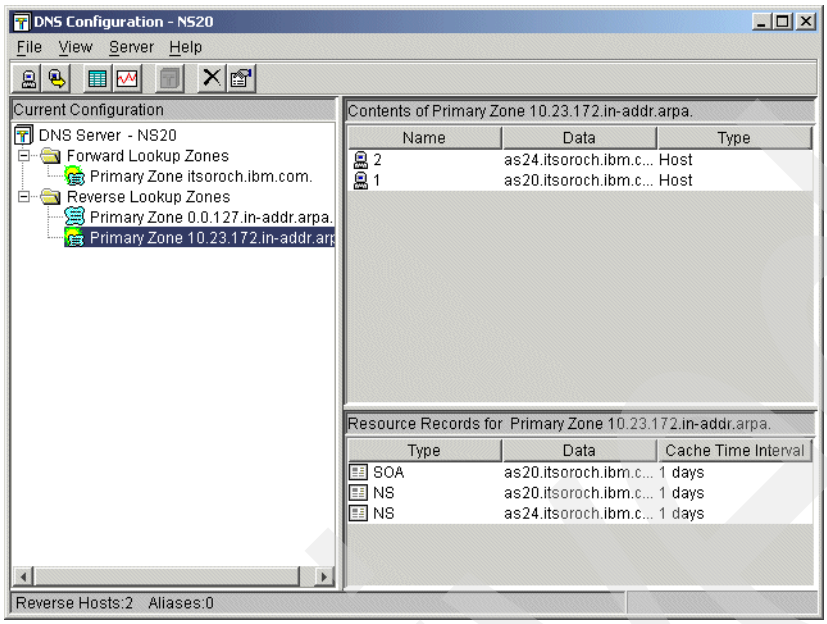


Figure 16-156 DNS Configuration window

16.5 Primary DDNS and DHCP servers, secondary DNS server Red Hat Linux 7.2

This is the procedure to configure a primary DDNS server and DHCP server on the same server, plus a secondary DNS server running under Red Hat Linux 7.2.

In this scenario, the DHCP server and primary DDNS server are configured on the same server, and the DHCP server updates A and PTR records to the DDNS server dynamically right after the DHCP server assigns a IP address to the client.

Our example procedure configures a secondary DNS server running under Red Hat Linux 7.2 for zone itsoroch.ibm.com. The secondary DDNS is updated by the primary DDNS every time the primary DDNS is dynamically updated by the DHCP server.

This scenario is based on scenario in 16.1, “Single DDNS and DHCP server on the same server” on page 368.

16.5.1 Scenario overview

You might choose this scenario if these conditions apply:

- If you have already configured DDNS and DHCP servers on AS20 using the procedure described in scenario number one (see 16.1, “Single DDNS and DHCP server on the same server” on page 368).
- If you need the secondary DNS server running under Red Hat Linux 7.2 as a fault-tolerant backup.

Sample network configuration

Figure 16-157 shows the sample network configuration of this scenario.

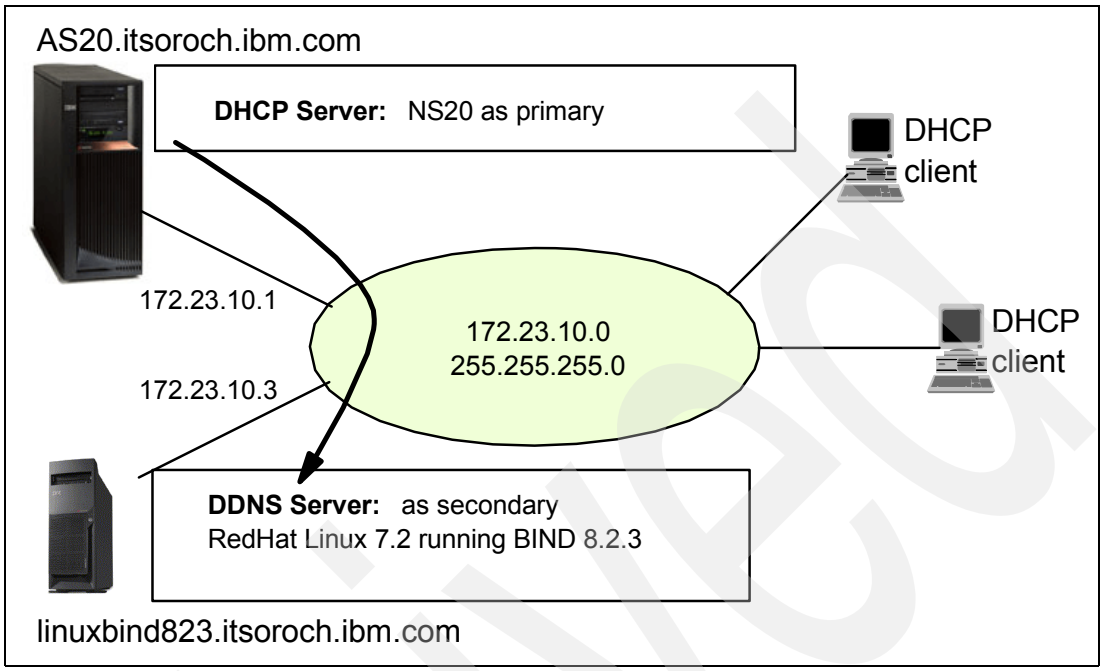


Figure 16-157 Primary DDNS server and DHCP server on the same server, plus secondary DNS server running under Red Hat Linux 7.2

16.5.2 Planning worksheet: secondary DDNS

Table 16-4 on page 448 shows the planning worksheet for preparing the required parameters to configure the secondary DNS server running under a Linux operating system. We have filled in our answers for each question in the adjacent Scenario answers column.

In this example, we used Red Hat Linux 7.2 as our Linux operating system.

Table 16-5 Planning worksheet for a secondary DNS server under Red Hat Linux 7.2

No	Questions to create a secondary DNS server running under Red Hat Linux 7.2	Scenario answers
1	What is the host name and IP address of the secondary DNS server running under Red Hat Linux 7.2?	linuxbind823.itsoroch.ibm.com 172.23.10.3
2	What is the name of the domain?	itsoroch.ibm.com
3	What is the address of the primary server for zone itsoroch.ibm.com?	172.23.10.1

16.5.3 Configuration: secondary DDNS

To implement the configuration in this scenario, perform the following steps:

- Step 1: Create named.conf file for secondary DNS server linuxbind823.
- Step 2: Add Host and NS entries on primary DDNS server NS20 on AS20.
- Step 3: Test the configuration.

Step 1: Create named.conf file for secondary DNS server linuxbind823

To create the named.conf file for secondary DNS server linuxbind823 on a Linux 7.2 system, perform the following steps:

1. On your workstation, copy the contents shown in Figure 16-158 and paste it into a new text file. Save the new text file as **named.conf**.

```
## named.conf - configuration for bind

options {
   directory "/var/named/";
};

zone "itsoroch.ibm.com" {
   type slave;
   file "itsoroch.ibm.com.zone";
   masters {
       172.23.10.1;
      };
};

zone "10.23.172.in-addr.arpa" {
   type slave;
   file "10.23.172.in-addr.arpa.zone";
   masters {
       172.23.10.1;
      };
};

zone  "." {
   type hint;
   file  "named.ca";
};

zone  "0.0.127.in-addr.arpa" {
   type master;
   file  "0.0.127.in-addr.arpa.zone";
};

zone  "localhost" {
   type master;
   file  "localhost.zone";
};
```

Figure 16-158 Sample named.conf file

Note: This sample named.conf file includes the minimum required parameters to run secondary server. You may need to consider other parameters to configure the DNS configuration as required.

There are two ways to transfer the named.conf file from your workstation to the Linux system. The first is to use a diskette and the second is to FTP the file.

Transfer the named.conf file via diskette

This procedure transfers the named.conf file onto diskette that you can move from your workstation to the Linux system:

1. Open a command shell and type:

   ```
   cp /mnt/floppy/named.conf /etc
   ```

   If you receive a message that the system is going to overwrite named.conf file, reply **Y** to continue.

2. Type the following command to ensure the permission of the named.conf file:

   ```
   cd /etc
   ls -l named.conf
   ```

   -rw-r--r-- named.conf is the correct permission of named.conf file. If the permission level is different, type the following command to change the permission level:

   ```
   chmod 644 named.conf
   ```

   This command changes the permission level of named.conf. Type the following command to ensure that the permission level was changed as expected:

   ```
   ls -l named.conf
   ```

   You should see `-rw-r--r-- named.conf`

Transfer the named.conf file via FTP

Another way to transfer the named.conf file is to use the **ftp** command to send it from your workstation to the Linux system. Start the FTP daemon (server) on the Linux system to establish an FTP session between it and your workstation.

If you use FTP to transfer the named.conf file, ensure the permission level of named.conf. If the permission level is not as expected, use the **chmod** command to change the permission level. Refer to the instructions in the previous section, “Transfer the named.conf file via diskette.”

Step 2: Add Host and NS entries on primary DDNS server NS20 on AS20

Add Host and NS entries of secondary DNS server linuxbind823 on primary DDNS server NS20 using the following steps:

1. On Primary Zone itsoroch.ibm.com on DNS NS20, add Host entry for linuxbind823.itsoroch.ibm.com with IP address 172.23.10.3:
   a. Right-click **Primary Zone itsoroch.ibm.com** and choose **New** → **Host** to open the window to enter the Host entry.
2. You must add a Name Server (NS) resource on Primary Zone itsoroch.ibm.com:
   a. Right-click **Primary Zone itsoroch.ibm.com** and choose **Properties**.
   b. In the Primary Zone Properties window, click the **Resources** tab.
   c. Click **Add** and choose **Name server (NS)** from the list.
   d. Type `linuxbind823.itsoroch.ibm.com` in the field. Type `1` and choose **days** for Cache time interval. Click **OK**.

e. In the next window, type `172.23.10.3` as an IP address for name server linuxbind823.itsoroch.ibm.com in the next window. Click **OK**. Your DNS Configuration window should appear, as shown in Figure 16-159.

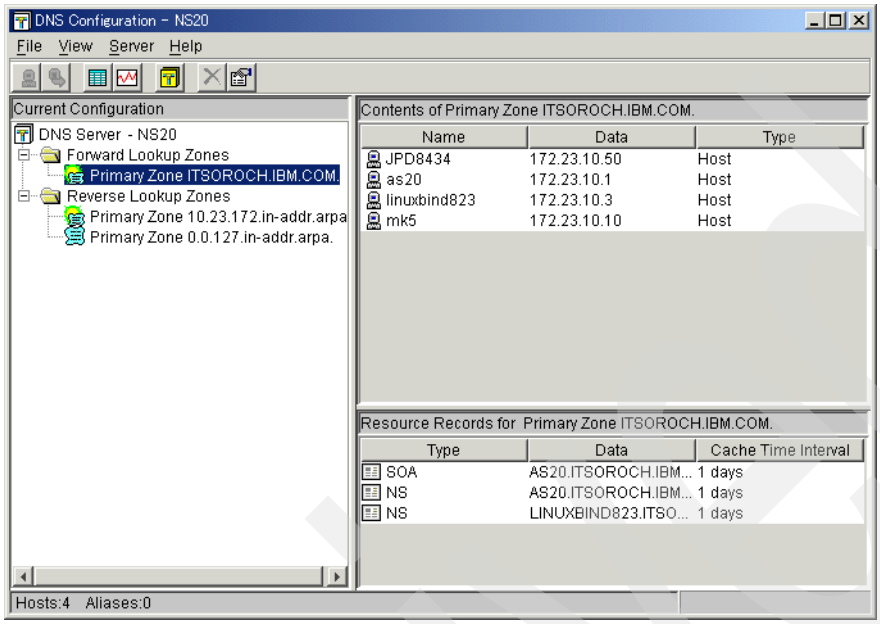


Figure 16-159 DNS Configuration window

3. On Primary Zone 10.23.172.in-addr-arpa on DNS NS20, add Host entry for linuxbind823.itsoroch.ibm.com with IP address 172.23.10.3:
   a. Right-click **Primary Zone 10.23.172.in-addr.arpa** and choose **New** → **Host** to open the window to enter the Host entry.
4. You also need to add Name Server (NS) resource on Primary zone 10.23.172.in-addr.arpa:
   a. Right-click **Primary Zone 10.23.172.in-addr.arpa** and choose **Properties**.
   b. In the Primary Zone Properties window, click the **Resources** tab.
   c. Click **Add** and choose **Name server (NS)** from the list.
   d. Type `linuxbind823.itsoroch.ibm.com.` in the field. Type `1` and choose **days** for Cache time interval. Click **OK**.

e. In the Primary Zone Properties window, click **OK**. A window should appear as shown in Figure 16-160.

f. To save the configuration, select **File** → **Save Configuration**.

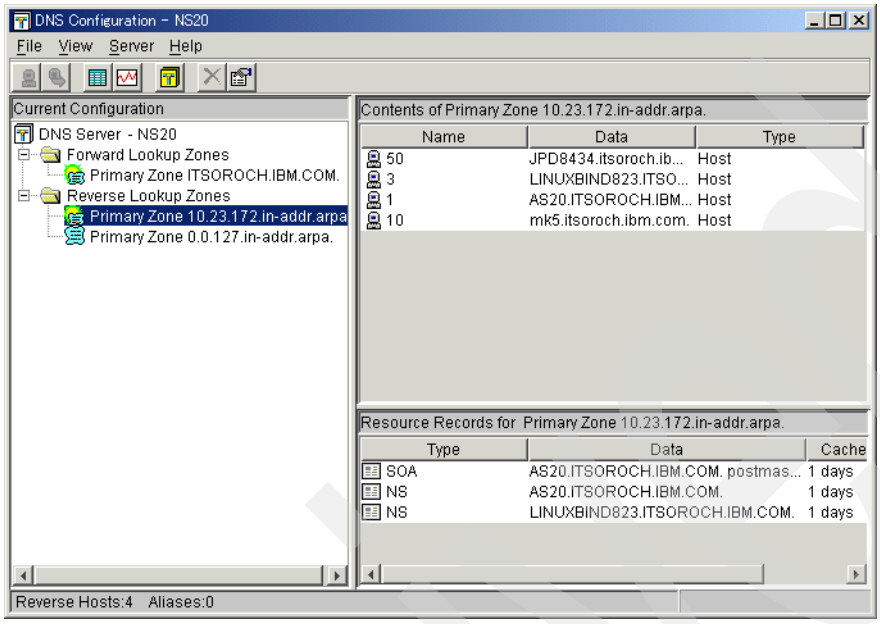


Figure 16-160 DNS Configuration window

Step 3: Test the configuration

In this step, start the secondary DDNS on the Linux operating system to confirm that the secondary DDNS is working as expected:

1. Open the command shell on the Linux operating system. To start the secondary DDNS server, type the following command:

   ```
   /sbin/service named start
   ```

   If named (Secondary DDNS server - BIND) brought up successfully, you will see the OK message on the panel. If named did not bring up successfully, check the contents of named.conf, opening it using the Kate editor.

   Tip: The Kate editor is installed with the KDE environment. If you haven't installed KDE environment, install KDE first.

   On the Kate editor panel, compare the contents with the named.conf file shown in Figure 16-158 on page 462 and change any discrepancies. In the Kate editor, use the Tab key to move the cursor on the panel. You should not use the spacebar to move the cursor on the panel. Now try to start the secondary server again.

2. Use the test procedure in "Step 4: Test the configuration" on page 396. In this test procedure, you issue the **`ipconfig /release`** and **`ipconfig /renew`** commands on a Windows 2000 workstation to lease or release an IP address. You can use the **`nslookup`** command to confirm that A and PTR records are updated to the secondary DDNS server dynamically.

   Tip: Be patient. In our lab environment, the dynamic update from Primary DDNS server to the secondary DDNS server sometimes took several minutes.

3. The alternative way to check whether A and PTR records are updated dynamically is to observe A and PTR record files. Type this command to observe A or PTR record files:

   `cd /var/named`

   `cat itsoroch.ibm.com.zone` (A record)

   `cat 10.23.172.in-addr.arpa.zone` (PTR record)

   Figure 16-161 shows the sample file contents of itsoroch.ibm.com.zone while Figure 16-162 shows the sample file contents of 10.23.172.in-addr.arpa.zone.

```
$ORIGIN .
$TTL 86400; 1 day
itsoroch.ibm.comIN SOAAS20.ITSOROCH.IBM.COM. postmaster.AS20.ITSOROCH.IBM.COM. (
             29         ; serial
             10800      ; refresh (3 hours)
             3600       ; retry (1 hour)
             604800     ; expire (1 week)
             86400      ; minimum (1 day)
             )
          NS AS20.ITSOROCH.IBM.COM.
          NS LINUXBIND823.ITSOROCH.IBM.COM.
$ORIGIN itsoroch.ibm.com.
as20      A  172.23.10.1
JPD8434   A  172.23.10.50
          TXT"AS400DHCP:1-0x0006290b72fa"
linuxbind823A172.23.10.3
mk5       A  172.23.10.10
          TXT"AS400DHCP:1-0x00062914b8e8"
```

Figure 16-161 itsoroch.ibm.com.zone file contents (A record)

```
$ORIGIN .
$TTL 86400; 1 day
10.23.172.in-addr.arpaIN SOAAS20.ITSOROCH.IBM.COM. postmaster.AS20.ITSOROCH.IBM.COM. (
             27         ; serial
             10800      ; refresh (3 hours)
             3600       ; retry (1 hour)
             604800     ; expire (1 week)
             86400      ; minimum (1 day)
             )
          NS AS20.ITSOROCH.IBM.COM.
          NS LINUXBIND823.ITSOROCH.IBM.COM.
$ORIGIN 10.23.172.in-addr.arpa.
1         PTRAS20.ITSOROCH.IBM.COM.
10        PTRmk5.itsoroch.ibm.com.
          TXT"AS400DHCP:1-0x00062914b8e8"
3         PTRLINUXBIND823.ITSOROCH.IBM.COM.
50        PTRJPD8434.itsoroch.ibm.com.
          TXT"AS400DHCP:1-0x0006290b72fa"
```

Figure 16-162 10.23.172.in-addr.arpa.zone file contents (PTR record)

16.6 Split DNS: Private and Public DNS with masquerade NAT

We introduced Dynamic Domain Name System (Dynamic DNS) in Chapter 8, “Dynamic Domain Name System (Dynamic DNS)” on page 133. Because your DNS server contains

sensitive data, you must seriously consider the security of your configuration. While the DNS server does not contain any direct means to access any of the systems on your network, it would provide a would-be hacker with your network's topology. This section describes a *split DNS* (that is, two DNS servers running on the same System i that can help prevent a zone transfer type of attack). We demonstrate how to place a public DNS between your private DNS server and the public network.

At V5R1 and earlier releases, this configuration would require two systems (or two System i Logical Partitions, or LPARs). Starting with V5R2, you can create multiple DNS instances on the same system.

In this scenario, we implement a network solution with split DNS and masquerade Network Address Translation (NAT). A split DNS prohibits any query accesses from a public network to a private DNS. A masquerade NAT conceals the private IP addresses of each user.

16.6.1 Scenario overview

You might choose this scenario if these conditions apply:

- If there is a need to conceal your private network from a public network
- If there is one System i available and you need to configure split DNS on the single server

Sample network configuration

Figure 16-163 shows the sample network configuration of this scenario. The client gets access to the Web servers on the Internet through a masquerade NAT. The client can ask queries of the private DNS. If the private DNS does not have an answer, it forwards the query to the public DNS.

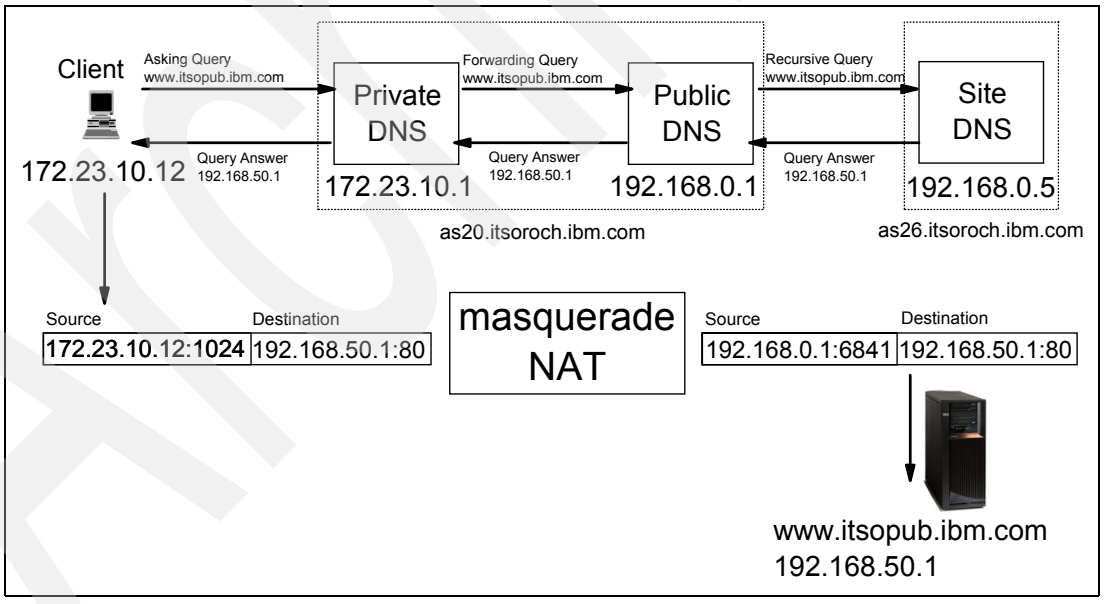


Figure 16-163 Sample network configuration: split DNS server scenario

In the example shown in Figure 16-163, Client asks for the IP address of www.itsopub.ibm.com. Because Private DNS does not have a record for this URL, it forwards the query to Public DNS. Public DNS handles recursive queries, which means that it resolves requests for an IP address from www.itsopub.ibm.com by asking the site DNS. This site DNS can be your Internet Service Provider's DNS server.

After Public DNS receives the answer 129.168.50.1 from Site DNS, Public DNS sends the query answer to Private DNS. Private DNS sends the query answer to Client. Now Client knows the IP address of www.itsopub.ibm.com.

Using the masquerade NAT, the private IP address is concealed from Public DNS. In this example, Client's private IP address 172.23.10.12 is concealed by masquerade NAT, so only one IP address, 192.168.0.1, is exposed to Public DNS.

This scenario also includes an IP filter for clients. If you need to limit client Internet use, you can set the IP filter to prevent Internet access. In this scenario, TCP port 80 (commonly used for HTTP and Web traffic) is only used for client access to the Internet.

Split DNS functional diagram

Figure 16-164 shows the functional diagram of a split DNS. Private DNS I20 has a *forwarders* option. This enables I20 to forward the unresolved queries to Public DNS E20. In I20's configuration, only private IP addresses in subnet 172.23.10.0 are allowed to ask queries to I20. In E20's configuration, any clients/servers can ask queries to E20.

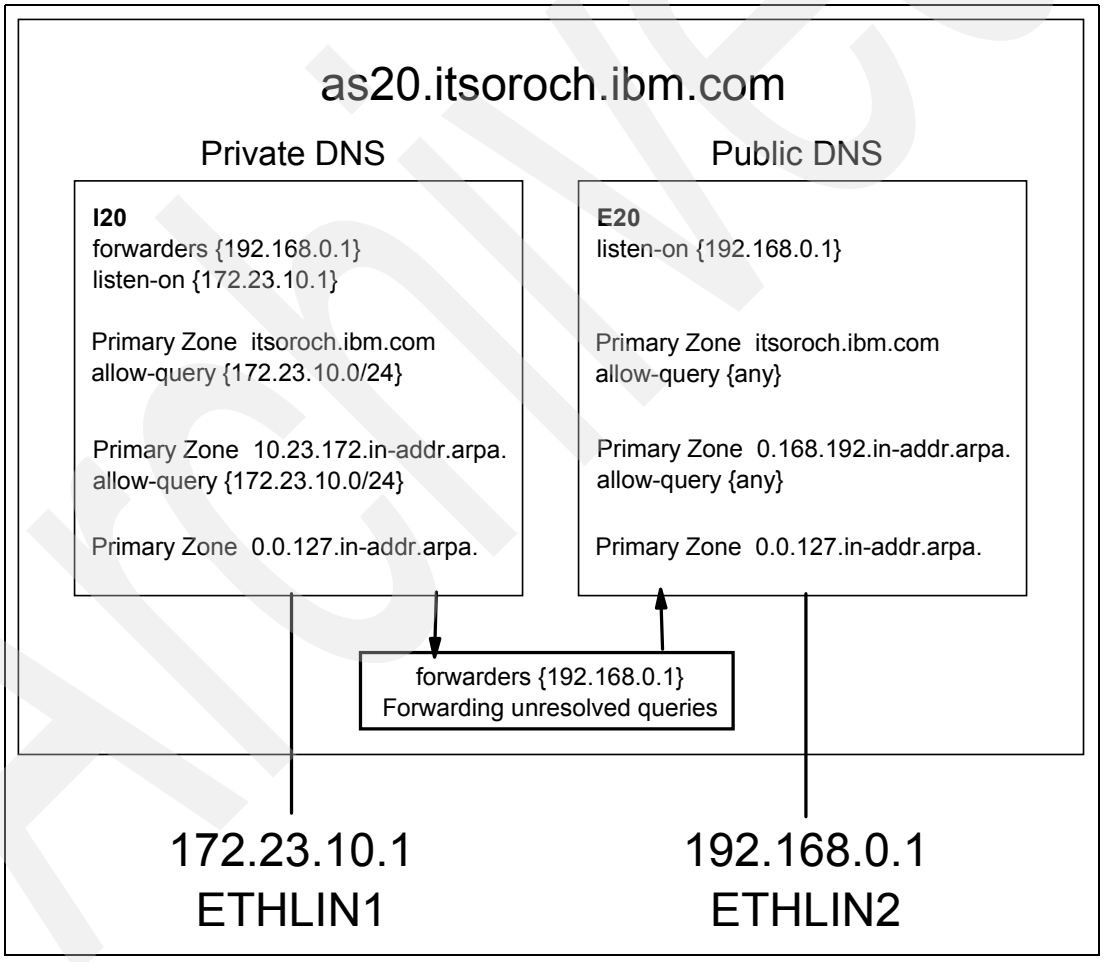


Figure 16-164 Split DNS functional diagram

16.6.2 Planning worksheet: split DNS with masquerade NAT

Table 16-6 shows the planning worksheet for preparing the required parameters to configure the split DNS with masquerade NAT scenario. We have filled in our answers for each question in the adjacent Scenario answers column.

Table 16-6 Planning worksheet for the split DNS with masquerade NAT scenario

No.	Questions to create the split DNS with masquerade NAT	Scenario answers
1	What is the private DNS instance name? What is the public DNS instance name?	I20 E20
2	What is the Ethernet line name for Private DNS I20? What is the IP address for this line?	ETHLIN1 172.23.10.1
3	What is the Ethernet line name for Public DNS E20? What is the IP address for this line?	ETHLIN2 192.168.0.1
4	What is the Site DNS IP address and fully qualified domain name? If you are using an ISP, the site DNS is the DNS server that is provided by your Internet Service Provider and it is used to resolve queries for any servers on the Internet.	192.168.0.5 as26.itsoroch.ibm.com.
5	What is the subnet of your private network?	172.23.10.0 255.255.255.0
6	What is the fully qualified server name of the System i where you are going to create private and public DNS servers? What is the domain name?	as20.itsoroch.ibm.com. itsoroch.ibm.com

16.6.3 Configuration: split DNS with masquerade NAT

This scenario outlines the scenario of a split DNS with masquerade NAT in these steps:

- Step 1: Create the private DNS I20 on AS20
- Step 2: Create the public DNS E20 on AS20
- Step 3: Create the IP filter

Step 1: Create the private DNS I20 on AS20

In this step, we create the private DNS I20 on AS20:

1. In the iSeries Navigator window, expand **Network** → **Servers**. Right-click **DNS** and choose **New Name Server,** as shown in Figure 16-165.

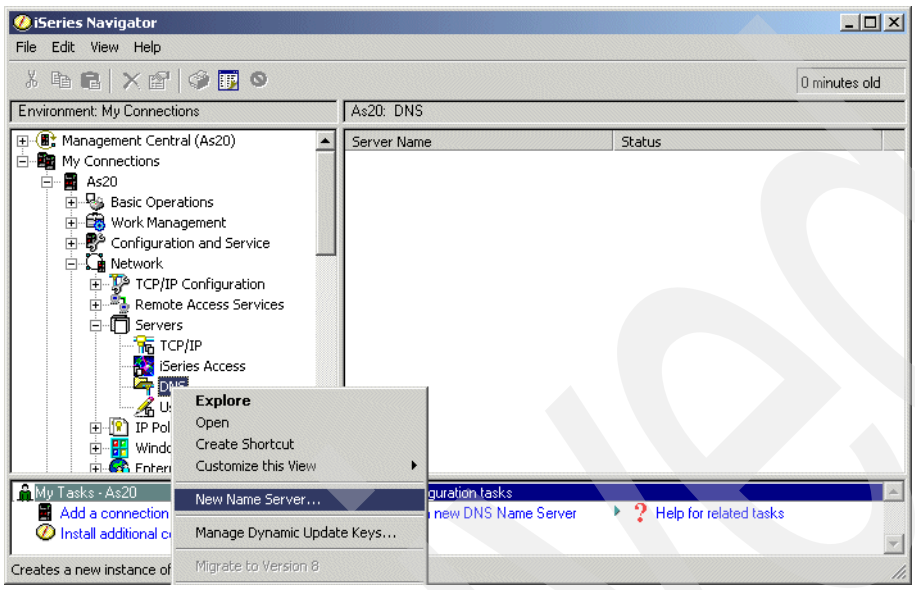


Figure 16-165 iSeries Navigator window

2. In the New DNS Configuration window, click **Next** as shown in Figure 16-166.

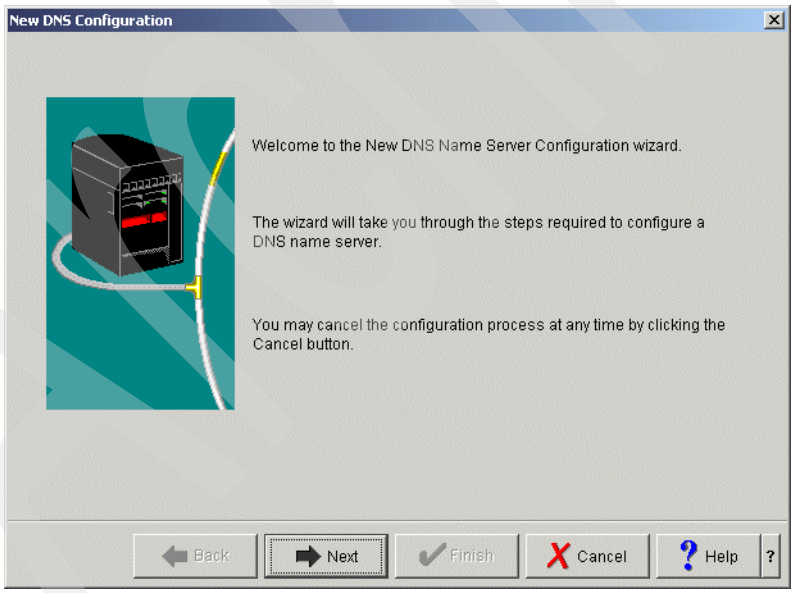


Figure 16-166 New DNS Configuration window

3. In the New DNS Configuration window (Figure 16-167), type `I20` (answer 1 in Table 16-6 on page 469). Click **Next** to continue.

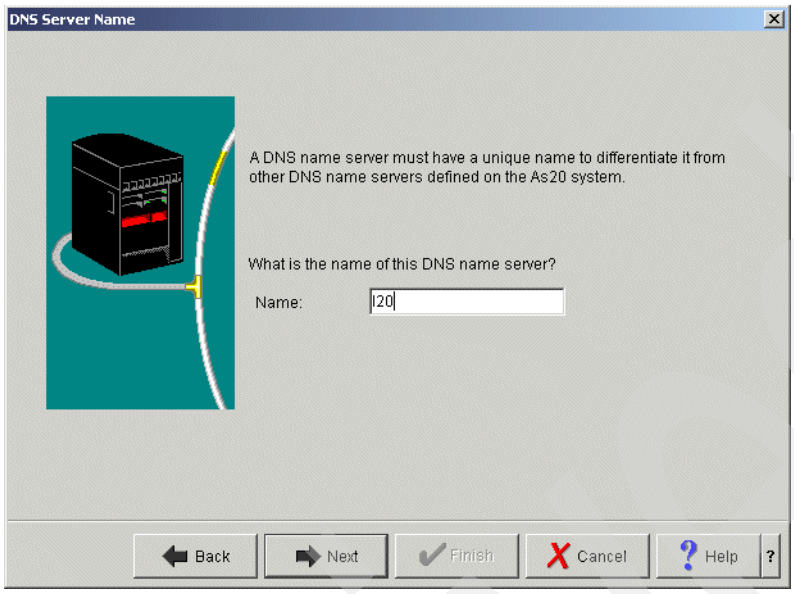


Figure 16-167 New DNS Configuration window

4. In the Listen On IP Addresses window, select **172.23.10.1** (answer 2 in Table 16-6 on page 469), as shown in Figure 16-168. Click **Next** to continue.

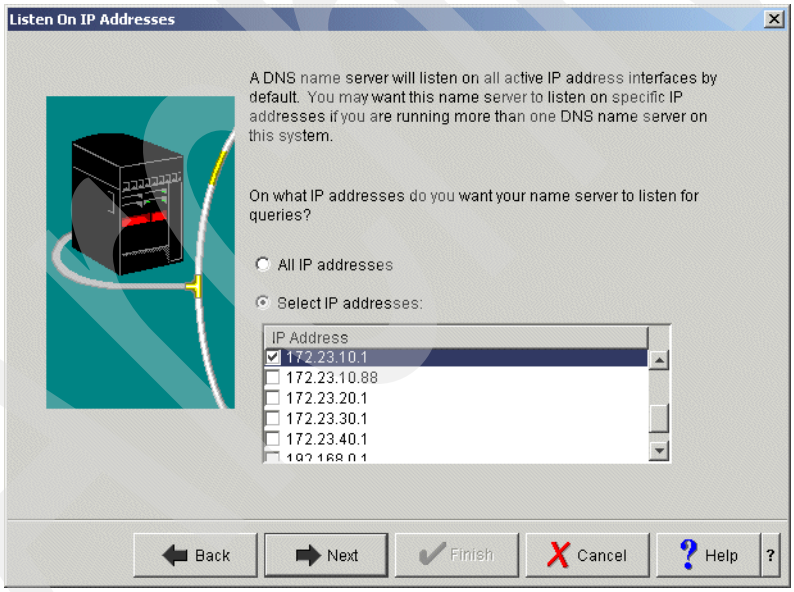


Figure 16-168 Listen On IP Address

5. In the Root Servers window (Figure 16-169) click **Next** to continue.

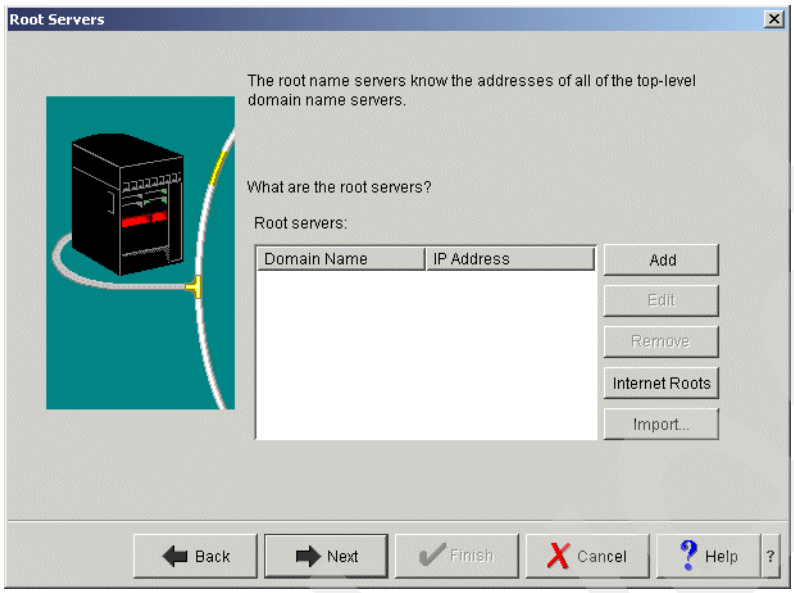


Figure 16-169 Root Servers window

6. In the Start DNS Name Server window (Figure 16-170), click **Yes**. Click **Next** to continue.

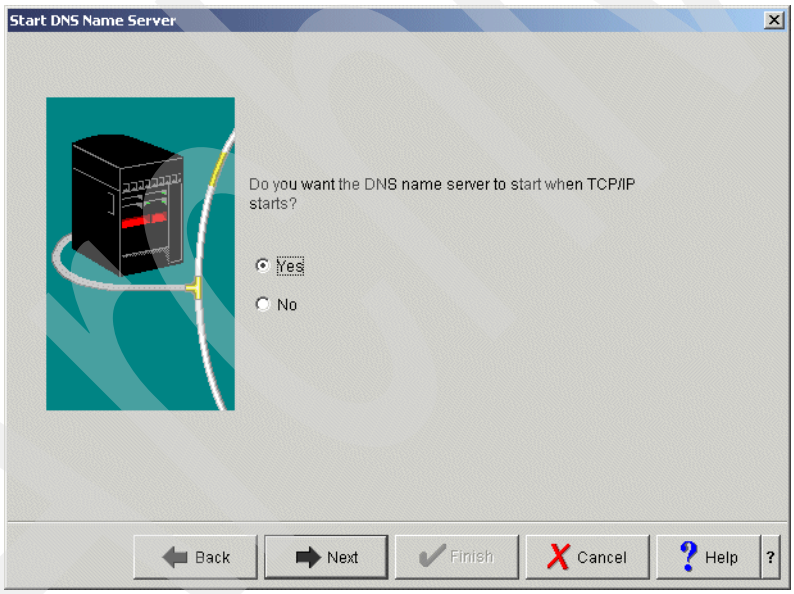


Figure 16-170 Start DNS name server window

7. In the Summary window (Figure 16-171), click **Finish**.

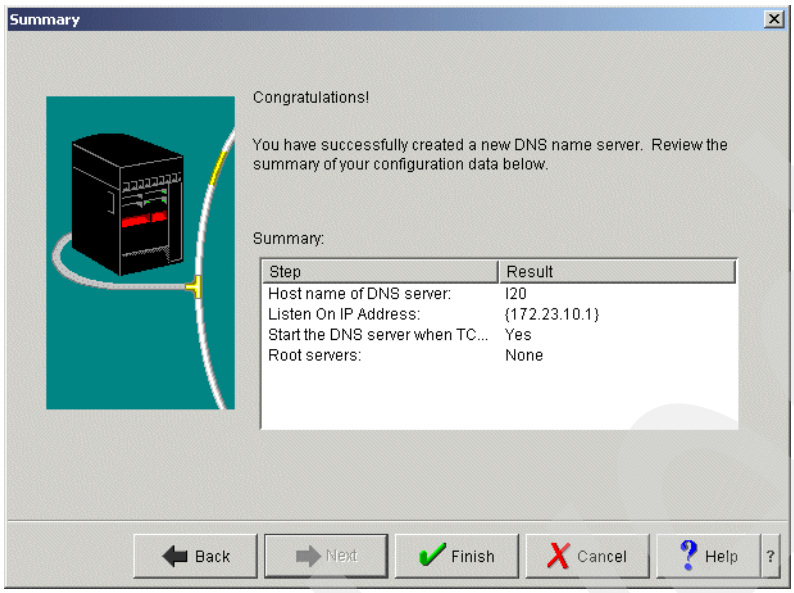


Figure 16-171 Summary window

8. This returns you to the iSeries Navigator window. Click **I20**, and from the context menu, choose **Configuration**, as shown in Figure 16-172.

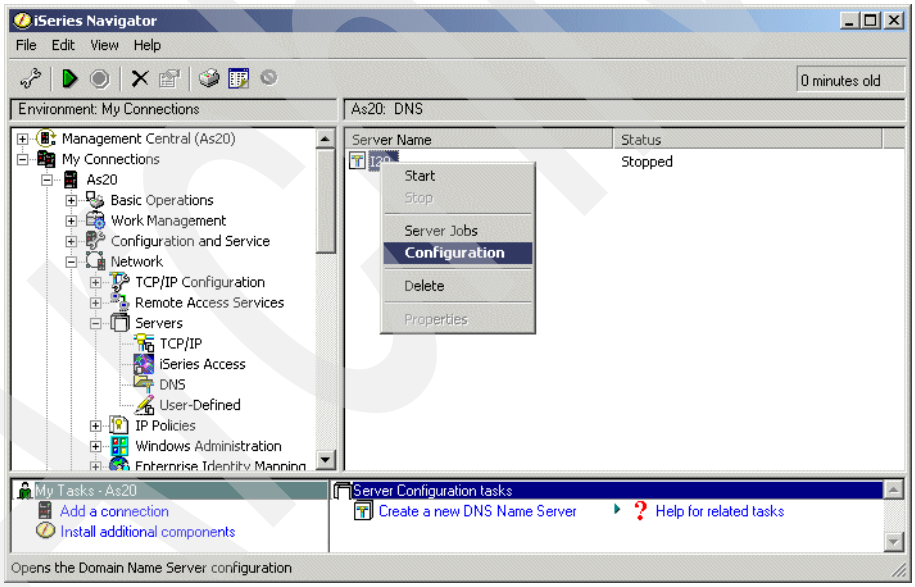


Figure 16-172 iSeries Navigator window

9. In the DNS Configuration window, right-click **DNS Server I20** and choose **Properties**, as shown in Figure 16-173.

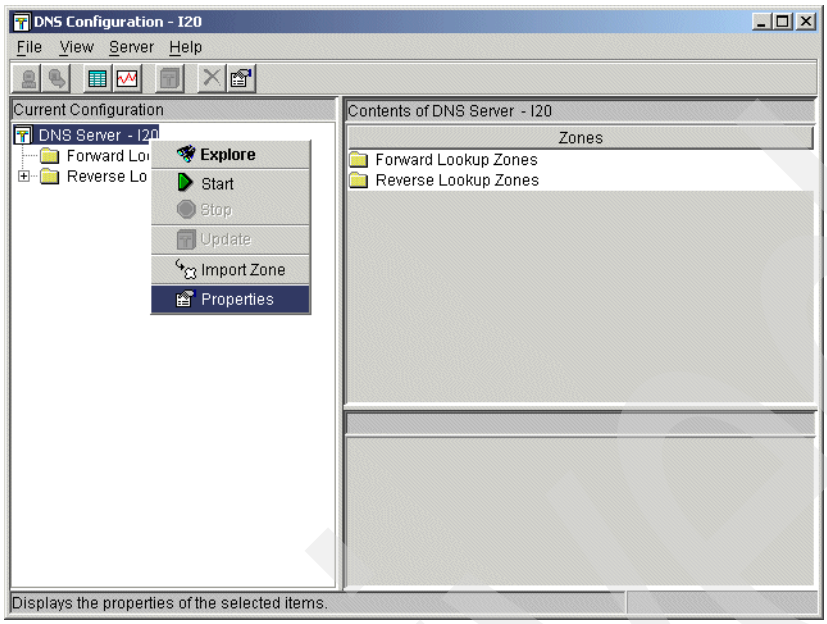


Figure 16-173 DNS Configuration window

10. In the Server Properties window, click the **Options** tab. Expand **Forwarding** and click **forward**. Select **Only - Query only forwarding**, as shown in Figure 16-174.

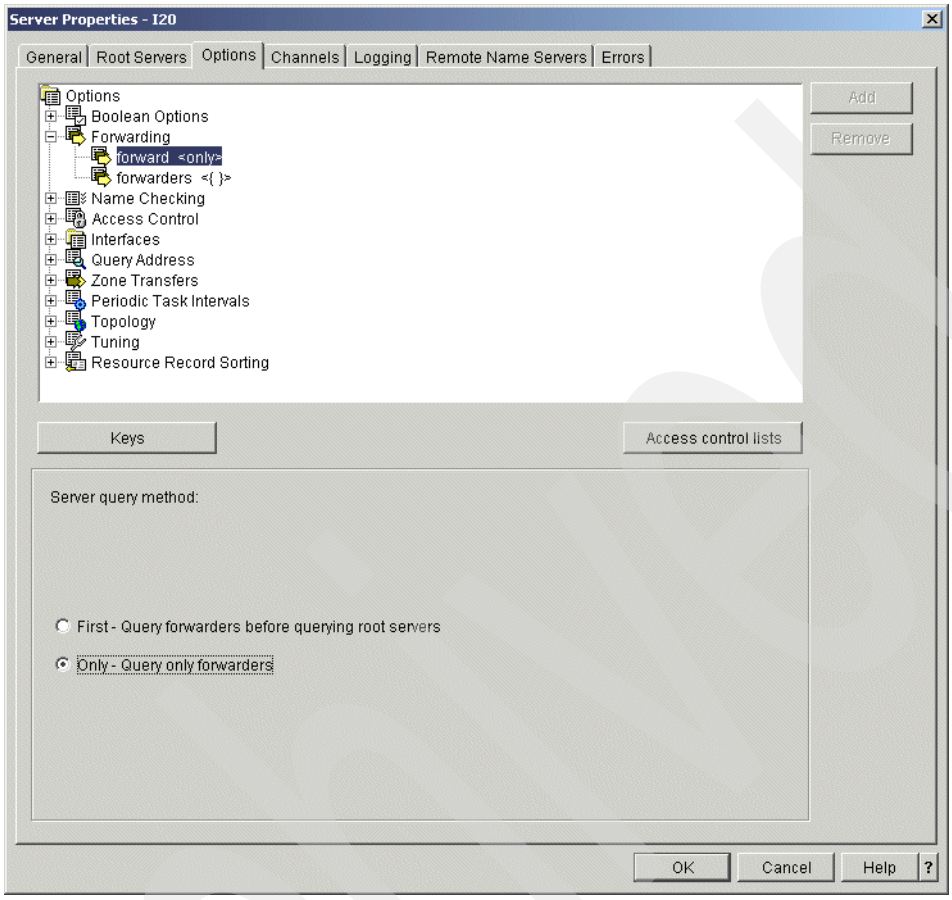


Figure 16-174 Server properties window

11. In the Server Properties window (Figure 16-175), click **forwarders**. Click **Add** and type `192.168.0.1` (answer 3 in Table 16-6 on page 469) in the IP Address field. Click **OK** to continue.

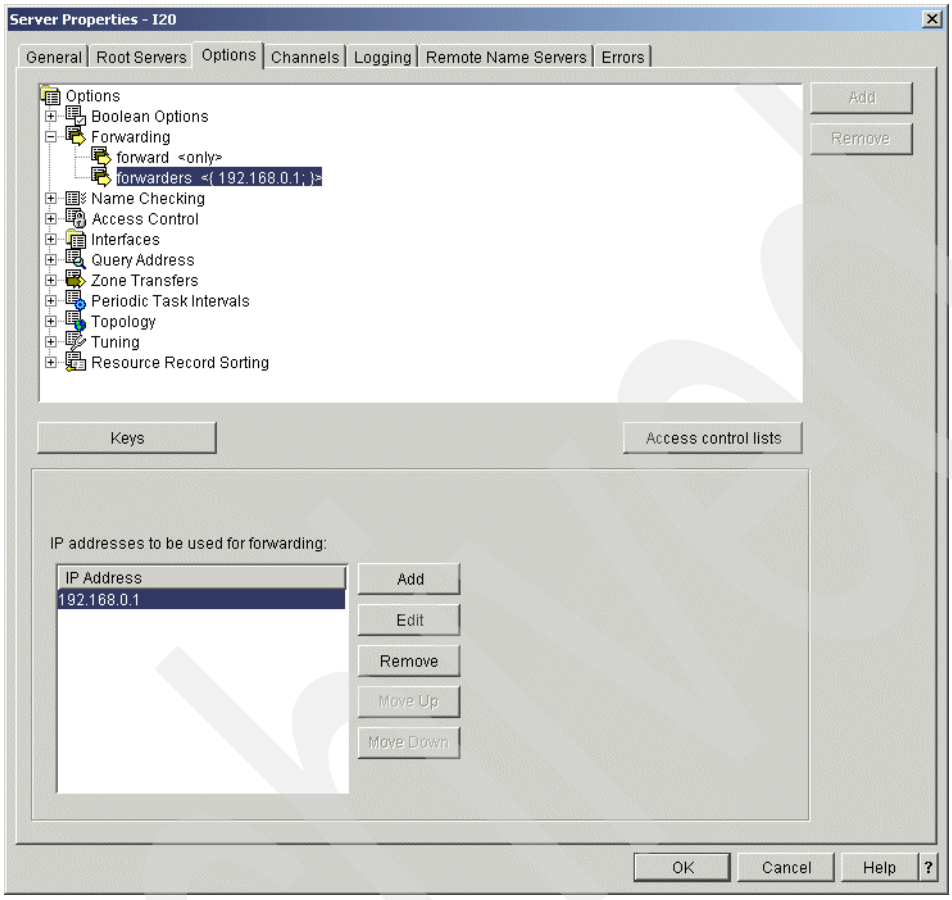


Figure 16-175 Server Properties window

12. In the DNS Configuration window, right-click **Forward Lookup Zone** and choose **New Primary Zone**, as shown in Figure 16-176.

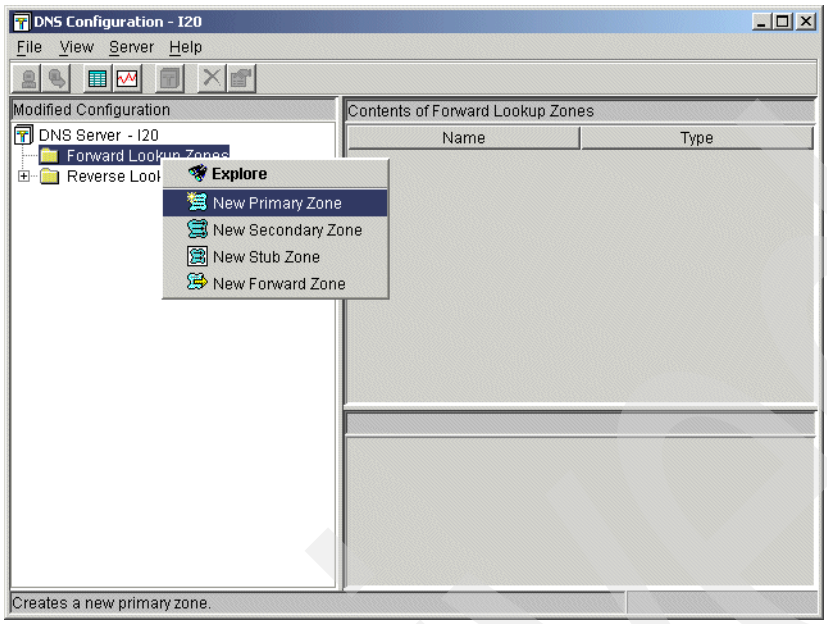


Figure 16-176 DNS Configuration window

13. In the Zone Domain Name window, type `itsoroch.ibm.com.` (answer 6 in Table 16-6 on page 469) in the Fully qualified domain name field, as shown in Figure 16-177.

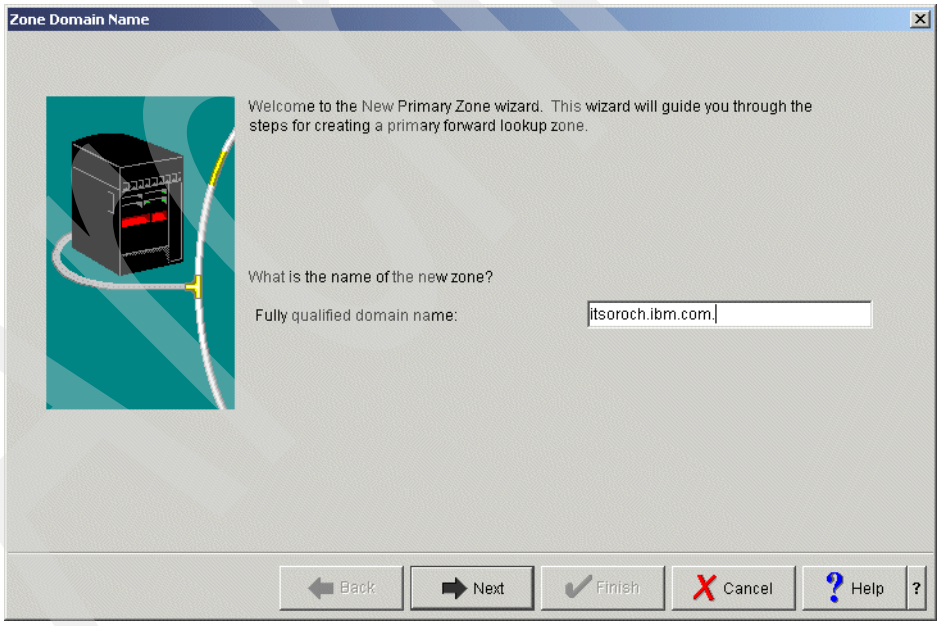


Figure 16-177 Zone Domain name window

14. In the Name Servers window, select **as20.itsoroch.ibm.com.** (answer 6 in Table 16-6 on page 469) and click **Edit**, as shown in Figure 16-178.

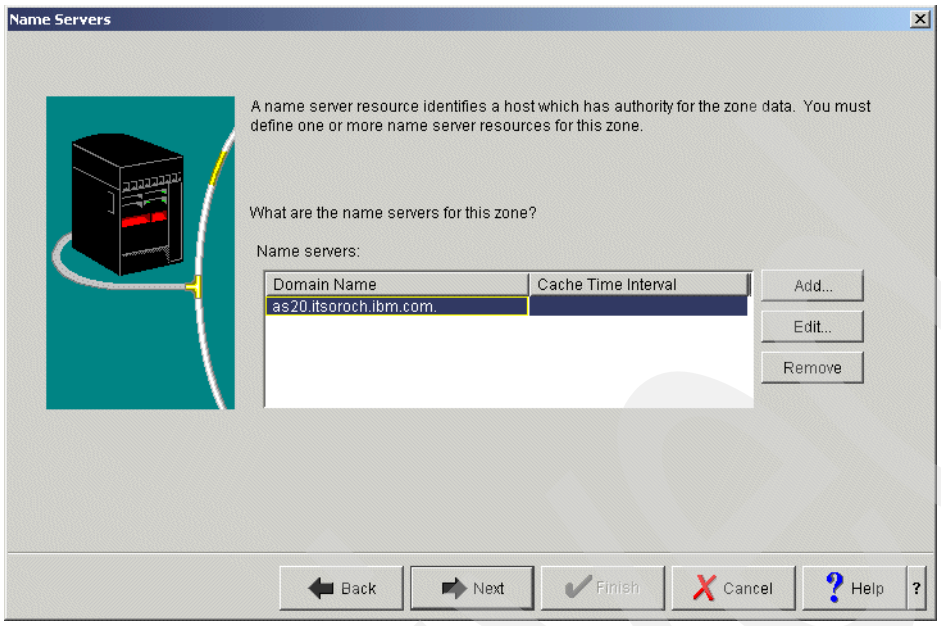


Figure 16-178 Name servers window

15. This opens the Edit Name Server window. Select **Cache time interval**. Type 1 and choose **days**, as shown in Figure 16-179. Click **OK** to continue.

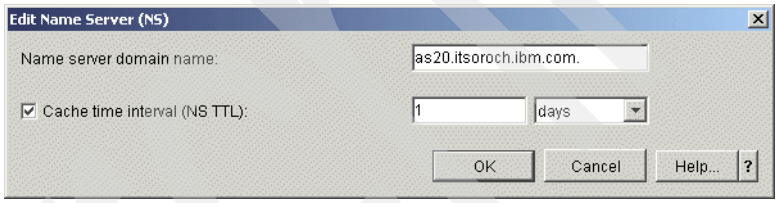


Figure 16-179 Edit Name Server window

16. In the Name Server IP Addresses window (Figure 16-180), click **Add** and type 172.23.10.1 (answer 2 in Table 16-6 on page 469), as shown in Figure 16-180. Click **OK** to continue.

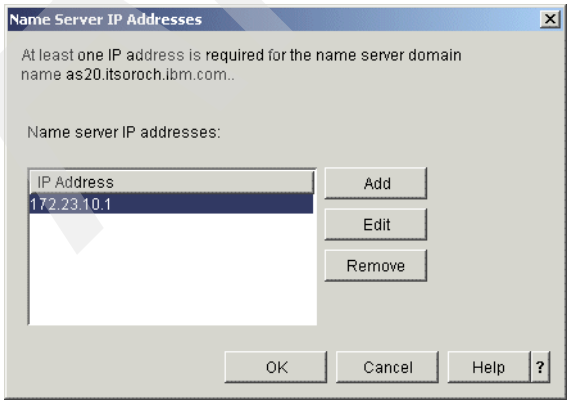


Figure 16-180 Name Server IP Address window

17. In the Static or Dynamic Zone window, choose **Perform static updates**, as shown in Figure 16-181. Click **Next** to continue.

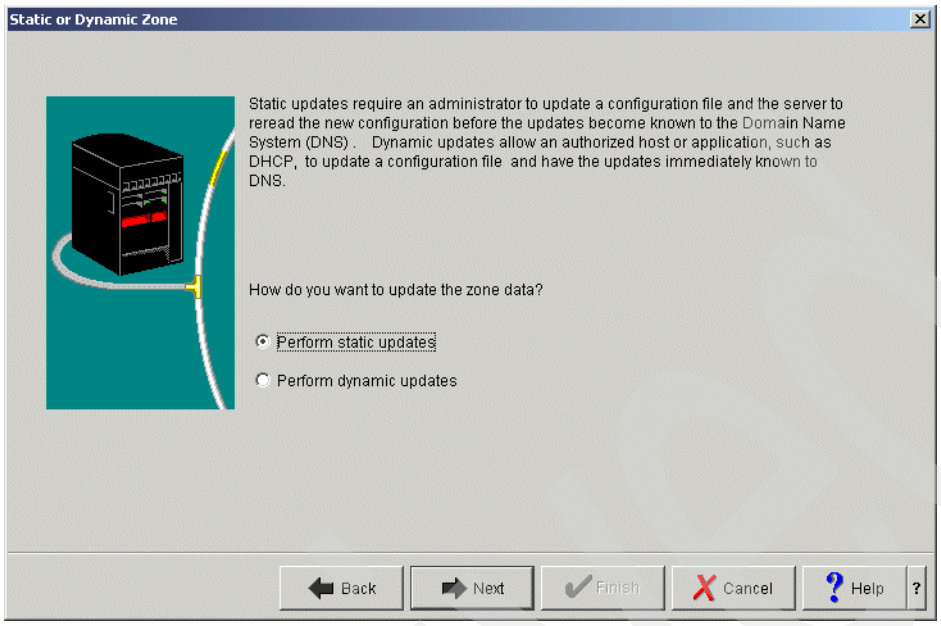


Figure 16-181 Static or Dynamic Zone

18. In the Summary window (Figure 16-182), click **Finish**.

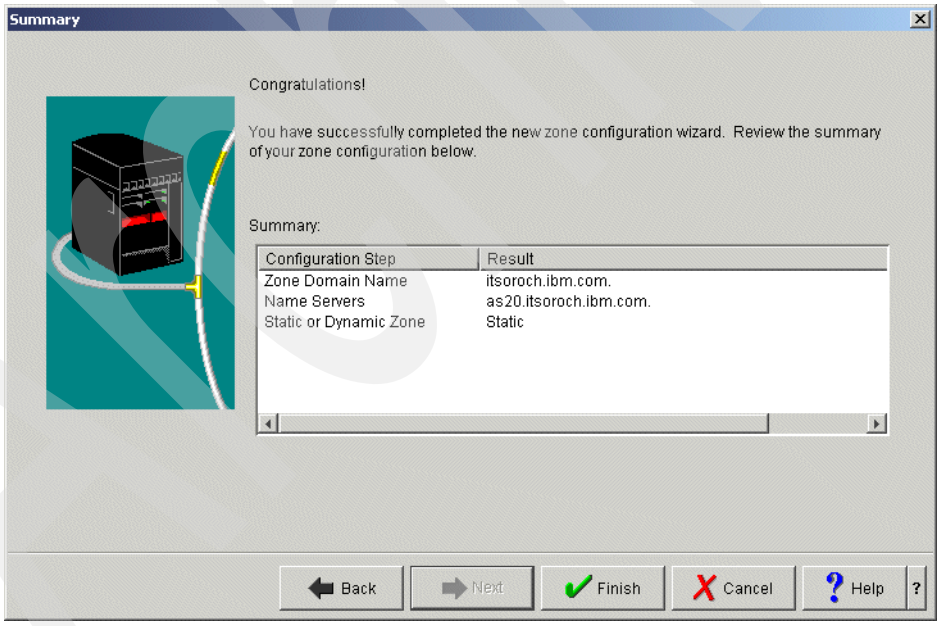


Figure 16-182 Summary window

19. In the DNS Configuration window, right-click **Primary Zone itsoroch.ibm.com and** choose **Properties**, as shown in Figure 16-183.

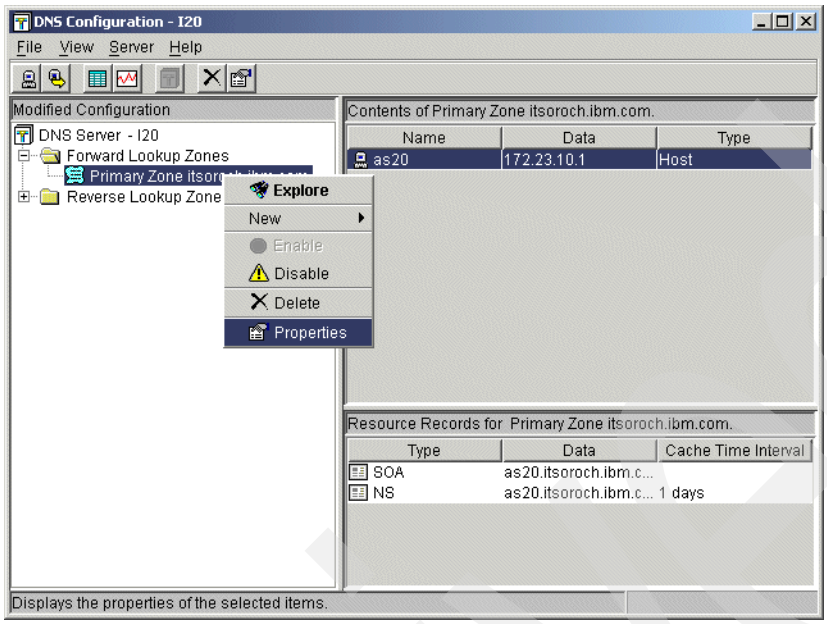


Figure 16-183 DNS Configuration window

20. In the Primary Zone Properties window, click the **Options** tab. Expand **Access Control** and click **allow-query**. Select **IP Prefix** as the Match list element type, as shown in Chapter 16-184, “Primary Zone Properties window” on page 480. Click **Add**.

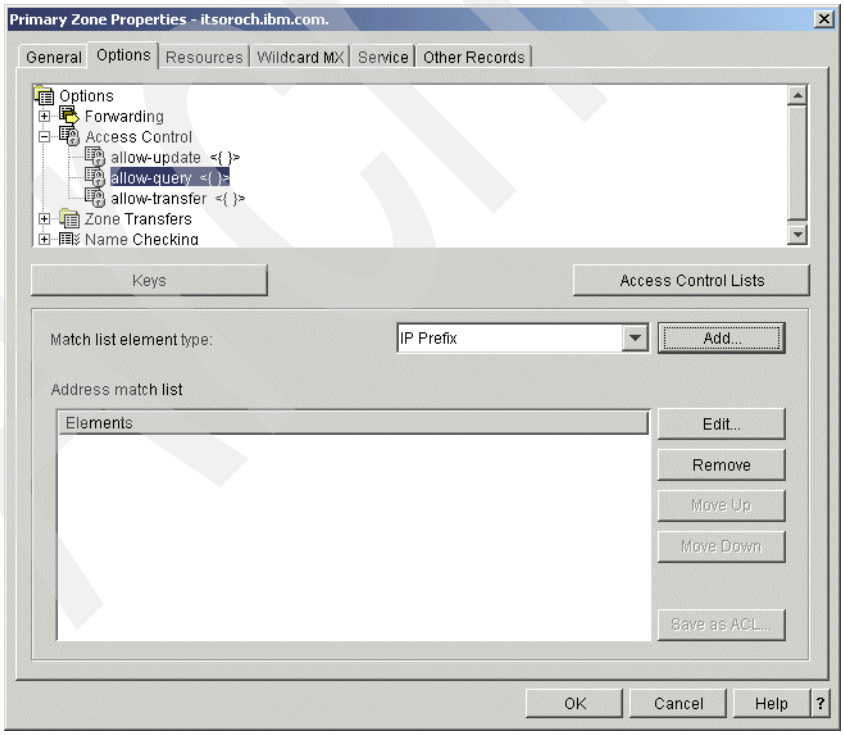


Figure 16-184 Primary Zone Properties window

21. This opens the IP Prefix window. Type `172.23.10.0` (answer 5 in Table 16-6 on page 469) in the IP Network field, as shown in Figure 16-185. Click **OK**.

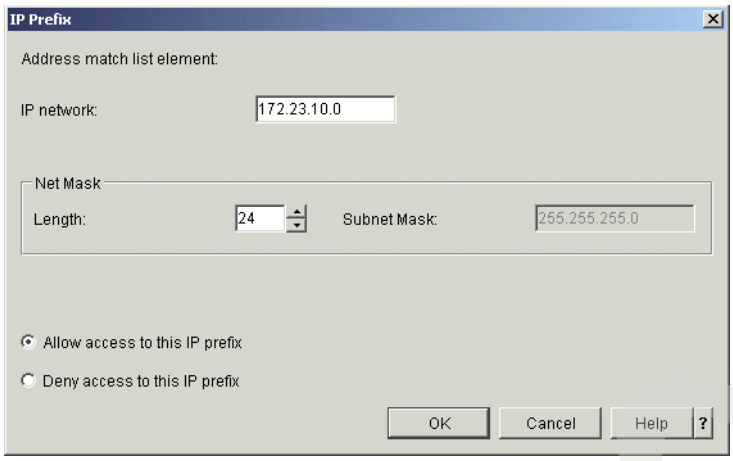


Figure 16-185 IP Prefix window

22. Returning to the Primary Zone Properties window, click the **Resources** tab. Select **SOA** and click **Edit** to open the Add/Edit resource window.

23. Select **Start of Authority cache time interval**. Type **1** and select **days**, as shown in Figure 16-186. Click **OK** to continue.

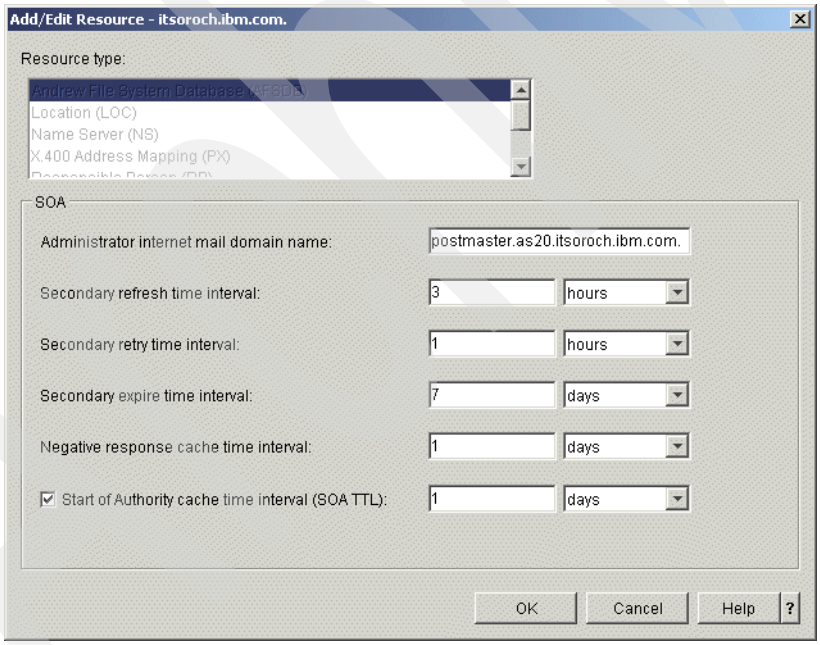


Figure 16-186 Add/Edit Resource window

24. In the DNS Configuration window, right-click **Reverse Lookup Zones** and choose **New Primary Zone**, as shown in Figure 16-187.

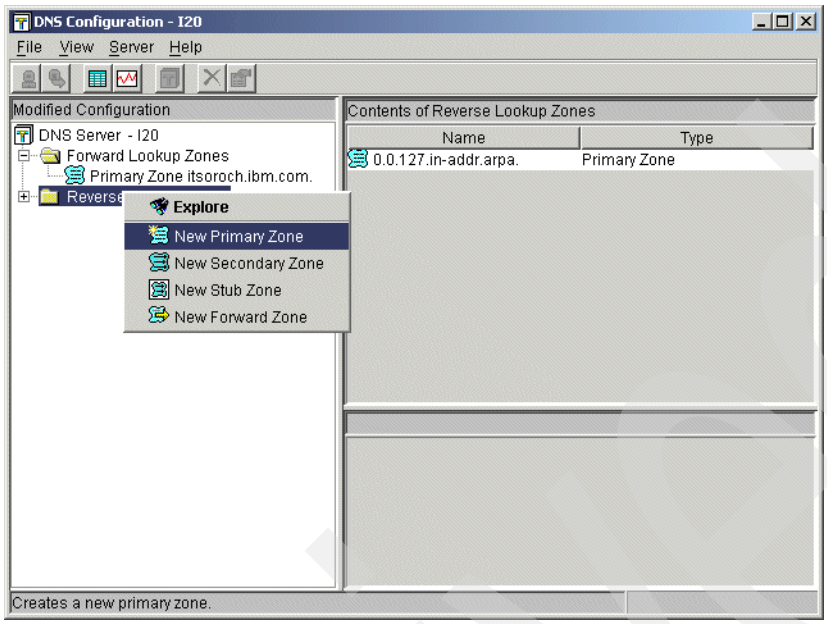


Figure 16-187 DNS Configuration window

25. In the Zone Domain Name window, type `10.23.172.in-addr.arpa.`, as shown in Figure 16-188. Click **Next** to continue.

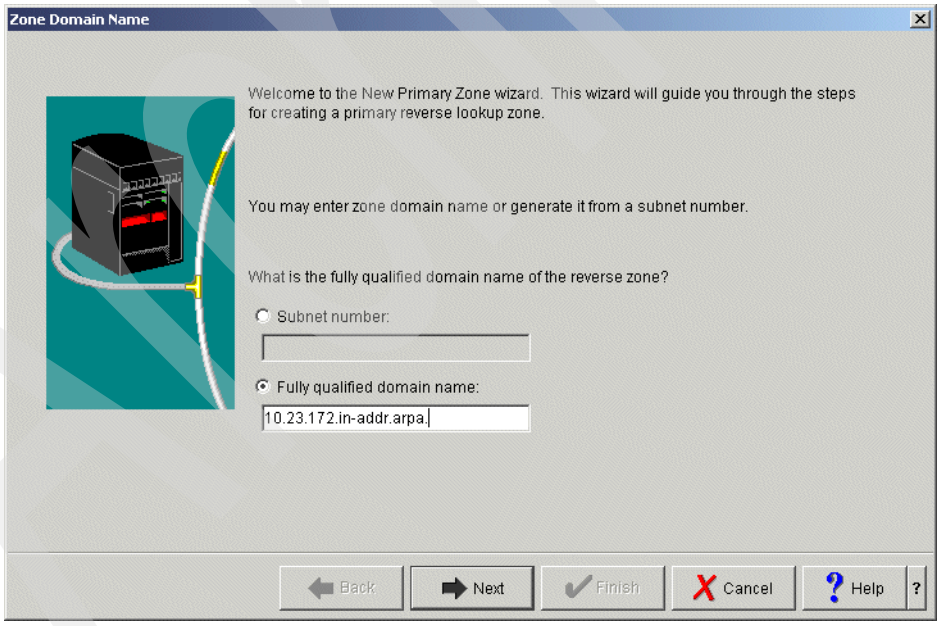


Figure 16-188 Zone Domain Name window

26. In the Name servers window, select **as20.itsoroch.ibm.com.**, as shown in Figure 16-189. Click **Edit**.

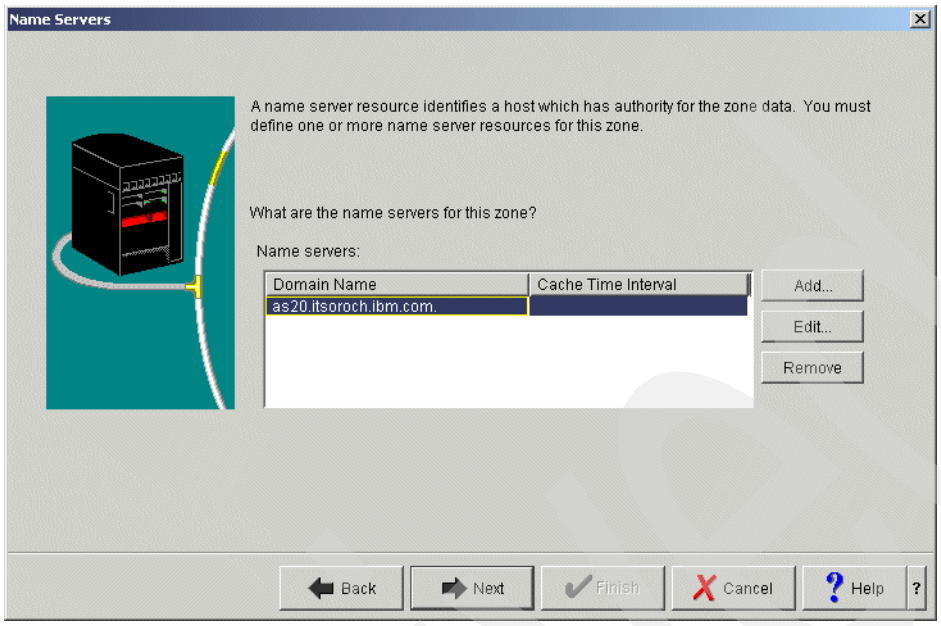


Figure 16-189 Name Servers window

27. In the Edit Name Server window, select **Cache time interval**. Type 1 and choose **days**, as shown in Figure 16-190. Click **OK** to continue.

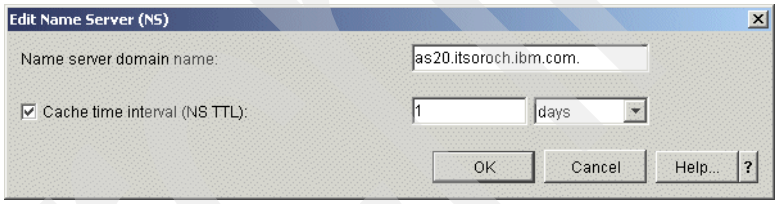


Figure 16-190 Edit Name Server window

28. In the Static or Dynamic Zone window, select **Perform static updates**, as shown in Figure 16-191. Click **Next** to continue.

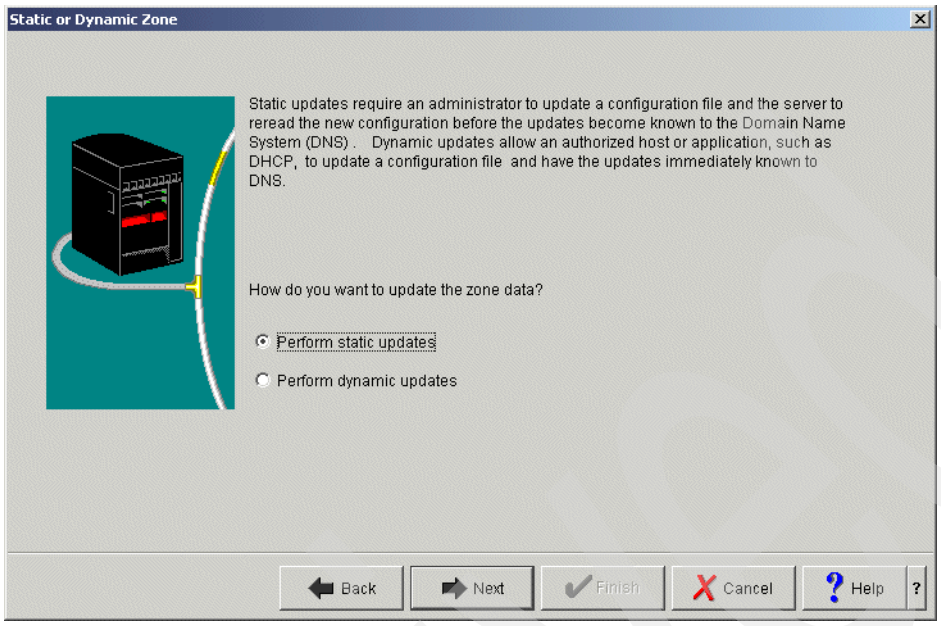


Figure 16-191 Static or Dynamic Zone window

29. In the Summary window (Figure 16-192), click **Finish**.

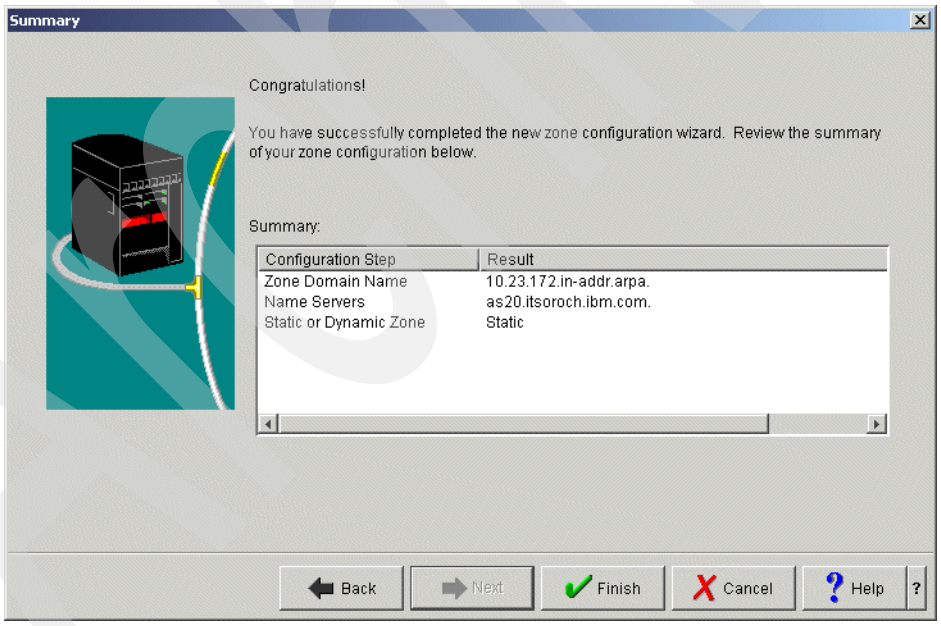


Figure 16-192 Summary window

30. In the DNS Configuration window, right-click **Primary Zone 10.23.172.in-addr.arpa** and choose **Properties** (Figure 16-193).

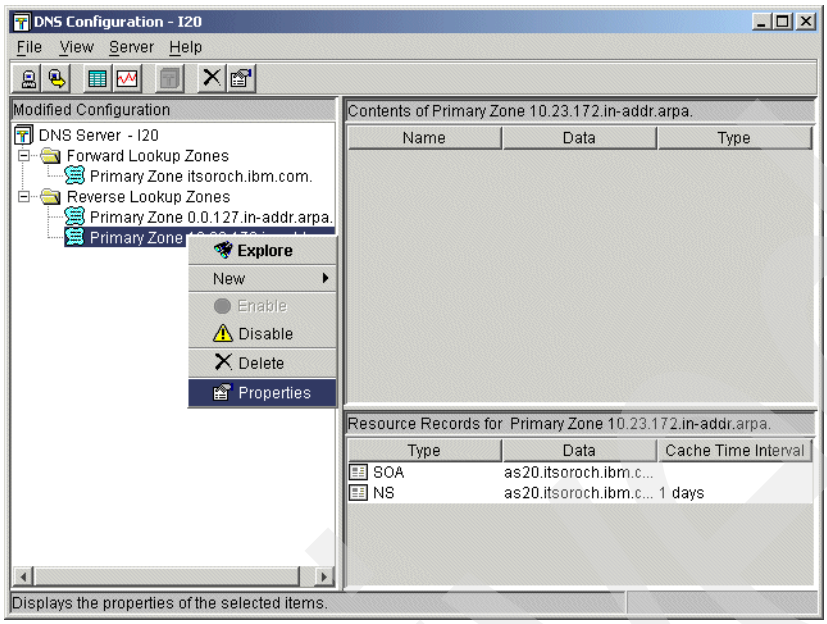


Figure 16-193 DNS Configuration window

31. In the Primary Zone Properties window, select **allow-query**. Choose **IP Prefix** as the Match list element type (Figure 16-194). Click **Add** to open the IP Prefix window.

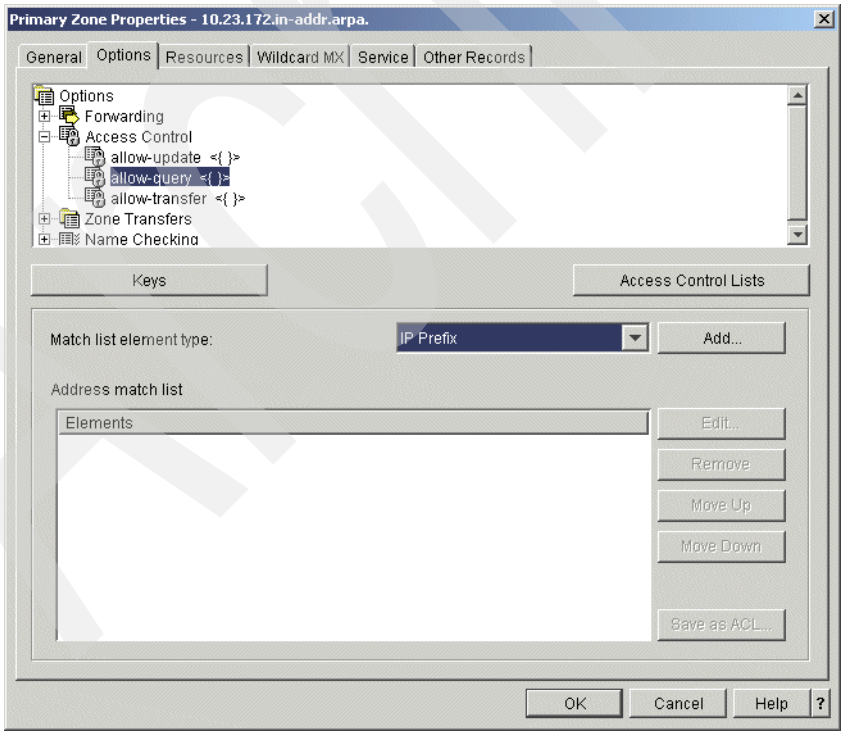


Figure 16-194 DNS Configuration window

32. Type `172.23.10.0` (answer 5 in Table 16-6 on page 469) in the IP network field, as shown in Figure 16-195. Click **OK** to return to the Primary Zone Properties window.

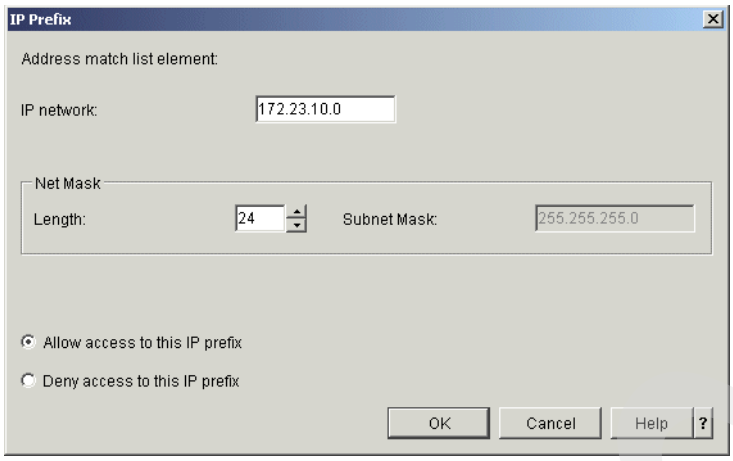


Figure 16-195 IP Prefix window

33. Click the **Resources** tab. Select **SOA** and click **Edit** to open the Add/Edit resource window.

34. Check **Start of Authority cache time interval**. Type 1 and select **days**, as shown in Figure 16-196. Click **OK** to continue.

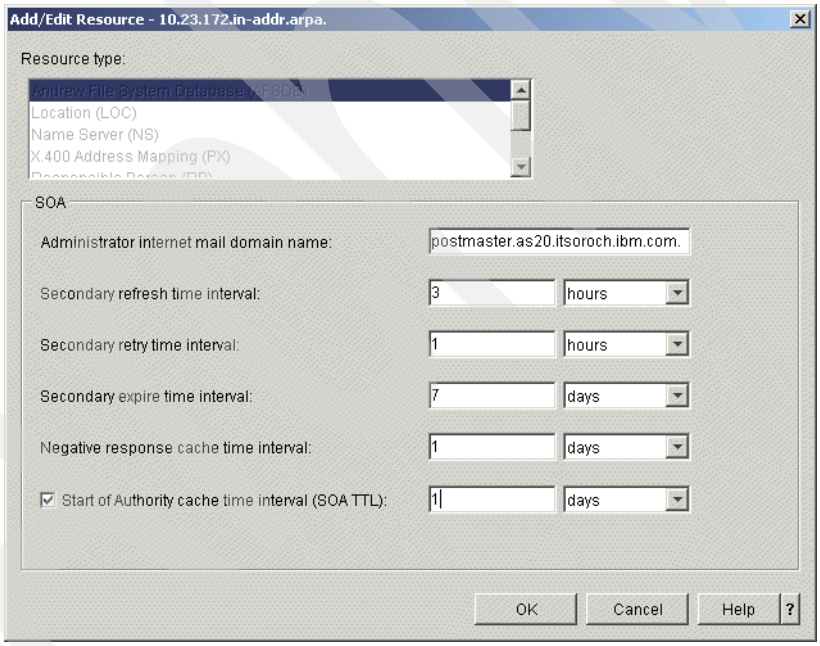


Figure 16-196 Add/Edit Resource window

35. In the DNS Configuration window, right-click **Primary Zone 10.23.172.in-addr.arpa.** and choose **New** → **Host**, as shown in Figure 16-197.

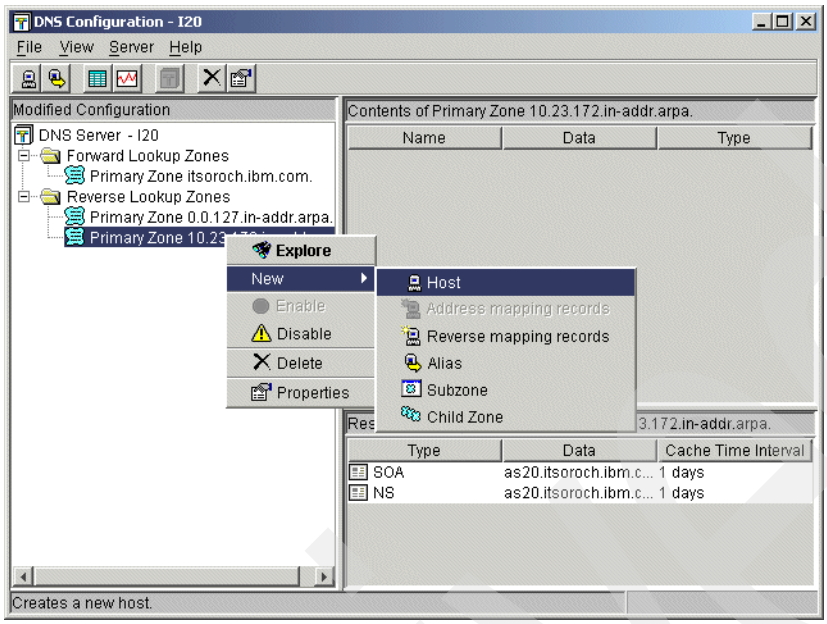


Figure 16-197 DNS Configuration window

36. In the New Host window, select **Domain Name** and type `1.10.23.172.in-addr.arpa.`, as shown in Figure 16-198. Click **Next**.

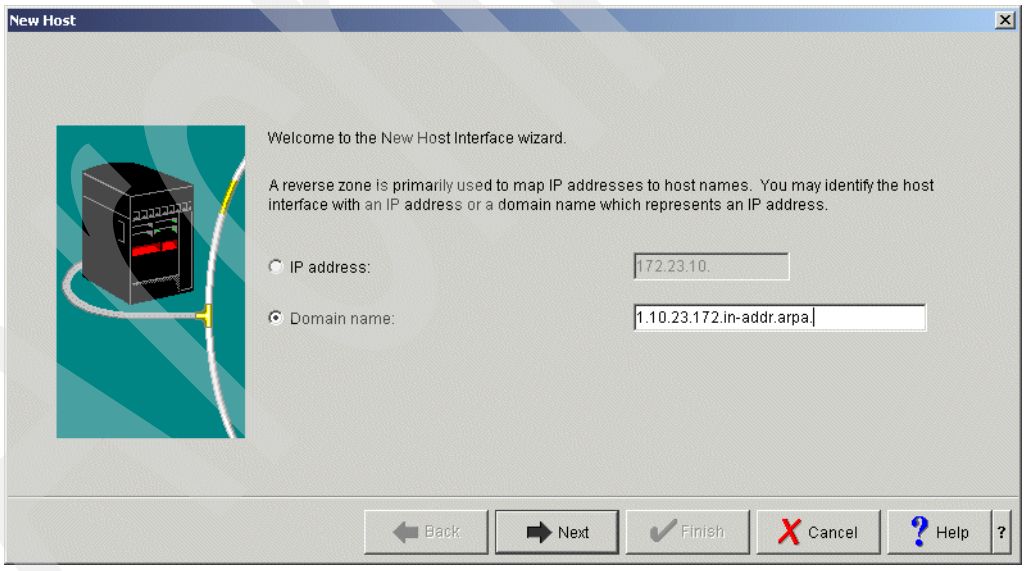


Figure 16-198 New Host window

37. In the New Host Resources window click **Add**, as shown in Figure 16-199, to open the Add/Edit Resource window.

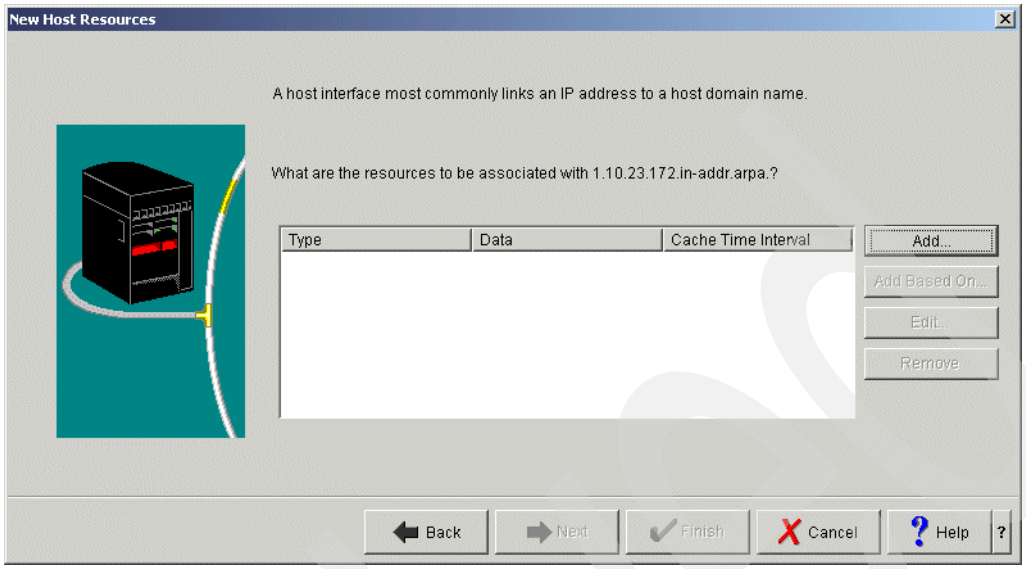


Figure 16-199 New Host Resources window

38. Type `as20.itsoroch.ibm.com.` (answer 6 in Table 16-6 on page 469) in the Fully qualified host domain name field. Check **Cache time interval**. Type 1 and choose **days**, as shown in Figure 16-200. Click **OK** to continue.

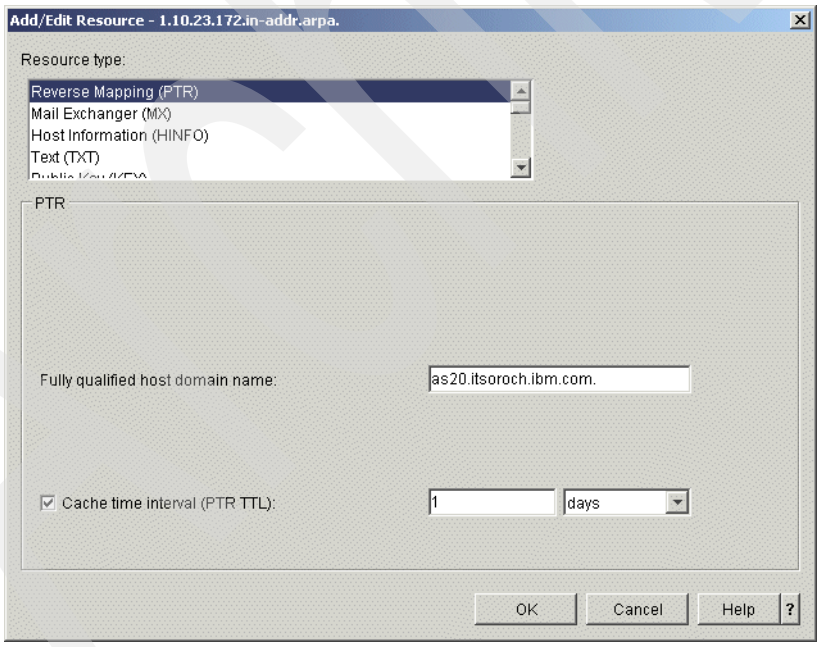


Figure 16-200 Add/Edit Resource window

39. In the New Host Resources window (Figure 16-201), click **Finish**.

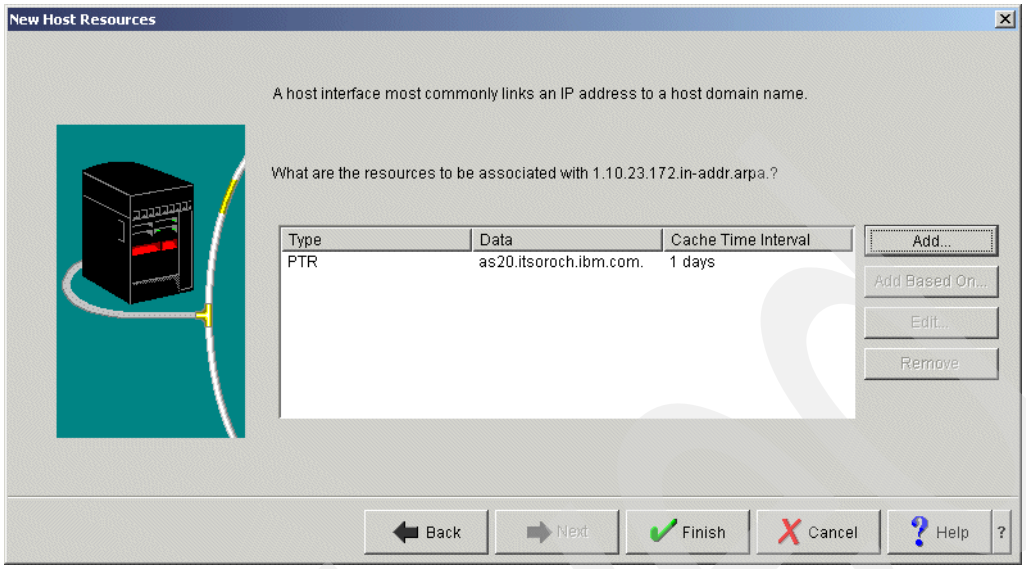


Figure 16-201 New Host Resources window

40. In the DNS Configuration window, choose **File** → **Save Configuration** to save the configuration, as shown in Figure 16-202.

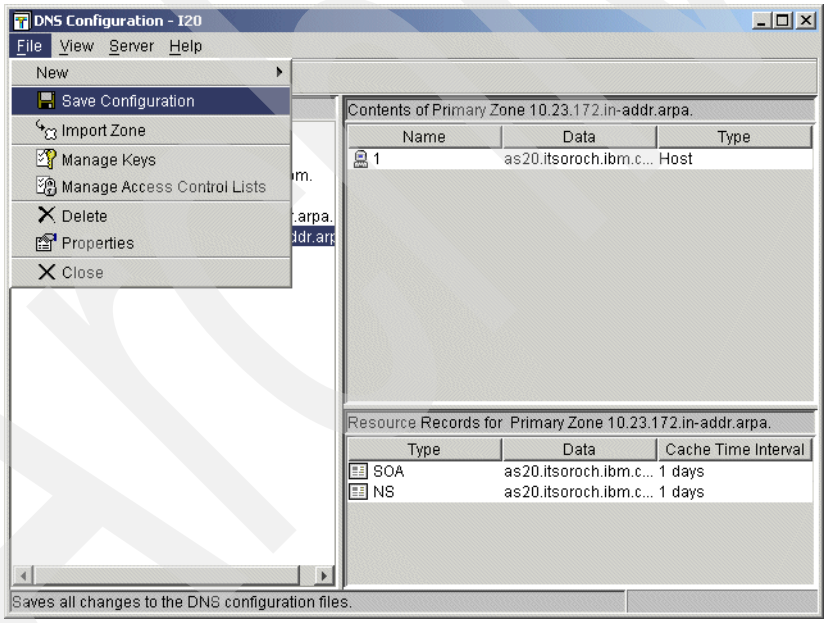


Figure 16-202 DNS Configuration window

Step 2: Create the public DNS E20 on AS20

In this step, we create the public DNS E20 on AS20.

1. In the iSeries Navigator window, expand **AS20** → **Network** → **Servers**. Right-click **DNS** and choose **New Name Server**, as shown in Figure 16-203.

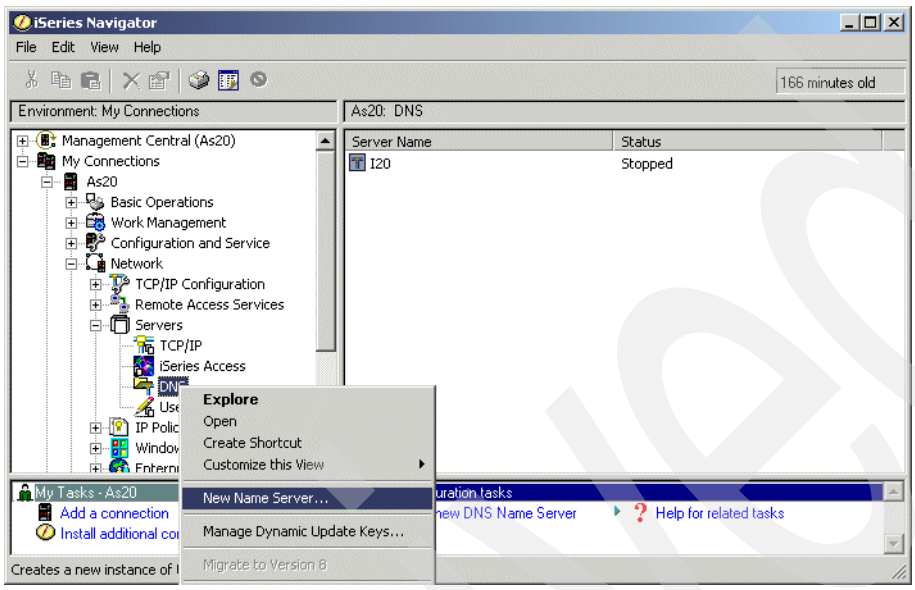


Figure 16-203 iSeries Navigator window

2. In the New DNS Configuration window click **Next**, as shown in Figure 16-204.

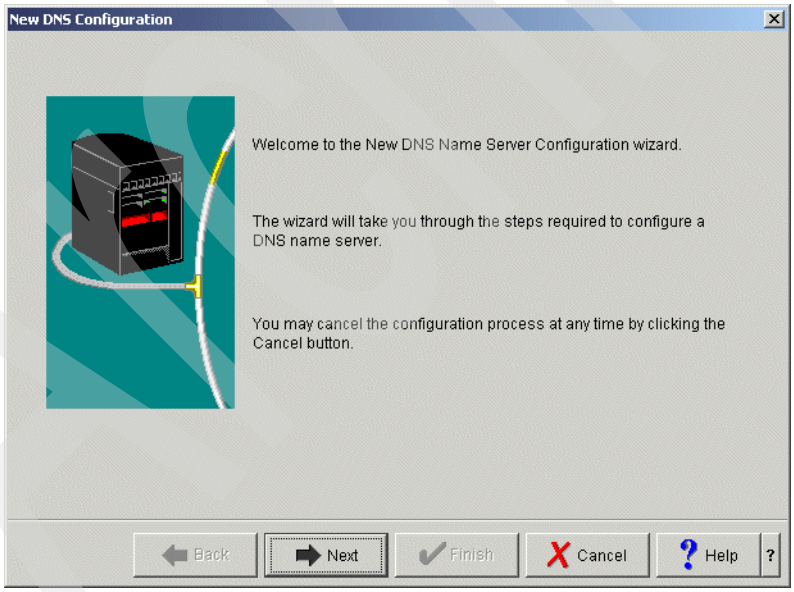


Figure 16-204 New DNS Configuration window

3. In the DNS Server Name window, type E20 (answer 1 in Table 16-6 on page 469) in the Name field, as shown in Figure 16-205. Click **Next** to continue.

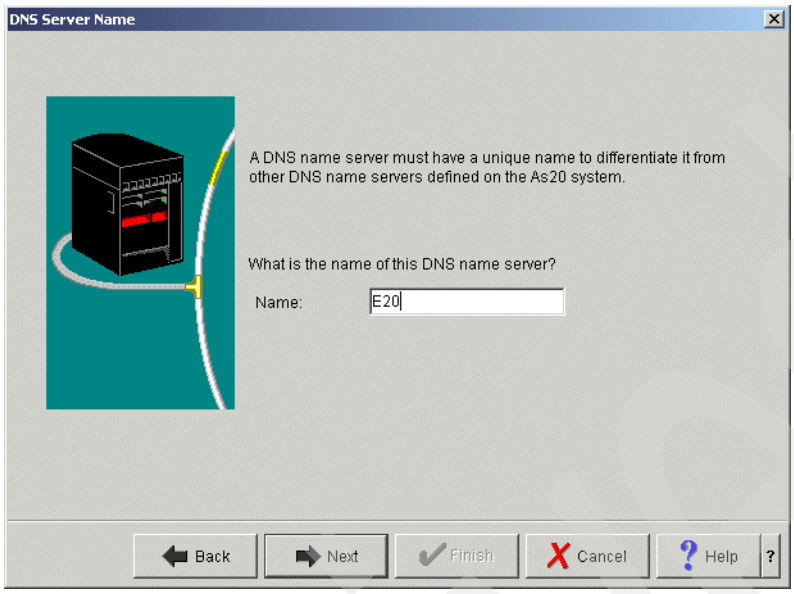


Figure 16-205 DNS Server Name window

4. In the Listen On IP address window, choose **192.168.0.1** (answer 3 in Table 16-6 on page 469), as shown in Figure 16-206. Click **Next** to continue.

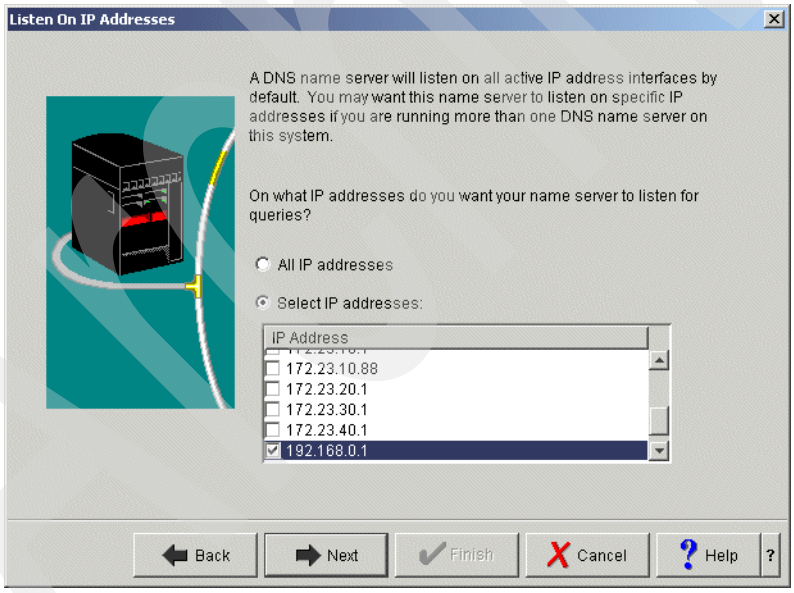


Figure 16-206 Listen On IP Addresses window

5. In the Root Servers window (Figure 16-207), click **Next** to continue.

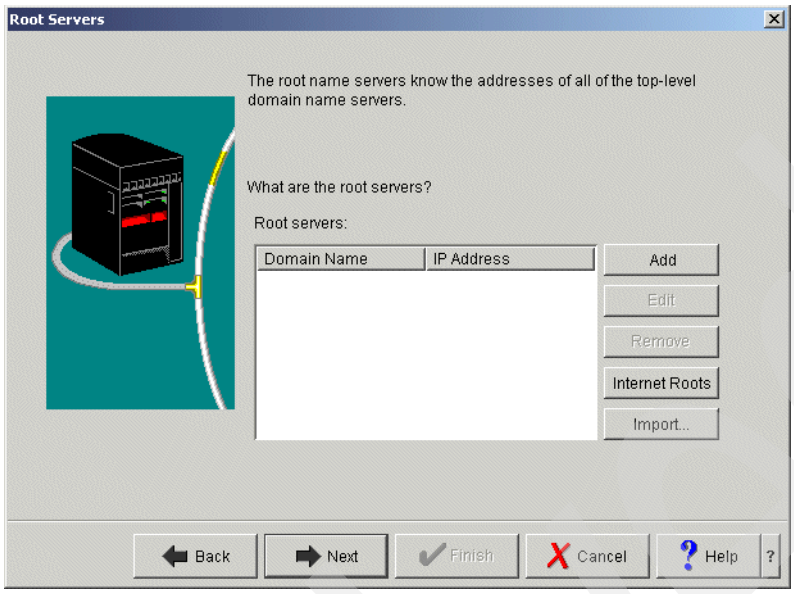


Figure 16-207 Root Servers window

6. In the Start DNS Name Server window select **Yes** (Figure 16-208). Click **Next**.

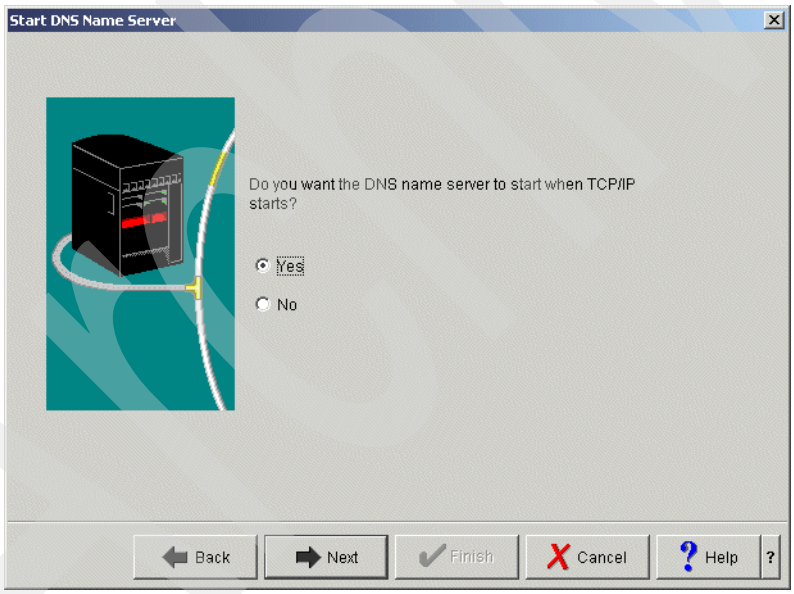


Figure 16-208 Start DNS Name Server window

7. In the Summary window (Figure 16-209), click **Finish**.

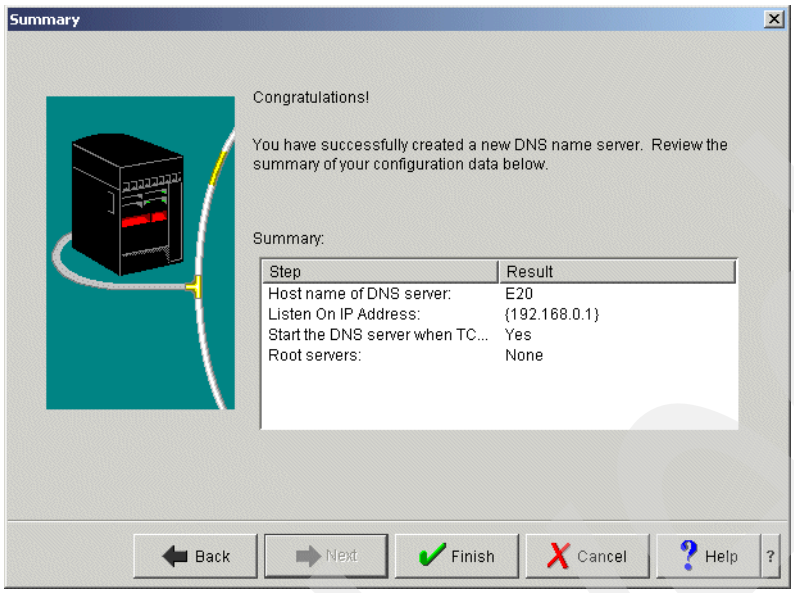


Figure 16-209 Summary window

8. Returning to the iSeries Navigator window (Figure 16-210), right-click **E20** and choose **Configuration**.

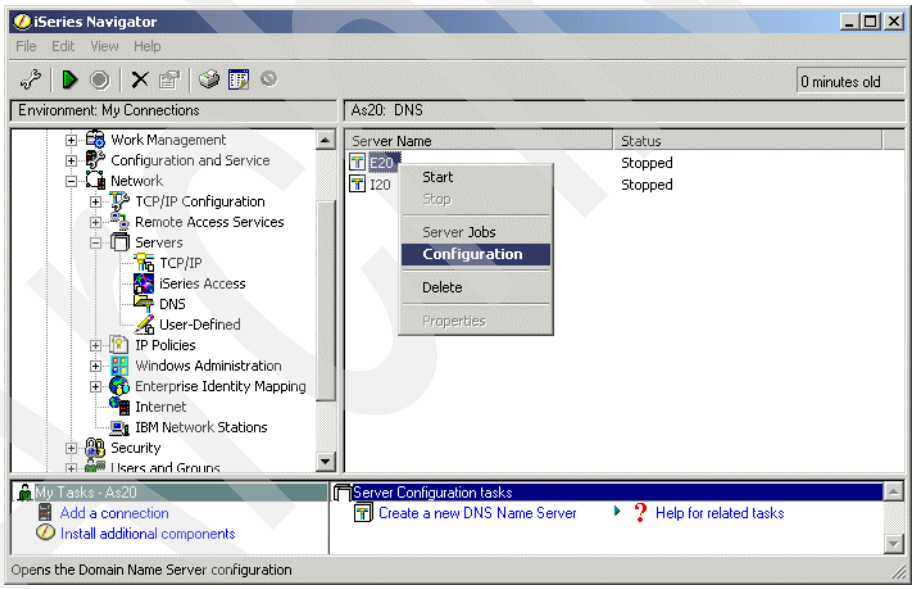


Figure 16-210 iSeries Navigator window

9. In the DNS Configuration - E20 window, right-click **DNS Server E20** and choose **Properties**, as shown in Figure 16-211.

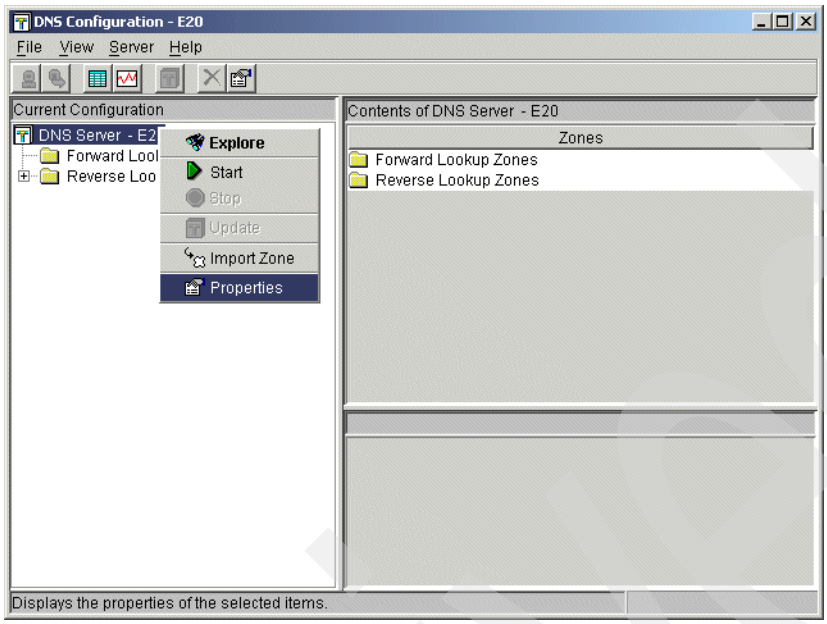


Figure 16-211 DNS Configuration - E20 window

10. In the Options tab of the Server Properties - E20 window, expand **Forwarding** and choose **forward**. Select **Only** as the Server query method (Figure 16-212).

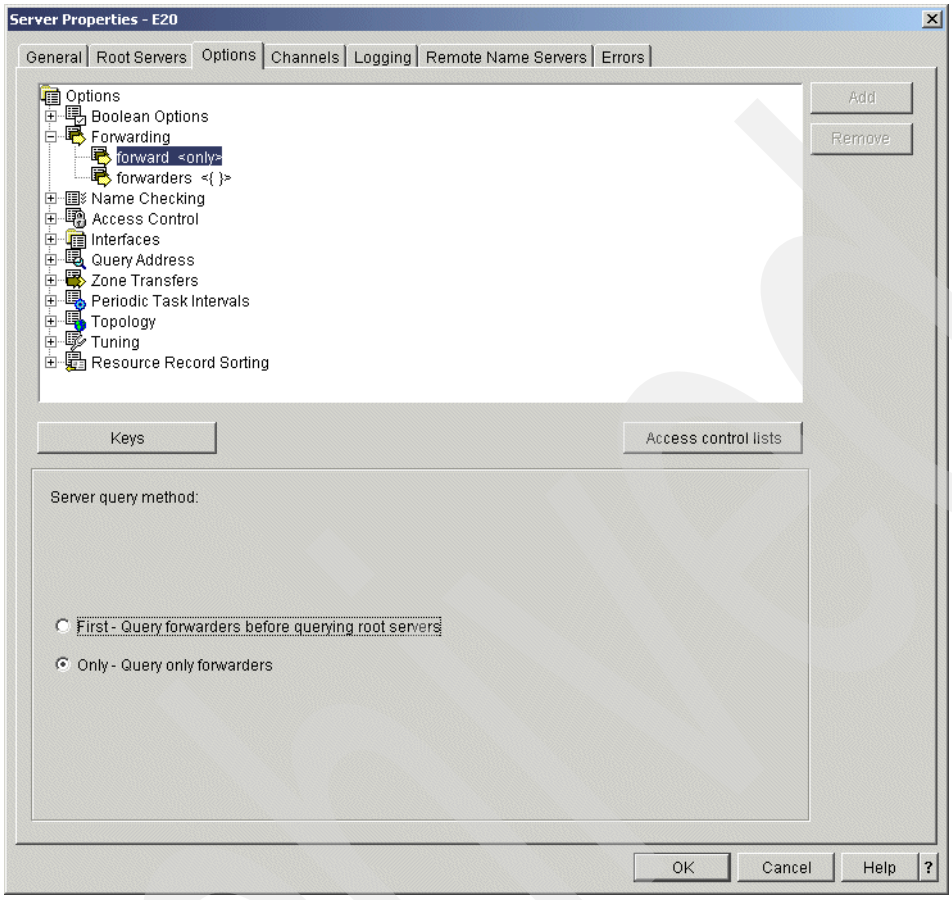


Figure 16-212 Server Properties - E20 window

11. In the same window, select **forwarders**. In the lower pane, click **Add** and type `192.168.0.5` (answer 4 in Table 16-6 on page 469), as shown in Figure 16-213. Click **OK** to continue.

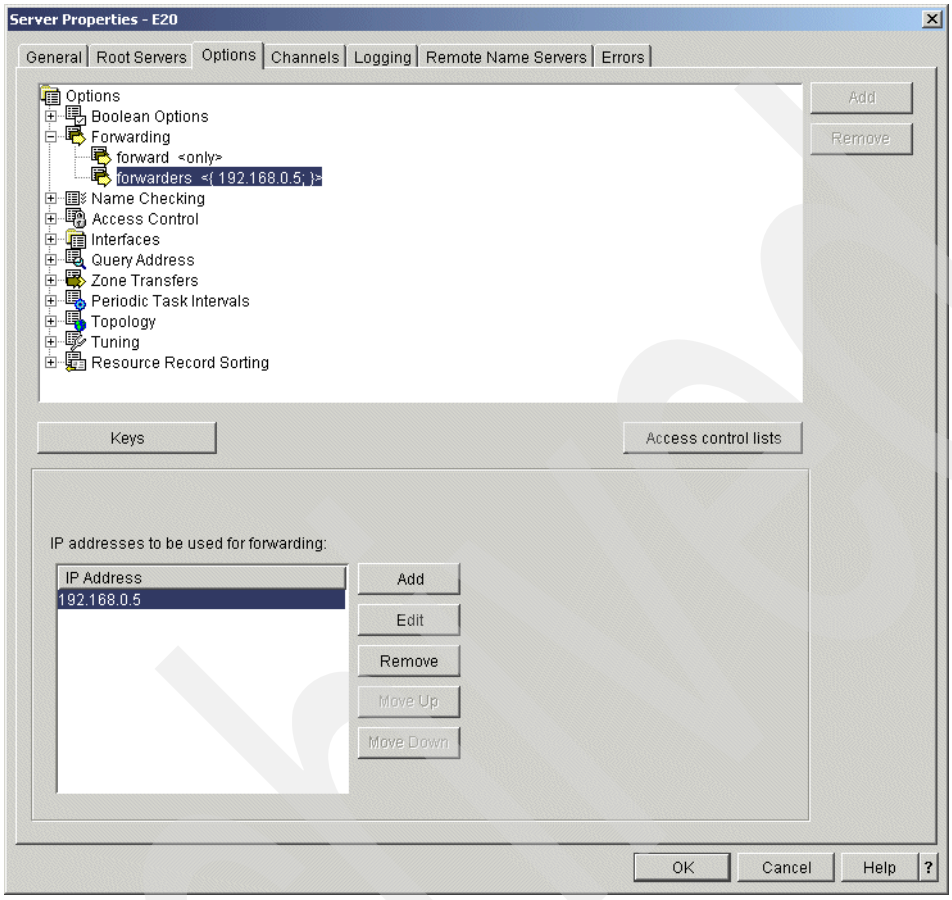


Figure 16-213 Server Properties - E20 window

a. In the DNS Configuration window, right-click **Forward Lookup Zones** and choose **New Primary Zone**, as shown in Figure 16-214.

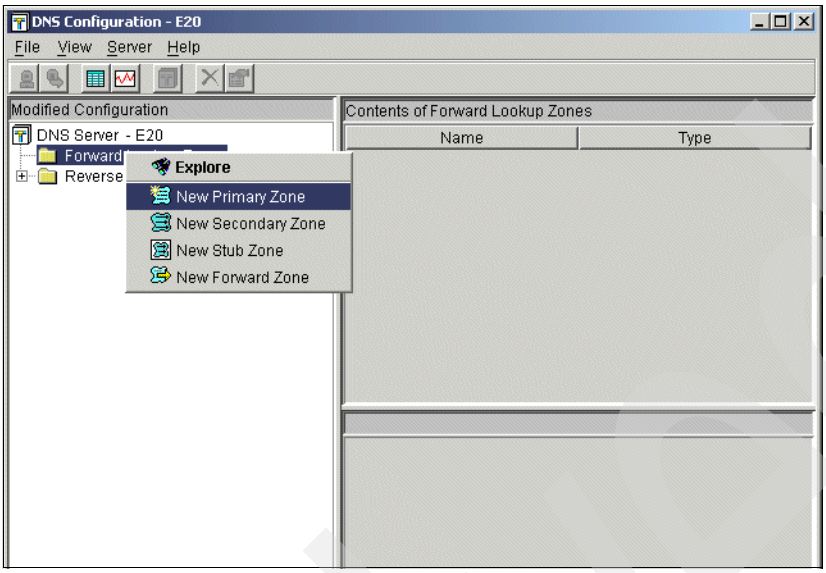


Figure 16-214 DNS Configuration - E20 window

12. In the Zone Domain Name window, type `itsoroch.ibm.com.` (answer 6 in Table 16-6 on page 469), as shown in Figure 16-215. Click **OK** to continue.

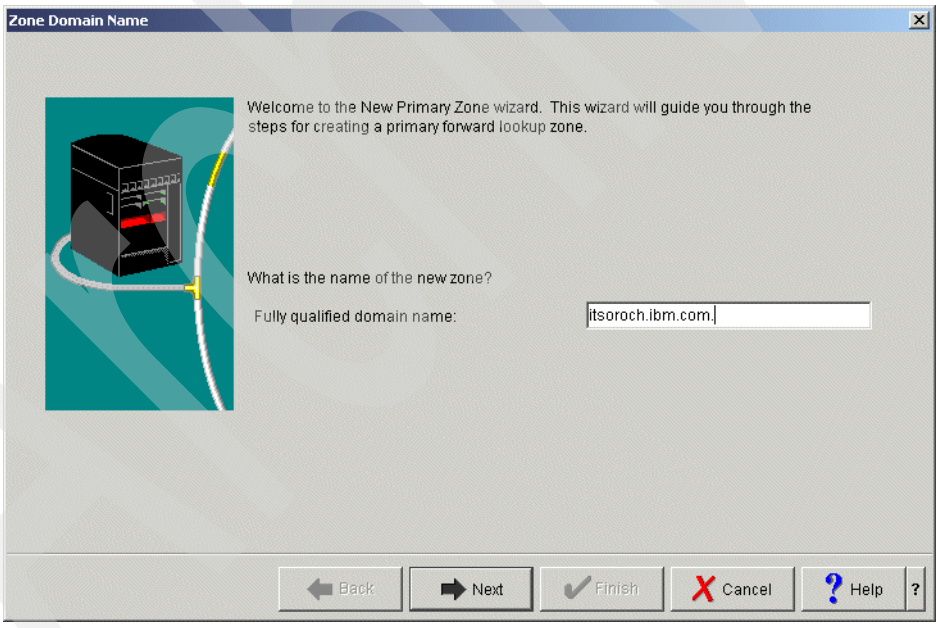


Figure 16-215 Zone Domain Name window

13. In the Name Servers window, select **as20.itsoroch.ibm.com**. Click **Edit** (Figure 16-216).

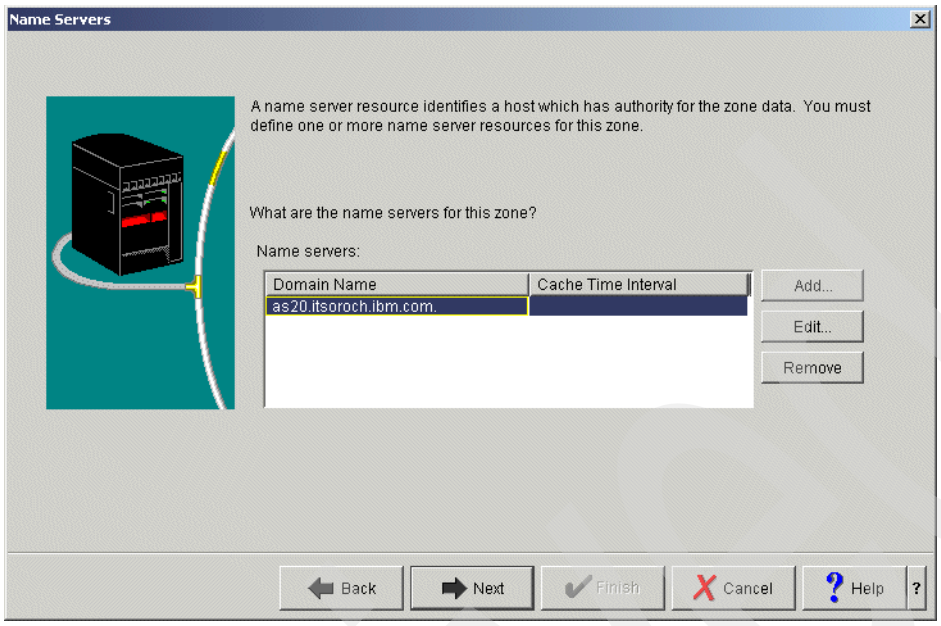


Figure 16-216 Name Servers window

14. This opens the Edit Name Server window. Select Cache time interval. Type 1 and choose **days**, as shown in Figure 16-217.

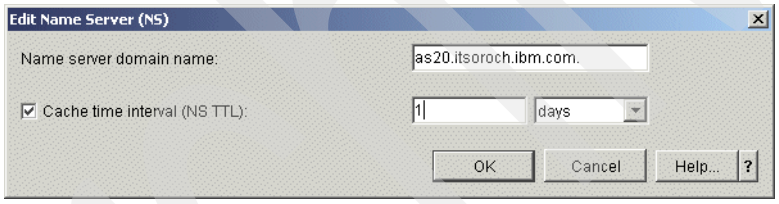


Figure 16-217 Edit Name server window

15. In the Name Server IP Addresses window, click **Add**. Type 192.168.0.1 (answer 3 in Table 16-6 on page 469) and click **OK**, as shown in Figure 16-218.

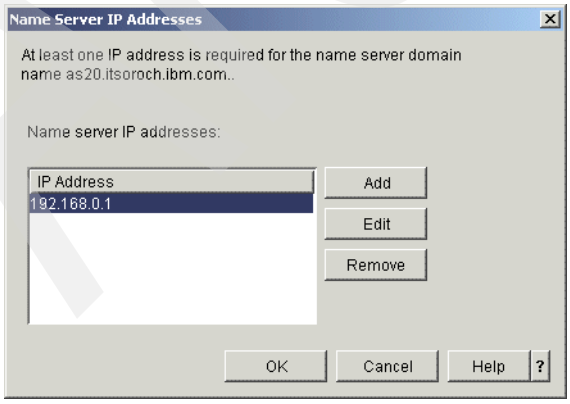


Figure 16-218 Name Server IP Addresses window

16. In the Static or Dynamic Zone window, select **Perform static updates** (Figure 16-219).

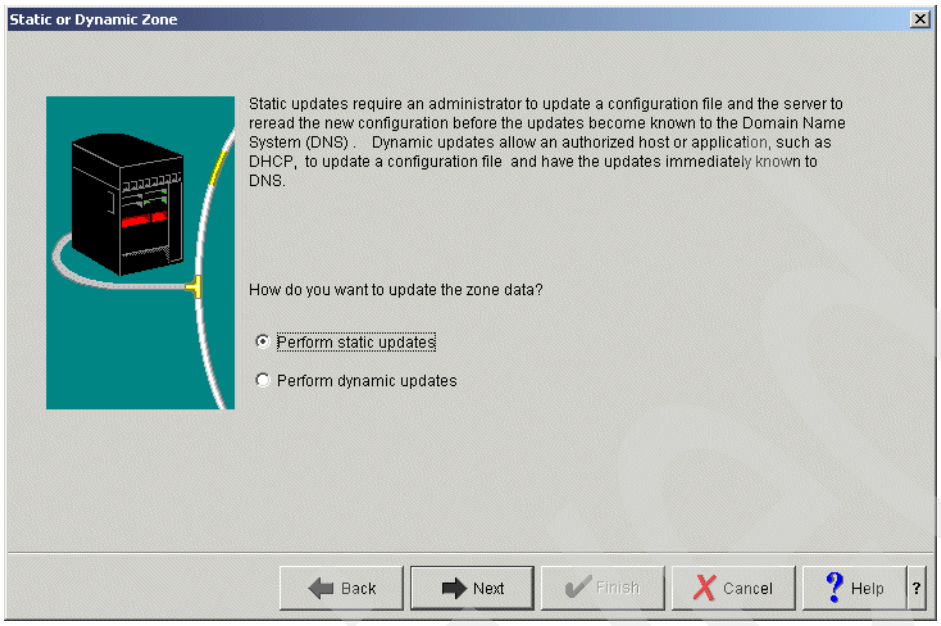


Figure 16-219 Static or Dynamic Zone window

17. In the Summary window (Figure 16-220), click **Finish**.

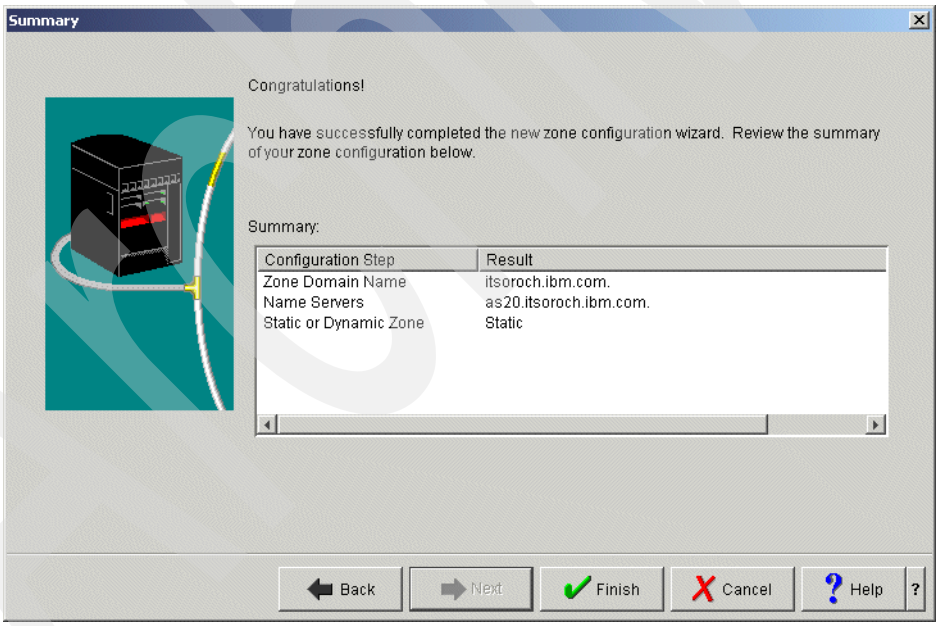


Figure 16-220 Summary window

18. Returning to the DNS Configuration - E20 window, right-click host name **AS20** and choose **Properties**, as shown in Figure 16-221.

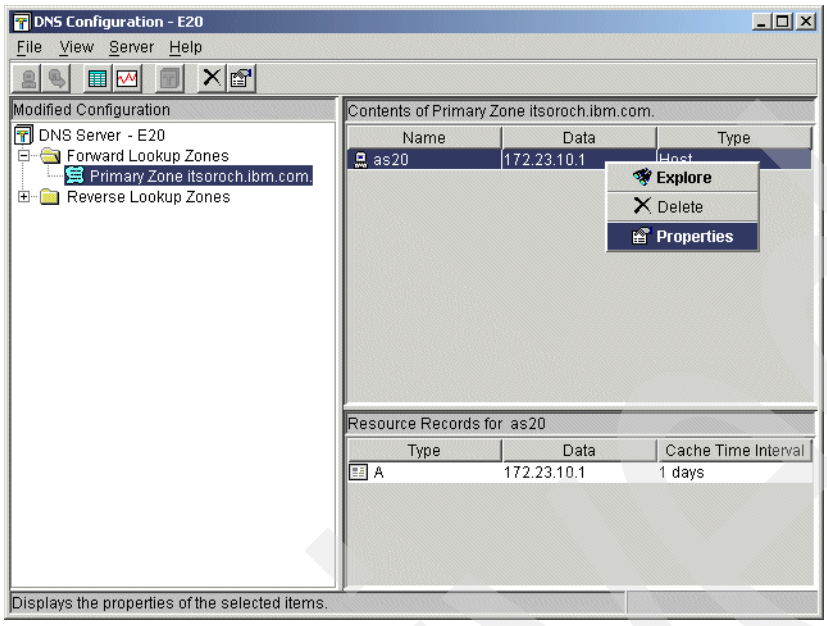


Figure 16-221 DNS Configuration window

19. The Host Properties window opens (Figure 16-222). Choose Type **A** and click **Edit**.

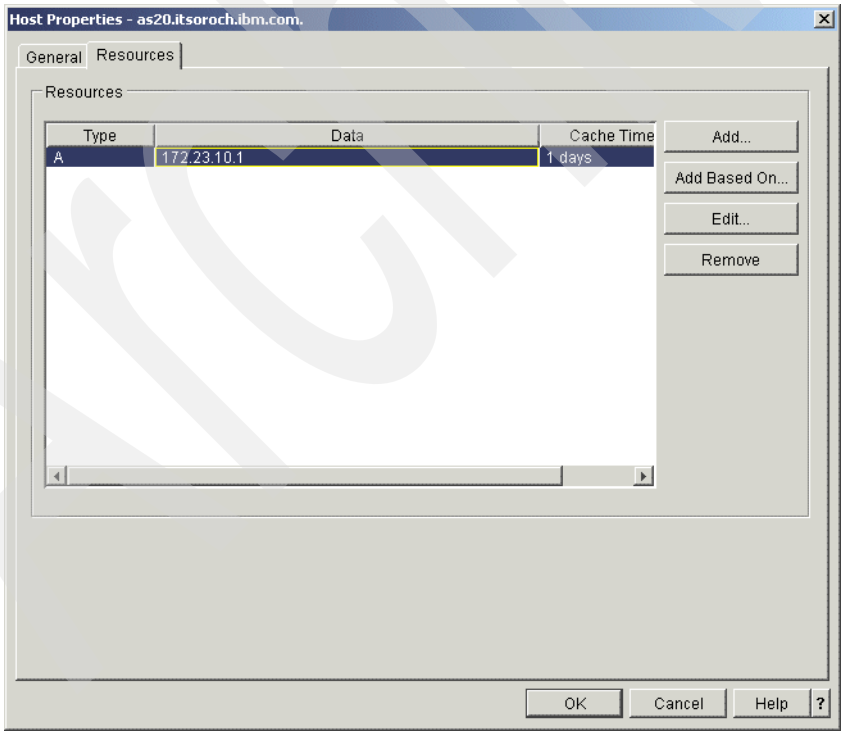


Figure 16-222 Host Properties window

20. In the Add/Edit Resource window, click **Cache time interval**. Type 1 and choose **days**, as shown in Figure 16-223. Click **OK** to continue.

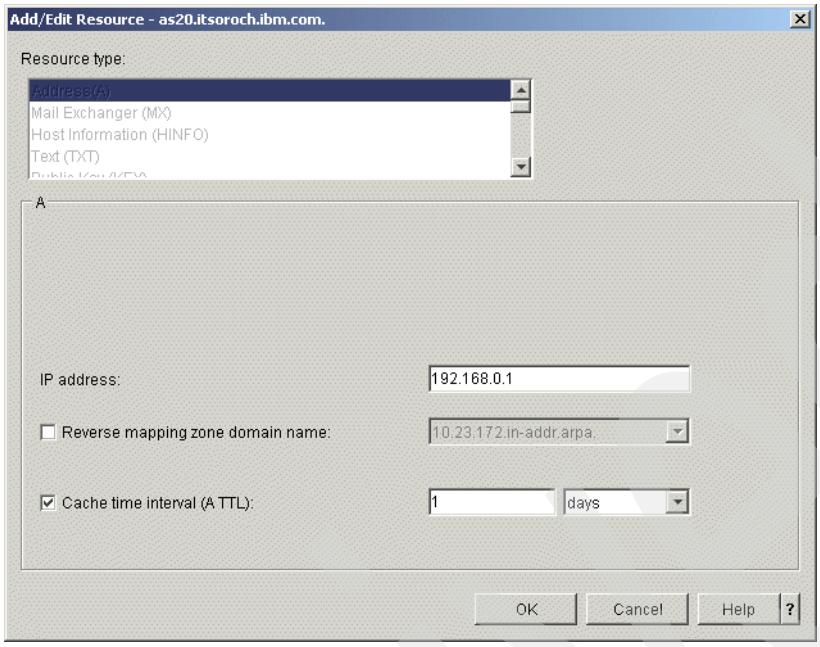


Figure 16-223 Add/Edit Resource window

21. In the DNS configuration - E20 window, right-click **Primary Zone itsoroch.ibm.com.** and choose **Properties**, as shown in Figure 16-224.

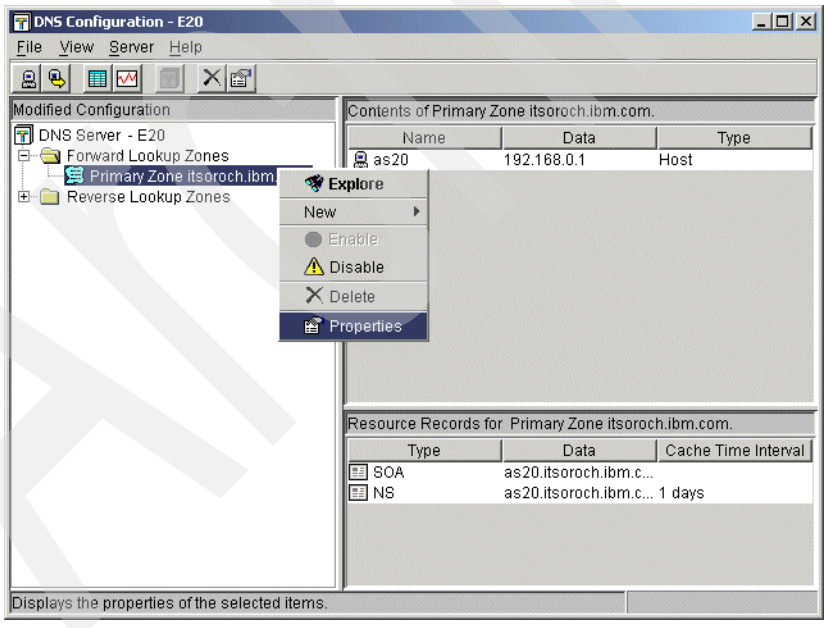


Figure 16-224 DNS configuration window

22. This opens the Primary Zone Properties window. Select **allow-query**. Choose **Access Control list** as the Match list element type, as shown in Figure 16-225. Click **Add**.

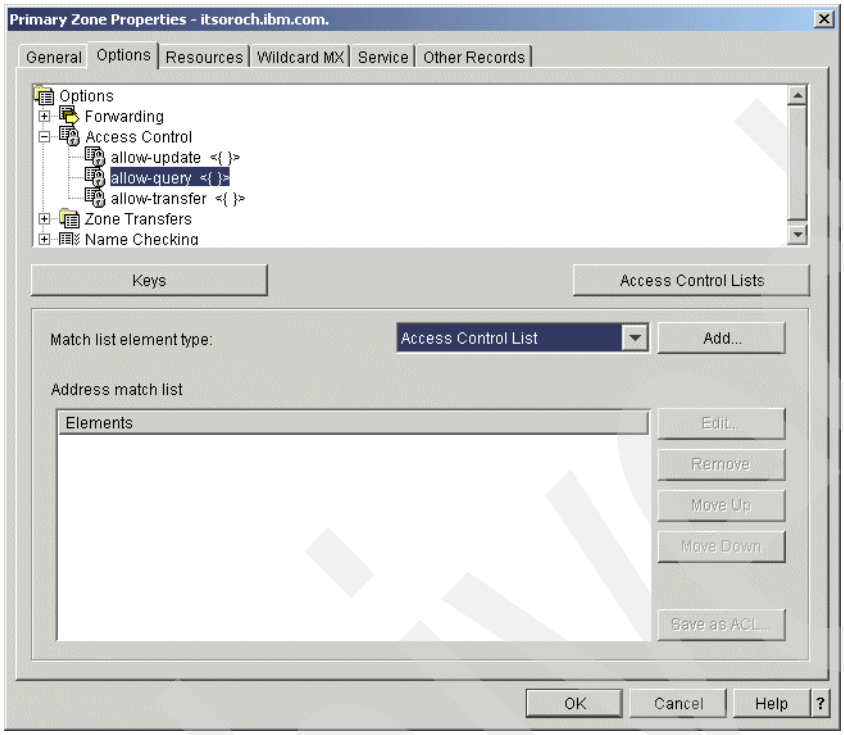


Figure 16-225 Primary Zone Properties window

23. In the Access Control list window, select **any**, as shown in Figure 16-226. Click **OK** to continue.

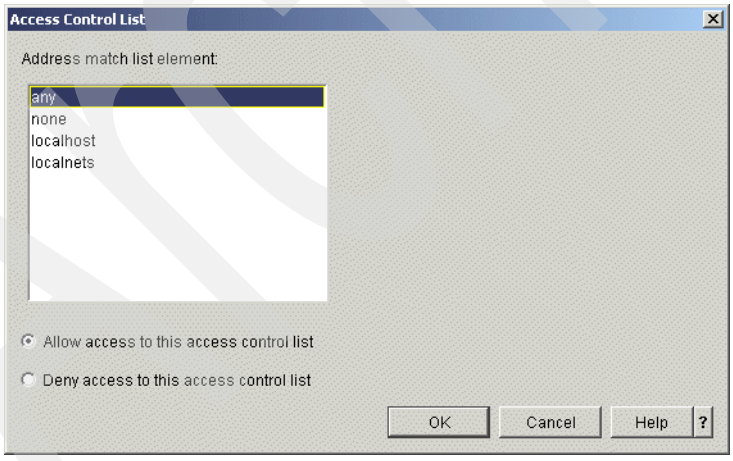


Figure 16-226 Access Control list window

24. In the Primary Zone Properties window, click the **Resources** tab. Select Type **SOA**, as shown in Figure 16-227. Click **Add**.

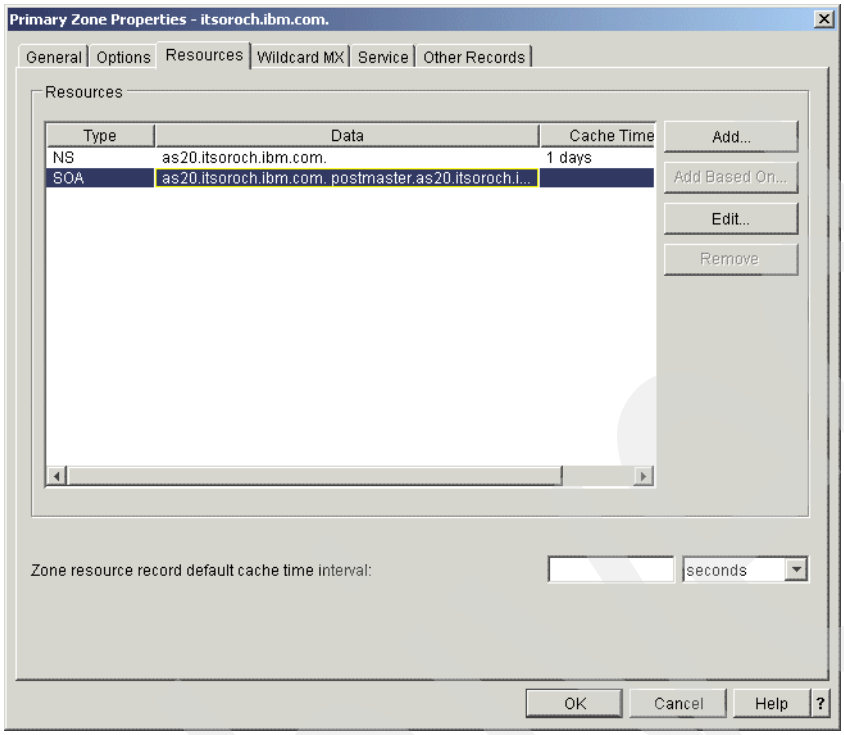


Figure 16-227 Primary Zone Properties window

25. In the Add/Edit Resource window, select **Start of Authority cache time interval (SOA TTL)**. Type 1 and choose **days** as shown in Figure 16-228. Click **OK**.

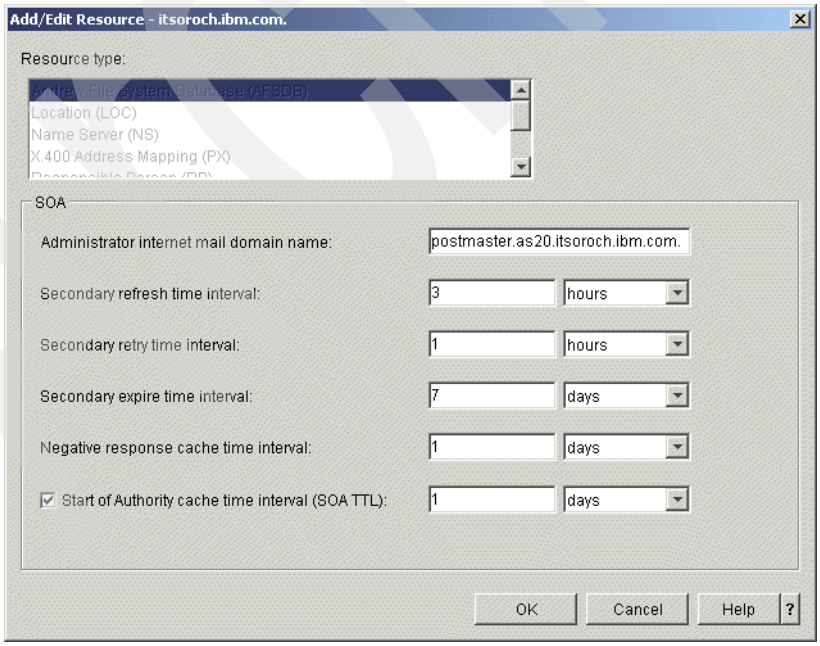


Figure 16-228 Add/Edit Resource window

26. Returning to the Primary Zone Properties - Resources window, click **Add** to open its Add/Edit Resource window. Choose **Name Server(NS)**. Type `as26.itsoroch.ibm.com`.

(answer 4 in Table 16-6 on page 469) as the Name server domain name. Check **Cache time interval**, type 1, and choose **days**, as shown in Figure 16-229.

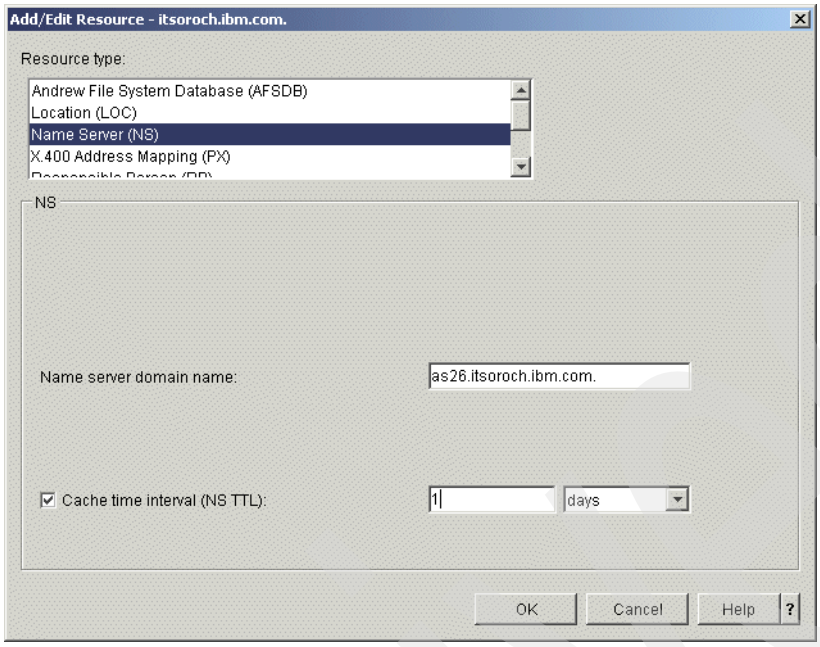


Figure 16-229 Primary Zone Properties window

27. In the Name Server IP Addresses window, click **Add**. Type 192.168.0.5 (answer 4 in Table 16-6 on page 469) and click **OK**, as shown in Figure 16-230.

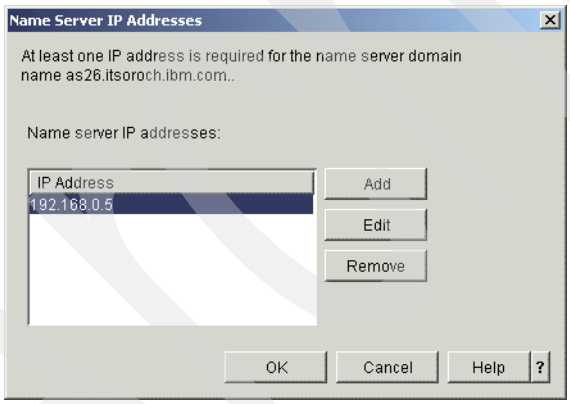


Figure 16-230 Name Server IP Address

28. In the DNS Configuration window, right-click **Reverse Lookup Zones** and choose **New Primary Zone**, as shown in Figure 16-231.

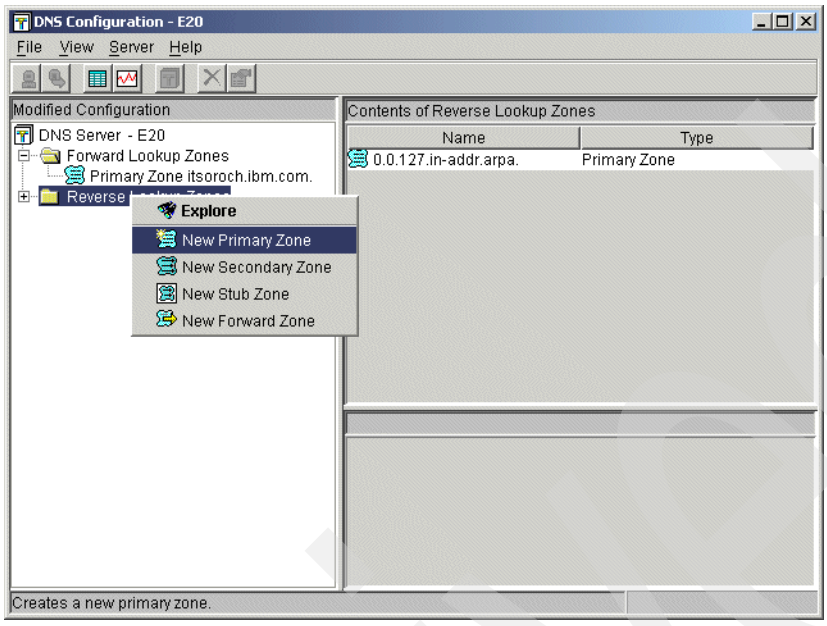


Figure 16-231 DNS Configuration window

29. In the Zone Domain Name window, type `0.168.192.in-addr.arpa.` in the Fully qualified domain name field, as shown in Figure 16-232. Click **Next** to continue.

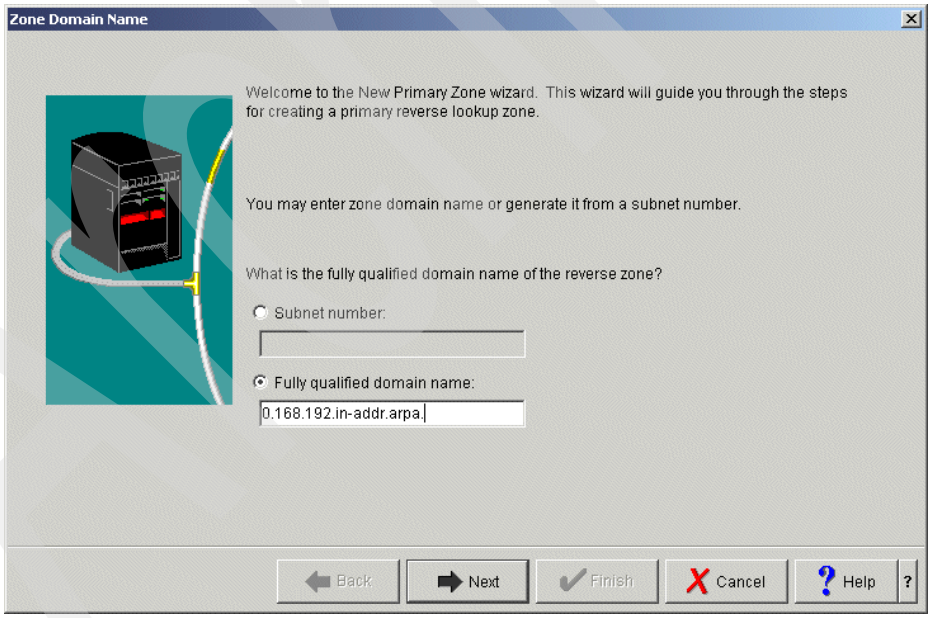


Figure 16-232 Zone Domain Name window

30. In the Name Servers window (Figure 16-233), select **as20.itsoroch.ibm.com.** Click **Edit**.

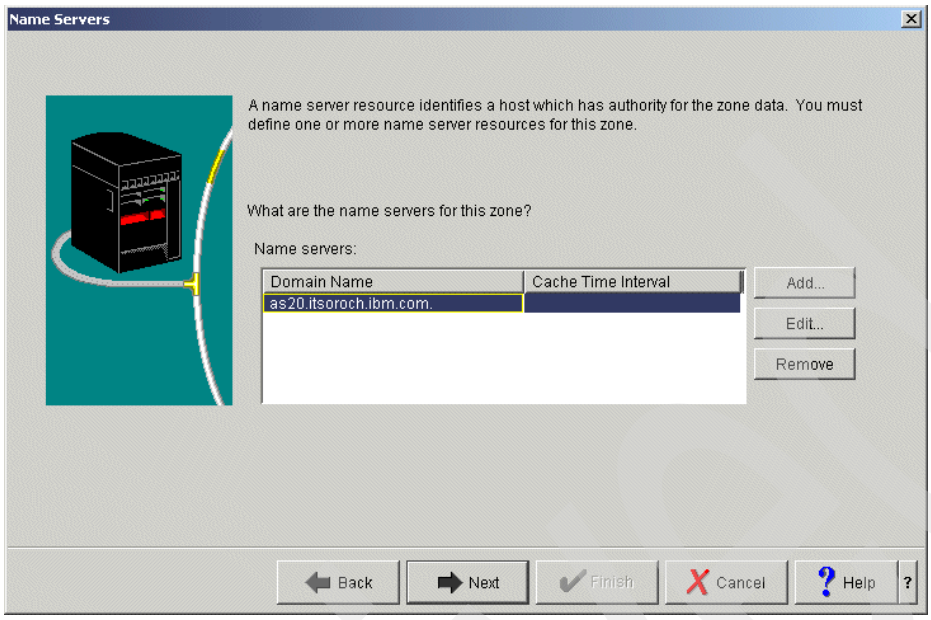


Figure 16-233 Name Servers window

31. In the Edit Name Server window, select **Cache time interval (NS TTL)**. Type 1 and choose **days**, as shown in Figure 16-234. Click **Next** to continue.

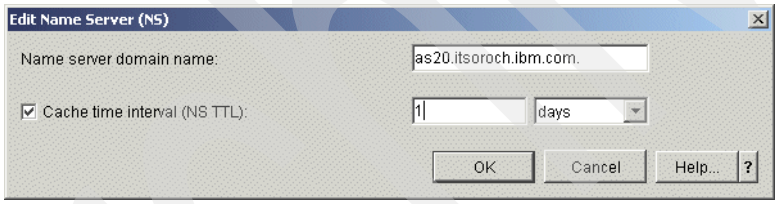


Figure 16-234 Edit Name Server window

32. In the Static or Dynamic Zone window, select **Perform static updates** (Figure 16-235). Click **Next**.

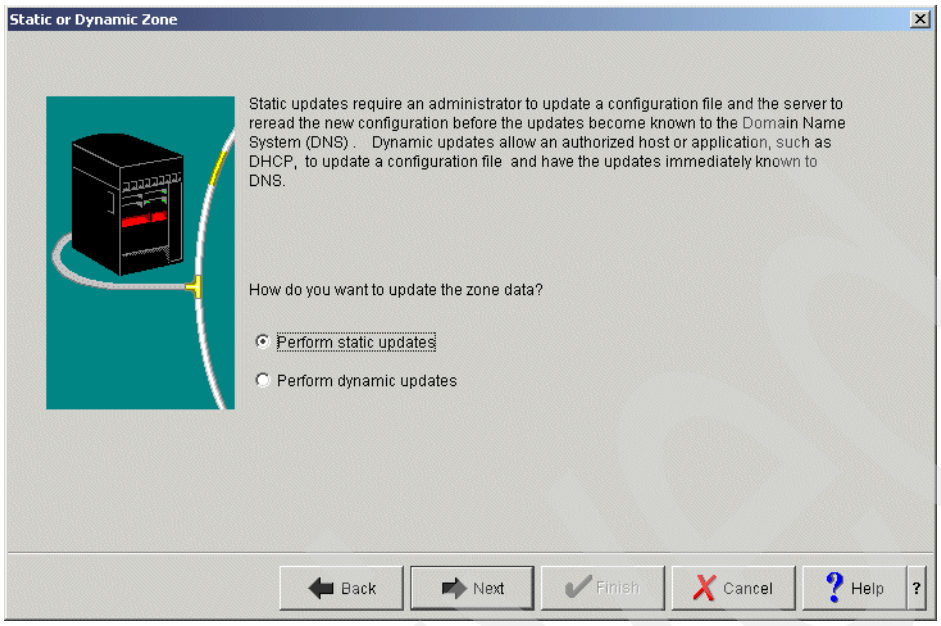


Figure 16-235 Static or Dynamic Zone window

33. In the Summary window (Figure 16-236), click **Finish**.

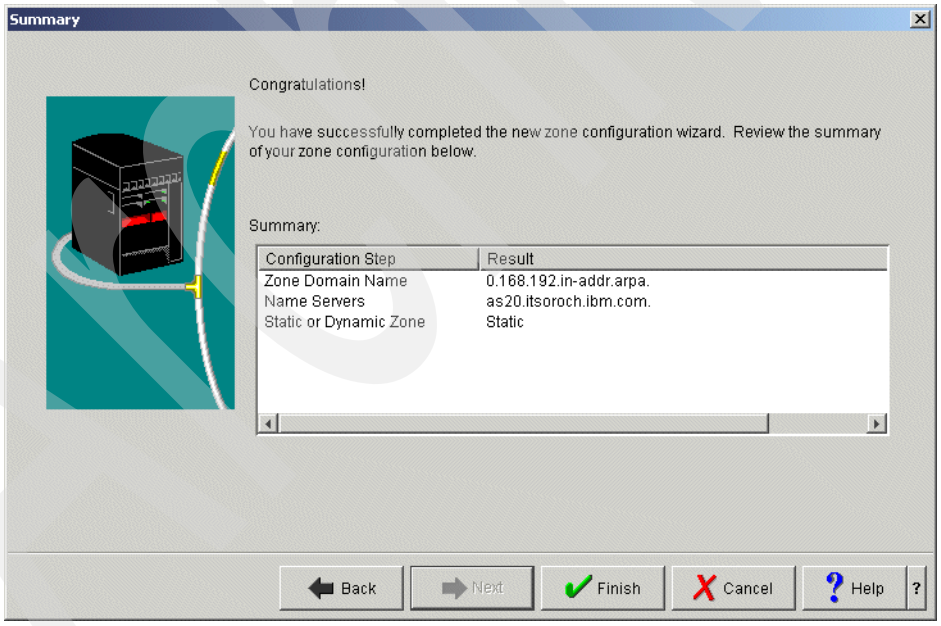


Figure 16-236 Summary window

34. In the DNS Configuration window, right-click **Primary Zone 0.168.192.in-addr.arpa.** and choose **Properties**, as shown in Figure 16-237.

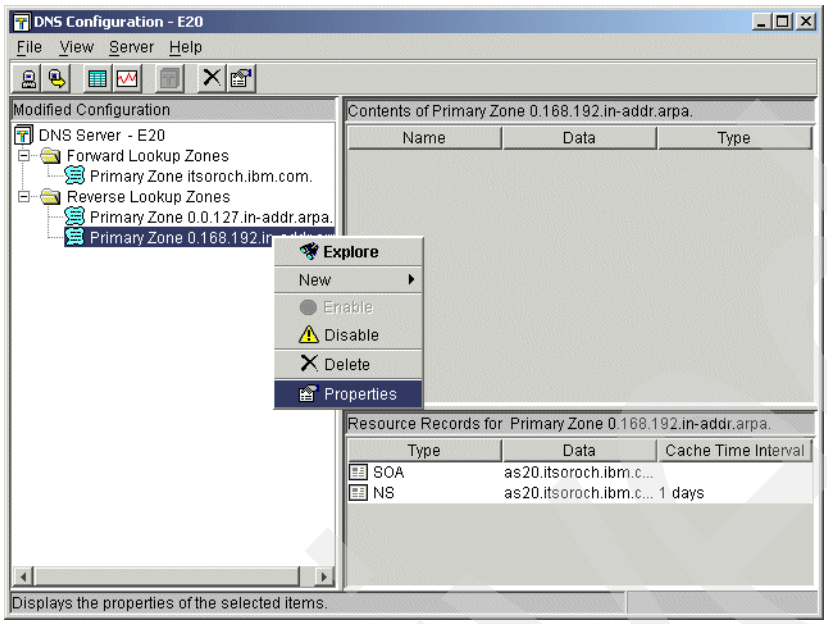


Figure 16-237 DNS Configuration window

35. In the Primary Zone Properties window, select **allow-query**. Choose **Access Control List** as the Matching list element type, as shown in Figure 16-238. Click **Add**.

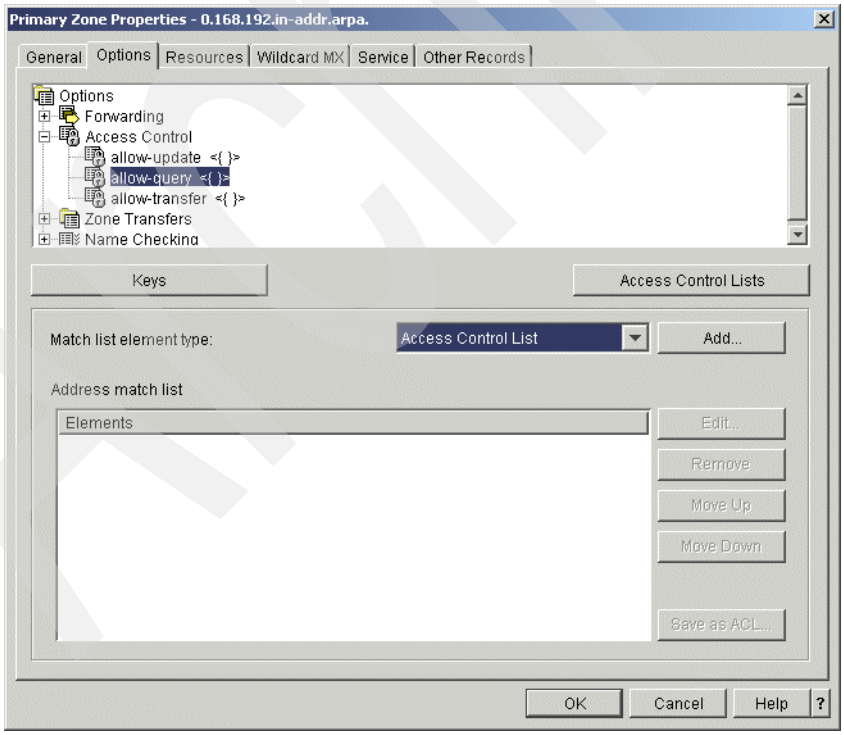


Figure 16-238 Primary Zone Properties window

36. In the Access Control List window, choose **any**, as shown in Figure 16-239, and click **OK**.

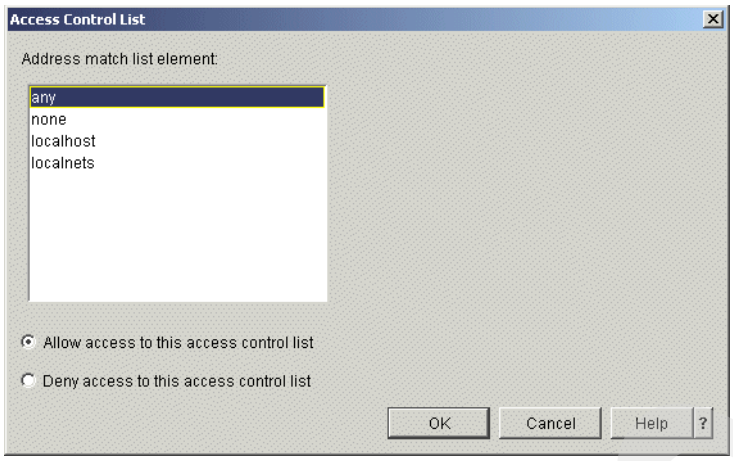


Figure 16-239 Access Control List

37. Back in the Primary Zone Properties window, click the **Resources** tab. Click **Add**, as shown in Figure 16-240.

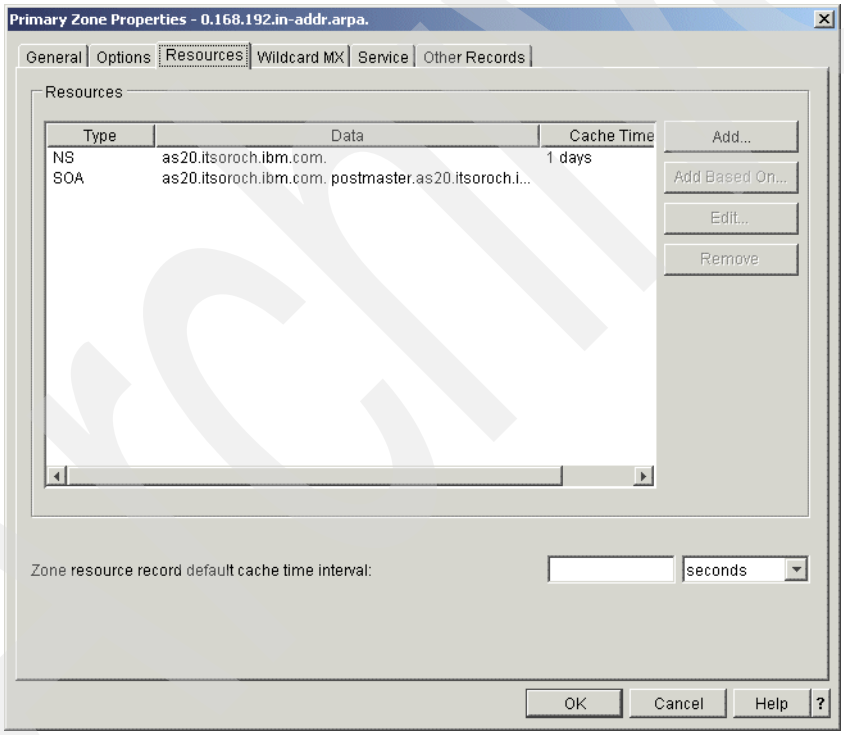


Figure 16-240 Primary Zone Properties window

38. This opens the Add/Edit Resource window. Choose **Name Server (NS)**. Type `as26.itsoroch.ibm.com.` (answer 4 in Table 16-6 on page 469) in the Name server domain name field. Click **Cache time interval (NS TTL)**, type `1` and choose **days**, as shown in Figure 16-241. Click **OK**.

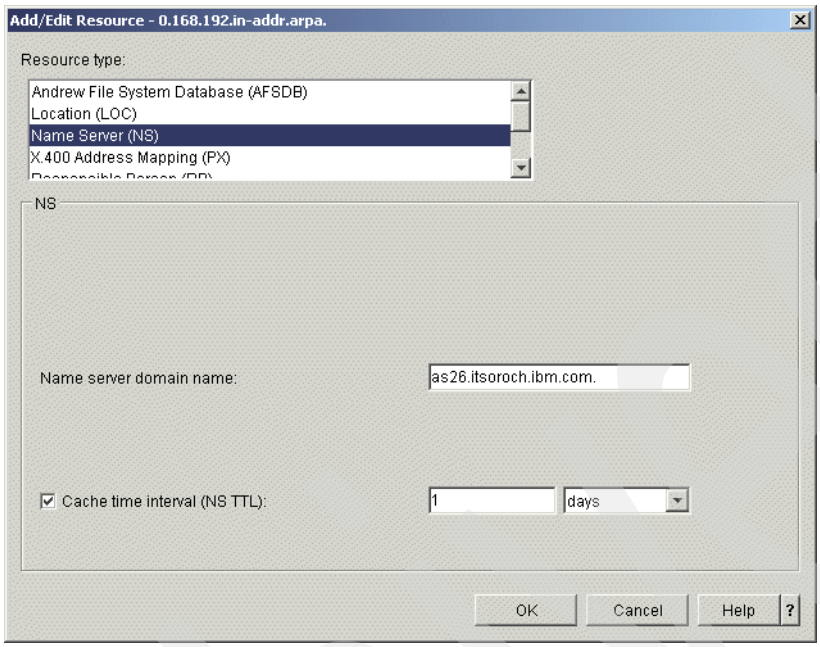


Figure 16-241 Add/Edit Resource window

39. Back in the Primary Zone Properties window (Figure 16-242), select Type **SOA** and click **Edit**.

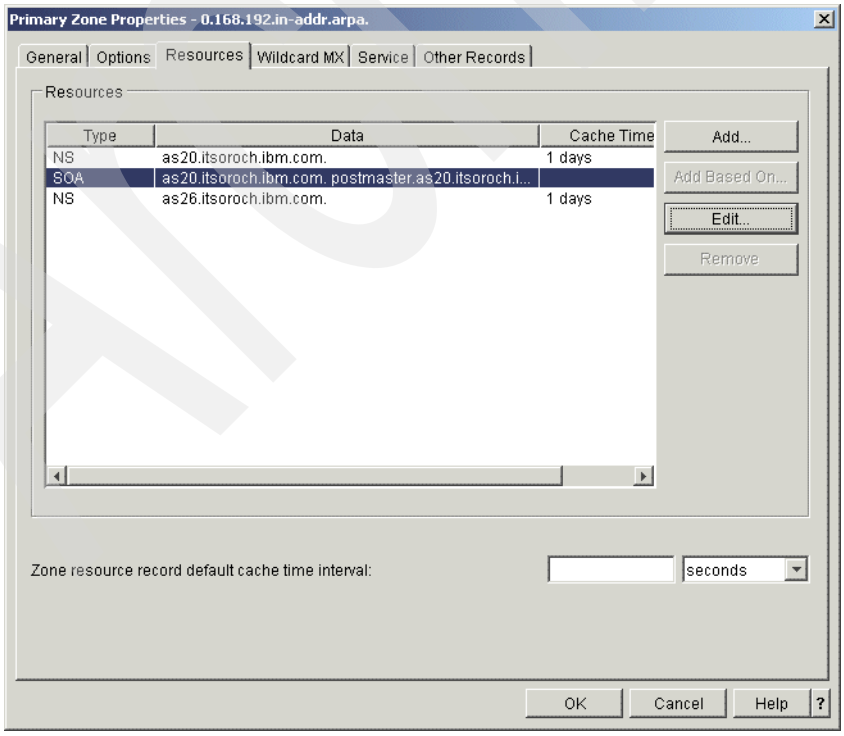


Figure 16-242 Primary Zone Properties window

40. In the Add/Edit Resource window, select **Start of Authority cache time interval (NS TTL)**. Type 1 and choose **days**, as shown in Figure 16-243. Click **OK** to continue.

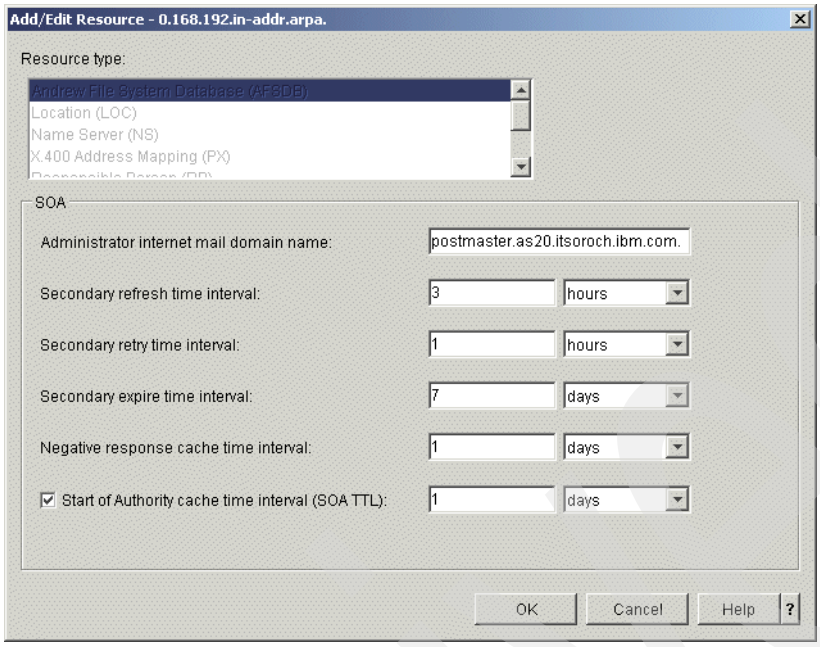


Figure 16-243 Add/Edit Resource window

41. In the DNS Configuration window, right-click **Primary Zone 0.168.192.in-addr.arpa.** and choose **New** → **Host**, as shown in Figure 16-244.

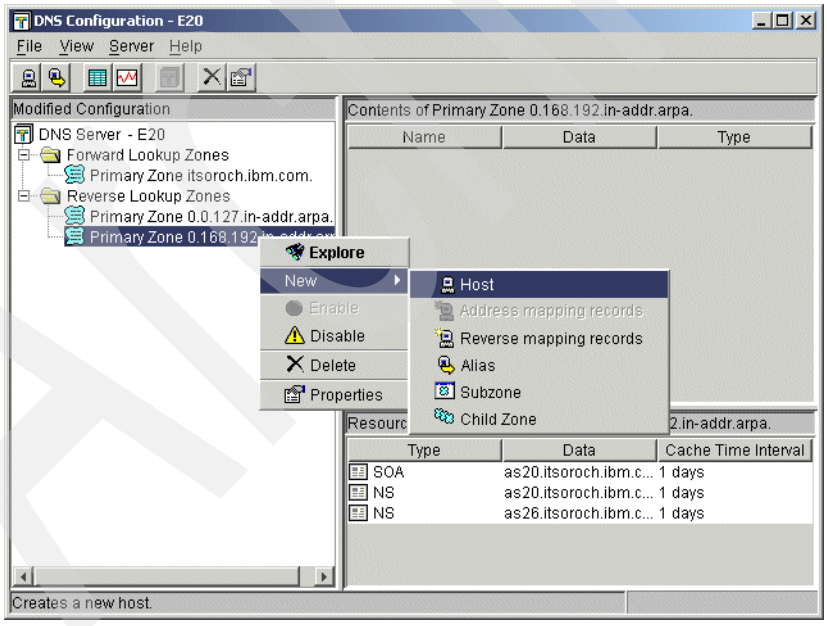


Figure 16-244 DNS Configuration window

42. In the New Host window, type `1.0.168.192.in-addr.arpa.` in the Domain name field, as shown in Figure 16-245. Click **Next** to continue.

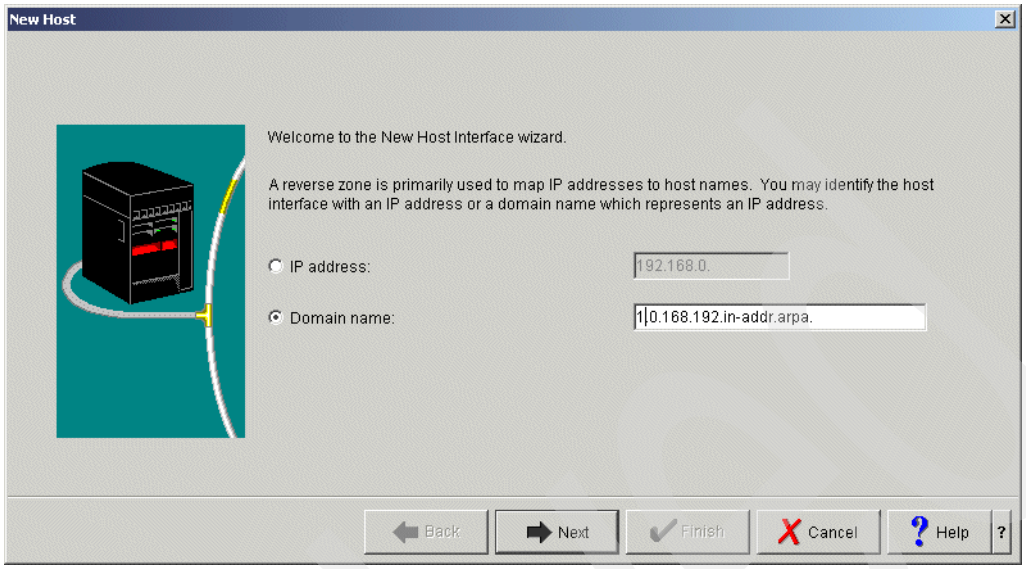


Figure 16-245 New Host window

43. In the New Host Resources window (Figure 16-246), click **Add** to open its Add/Edit Resources window.

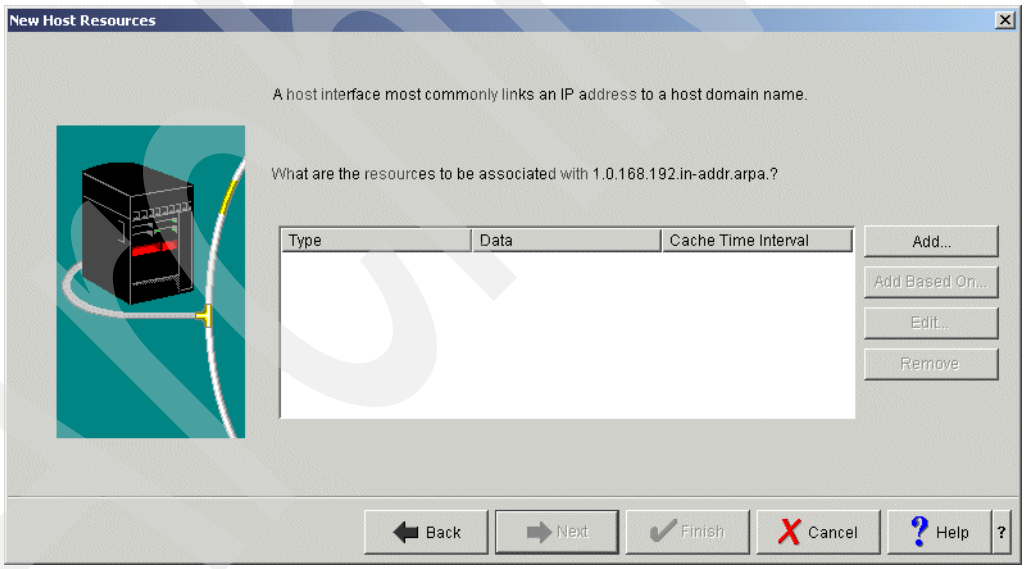


Figure 16-246 New Host Resources window

44. Type `as20.itsoroch.ibm.com.` (answer 6 in Table 16-6 on page 469) as the Fully qualified host domain name. Check **Cache time interval (PTR TTL)**. Type `1` and choose **days**, as shown in Figure 16-247. Click **OK** to continue.

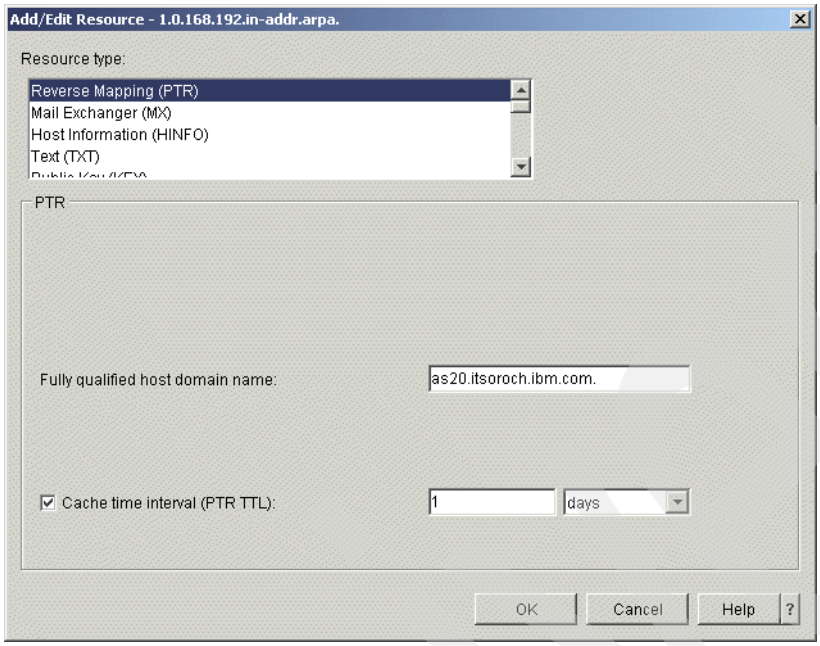


Figure 16-247 Ass/Edit Resource window

45. In the New Host Resources window (Figure 16-248), click **Finish**.

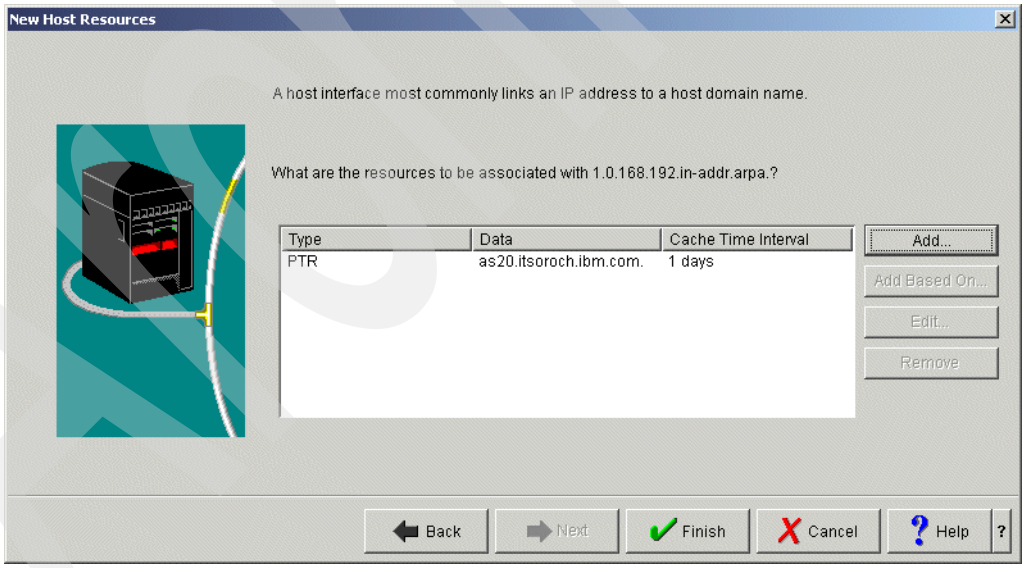


Figure 16-248 New Host Resources window

46. Repeat the procedure from step 41 on page 511 to step 45 to add a PTR record for as26.itsoroch.ibm.com. Use the IP address 192.168.0.5 (answer 4 in Table 16-6 on page 469).

47. In the DNS Configuration window, choose **File** → **Save Configuration** to save the configuration, as shown in Figure 16-248 on page 513.

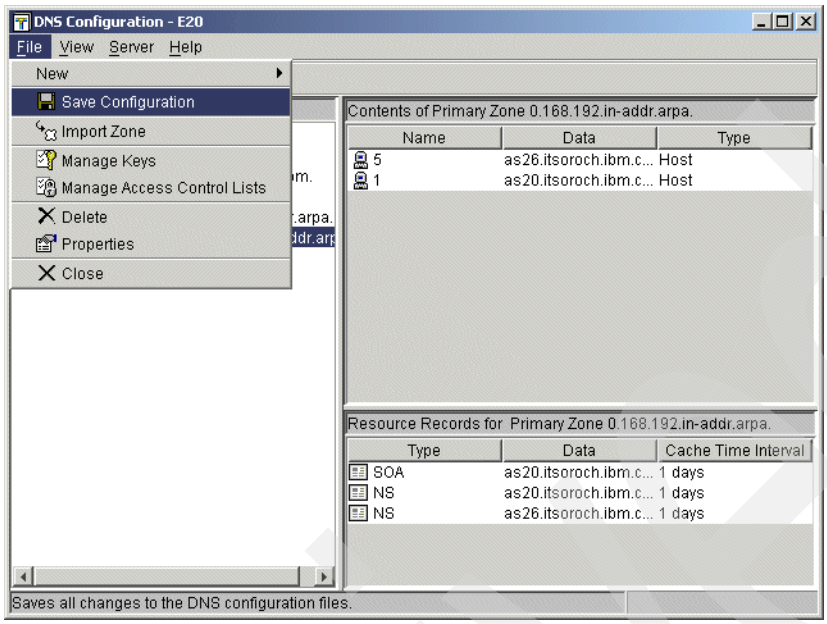


Figure 16-249 DNS Configuration window

Step 3: Create the IP filter

This is the procedure to create the IP filter, which includes a masquerade NAT definition and IP filter for the public DNS:

1. Copy the packet filter rule sentences shown in Figure 16-250. You will paste these sentences in a later step.

```
# ------------------------------------------------------------------------------------------------------
# Statements to hide 172.23.10.2 - 172.23.10.254 behind 192.168.0.1
# This is the masquerade NAT definition
# ------------------------------------------------------------------------------------------------------
ADDRESS HIDE1   IP = 172.23.10.2 THROUGH 172.23.10.254 TYPE = TRUSTED
ADDRESS BEHIND1   IP = 192.168.0.1 TYPE = BORDER
HIDE HIDE1   BEHIND BEHIND1

# ------------------------------------------------------------------------------------------------------
# Statements to allow TCP port80(Web) for 172.23.10.2 - 172.23.10.254
# This allows the TCP port80(web) traffic between internal clients and any Web servers on the Internet
# If you need any other protocols(telnet, ftp, etc), add the filter rules as required.
# ------------------------------------------------------------------------------------------------------

FILTER SET nat   ACTION = PERMIT   DIRECTION = OUTBOUND   SRCADDR = HIDE1  DSTADDR = *   PROTOCOL = TCP
DSTPORT = 80   SRCPORT = *   JRN = OFF
FILTER SET nat   ACTION = PERMIT   DIRECTION = INBOUND   SRCADDR = *   DSTADDR = HIDE1   PROTOCOL = TCP
DSTPORT = *   SRCPORT = 80   JRN = OFF

# ------------------------------------------------------------------------------------------------------
# Statements to allow UDP port53(DNS) for 192.168.0.1
# This allows the UDP port53(DNS) traffic between External DNS server E20 and any other servers
# on the Internet
# ------------------------------------------------------------------------------------------------------

FILTER SET dns   ACTION = PERMIT   DIRECTION = OUTBOUND   SRCADDR = BEHIND1 DSTADDR = * PROTOCOL = UDP
DSTPORT = 53   SRCPORT = *   JRN = OFF
FILTER SET dns   ACTION = PERMIT   DIRECTION = OUTBOUND   SRCADDR = BEHIND1 DSTADDR = * PROTOCOL = UDP
DSTPORT = *   SRCPORT = 53   JRN = OFF
FILTER SET dns   ACTION = PERMIT   DIRECTION = INBOUND   SRCADDR = * DSTADDR = BEHIND1 PROTOCOL = UDP
DSTPORT = 53   SRCPORT = *   JRN = OFF
FILTER SET dns   ACTION = PERMIT   DIRECTION = INBOUND   SRCADDR = * DSTADDR = BEHIND1 PROTOCOL = UDP
DSTPORT = *   SRCPORT = 53   JRN = OFF

# ------------------------------------------------------------------------------------------------------
# Filter Interface statement
# This defines the relationship between Filter rules(nat,dns) and Interface ETHLIN1
# ------------------------------------------------------------------------------------------------------

FILTER_INTERFACE   LINE = ETHLIN2 SET = nat, dns
```

Figure 16-250 Sample IP Filter AS20

2. In the iSeries Navigator window, expand **AS20** → **Network** → **IP Policies** and right-click **Packet Rules** and choose **Rules Editor** (Figure 16-251).

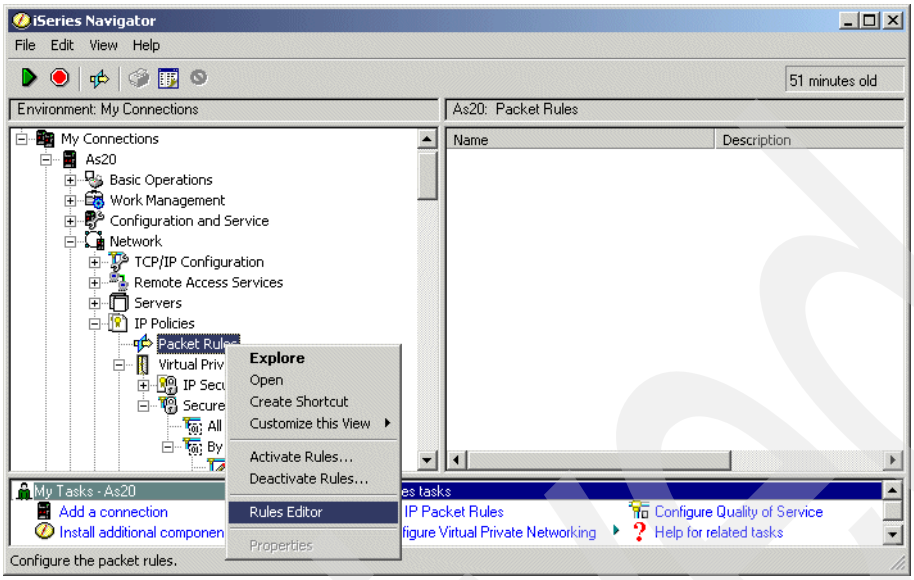


Figure 16-251 iSeries Navigator window

3. In the Welcome Packet Rules Configuration window, check **Create a new packet rules file** (Figure 16-252). Click **OK**. In the Getting Started window, click **OK** to continue.

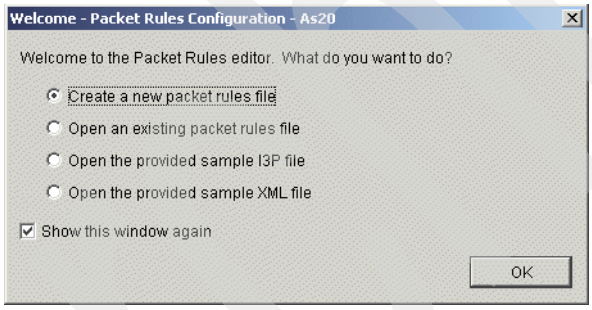


Figure 16-252 Welcome Packet Rules Configuration window

4. In the Packet Rules Editor window, select **Edit** → **Paste** to copy the sentences from Figure 16-250 on page 515 into the new file. Figure 16-253 shows the result.

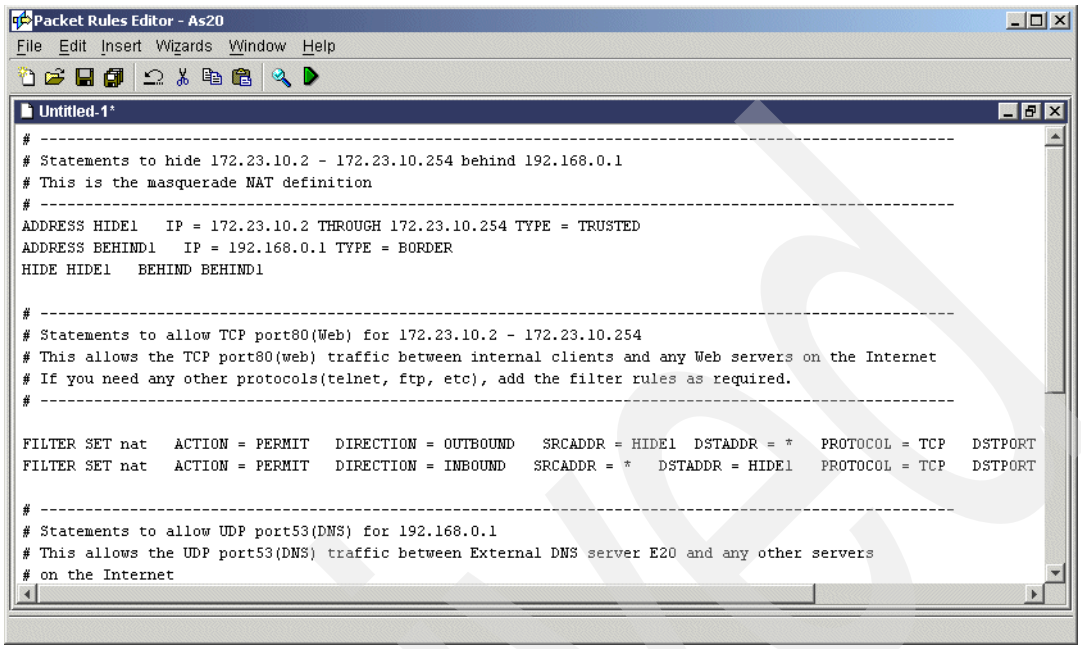


Figure 16-253 Packet Rules Editor window

5. Choose **File** → **Verify Rules** (Figure 16-254). Click **Yes** on the Save Required window.

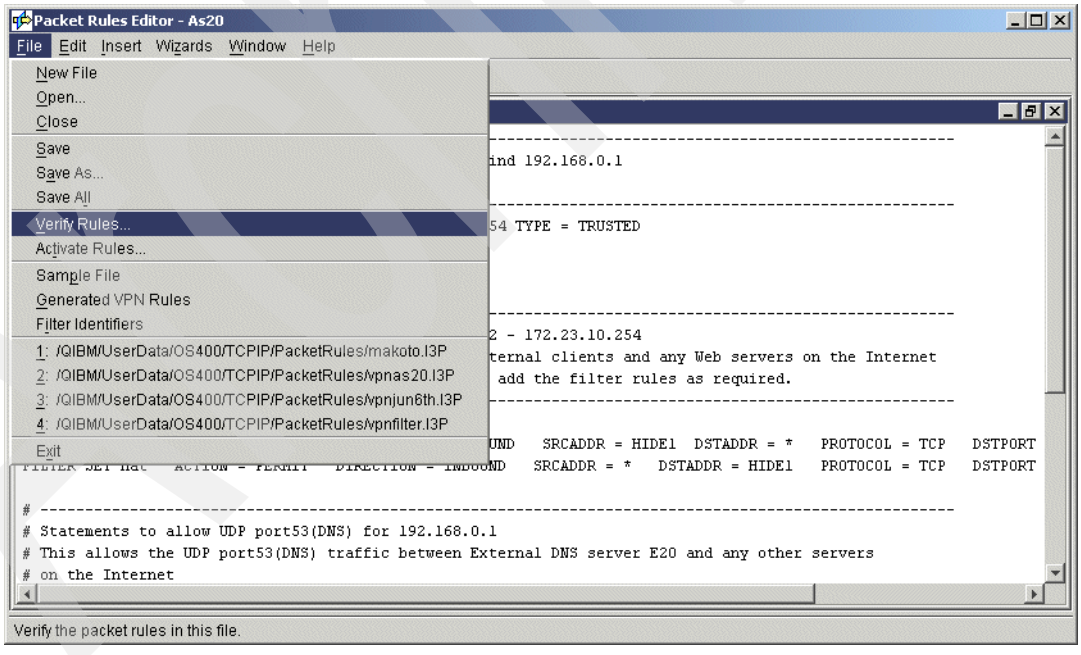


Figure 16-254 Packet Rules Editor window

6. In the Save File window, type the filter file name `split_nat`. Make sure that split_nat.i3p is being saved to the PacketRules directory, as shown in Figure 16-255. All custom filter rules must be saved in the /QIBM/UserData/OS400/TCPIP/PacketRules directory.

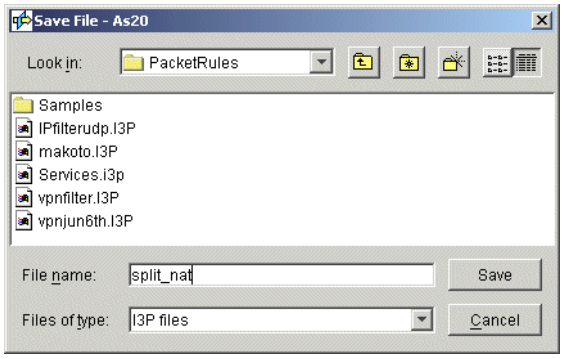


Figure 16-255 Save Files window

7. In the Verify Packet Rules window, select **Verify only selected file**. Select **Verify these rules on the following interface** and choose **ETHLIN2** (answer 3 in Table 16-6 on page 469), as shown in Figure 16-256. Click **OK** to continue.

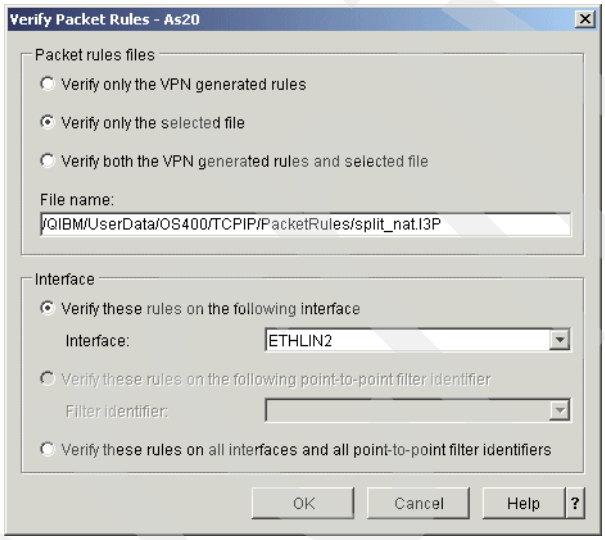


Figure 16-256 Verify Packet Rules window

8. If the Packet Rule is verified successfully, the message `The rules were successfully verified` will appear in the bottom pane of the Packet Rules Editor, as shown in Figure 16-257. If there is an error, read the error message and fix it before you proceed. Select **File** → **Activate Rules**.

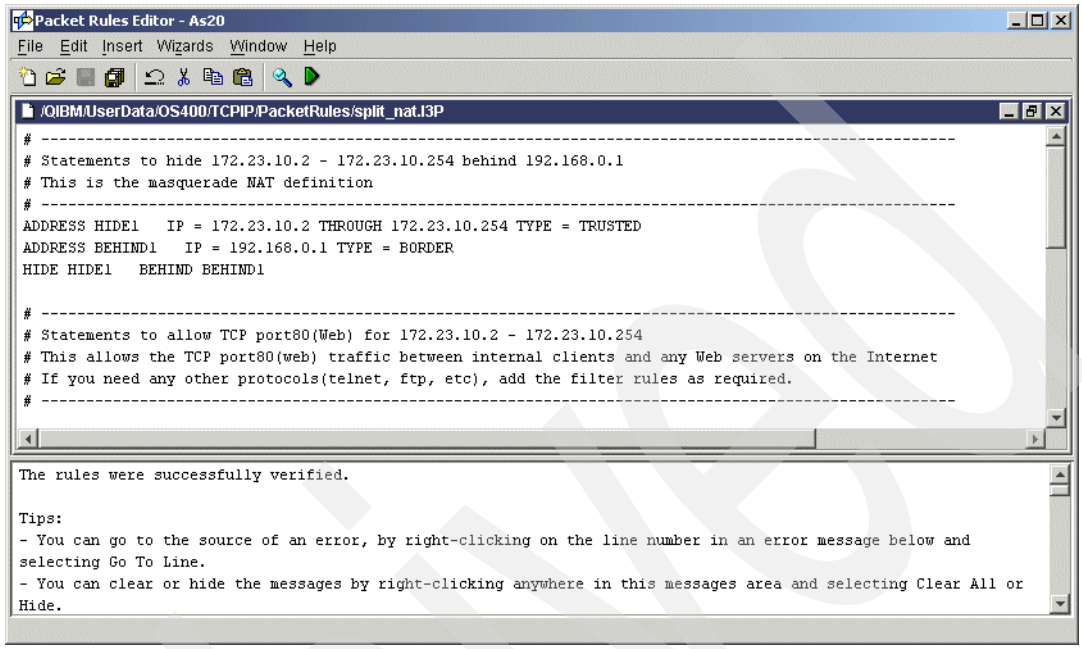


Figure 16-257 Packet Rules Editor

9. In the Activate Packet Rules window, select **Activate only the selected file**. Select **Activate these rules on the following interface** and select **ETHLIN2**, as shown in Figure 16-258. Click **OK** to continue.

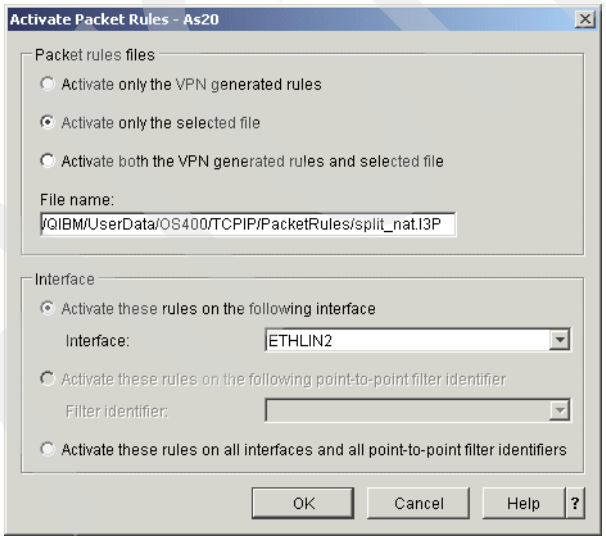


Figure 16-258 Activate Packet Rules window

10. If the Packet Rule is activated successfully, the message `The rules were successfully activated` will appear in the bottom pane of the Packet Rules Editor, as shown in Figure 16-259.

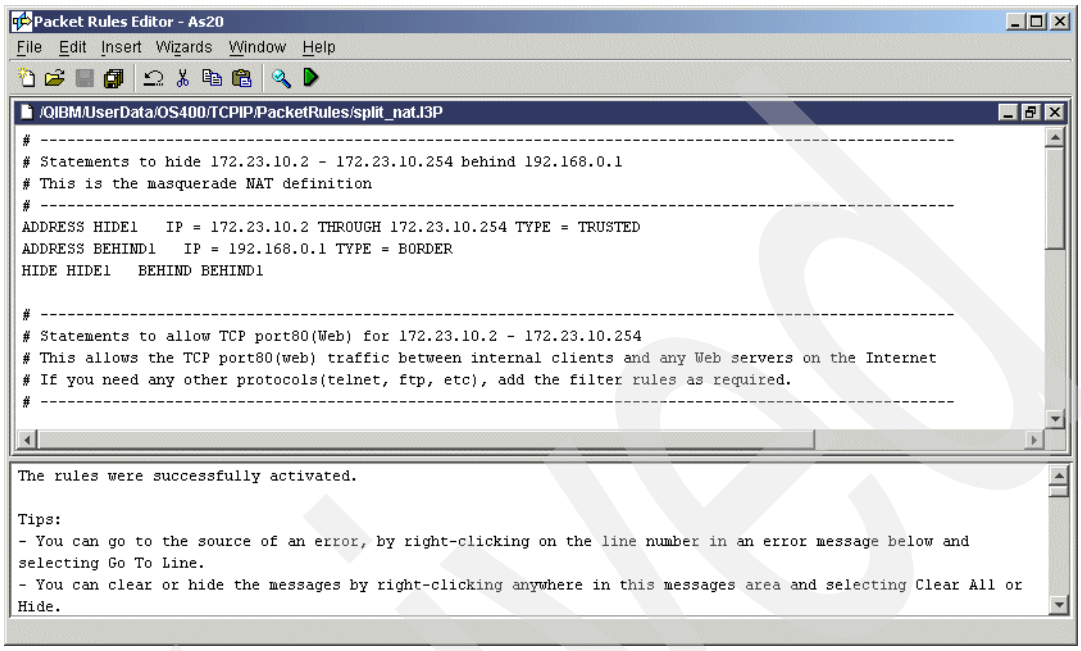


Figure 16-259 Packet Rules Editor window

17

Dynamic PPP scenarios

This chapter contains PPP scenarios referring to the cases explained in Chapter 5, “Point-to-Point Protocol (PPP)” on page 87. Each sample scenario consists of four sections:

- The scenario overview. This includes the conditions in which you would choose the scenario and a sample network configuration.
- A planning worksheet. This worksheet helps you prepare the required parameters that you will need to configure the sample configuration.
- A step-by-step guide to configuring your sample configuration.
- How to test the sample configuration.

This chapter contains the following sample configurations:

17.1 PPPoE branch office with secured connection

In this scenario, we configure the PPPoE branch office with a secured connection scenario.

17.1.1 Scenario overview

You might choose this scenario if you need to connect two System i servers in branch offices over the Internet (public network) with a secured connection.

Note: In some DSL or cable services, the user-side IP address of the PPPoE connection varies each time PPP authentication takes place. In this case, it is difficult to keep current IP addresses updated in the VPN configuration.

Sample network configuration

Figure 17-1 shows the sample network configuration of this scenario.

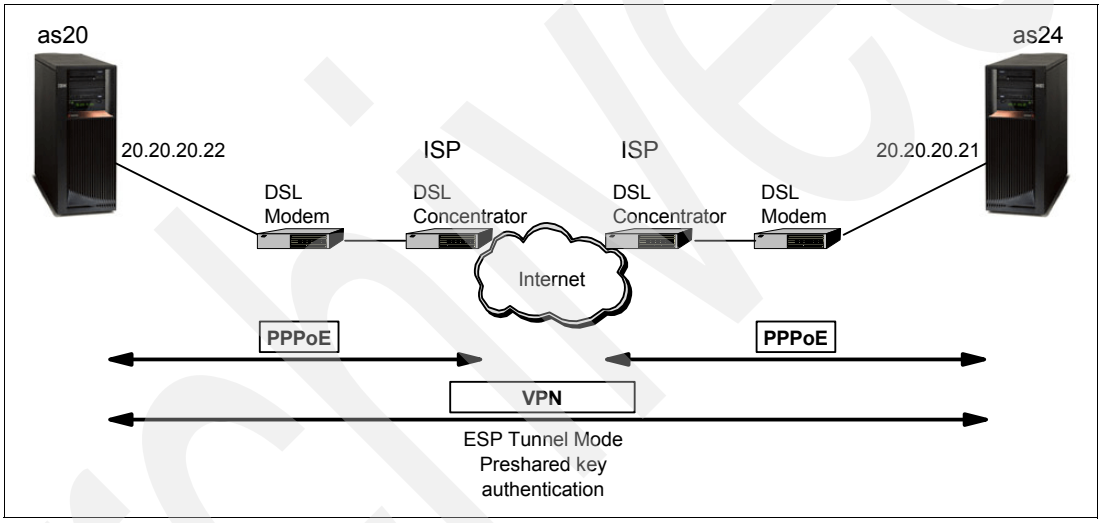


Figure 17-1 Sample network configuration: PPPoE branch office with secured connection scenario

17.1.2 Planning worksheet: PPPoE branch office with secured connection

Table 17-1 shows the planning worksheet for preparing the required parameters to configure the PPPoE Branch office with secured connection scenario. We have filled in our answers for each question in the adjacent Scenario answers column.

Table 17-1 Planning worksheet for PPPoE branch office with secured connection scenario

No.	Questions to create PPPoE branch office with secured connection scenario	Scenario answers
1	What is the user ID and password to log into your Internet Service Provider? Which authentication method is used?	User ID = itso Password = password PAP authentication
2	What is the user-side fixed IP address (AS20 side) that is provided by your Internet Service Provider?	20.20.20.22
3	What is the fixed IP address on the AS24 side?	20.20.20.21

No.	Questions to create PPPoE branch office with secured connection scenario	Scenario answers
4	What DNS server IP address is provided by your Internet Service Provider?	192.168.100.1
5	Which Feature Code (F/C) 2838 or 2849 resource will you use for this PPPoE connection? Note: 1. This 2838 or 2849 resource can only be used by PPPoE if it is not in use at the same time by IPv6 or IPv4 interfaces. Note that the sharing restriction has been removed in V5R4. 2. F/C 2838 and 2849 are the only hardware features that support PPPoE host (client configurations) in V5R2 and V5R3. Note that the list of supported resources has been extended in V5R4.	CMN07
6	What is the PPPoE connection name as an originator connection profile on AS20 and AS24?	PPPOE1
7	What is the physical line name for this PPPoE connection on AS20?	PPPOE2
8	What is the Ethernet speed and mode for PPPOE2 on AS20?	speed = 10 Mbps mode = half
9	What is the pre-shared key for IKE Phase1 negotiation?	makoto
10	What is the encryption algorithm for ESP protocol? What is the authentication algorithm for ESP protocol?	DES-CBC MD5
11	What is the hash algorithm for IKE authentication? What is the Diffie-Hellman Group for IKE authentication? What is the IKE key expire hours?	MD5 Group1(default 768 MODP) 2 hours
12	Which mode is used for IKE negotiation?	IKE Main mode
13	What mode do you use for ESP protocol?	ESP Tunnel mode
14	What is the VPN endpoint IP address for AS20 side?	20.20.20.22
15	What is the VPN endpoint IP address for AS24 side?	20.20.20.21
16	What is the local data endpoint IP address for AS20 side?	20.20.20.22
17	What is the local data endpoint IP address for AS24 side?	20.20.20.21

Note: The algorithms shown in this example are not necessarily the recommended algorithms. Stronger algorithms are available on System i. We only use them in this example since they should be supported on the majority of operating systems that support IKE and IPSec.

17.1.3 Configuring the PPPoE branch office with secured connection scenario

In this scenario, we create the scenario for the PPPoE branch office with secured connection in the following steps:

- Step 1: Creating the PPPoE definition on AS20
- Step 2: Test the PPPoE connection on AS20
- Step 3: Create VPN connection on the PPPoE definition on AS20
- Step 4: Create VPN connection on the PPPoE definition on AS24
- Step 5: Start the VPN connection

Note: In this scenario, it is assumed that the configuration for AS24 is very similar to the configuration steps we detail for AS20. After you have completed and tested the configuration for AS20, you should use the procedure outlined in “Step 1: Creating the PPPoE definition on AS20” on page 524 to complete the configuration for AS24.

Step 1: Creating the PPPoE definition on AS20

In this step, we create the PPPoE definition on AS20:

1. In the iSeries Navigator window, expand **Network** → **Remote Access Services**. Right-click **Originator Connection Profiles** and select **New Profile**, as shown in Figure 17-2.

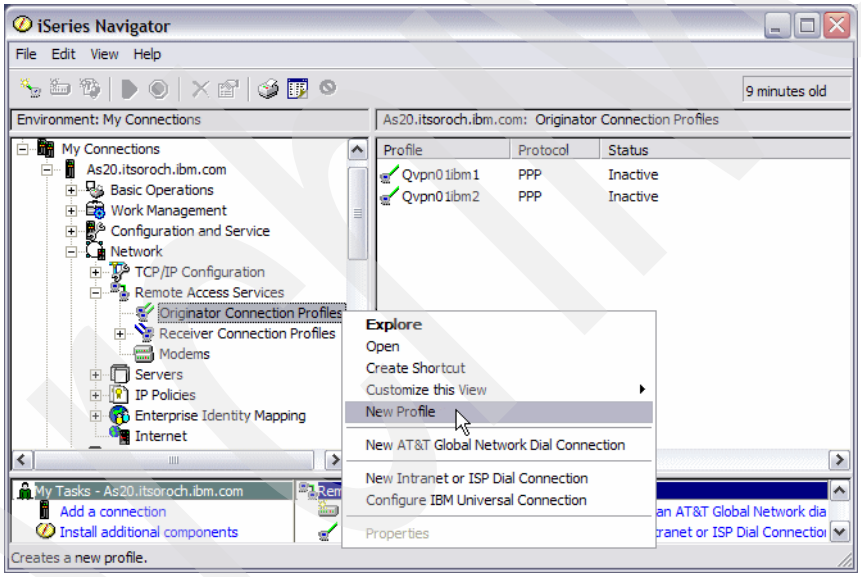


Figure 17-2 iSeries Navigator window

2. In the New Point-to-Point Connection Profile setup window, select Protocol type **PPP**. For Connection type, select **PPP over Ethernet**, as shown in Figure 17-3. Click **OK** to continue.

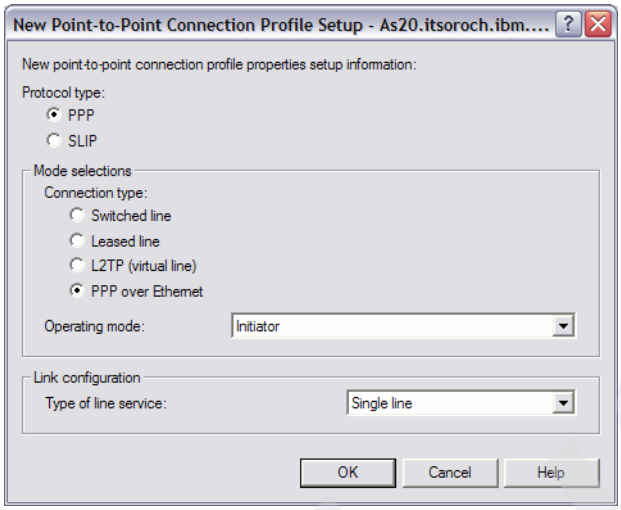


Figure 17-3 New Point-to-Point Connection Profile Setup window

3. In the New Point-to-Point Profile Properties window, click the **General** tab. Enter the PPPoE connection name `PPPoE1` in the Name field (answer 6 in Table 17-1 on page 522). Enter a description of the connection in the description field if needed. If you want the connection to automatically start with TCP/IP, select **Start profile with TCP**. Click the **Connection** tab to continue.

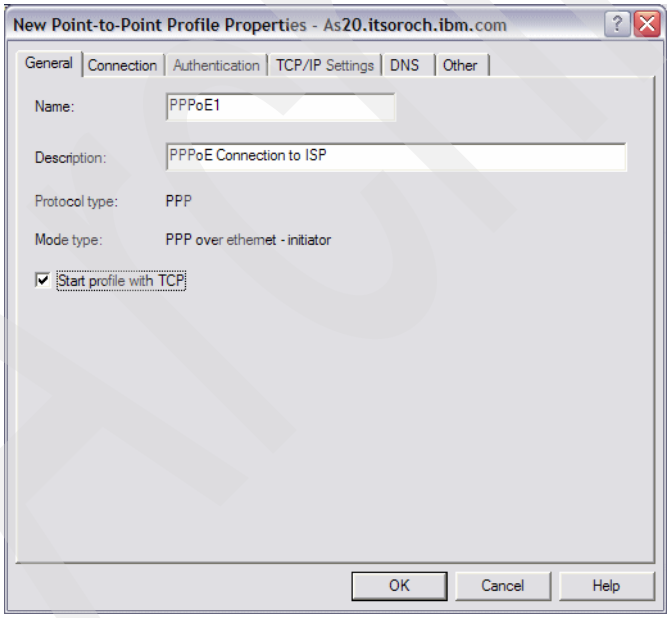


Figure 17-4 New Point-to-Point Profile Properties: General tab

4. On the Connection tab, choose **PPPOE1** from the PPPoE virtual line name pull-down menu, as shown in Figure 17-5.

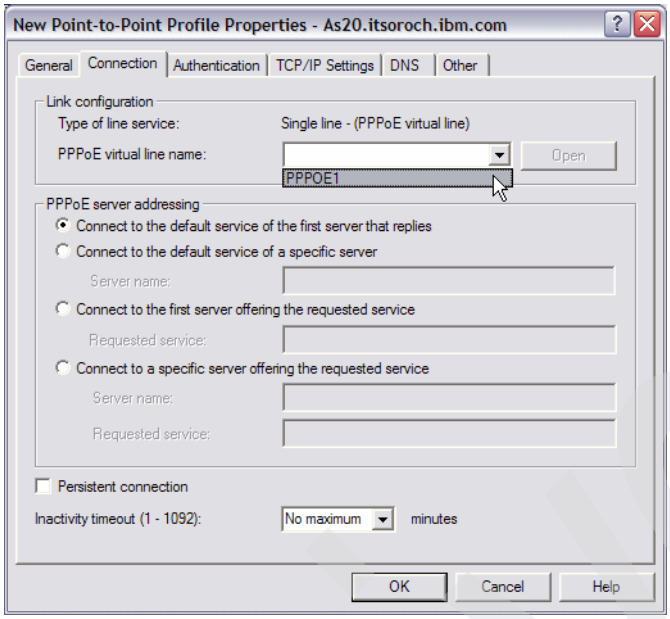


Figure 17-5 New Point-to-Point Properties: Connection tab

5. The window shown in Figure 17-6 opens. Enter a PPPoE line description if needed. Click the **Link** tab to continue.

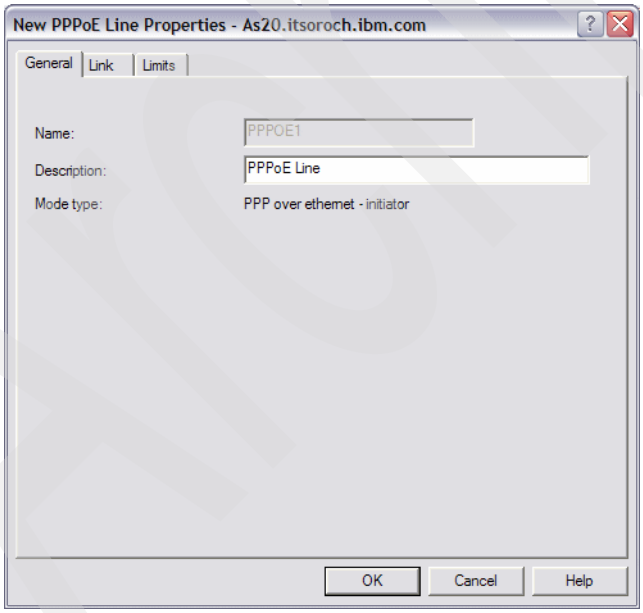


Figure 17-6 New PPPoE Line Properties: General tab

6. On the Link tab, choose **PPPOE2** in the Physical line name field (answer 7 in Table 17-1 on page 522), as shown in Figure 17-7. Click **New**.

> **Note:** Starting in V5R4, you can share the same Ethernet line for PPPoE, IPv4, and IPv6 network traffic if desired.

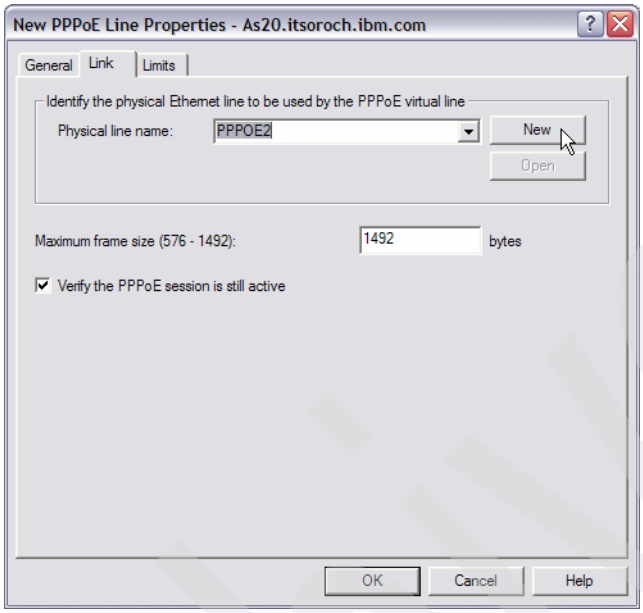


Figure 17-7 New PPPoE Line Properties: Link tab

7. In the New Ethernet Line Properties window, enter a line Description if needed. Select **CMN07** in the Hardware resource list (answer 5 in Table 17-1 on page 522), as shown in Figure 17-8. Click the **Link** tab to continue.

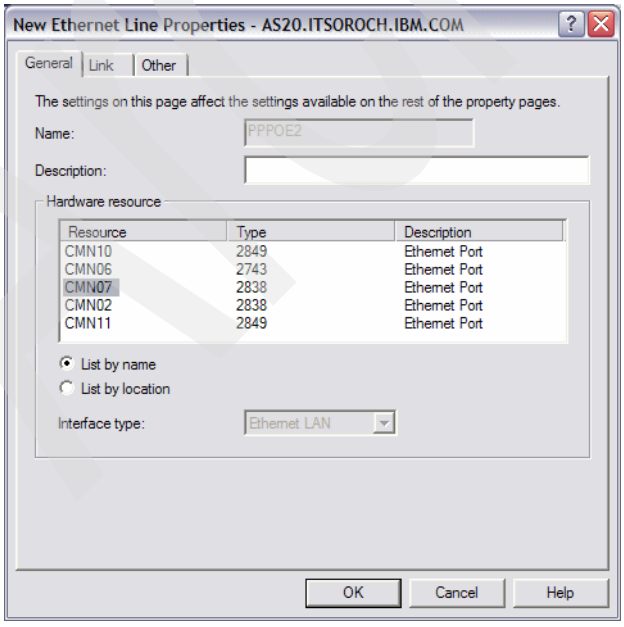


Figure 17-8 New Ethernet Line Properties: General tab

8. On the Link tab, select the Line speed **10 Mb/second**. Choose **Half** from the Duplex pull-down menu (answer 8 in Table 17-1 on page 522), as shown in Figure 17-9. Click **OK** to continue.

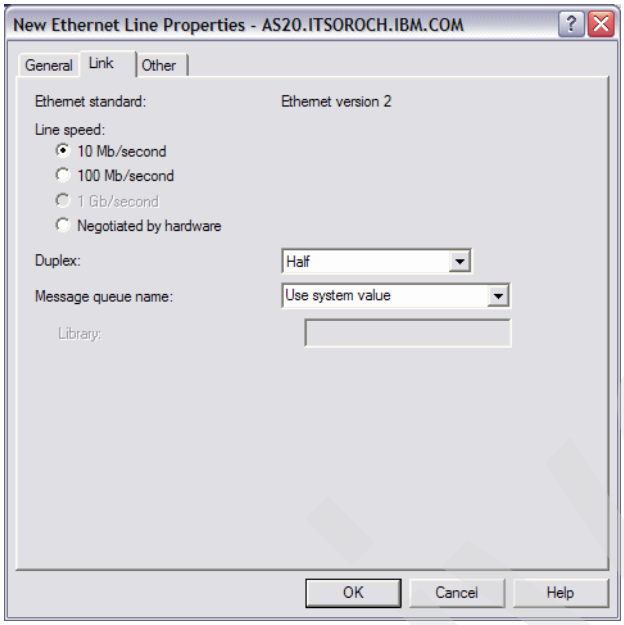


Figure 17-9 New Ethernet Line Properties: Link tab

9. Click the **Limits** tab. For this example, keep the default value, as shown in Figure 17-10. Click **OK** to continue.

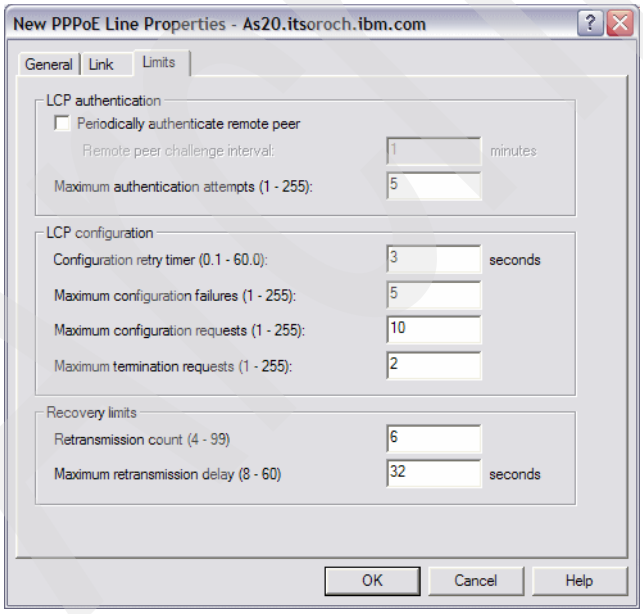


Figure 17-10 New PPPoE Line Properties: Limits tab

10. In the New Point-to-Point Profile Properties window, click the **Authentication** tab. Check **Allow the remote system to verify the identity of this iSeries server**. Select **Require unencrypted password (PAP)**. Enter `itso` in the User name field. Enter `password` in the Password field (answer 1 Table 17-1 on page 522), as shown in Figure 17-11.

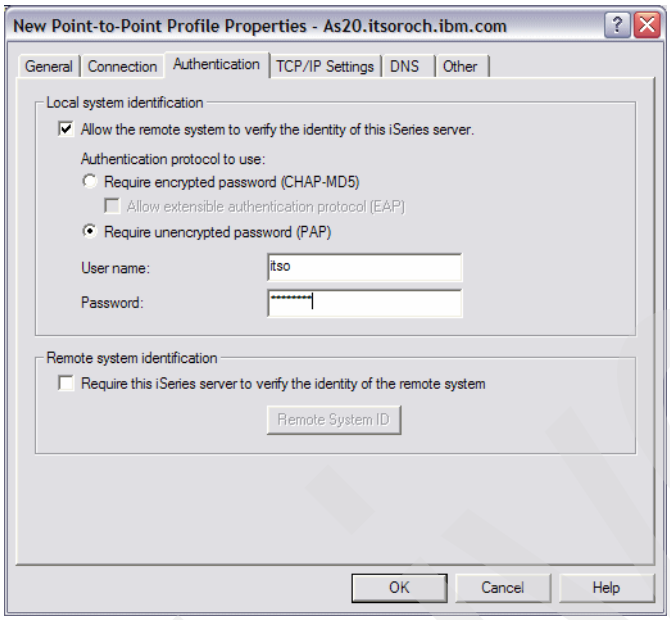


Figure 17-11 New Point-to-Point Profile Properties: Authentication tab

11. Click the **TCP/IP Settings** tab. When the Password confirmation window pops up, enter the password again (`password`) and click **OK**.

12. On the TCP/IP Settings tab, keep the default values, as shown in Figure 17-12, for our sample configuration. Click the **DNS** tab to continue.

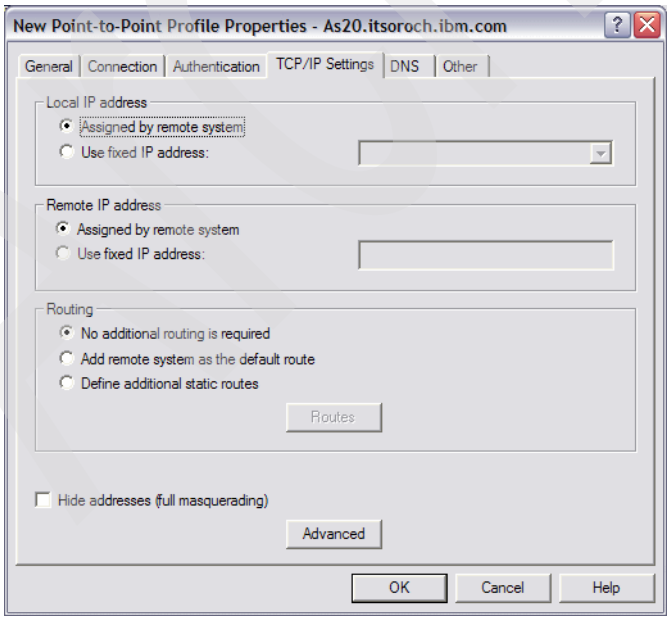


Figure 17-12 New Point-to- Point Profile Properties: TCP/IP Settings tab

13. On the DNS tab, select **IP address** and enter `192.168.100.1` (answer 4 in Table 17-1 on page 522), as shown in Figure 17-13. Click **OK**.

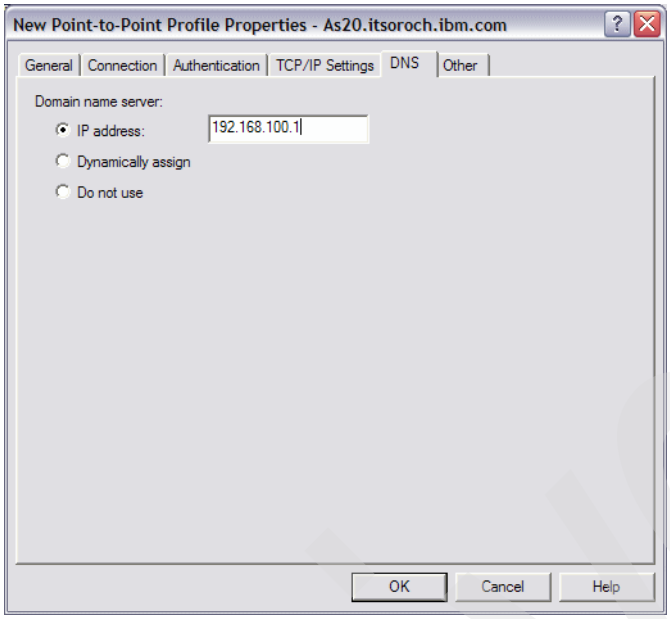


Figure 17-13 New Point-to-Point Profile Properties: DNS tab

Step 2: Test the PPPoE connection on AS20

In this step, we test the PPPoE connection that we created in the previous step:

1. In the iSeries Navigator window, under Originator Connection Profiles for your server, right-click **PPPOE1**, which you created in the previous step and choose **Start**, as shown in Figure 17-14.

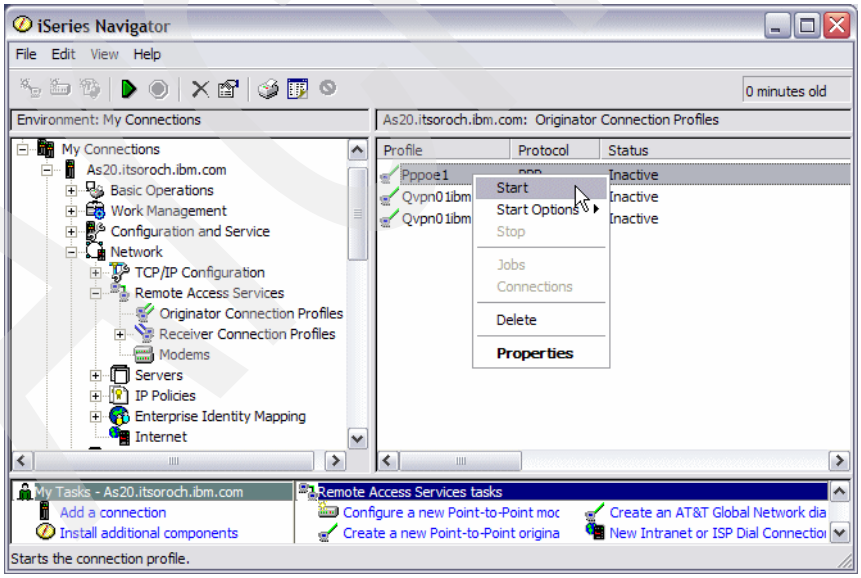


Figure 17-14 iSeries Navigator window: Originator Connection Profiles: PPPoE1

2. PPPOE1 Status will change as it is being activated, as shown in Figure 17-15. Wait for the status to show Active (for a successful connection) or Ended (for a failing connection).

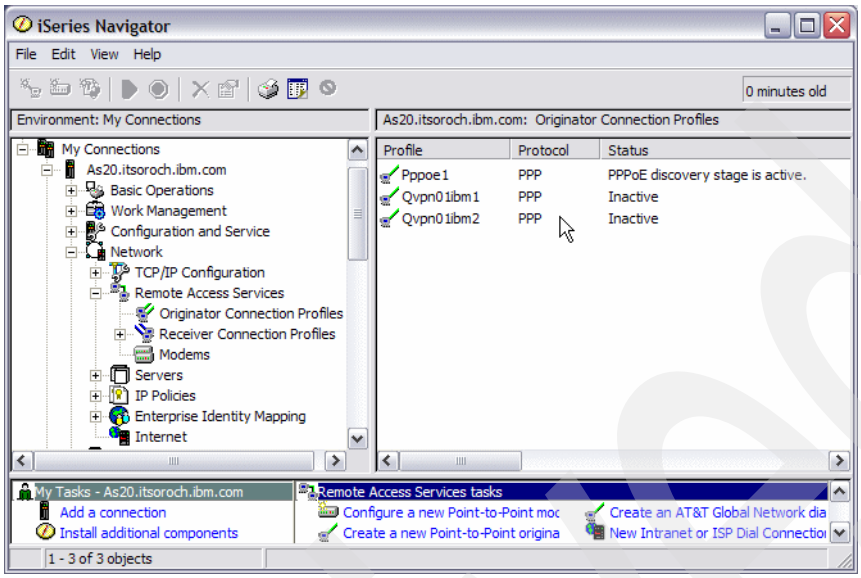


Figure 17-15 iSeries Navigator window: Originator Connection Profiles: PPPoE1 activating

3. If the connection does not become active, on the iSeries Navigator window right-click **PPPOE1** and choose **Connections**, as shown in Figure 17-16.

> **Note:** In releases prior to V5R4, instead of selecting Connections you would review the errors by selecting **Jobs**, finding the appropriate job, and reviewing its **Job Log**.

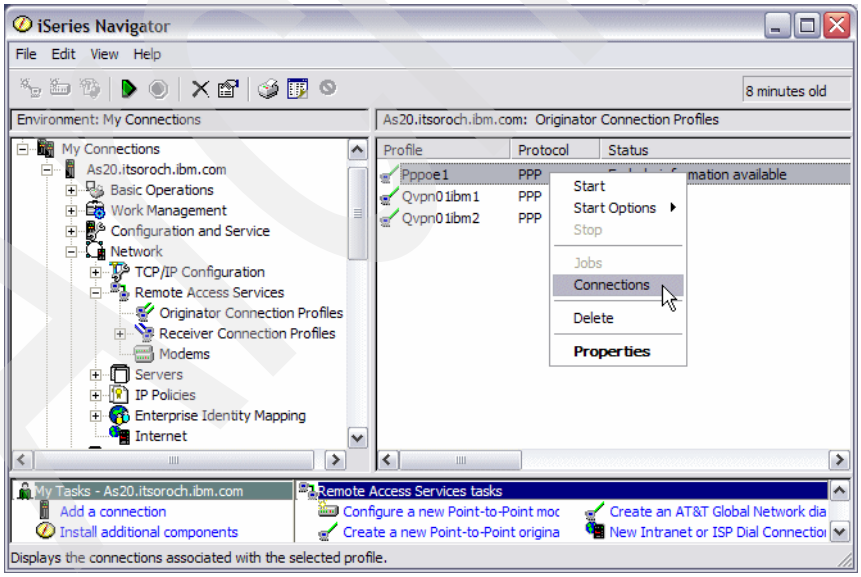


Figure 17-16 iSeries Navigator window: Originator Connection Profiles: PPPoE1 Connections

4. In the Pppoe1 Connections window, select the connection whose status shows an error and click **Message Log**, as shown in Figure 17-17.

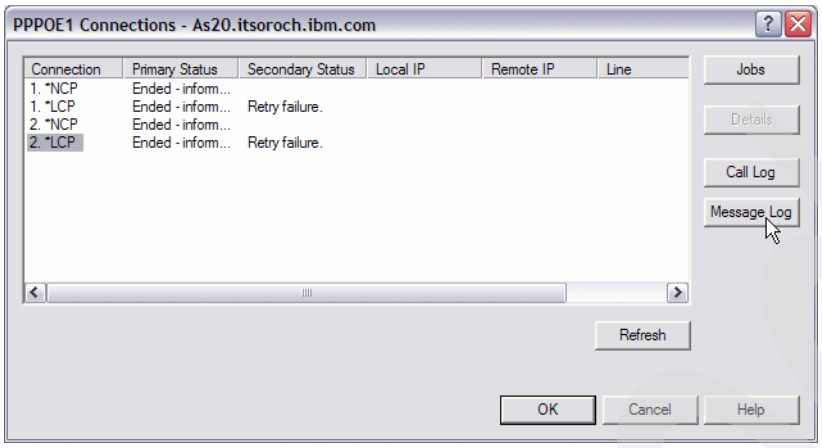


Figure 17-17 Pppoe1 Connections window

Figure 17-18 shows a failing scenario when attempting to start the PPPoE1 originator connection profile. If you are not able to determine the problem based on the information shown in the message log, contact IBM Support for further assistance. Here is a short description of some of the messages you should receive in a successful connection:

- TCP82A1 PPPoE peer discovery complete

  This means that the PPPoE discovery state (Ethertype=8863) was completed.

- TCP8342 TCP/IP point-to-point interface 20.20.20.22 added

  This means that IP address 20.20.20.22 was assigned from the ISP to the System i.

- TCP8346 TCP/IP point-to-point route to destination 23.24.25.26 added

  This means that IP address 23.24.25.26 was assigned to the ISP side of the connection.

- TCP8344 TCP/IP point-to-point interface 20.20.20.22 started

  This means that point-to-point interface 20.20.20.22 was successfully started.

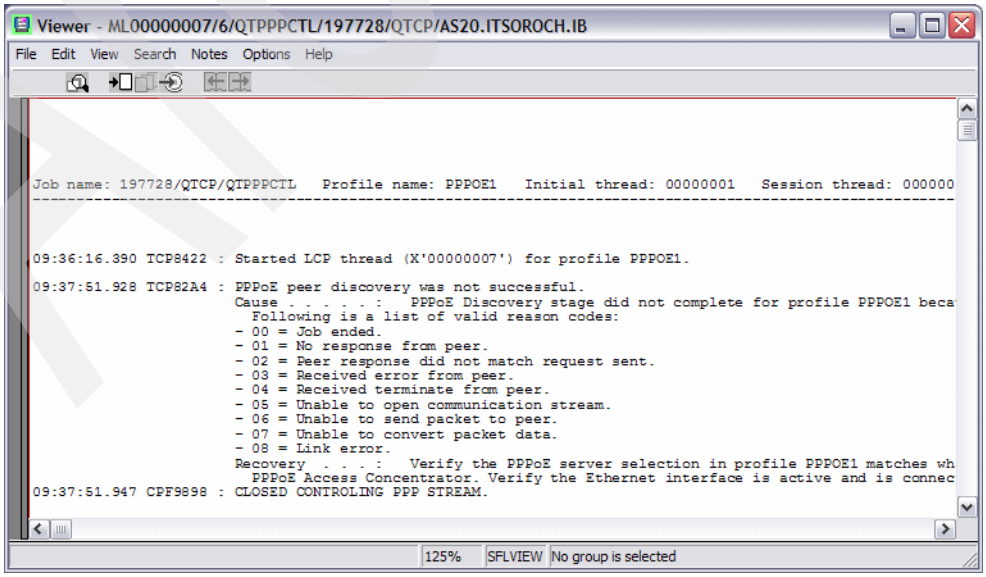


Figure 17-18 Message Log window

5. If the connection is successful, an additional test can be used to verify connectivity. PING from AS20 to AS24.

> **Note:** In this scenario, it is understood that the PPPoE definition on the AS24 side is already created, and fixed address 20.20.20.21 is assigned. If you have not created the PPPoE definition on the AS24 side, create the definition using the procedure outlined in “Step 1: Creating the PPPoE definition on AS20” on page 524.
>
> Make the AS24 side of the connection active and then return to this point to test the end-to-end connection.

On the AS20 command entry window, enter (as shown in Figure 17-19):

```
PING RMTSYS('20.20.20.21') LCLINTNETA('20.20.20.22')
```

If you successfully receive all PING replies, proceed to “Step 3: Create VPN connection on the PPPoE definition on AS20” on page 534. If there is a connection problem between AS20 and AS24, fix the problem before proceeding to the next step.

```
Previous commands and messages:
   > PING RMTSYS('20.20.20.21') LCLINTNETA('20.20.20.22')
     Verifying connection to host system 20.20.20.21.
     PING reply 1 from 20.20.20.21 took 24 ms. 256 bytes. TTL 63.
     PING reply 2 from 20.20.20.21 took 18 ms. 256 bytes. TTL 63.
     PING reply 3 from 20.20.20.21 took 20 ms. 256 bytes. TTL 63.
     PING reply 4 from 20.20.20.21 took 20 ms. 256 bytes. TTL 63.
     PING reply 5 from 20.20.20.21 took 20 ms. 256 bytes. TTL 63.
     Round-trip (in milliseconds) min/avg/max = 18/20/24.
     Connection verification statistics: 5 of 5 successful (100 %).

                                                                    Bottom
Type command, press Enter.
===>
```

Figure 17-19 PING command to verify connection between AS20 and AS24

Step 3: Create VPN connection on the PPPoE definition on AS20

In this step, we create a VPN connection on AS20:

1. In the iSeries Navigator window, expand **Network** → **IP Policies** → **Virtual Private Networking** → **IP Security Policies**. Right-click **Internet key Exchange policies** and choose **New Internet Key Exchange Policies**, as shown in Figure 17-20.

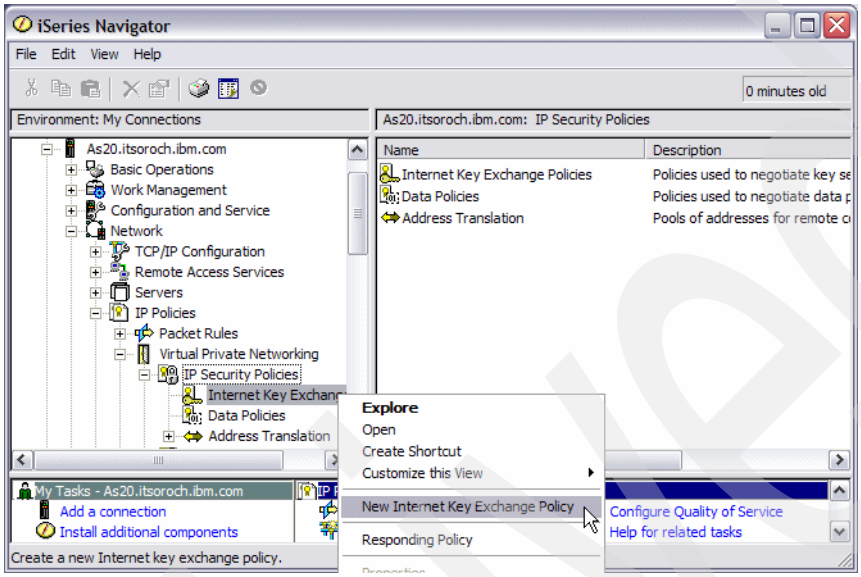


Figure 17-20 iSeries Navigator window: New Internet Key Exchange Policy

2. In the New Internet Key Exchange Policy window, enter `20.20.20.21` in the IP Address field (answer 15 in Table 17-1 on page 522), as shown in Figure 17-21.

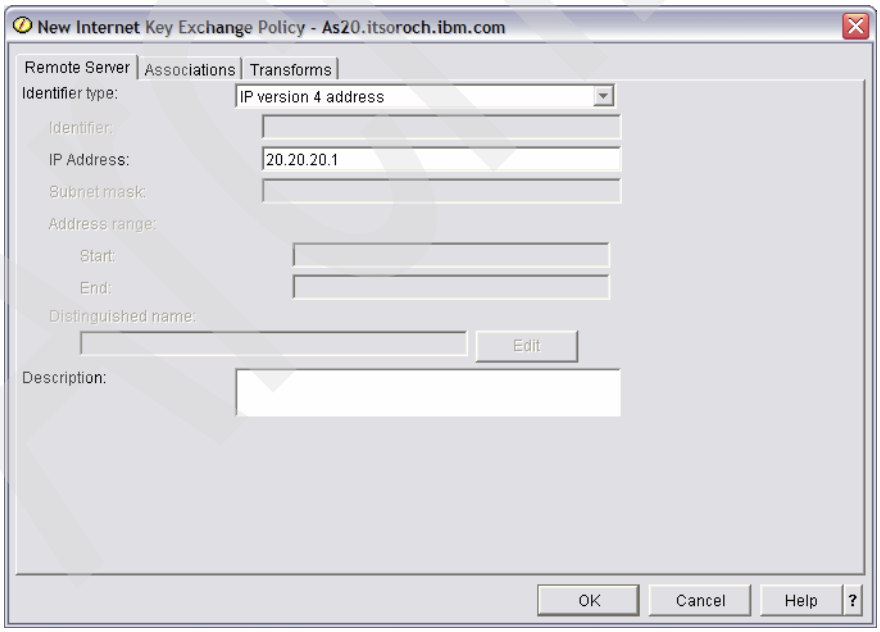


Figure 17-21 New Internet Key Exchange Policy window

3. Click the **Associations** tab. Check **Preshared key**, and enter `makoto` in the Key field (answer 9 in Table 17-1 on page 522). Enter `20.20.20.22` in the Local key server IP Address field (answer 14 in Table 17-1 on page 522), as shown in Figure 17-22.

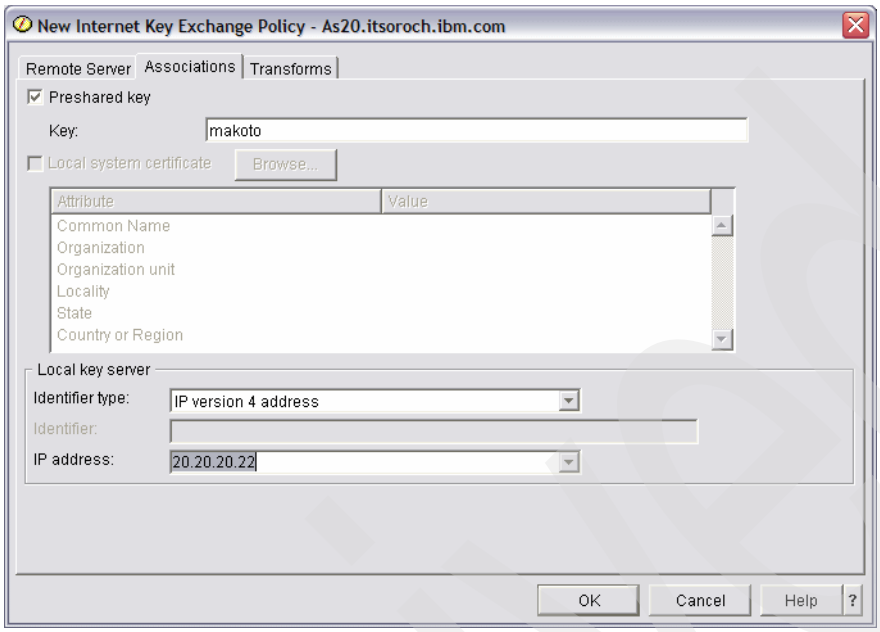


Figure 17-22 New Internet Key Exchange Policy window

4. Click the **Transforms** tab. Select **IKE main mode negotiation** (answer 12 in Table 17-1 on page 522), as shown in Figure 17-23. Click **Add** to continue.

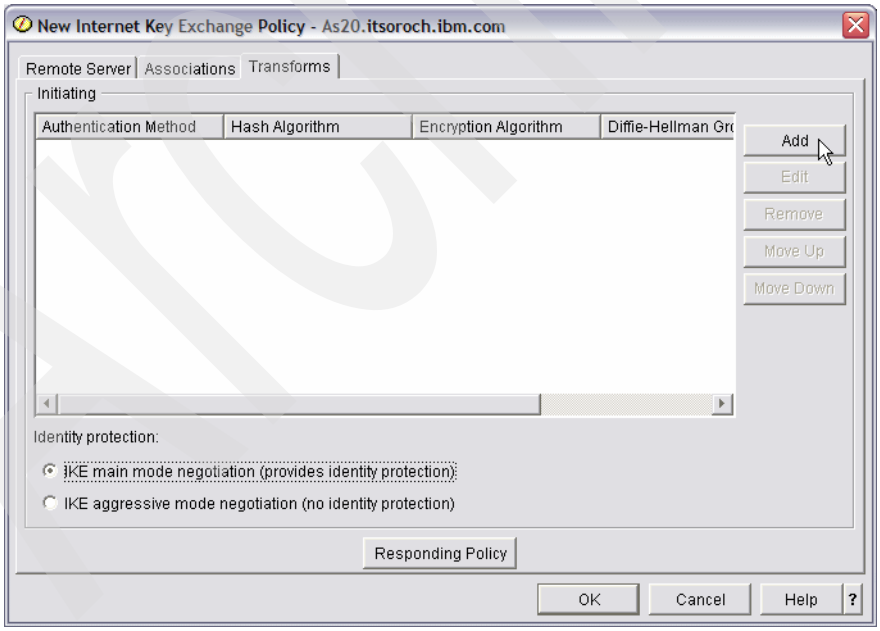


Figure 17-23 New Internet Key Exchange Policy window: Transforms tab

5. This opens the Internet Key Exchange Policy Transform window. From the pull-down menus, make these choices, as shown in Figure 17-24 on page 536:

Authentication method	Preshared key.
Hash algorithm	MD5 (answer 11 in Table 17-1 on page 522).

Encryption algorithm	DES-CBC (answer 10 in Table 17-1 on page 522).
Diffie-Hellman group	Group1 (default 768 bit MODP).
Expire IKE keys after	Enter 2, choose **hours** (answer 11 in Table 17-1 on page 522).

Click **OK** to continue.

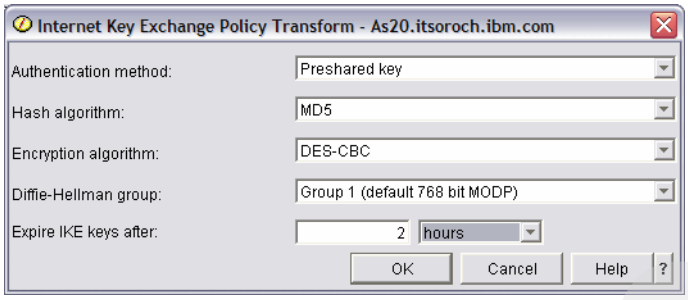


Figure 17-24 Internet Key Exchange Policy Transform window

6. Returning to the New Internet Key Exchange Policy window (Figure 17-25), click **Responding Policy**.

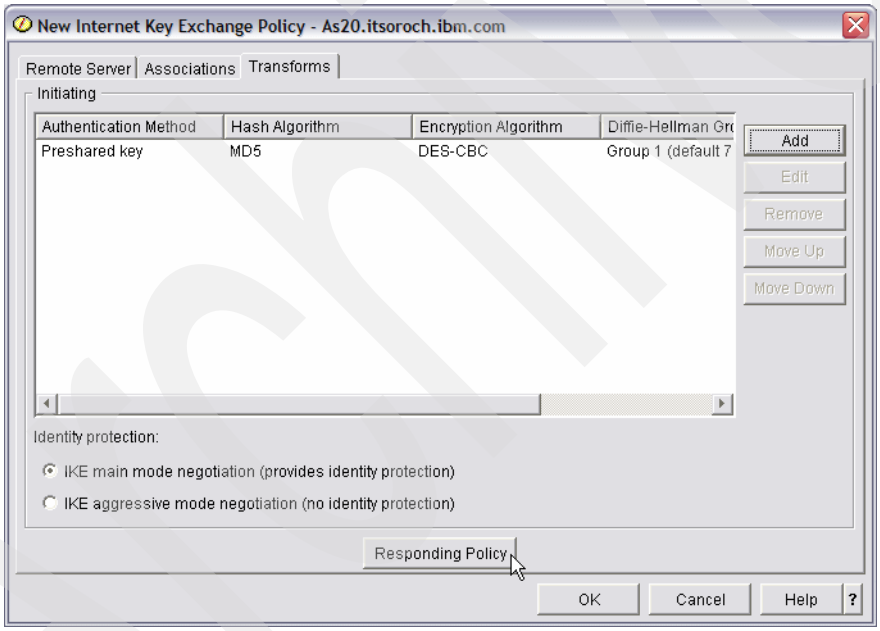


Figure 17-25 New Internet Key Exchange Policy window

7. In the Responding Internet Key Exchange Policy window, enter 2 and select **hours** in the Expire IKE keys after field (answer 11 in Table 17-1 on page 522). Click **OK** to continue.

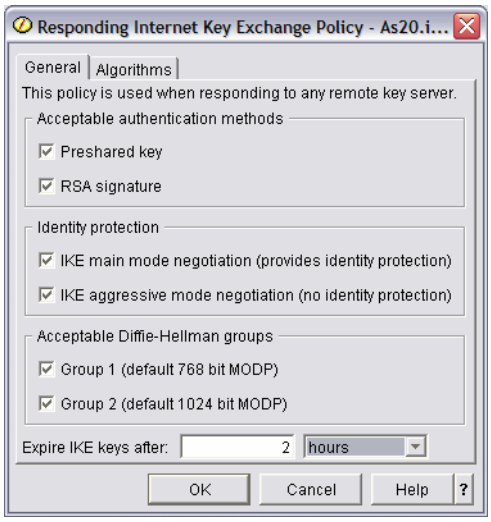


Figure 17-26 Responding Internet Key Exchange Policy window

8. In the New Internet Key Exchange policy window, click **OK**.
9. In the iSeries Navigator window, right-click **Data Policies** and choose **New Data Policy**, as shown in Figure 17-27.

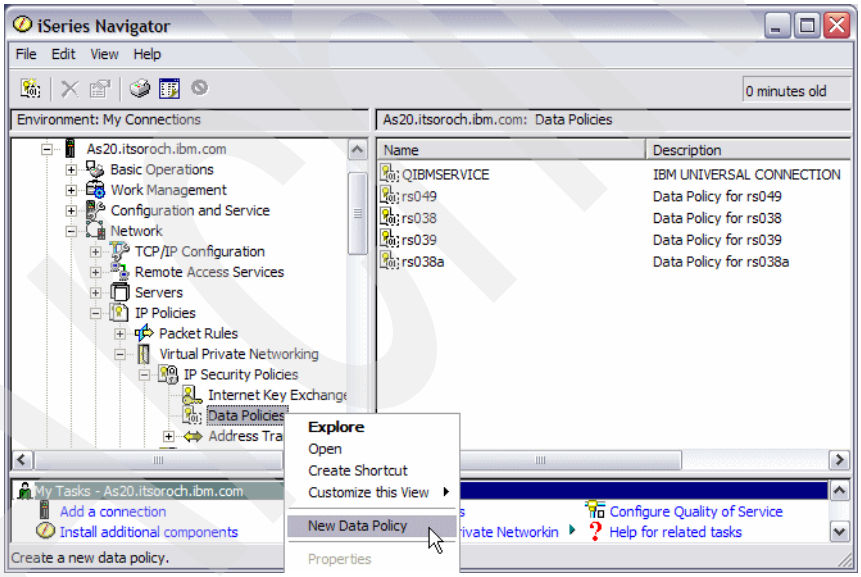


Figure 17-27 iSeries Navigator window

10. In the New Data Policy window, enter AS20 in the Name field. Make sure that **Group 1 (768 bit MODP)** is selected for the Diffie-Hellman group (answer 11 in Table 17-1 on page 522), as shown in Figure 17-28. Click the **Proposals** tab.

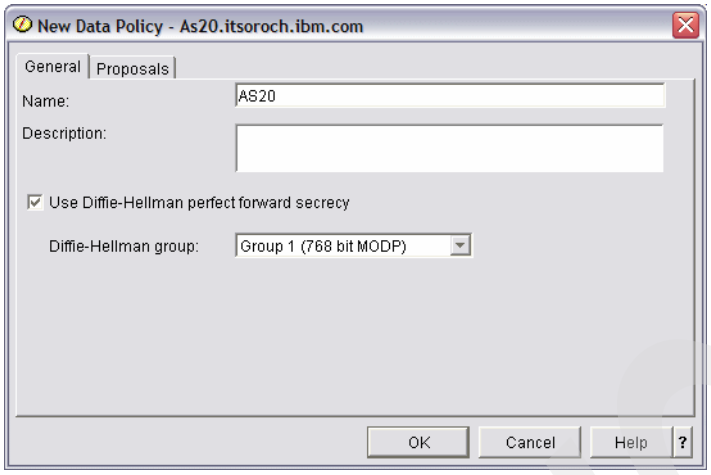


Figure 17-28 New Data Policy window

11. In the New Data Policy window click **Add**, as shown in Figure 17-29.

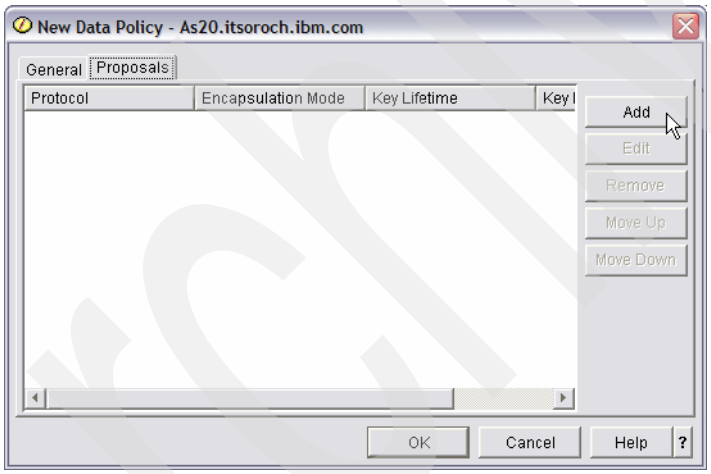


Figure 17-29 New Data Policy window

12. In the New Data Policy Proposal window, keep the default values shown in Figure 17-30.

> **Tip:** You could also set the Encapsulation mode to Transport because we are doing a host-to-host configuration. This, in fact, might be a preferred way to configure this host-to-host connection. Of course, if you use the Encapsulation mode of Transport, ensure that the same encapsulation mode that you select here is also selected on the remote system AS24.

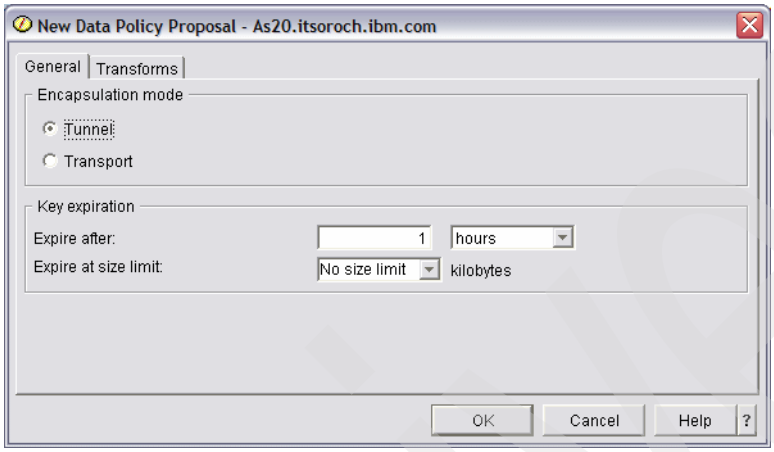


Figure 17-30 New Data Policy Proposal window

13. Click the **Transforms** tab. In the Transforms window, click **Add**.
14. In the Data Policy Transform window, make the following selections, as shown in Figure 17-31:

    Protocol Encapsulating Security Payload (ESP)
    Authentication algorithm MD5
    Encryption algorithm DES-CBC (answer 10 in Table 17-1 on page 522)

    Click **OK** to continue.

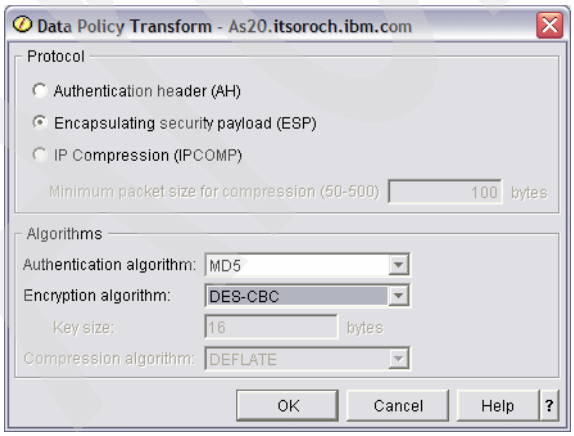


Figure 17-31 Data Policy Transform window

15. In the New Data Policy Proposal window, click **OK**.
16. In the New Data Policy window, click **OK**.

17. In the iSeries Navigator window, right-click **Virtual Private Networking** and choose **New Connection**, as shown in Figure 17-32.

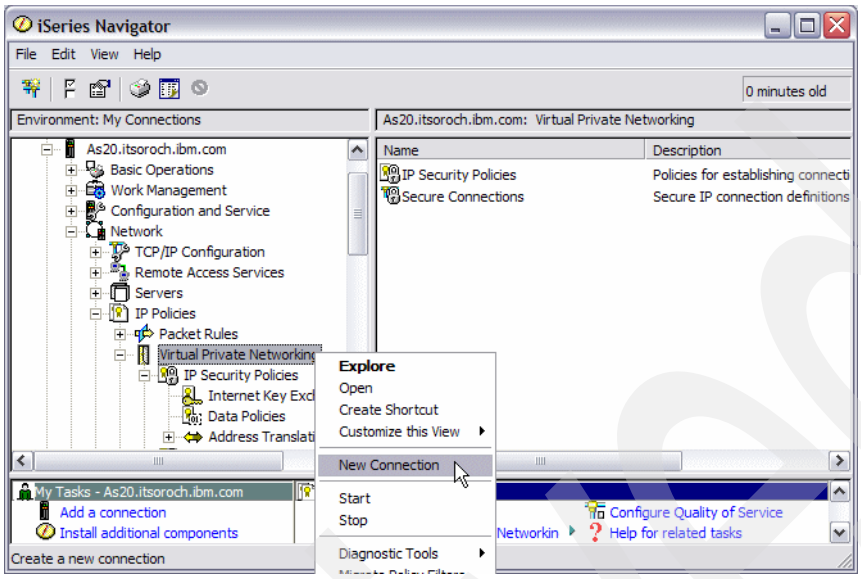


Figure 17-32 iSeries Navigator window

18. This opens the New Connection Wizard (Figure 17-33). Click **Next**.

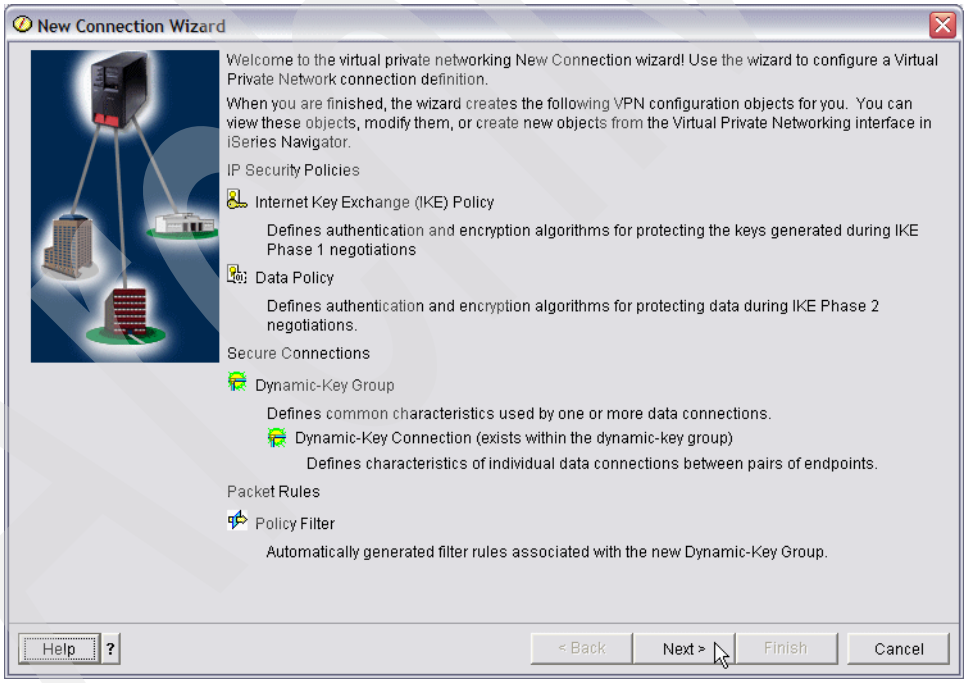


Figure 17-33 New Connection wizard

19. In the Connection Name window (Figure 17-34), enter AS20 in the Name field. Click **Next**.

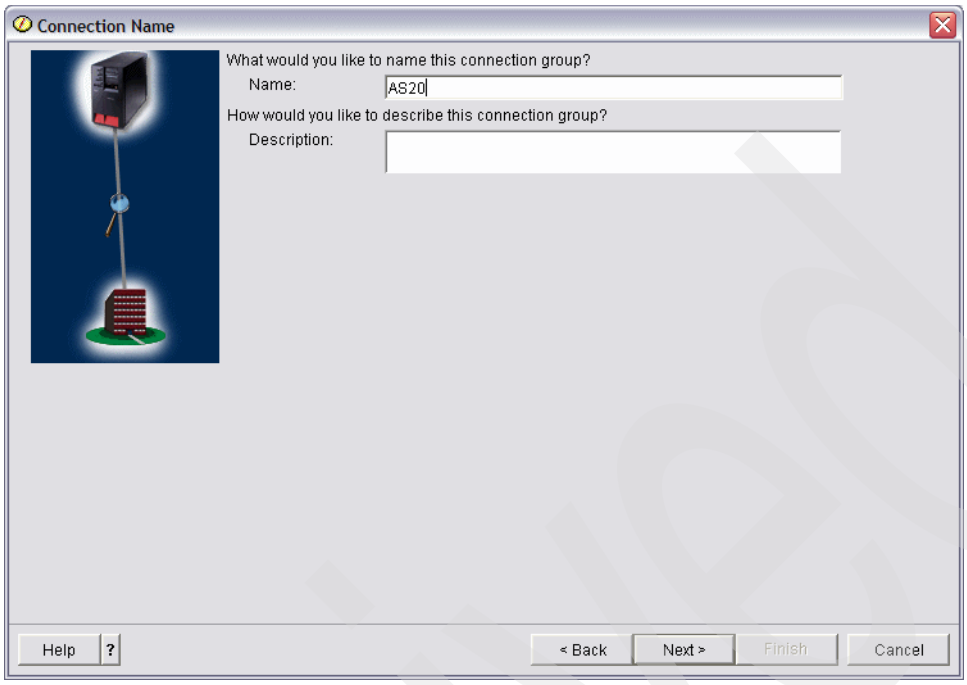


Figure 17-34 Connection Name window

20. In the Connection Scenario window, select **Connect your host to another host**, as shown in Figure 17-35. Click **Next** to continue.

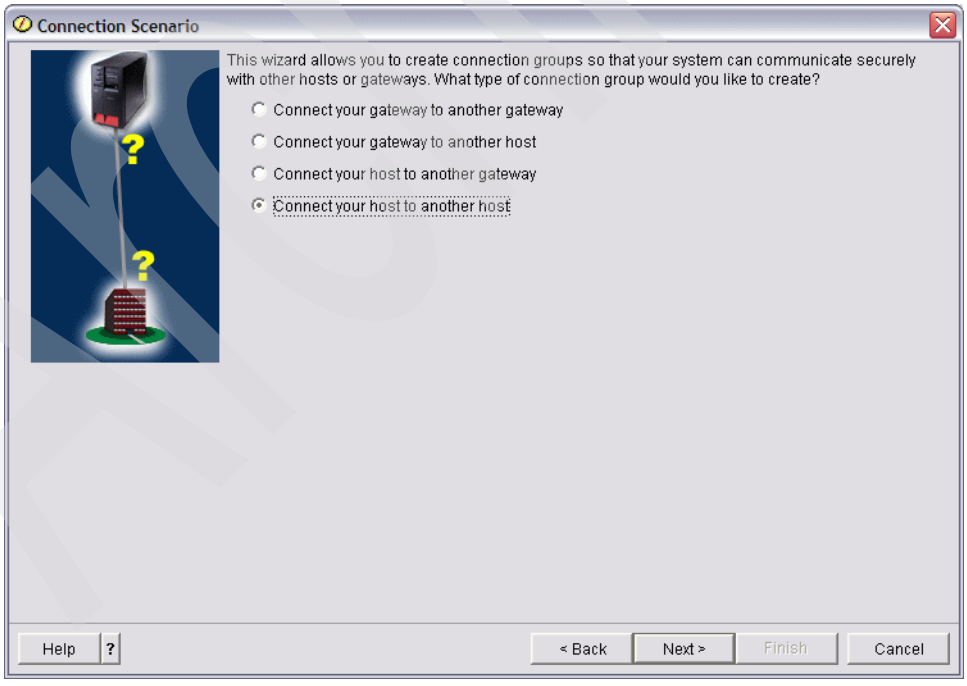


Figure 17-35 Connection scenario window

21. In the Internet Key Exchange Policy window, choose **20.20.20.21** as the Policy (answer 15 in Table 17-1 on page 522), as shown in Figure 17-36. Click **Next** to continue.

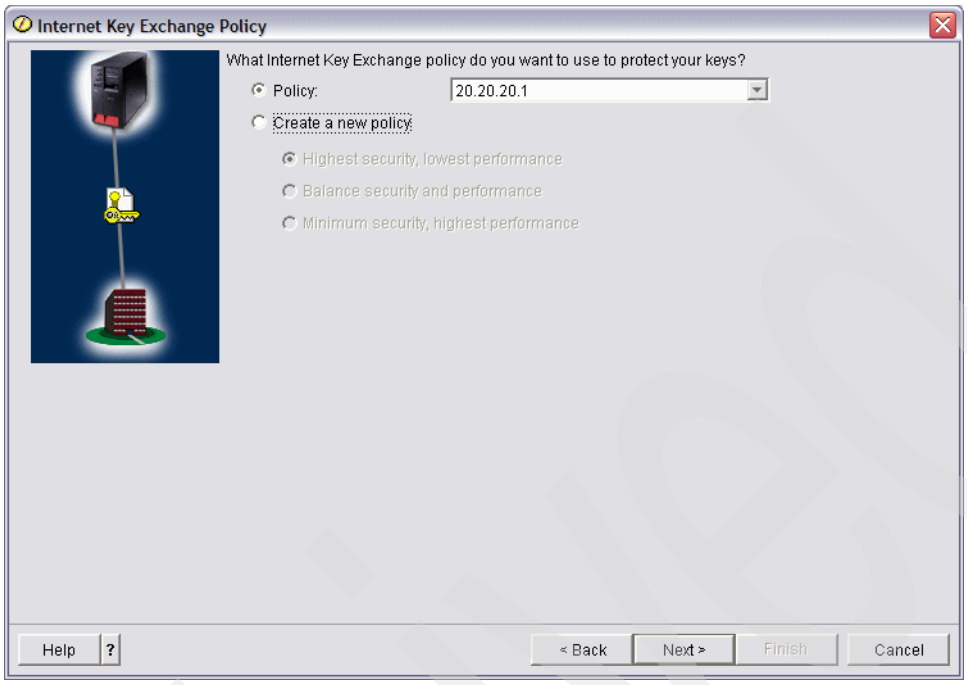


Figure 17-36 Internet Key Exchange Policy window

22. In the Data Services window, keep the default values shown in Figure 17-37. Click **Next**.

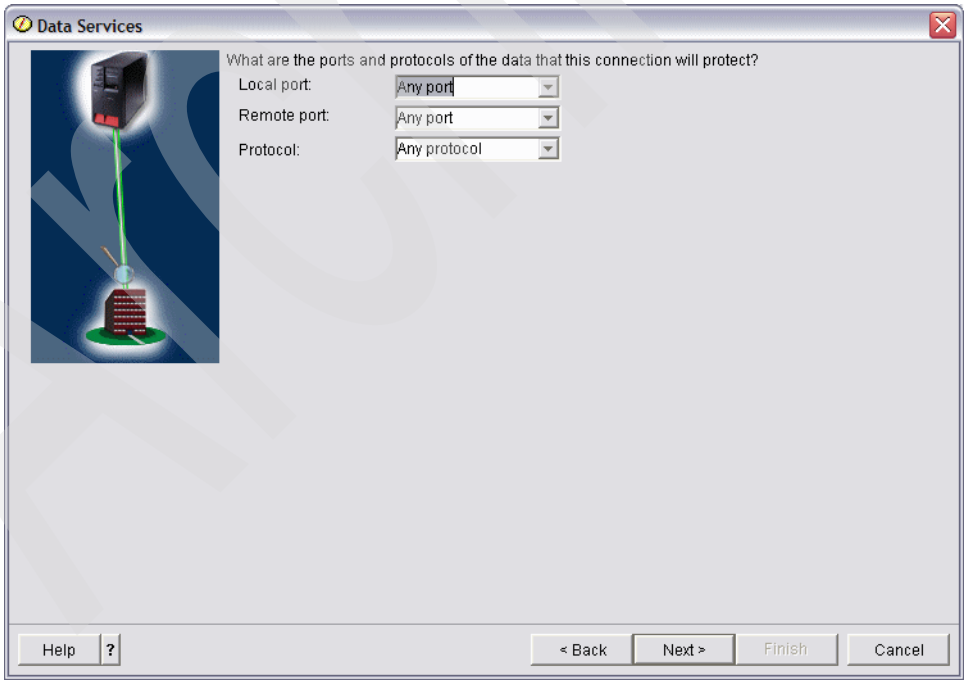


Figure 17-37 Data Services window

23. In the Data Policy window, choose **AS20** for the Policy name, as shown in Figure 17-38. Click **Next** to continue.

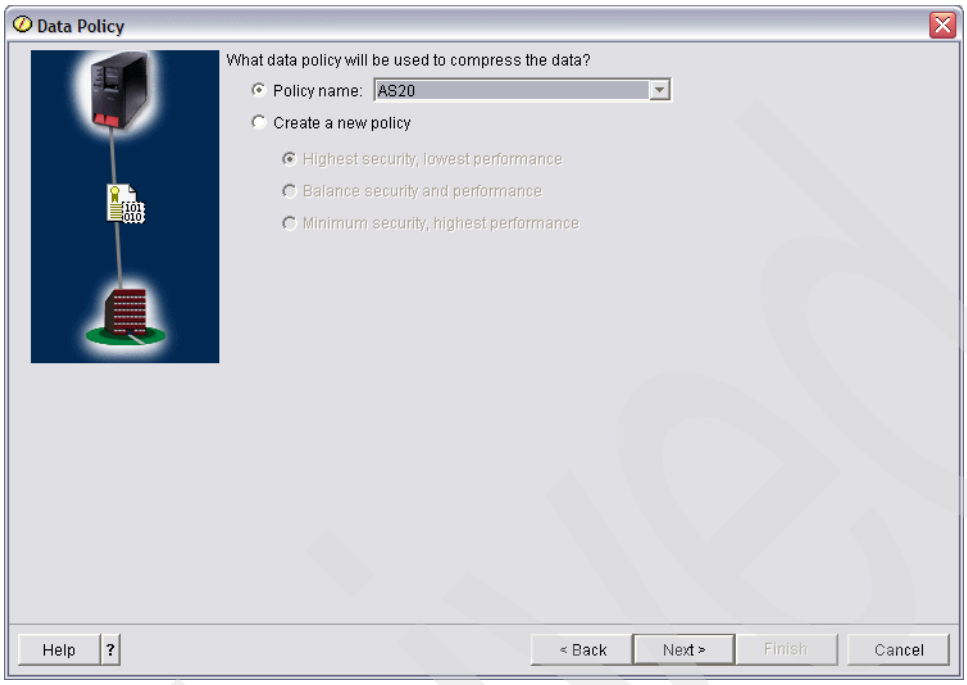


Figure 17-38 Data Policy window

24. In the Require Policy Filter window (Figure 17-39), select **This connection requires a policy filter**. Click **Next** to continue.

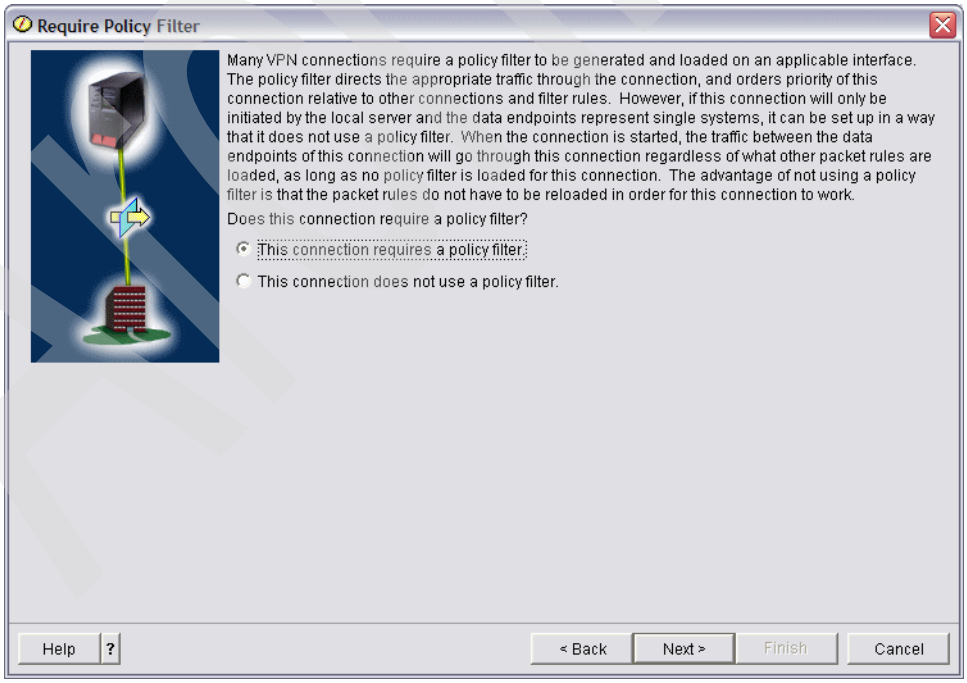


Figure 17-39 Require Policy filter window

25. In the Applicable Interfaces window, check **PPPOE1** (answer 6 in Table 17-1 on page 522), as shown in Figure 17-40. Click **Next** to continue.

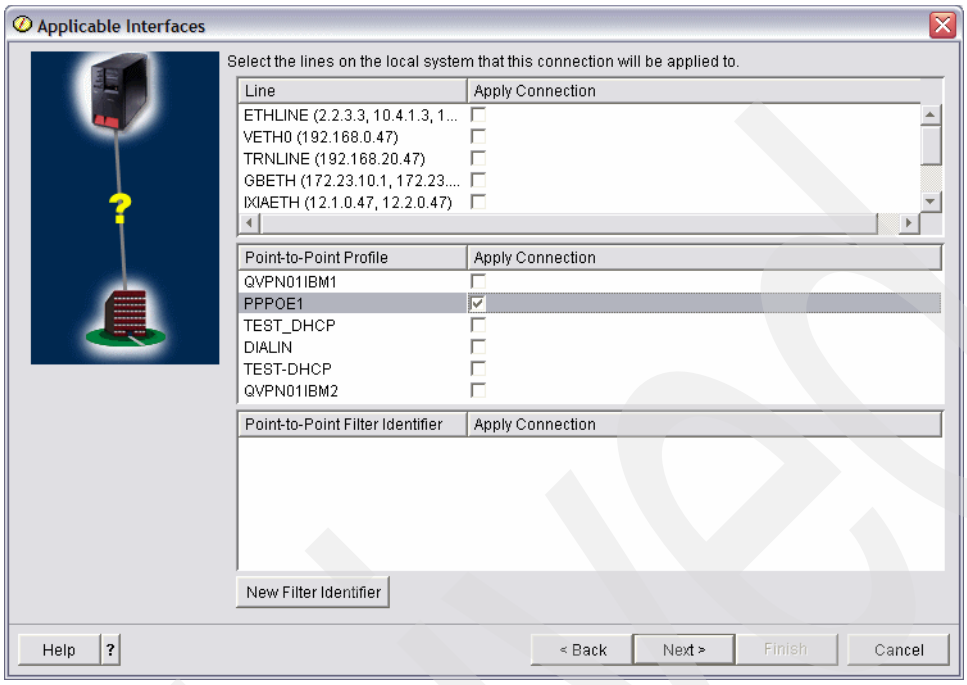


Figure 17-40 Applicable Interfaces window

26. In the New Connection Summary window click **Finish**, as shown in Figure 17-41.

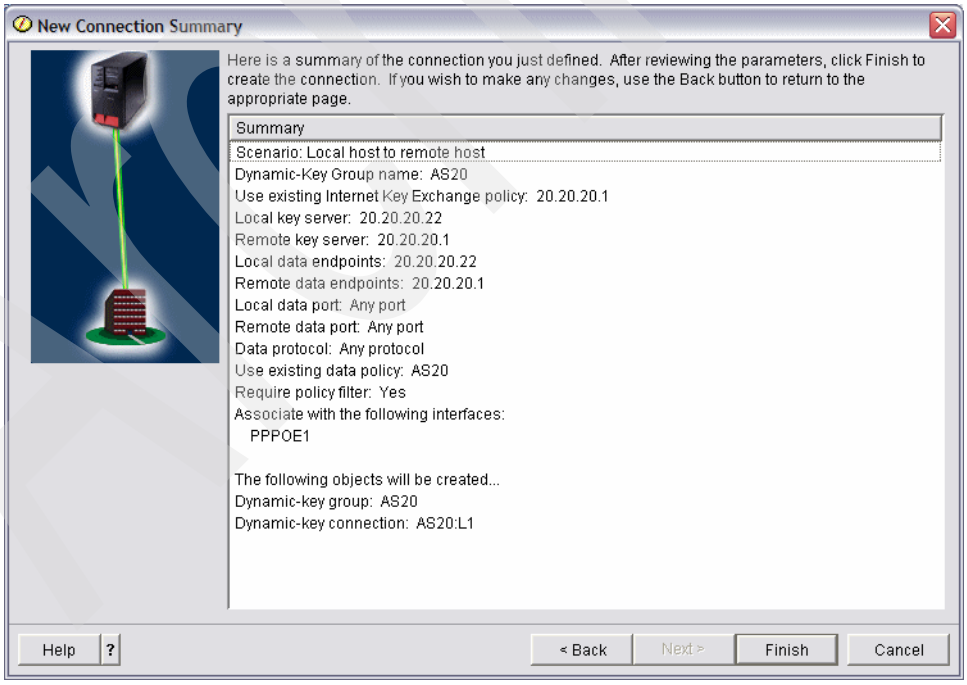


Figure 17-41 New Connection Summary window

27. The Activate Policy Filters window opens. If you want to activate the IP filter on PPP definition PPPOE1, select **Yes, activate the generated policy filters** and **Permit all other traffic**, as shown in Figure 17-42.

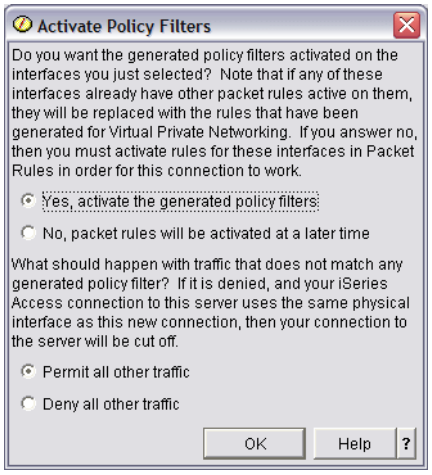


Figure 17-42 Activate Policy filters window

Tip: If you prefer not to activate IP filter now, we have provided some procedures to activate the IP filter manually in “Manually activate the IP filter” on page 546. The IP filter is required to activate the VPN connection. If you try to activate the VPN definition without activating the corresponding IP filter (the one that is created for AS20 VPN definition using PPPOE1 PPP definition), you will receive an error message that the IP policy filter is not activated.

Also, in some cases you might need to edit the packet rules. If so, see “Customizing the packet rules” on page 549.

Manually activate the IP filter

This is the procedure for activating the IP filter manually. The IP filter for VPN connection is created through iSeries Navigator:

1. To open the VPN filter, expand **IP Policies** and right-click **Packet Rules** and choose **Rules Editor**, as shown in Figure 17-43.

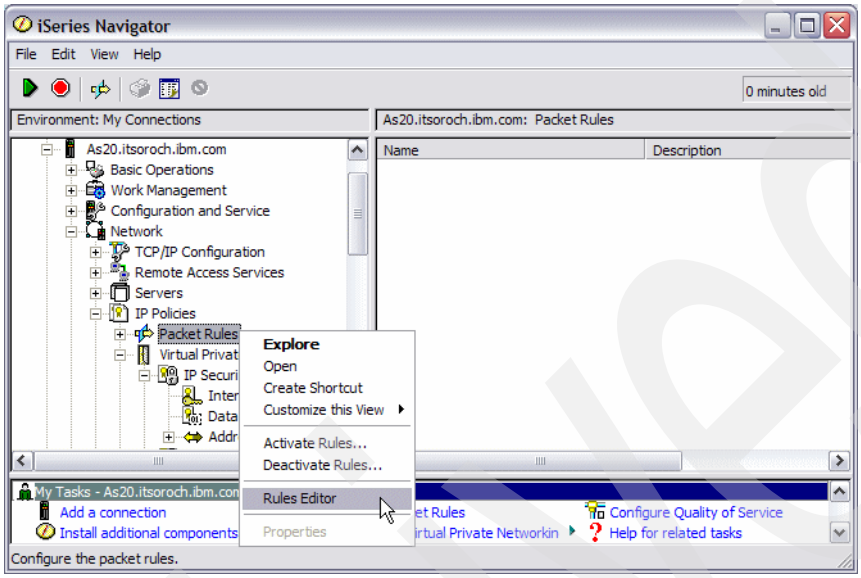


Figure 17-43 iSeries Navigator window

2. In the Welcome Packet Rules Configuration window, select **Open an existing packet rules file**, as shown in Figure 17-44. Click **OK** to open the Open File window.

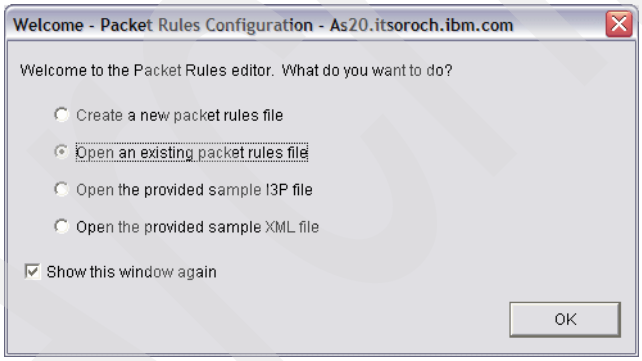


Figure 17-44 Welcome Packet Rules Configuration window

3. Choose **QIBM/UserData/OS400/TCPIP/OPNAVRULES/VPNPOLICYFILTERS.I3P** as shown in Figure 17-45. Click **OK** on the Important pop-up to hide it. Click **OK** on the Getting Started pop-up to hide it.

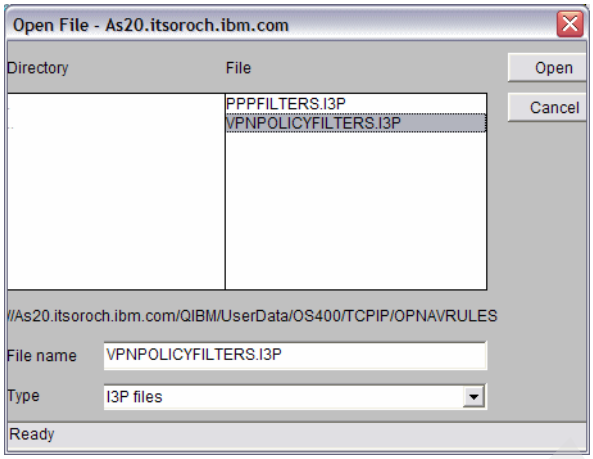


Figure 17-45 Open file window

4. Figure 17-46 shows the sample IP filters on the AS20 side. This IP Filter file is created by iSeries Navigator. This filter file is not permitted to have the contents altered. If you prefer creating custom filters yourself, refer to “Customizing the packet rules” on page 549.

```
FILTER SET PreIPsecPermitAllIKE   ACTION = PERMIT   DIRECTION = *   SRCADDR = *   DSTADDR = *   PROTOCOL
= UDP   DSTPORT = 500   SRCPORT = 500   JRN = OFF    FRAGMENTS = NONE
FILTER SET OPNAVPermitNonVPN   ACTION = PERMIT   DIRECTION = *   SRCADDR = *   DSTADDR = *   PROTOCOL =
*   DSTPORT = *   SRCPORT = *   JRN = OFF    FRAGMENTS = NONE
FILTER SET OPNAV2   ACTION = IPSEC   DIRECTION = OUTBOUND   SRCADDR = 20.20.20.22   DSTADDR = 20.20.20.1
PROTOCOL = *   DSTPORT = *   SRCPORT = *   CONNECTION_DEFINITION = AS20   JRN = OFF
FILTER_INTERFACE   INTERFACE = PPPOE1   SET = PreIPsecPermitAllIKE, OPNAV2, OPNAVPermitNonVPN
INCLUDE   FILE = /QIBM/UserData/OS400/TCPIP/OPNAVRULES/PPPFILTERS.I3P
```

Figure 17-46 Sample IP Filter AS20

Here are the meanings of each FILTER SET sentence:

– FILTER SET PreIPsecPermitAllIKE

  This allows IKE packets (UDP Port 500) for any destination IP addresses and any source IP addresses. IKE (Internet Key Exchange protocol) is required for the negotiation to exchange hashed keys before starting ESP-encrypted communication.

– FILTER SET OPNAVPermitNonVPN

  This allows non-VPN traffic to go through on the PPPOE1 interface. Without having this sentence, all non-VPN traffic on PPPOE1 interface would be cut off.

– FILTER SET OPNAV2

  This is the specified sentence for VPN connection. IPSEC means that this filter allows IPSEC-related protocols. If you are using ESP protocol, this filter allows the IP packets for which IP protocol number is 52 (ESP protocol).

– FILTER_INTERFACE INTERFACE = PPPOE1

  This sentence specifies that the PPPOE1 PPP definition has three filter sets being activated: PreIPsecPermitAllIKE, OPNAV2, and OPNAVPermitNonVPN.

– INCLUDE FILE

  This sentence is always seen on the bottom of the IP filter sets. It is used to include the other filter file PPPFILTERS.I3P.

5. To activate the filter manually, select **File** → **Activate Rules**, as shown in Figure 17-47.

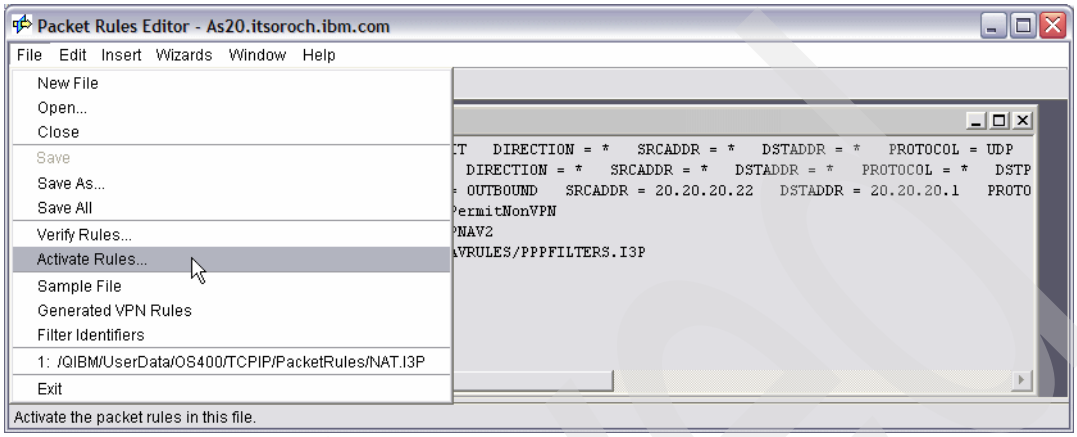


Figure 17-47 Activating rules

6. In the Activate Packet Rules window, choose **PPPOE1** (answer 6 in Table 17-1 on page 522) as the Interfaces selection, as shown in Figure 17-48. Click **OK** to continue. If the Packet Rule is activated, you will see the message `The rules were successfully activated` in the bottom pane of the window.

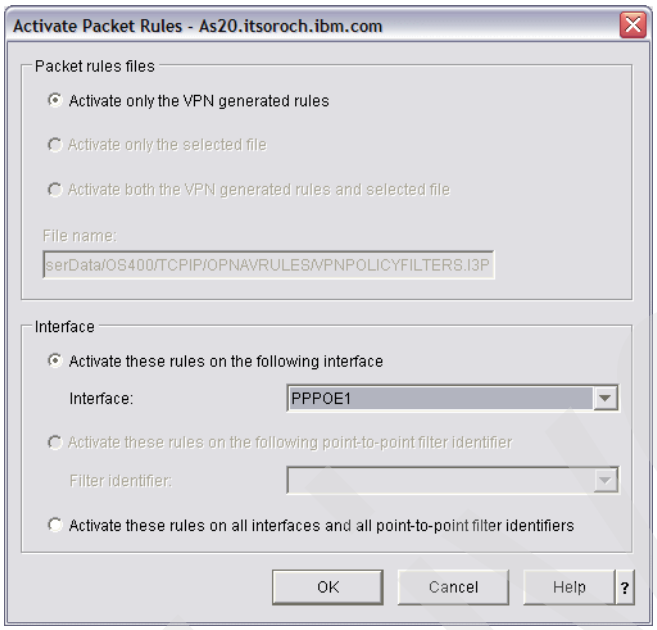


Figure 17-48 Activate Packet Rules window

Customizing the packet rules

This is the procedure for creating the custom packet rule. If for some reason you need to alter the packet rule contents, use this procedure:

1. Copy the packet filter rule sentences shown in Figure 17-49. You will paste these sentences in the next step.

```
FILTER SET PreIPsecPermitAllIKE   ACTION = PERMIT   DIRECTION = *   SRCADDR = *   DSTADDR = *   PROTOCOL
= UDP   DSTPORT = 500   SRCPORT = 500   JRN = OFF    FRAGMENTS = NONE

FILTER SET OPNAVPermitNonVPN   ACTION = PERMIT   DIRECTION = *   SRCADDR = *   DSTADDR = *   PROTOCOL =
*   DSTPORT = *   SRCPORT = *   JRN = OFF    FRAGMENTS = NONE

FILTER SET OPNAV2 ACTION = IPSEC   DIRECTION = OUTBOUND   SRCADDR = 20.20.20.22   DSTADDR = 20.20.20.21
PROTOCOL = *   DSTPORT = *   SRCPORT = *   CONNECTION_DEFINITION = AS20   JRN = OFF

FILTER_INTERFACE   INTERFACE = PPPOE1   SET = PreIPsecPermitAllIKE, OPNAV2, OPNAVPermitNonVPN

INCLUDE   FILE = /QIBM/UserData/OS400/TCPIP/OPNAVRULES/PPPFILTERS.I3P
```

Figure 17-49 Sample IP Filter AS20

2. In the Packet Rules Editor window, select **File** → **New file**. Select **Edit** → **Paste**. Your new file should resemble Figure 17-50.

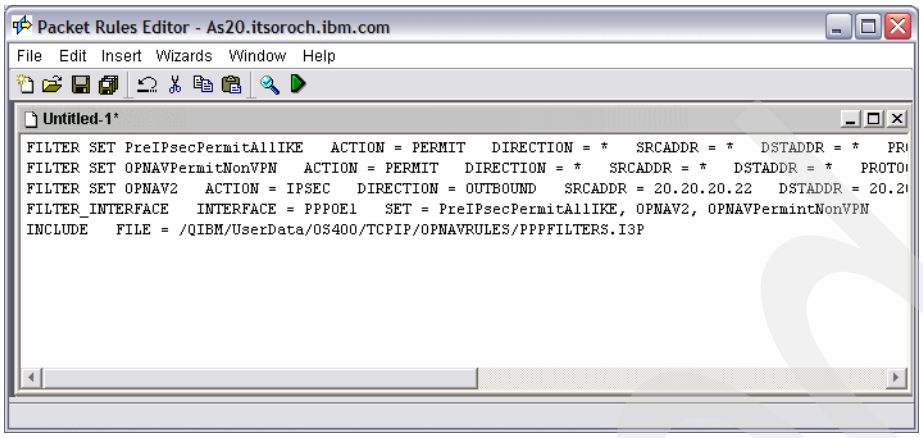


Figure 17-50 Packet Rules Editor window

3. In the Packet Rules Editor window choose **File** → **Verify Rules**, as shown in Figure 17-51. Click **Yes** on the Save Required window.

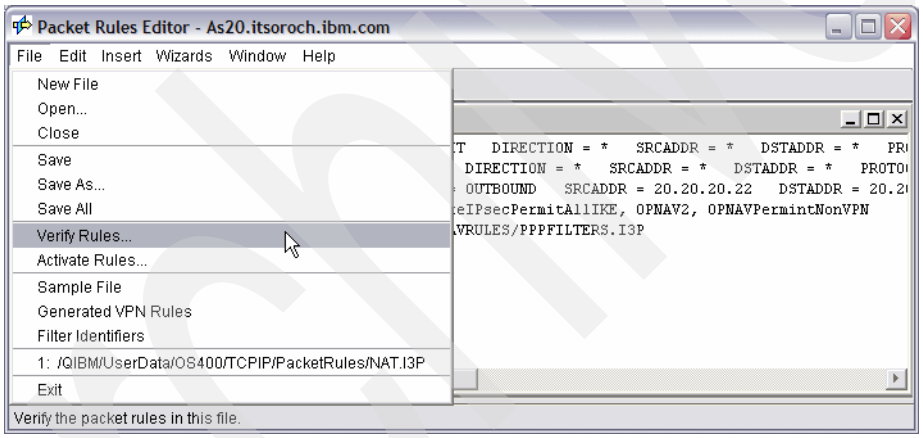


Figure 17-51 Packet Rules Editor window

4. In the Save File window, enter the filter file name `vpnas20`. Make sure that vpnas20.i3p saves under PacketRules directory, as shown in Figure 17-52. All custom filter rules must be saved under the /QIBM/UserData/OS400/TCPIP/PacketRules directory.

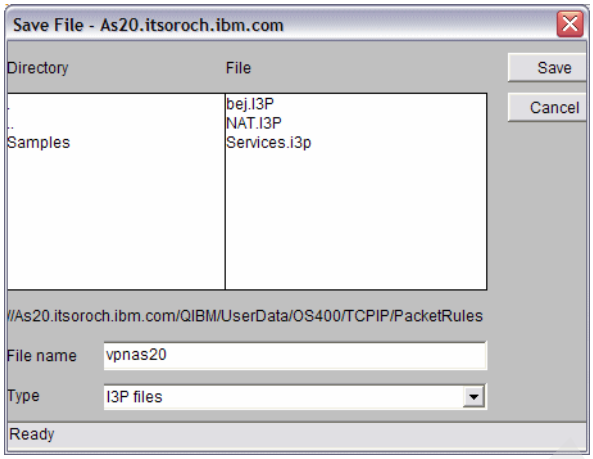


Figure 17-52 Save Files window

5. In the Verify Packet Rules window, select **Verify only the selected file** and **Verify these rules on the following interface**, and choose **PPPOE1** (answer 6 in Table 17-1 on page 522), as shown in Figure 17-53. Click **OK** to continue.

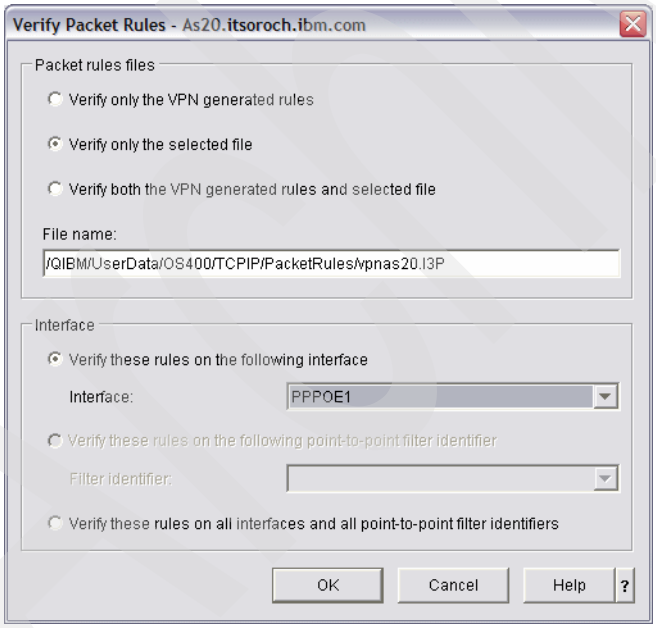


Figure 17-53 Verify Packet Rules window

6. If the Packet Rule is successfully verified, you will see the message `The rules were successfully verified` on the bottom pane as shown in Figure 17-54. If you receive an error, read the error message and fix it before you proceed. Select **File** → **Activate Rules**.

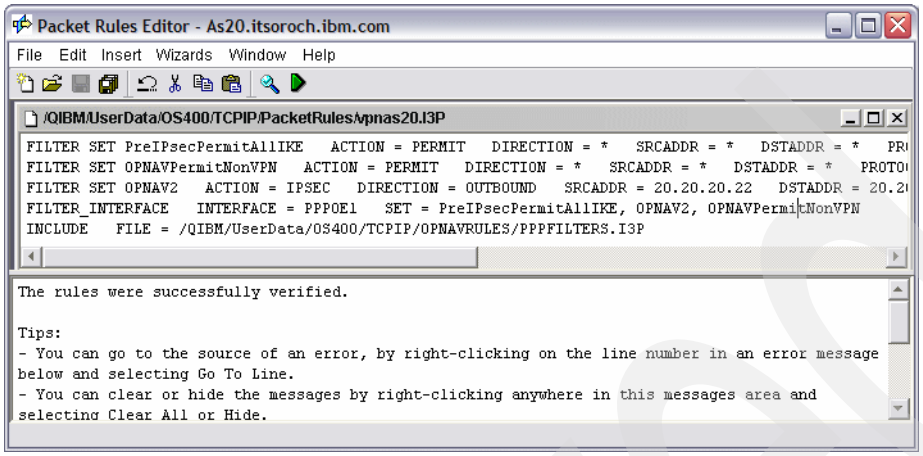


Figure 17-54 Packet Rules Editor

7. In the Activate Packet Rules window, select **Activate only selected file** and **Activate these rules on the following interface**, and select **PPPOE1** (answer 6 in Table 17-1 on page 522), as shown in Figure 17-55. Click **OK** to continue.

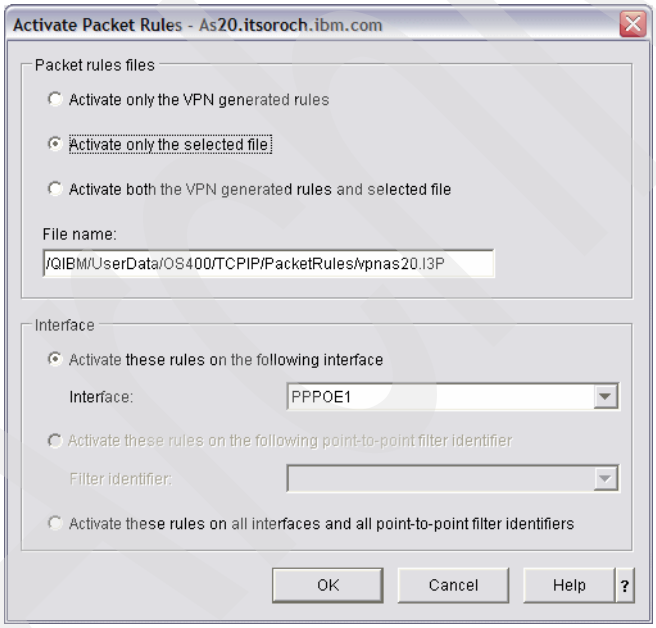


Figure 17-55 Activate Packet Rules window

8. If the Packet Rule is activated successfully, you will see the message `The rules were successfully activated` in the bottom pane, as shown in Figure 17-56. If you receive an error, read the error message and fix it before you proceed. Now you are ready to start VPN definition AS20.

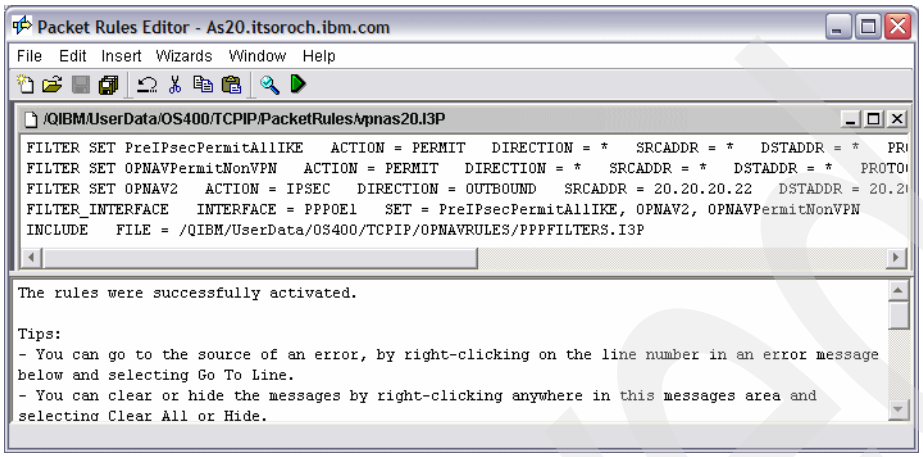


Figure 17-56 Packet Rules Editor window

Step 4: Create VPN connection on the PPPoE definition on AS24

In this step, you are going to create a VPN connection on AS24:

1. In the iSeries Navigator window, expand **Network** → **IP Policies** → **Virtual Private Networking** → **IP Security Policies**. Right-click **Internet key Exchange Policies** and choose **New Internet Key Exchange Policy**, as shown in Figure 17-57.

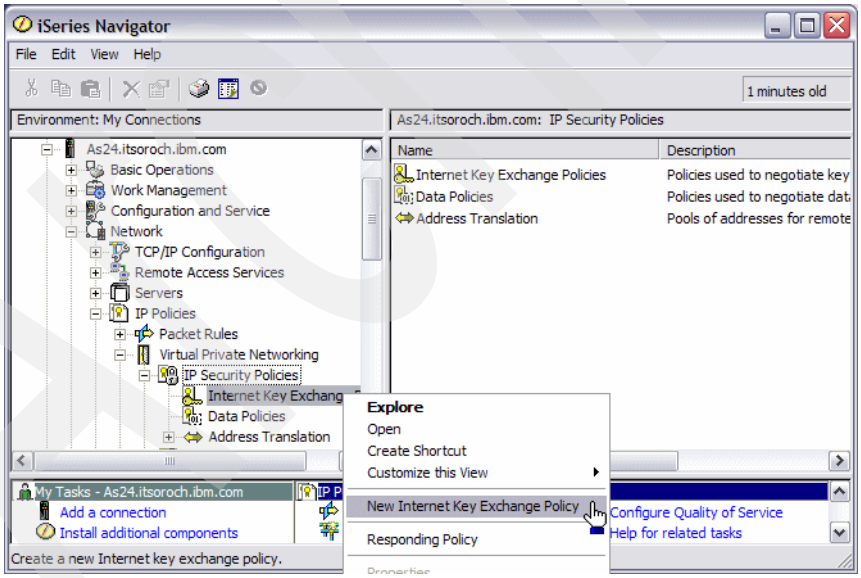


Figure 17-57 iSeries Navigator window: New Internet Key Exchange Policies

2. In the New Internet Key Exchange Policy window Remote tab, enter `20.20.20.22` in the IP Address field (answer 14 in Table 17-1 on page 522), as shown in Figure 17-58. Click the **Associations** tab.

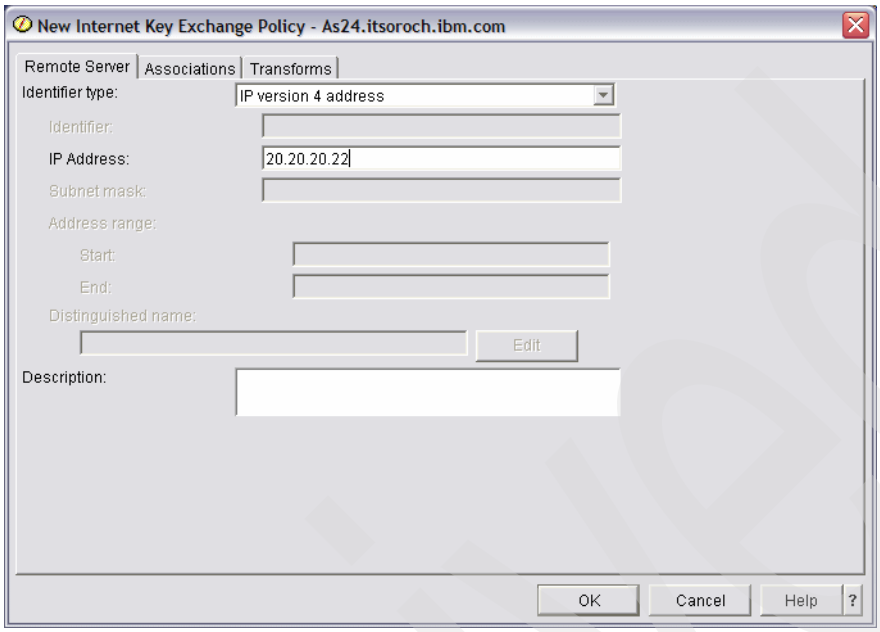


Figure 17-58 New Internet Key Exchange Policy window: Remote tab

3. On the Associations tab, check **Preshared key**. Enter `makoto` in the Key field (answer 9 in Table 17-1 on page 522). Enter `20.20.20.21` (answer 15 in Table 17-1 on page 522) in the Local key server IP address field, as shown in Figure 17-59. Click the **Transforms** tab.

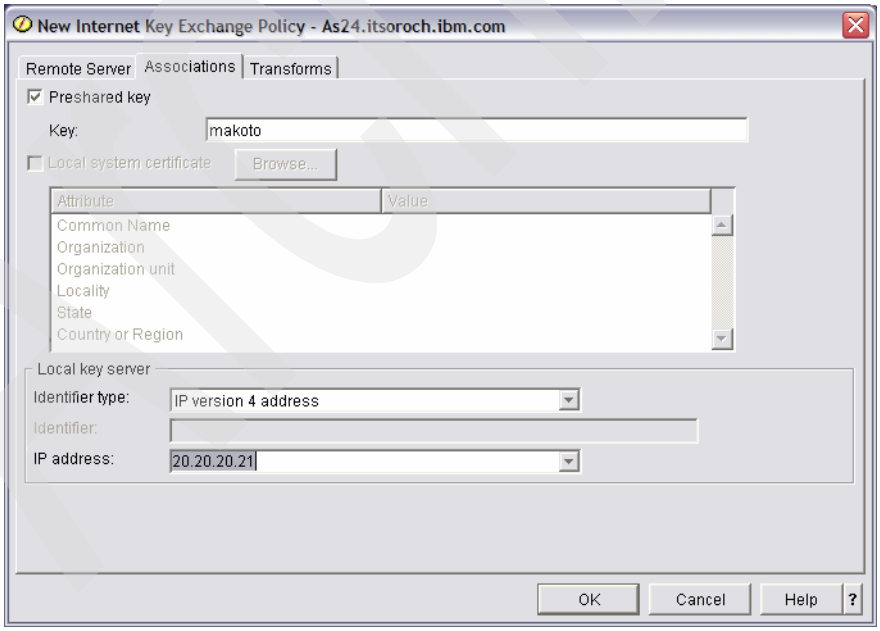


Figure 17-59 New Internet Key Exchange Policy window: Transforms tab

4. On the Transforms tab, select **IKE main mode negotiation** (answer 12 in Table 17-1 on page 522), as shown in Figure 17-60. Click **Add** to continue.

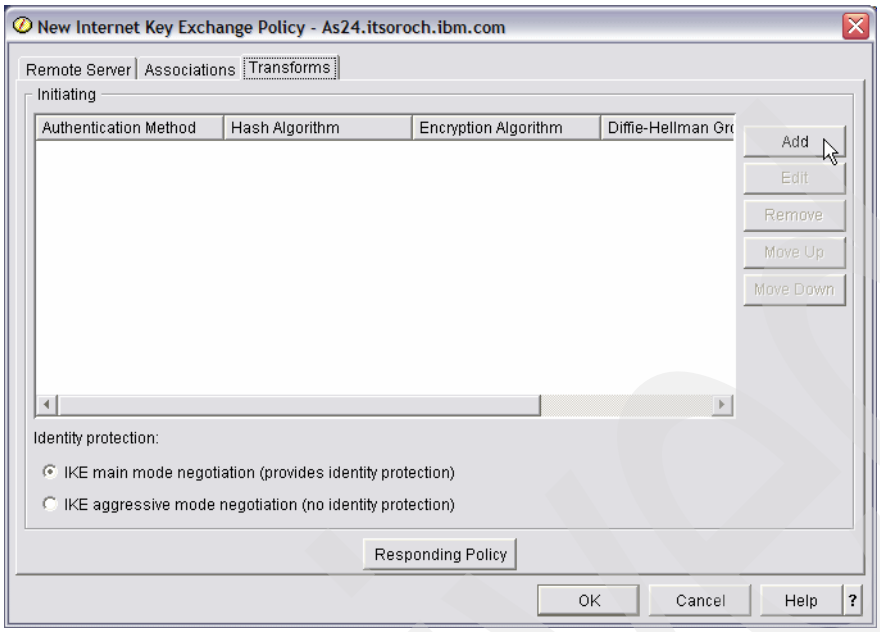


Figure 17-60 New Internet Key Exchange Policy window

5. This opens the Internet Key Exchange Policy Transform window. From the pull-down menus, make these choices, as shown in Figure 17-61:

Authentication method	Preshared key
Hash algorithm	MD5 (answer 11 in Table 17-1 on page 522).
Encryption algorithm	DES-CBC (answer 10 in Table 17-1 on page 522).
Diffie-Hellman group	Group1 (default 768 bit MODP).
Expire IKE keys after	Enter 2, choose **hours** (answer 11 in Table 17-1 on page 522).

Click **OK** to continue.

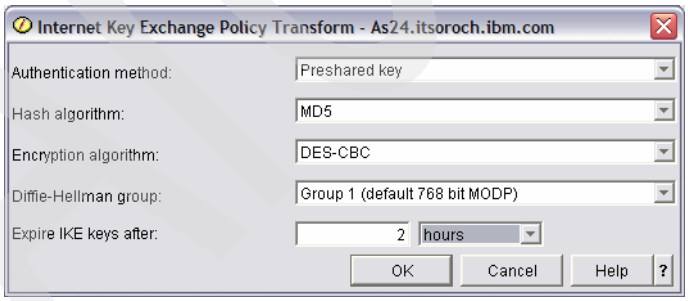


Figure 17-61 Internet Key Exchange Policy Transform window

6. In the Internet Key Exchange Policy window (Figure 17-62) click **Responding Policy**.

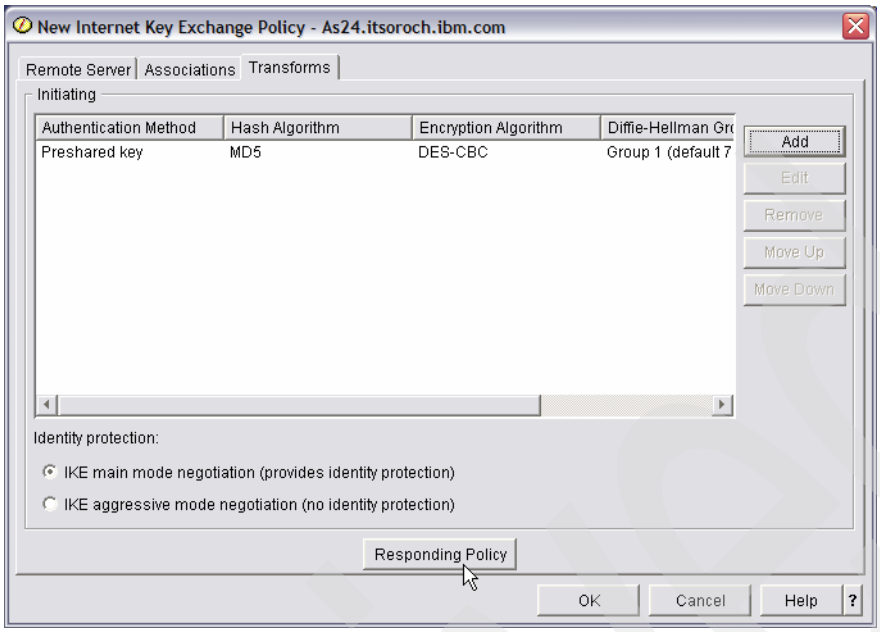


Figure 17-62 Internet Key Exchange Policy window

7. In the Responding Internet Key Exchange Policy window, enter 2 and select **hours** in the Expire IKE keys after field (answer 11 in Table 17-1 on page 522), as shown in Figure 17-63. Click **OK** to continue.

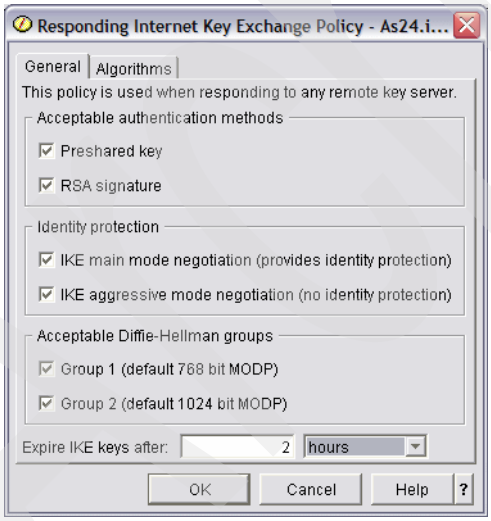


Figure 17-63 Responding Internet Key Exchange Policy window

8. In the New Internet Key Exchange policy window, click **OK**.

9. In the iSeries Navigator window, right-click **Data Policies** and choose **New Data Policy**, as shown in Figure 17-64.

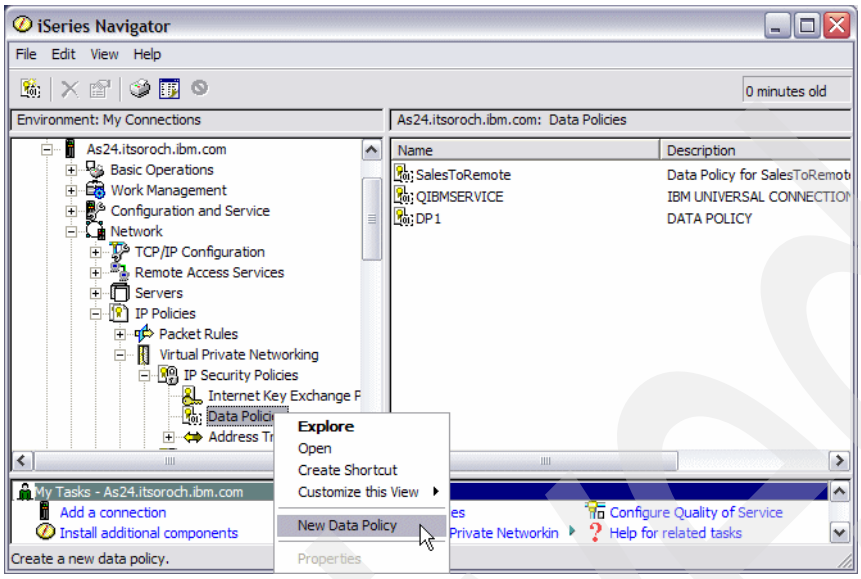


Figure 17-64 iSeries Navigator window

10. In the New Data Policy window General tab, enter AS24 in the Name field. Confirm that **Group 1 (768 bit MODP)** (answer 11 in Table 17-1 on page 522) is selected in the Diffie-Hellman group, as shown in Figure 17-65. Click the **Proposals** tab.

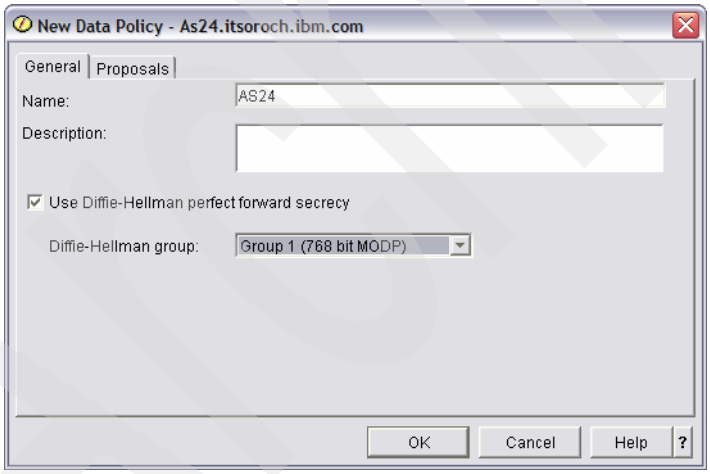


Figure 17-65 New Data Policy window

11. On the Proposals tab (Figure 17-66), click **Add**.

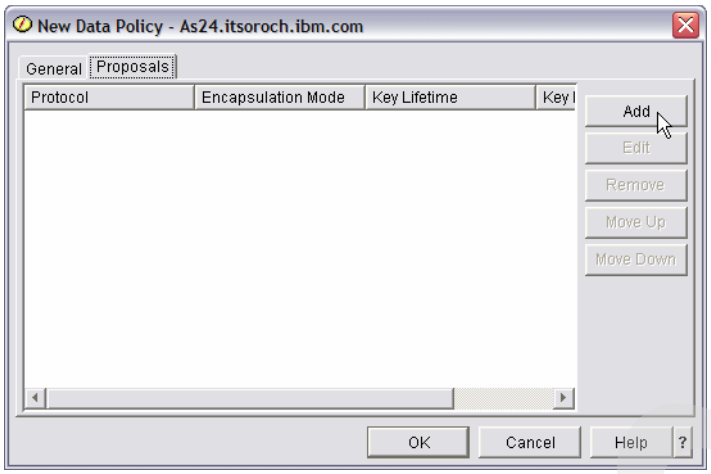


Figure 17-66 New Data Policy window: Proposals tab

12. In the New Data Policy Proposal window, keep the default values, as shown in Figure 17-67. Click the **Transforms** tab.

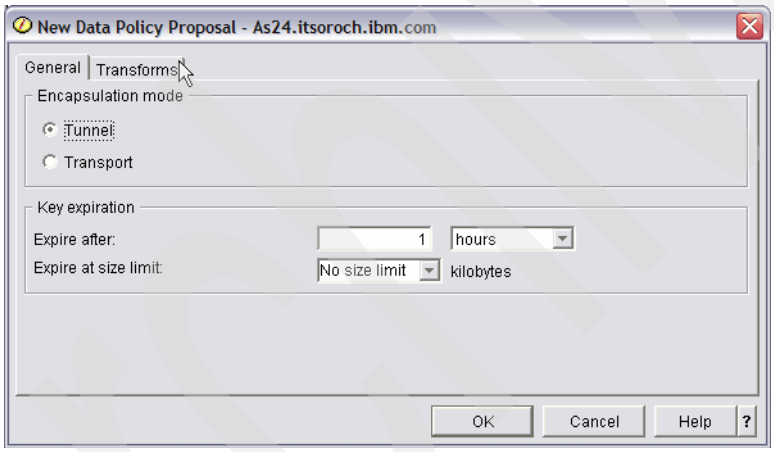


Figure 17-67 New Data Policy Proposal window

> **Tip:** You could also set the Encapsulation mode to Transport since we are doing a host-to-host configuration. This in fact might be a preferred way to configure this host-to-host connection. Of course, if you use the Encapsulation mode of Transport ensure that the same encapsulation mode that you select here is also selected on the remote system AS20.

13. On the Transforms tab, click **Add**.

14. This opens the Data Policy Transform window. Make these selections, as shown in Figure 17-68:

    Protocol Encapsulating Security Payload (ESP)
    Authentication algorithm MD5
    Encryption algorithm DES-CBC (answer 10 in Table 17-1 on page 522)

    Click **OK** to continue.

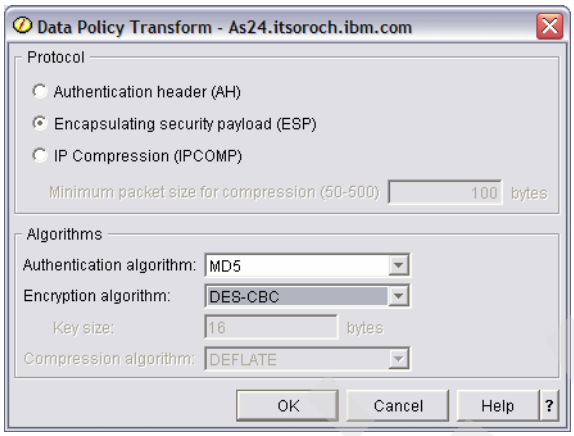


Figure 17-68 Data Policy Transform window

15. In the New Data Policy Proposal window, click **OK**.

16. In the New Data Policy window, click **OK**.

17. In the iSeries Navigator window, right-click **Virtual Private Networking** and choose **New Connection**, as shown in Figure 17-69.

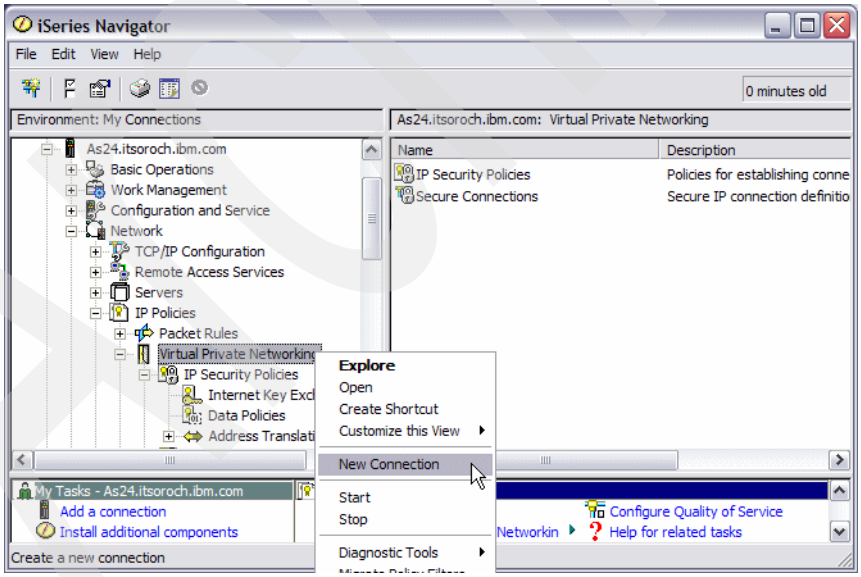


Figure 17-69 iSeries Navigator window

18. In the New Connection Wizard click **Next**, as shown in Figure 17-70.

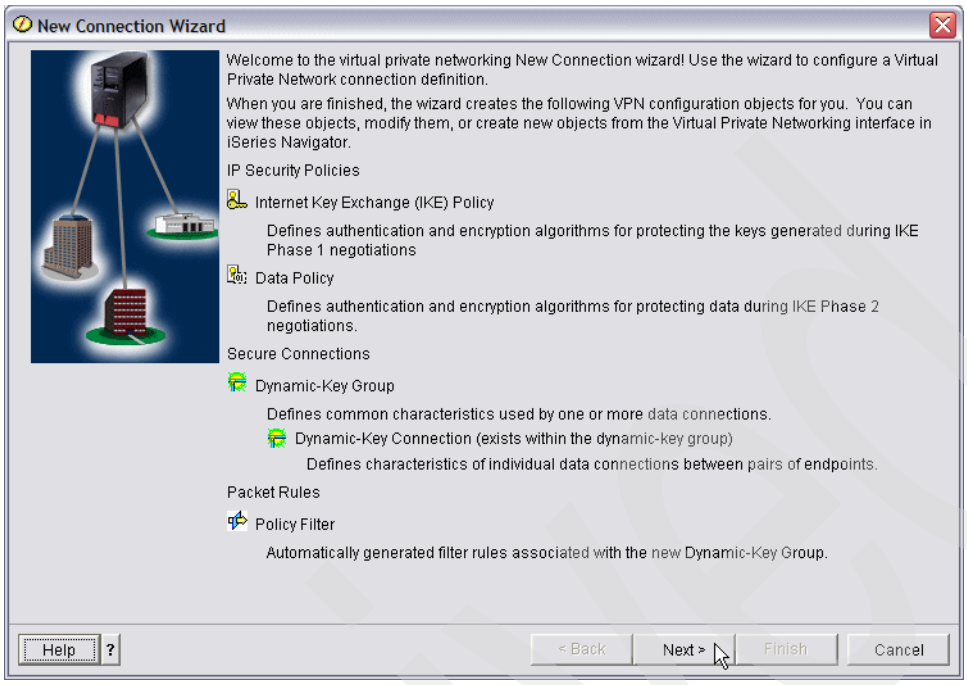


Figure 17-70 New Connection wizard

19. In the Connection Name window, enter AS24 in the Name field, as shown in Figure 17-71. Click **Next** to continue.

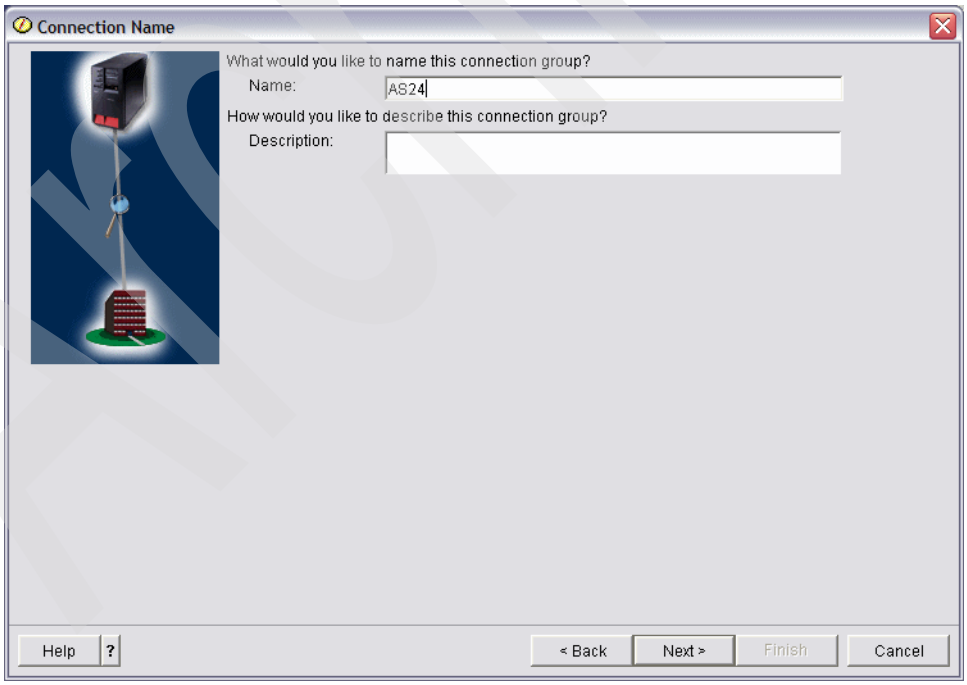


Figure 17-71 Connection Name window

20. In the Connection Scenario window, choose **Connect your host to another host**, as shown in Figure 17-72. Click **Next** to continue.

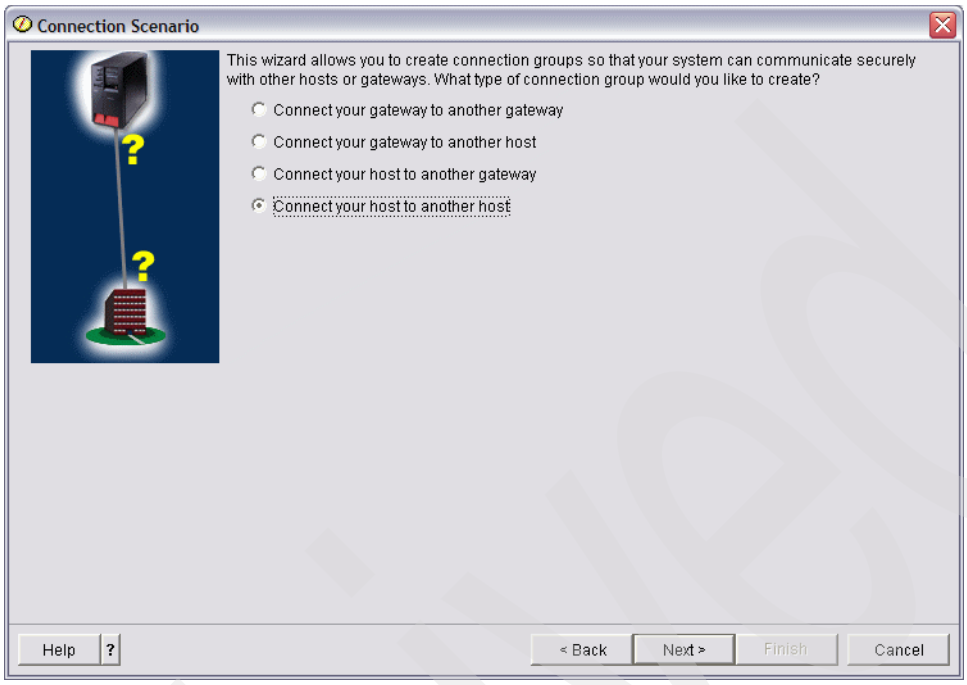


Figure 17-72 Connection scenario window

21. In the Internet Key Exchange Policy window, choose **20.20.20.22** (answer 14 in Table 17-1 on page 522) for the Policy selection, as shown in Figure 17-73. Click **Next** to continue.

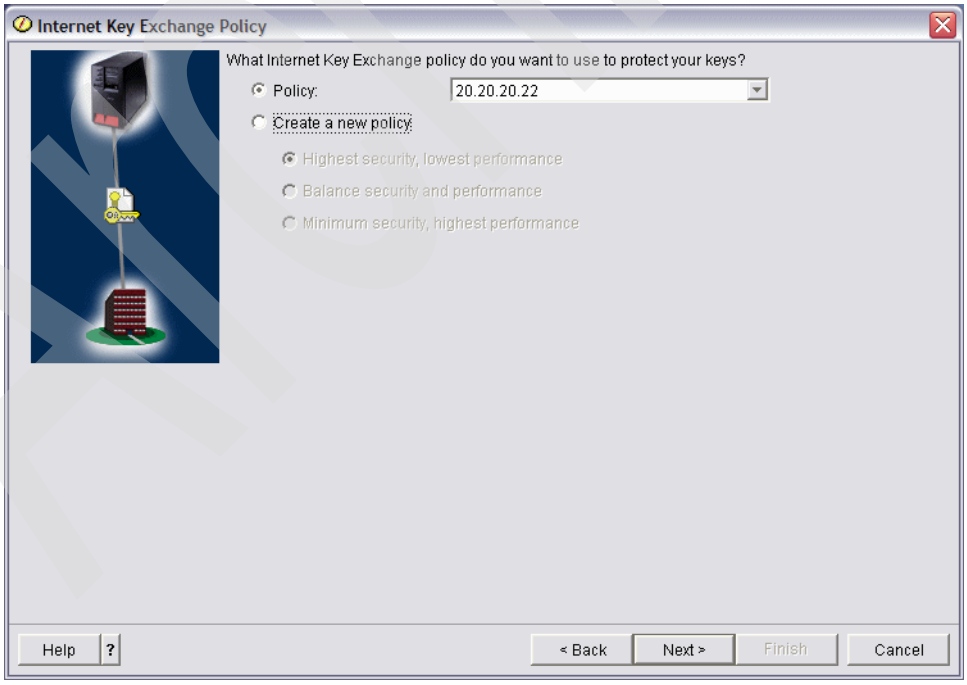


Figure 17-73 Internet Key Exchange policy window

22. In the Data Services window, keep the default values, as shown in Figure 17-74. Click **Next** to continue.

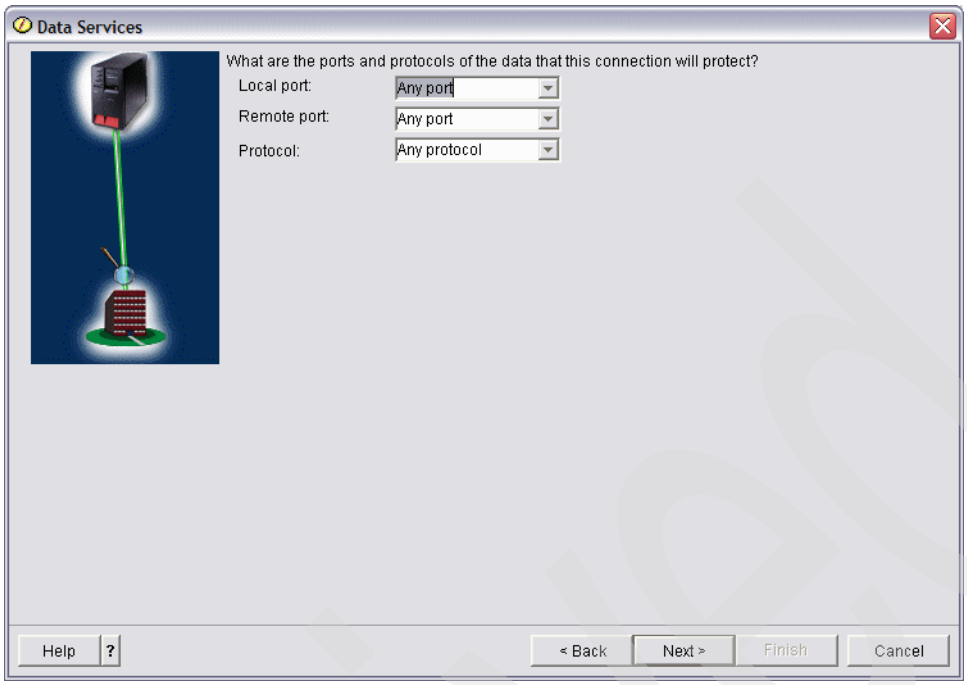


Figure 17-74 Data Services window

23. In the Data Policy window, choose **AS24** for the Policy name, as shown in Figure 17-75. Click **Next** to continue.

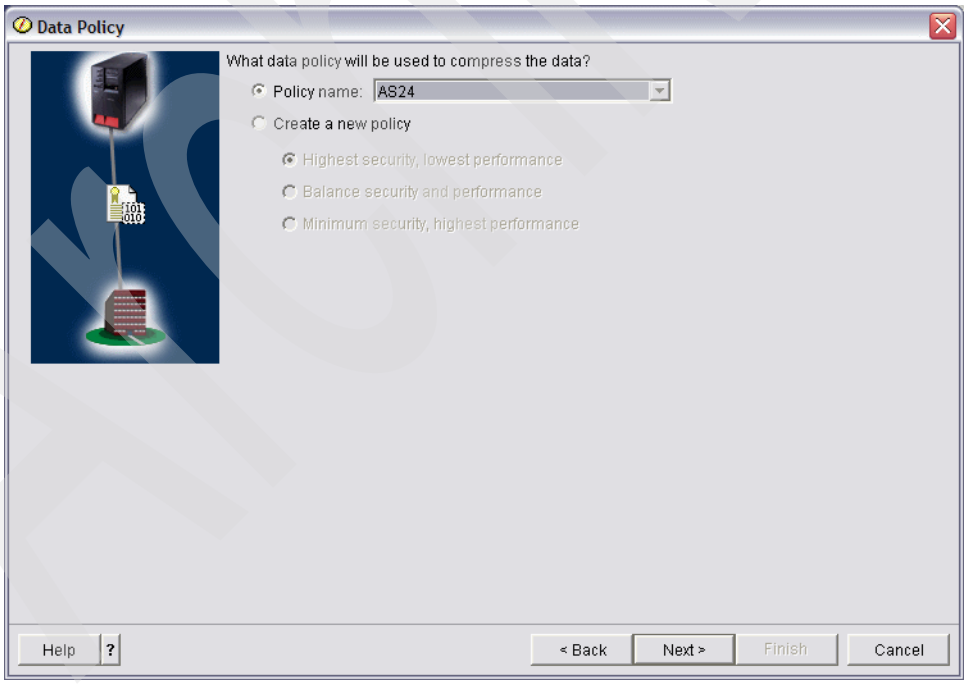


Figure 17-75 Data Policy window

24. In the Require Policy Filter window, select **This connection requires a policy filter**, as shown in Figure 17-76. Click **Next** to continue.

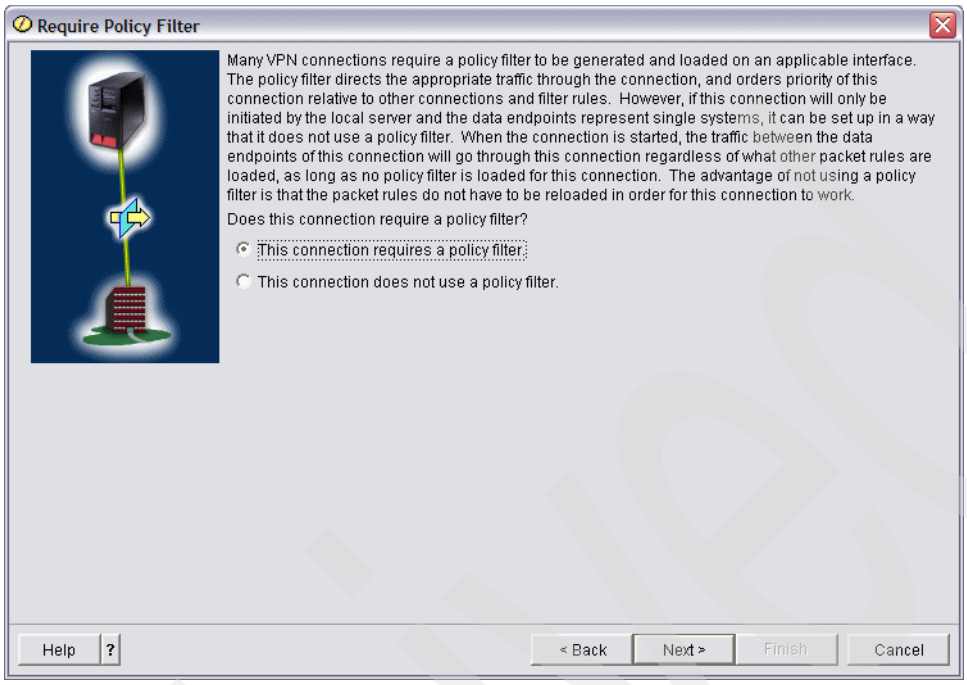


Figure 17-76 Require Policy Filter window

25. In the Applicable Interfaces window, check **PPPOE1** (answer 6 in Table 17-1 on page 522), as shown in Figure 17-77. Click **Next** to continue.

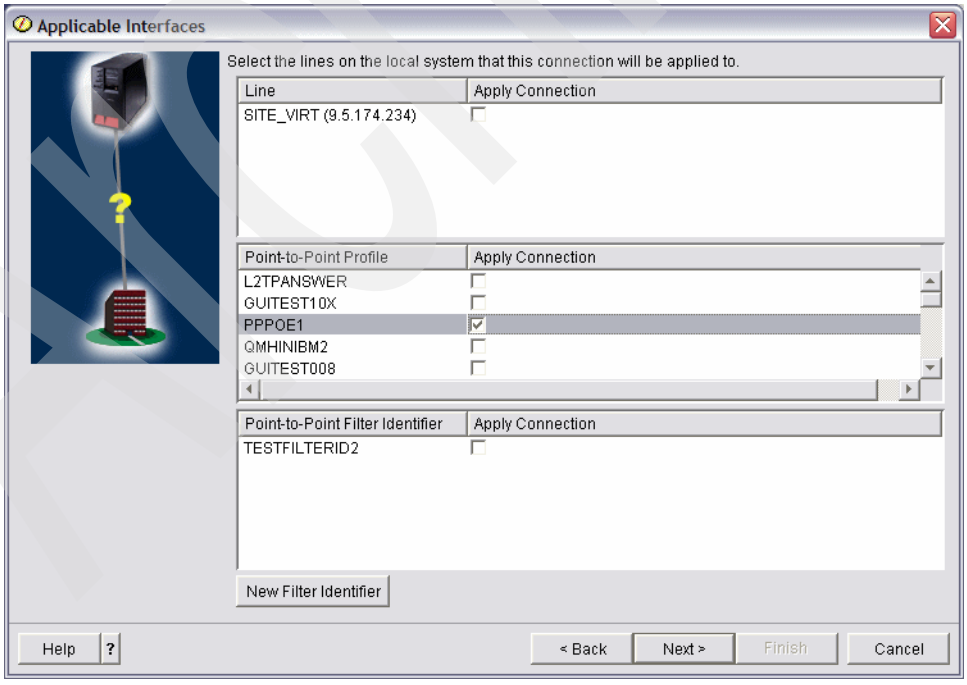


Figure 17-77 Applicable Interfaces window

26. In the New Connection Summary window, click **Finish** as shown in Figure 17-78.

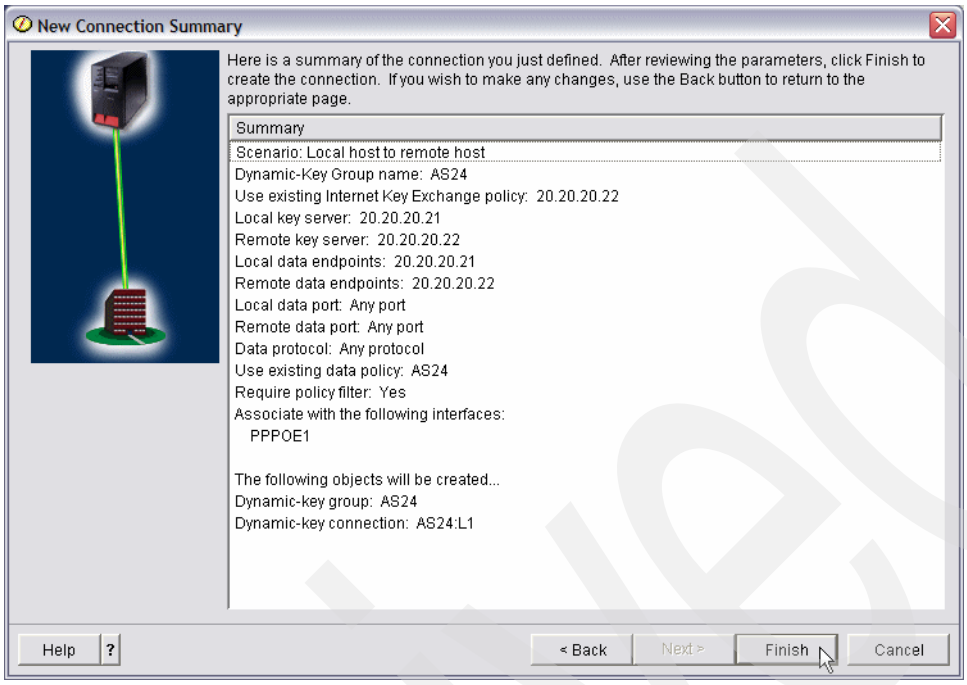


Figure 17-78 New Connection summary window

27. After you click **Finish**, the Activate Policy Filters window appears. To activate the IP filter on PPP definition PPPOE1, select **Yes, activate the generated policy filters** and **Permit all other traffic**, as shown in Figure 17-79.

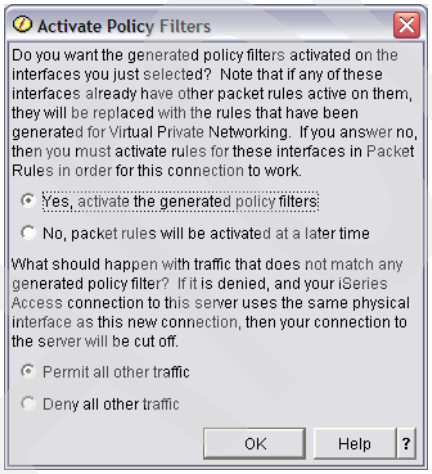


Figure 17-79 Activate Policy filters window

Note: If you prefer not to activate the IP filter now, read the procedure in the next step to activate the IP filter manually. The IP filter is required to activate the VPN connection. If you try to activate the VPN definition without activating the corresponding IP filter (which is created for the AS20 VPN definition using the PPPOE1 PPP definition), you will receive an error message that the IP policy filter is not activated.

28. Figure 17-80 shows the sample Packet Rule on the AS24 side, which is created by iSeries Navigator. This filter is not permitted to alter the contents. If you need to activate this Packet Rule manually, refer to “Manually activate the IP filter” on page 546. If you need to create a custom Packet Rule, refer to “Customizing the packet rules” on page 549.

```
FILTER SET PreIPsecPermitAllIKE   ACTION = PERMIT   DIRECTION = *   SRCADDR = *   DSTADDR = *   PROTOCOL
= UDP   DSTPORT = 500   SRCPORT = 500   JRN = OFF    FRAGMENTS = NONE

FILTER SET OPNAVPermitNonVPN   ACTION = PERMIT   DIRECTION = *   SRCADDR = *   DSTADDR = *   PROTOCOL =
*   DSTPORT = *   SRCPORT = *   JRN = OFF    FRAGMENTS = NONE

FILTER SET OPNAV6   ACTION = IPSEC   DIRECTION = OUTBOUND   SRCADDR = 20.20.20.21 DSTADDR = 20.20.20.22
PROTOCOL = *   DSTPORT = *   SRCPORT = *   CONNECTION_DEFINITION = AS24 JRN = OFF

FILTER_INTERFACE   INTERFACE = PPPOE1   SET = PreIPsecPermitAllIKE, OPNAV6, OPNAVPermitNonVPN

INCLUDE   FILE = /QIBM/UserData/OS400/TCPIP/OPNAVRULES/PPPFILTERS.I3P
```

Figure 17-80 Sample Packet Rule on AS24

Step 5: Start the VPN connection

In this procedure, we start the VPN connection on both AS20 and AS24.

Note: Assuming that the PPPoE connection is persistent, there is a option to start the VPN session automatically. You would select Start on-demand as a configuration option to cause a Dynamic-key connection. The VPN session will start when there is traffic that needs to be sent to the remote peer. This option is found when you expand **Network → IP Policies → Virtual Private Networking → Secure Connections**. Select **All Connections** and then right-click your connection and choose **Properties** from the context menu. The `Start on-demand` check box is on the General tab.

From the iSeries Navigator help, select **Start on-demand** if you want the connection to start automatically when outbound or inbound IP packets match the filter criteria for this connection. Connections with this attribute are referred to as on-demand. When data stops flowing through the connection, it automatically returns to its on-demand state after the active security associations are no longer valid.

For this attribute to take effect, you must also select **Start appropriate connections when the server starts** on the Virtual Private Networking - General properties dialog by expanding **Network → IP Policies**. Then right-click **Virtual Private Networking** and choose **Properties**.

The procedure is:

1. In the iSeries Navigator window, right-click the **AS20** VPN definition and choose **Start**, as shown in Figure 17-81.

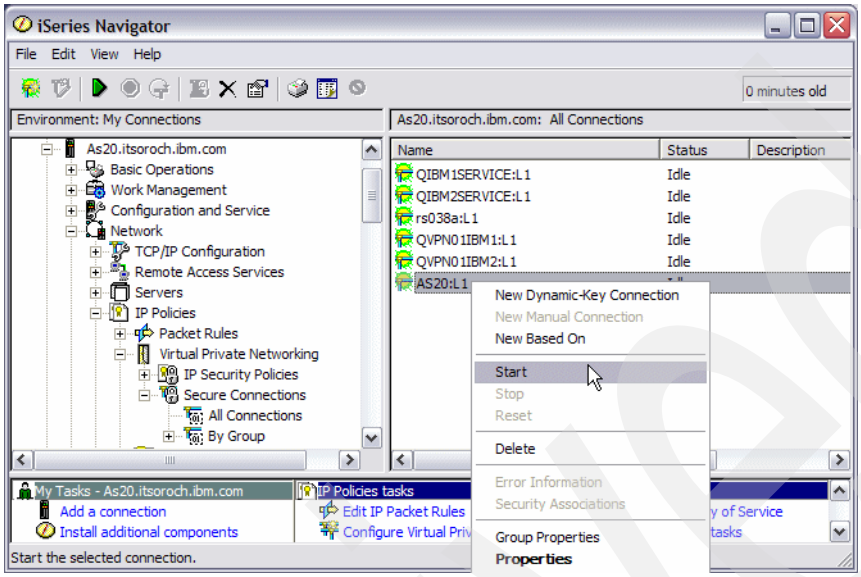


Figure 17-81 iSeries Navigator window: Start VPN connection AS20

2. In the iSeries Navigator window, right-click the **AS24** VPN definition and choose **Start**, as shown in Figure 17-82.

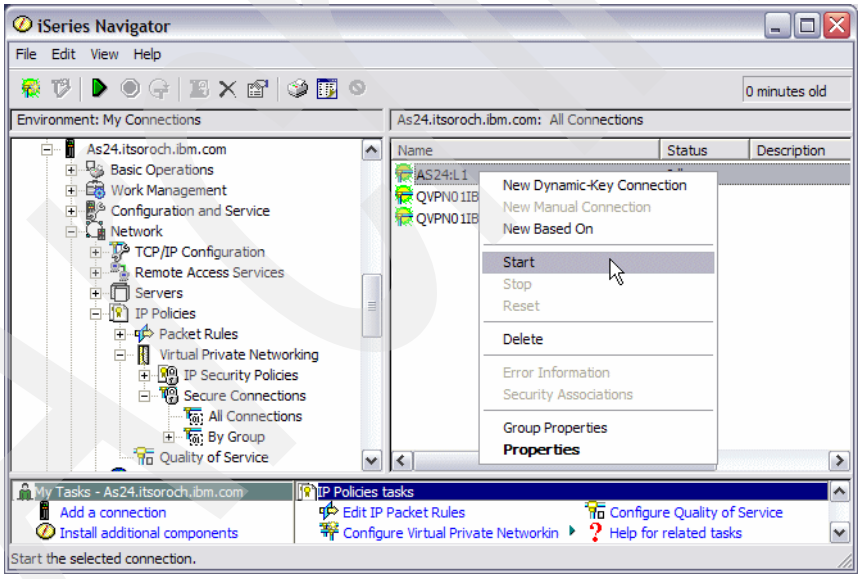


Figure 17-82 iSeries Navigator window: Start VPN connection AS24

3. If the VPN connection is established successfully between AS20 and AS24, the Status of each connection turns to Enabled, as shown for As20 in Figure 17-83 and for AS24 in Figure 17-84.

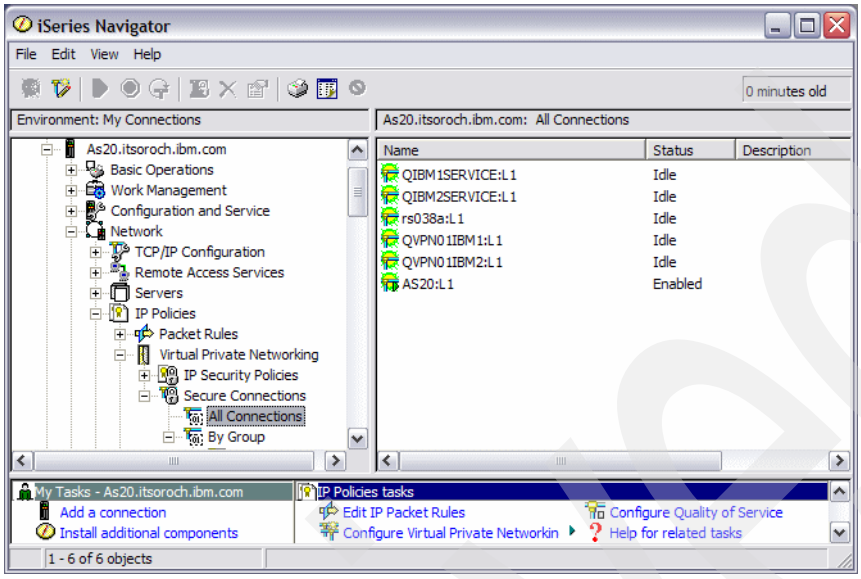


Figure 17-83 iSeries Navigator window: Enabled VPN connection for AS20

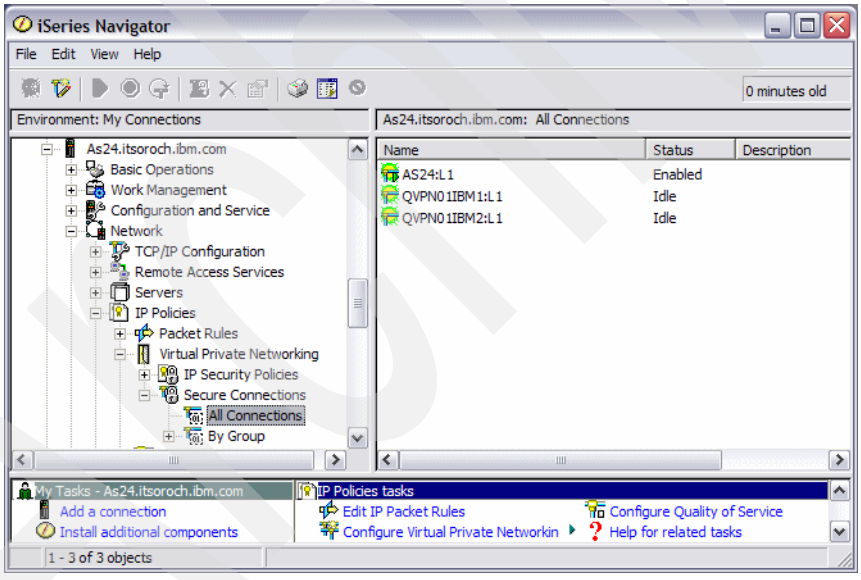


Figure 17-84 iSeries Navigator window: Enabled VPN connection for AS24

17.2 Dynamic resource sharing scenario

The PPP Dynamic Resource Sharing function provides the capability to designate an (analog) line resource as *shared*, enabling PPP dial profiles to *borrow* a line being used to listen for incoming calls in order to place an outgoing call. For more information about this topic, see “PPP Dynamic Resource Sharing for analog connections” on page 96.

In this scenario, we show how to configure dynamic resource sharing for both an Originator connection profile and a Receiver connection profile.

17.2.1 Scenario overview

You might choose this scenario if these conditions apply:

- If your communication application needs to have a Receiver connection profile running for incoming calls and an Originator profile running for outgoing calls.
- These calls do not have to be active at the same time.
- You have a limited number of switched line ports and you cannot afford to dedicate each port for either an originating or a receiving application.

Sample network configuration

Figure 17-85 shows the sample network configuration of this scenario.

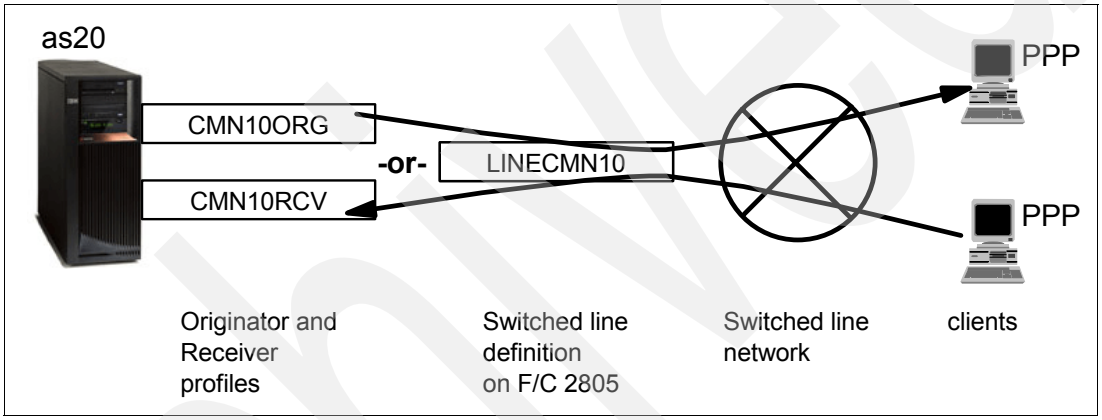


Figure 17-85 Sample network configuration: dynamic resource sharing scenario

17.2.2 Configuring dynamic resource sharing

For the configuration of this scenario we assume that your Originator profile CMN06ORG and your Receiver profile CMN06RCV have already been created and are working. To complete this scenario, these are the steps:

- Modifying the line definition to enable dynamic resource sharing
- Testing the dynamic resource sharing scenario

Modifying the line definition to enable dynamic resource sharing

To modify the line definition on your System i server to support dynamic resource sharing:

1. In your Originator profile window, click the **Connection** tab (Figure 17-86). Click **Open** under Link Configuration. In this example, the line resource to be shared between Originator profile and Receiver profile is LINECMN06.

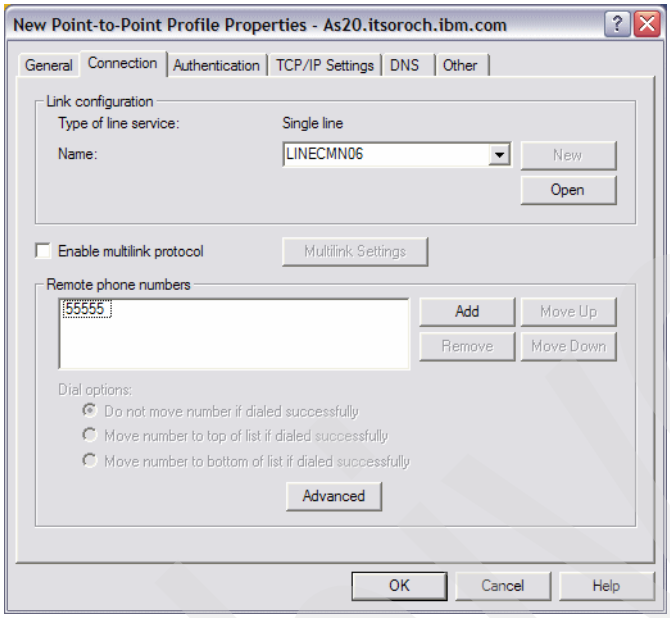


Figure 17-86 Originator profile window

2. In the Linecmn06 Properties window, make sure that your hardware resource is selected in the Hardware resource box, as shown in Figure 17-87. In this example, **CMN06** is selected. Click the **Modem** tab.

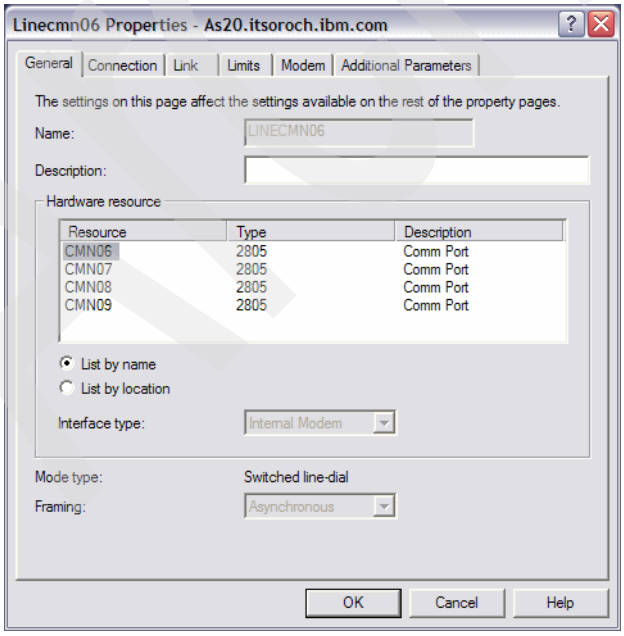


Figure 17-87 LINECMN06 Properties window

3. On the Modem tab check **Enable dynamic resource sharing**, as shown in Figure 17-88. This allows the line definition to be shared between the Originator profile and the Receiver profile. Click **OK** to save the configuration.
4. In your Originator profile window, click **OK** to save the configuration.

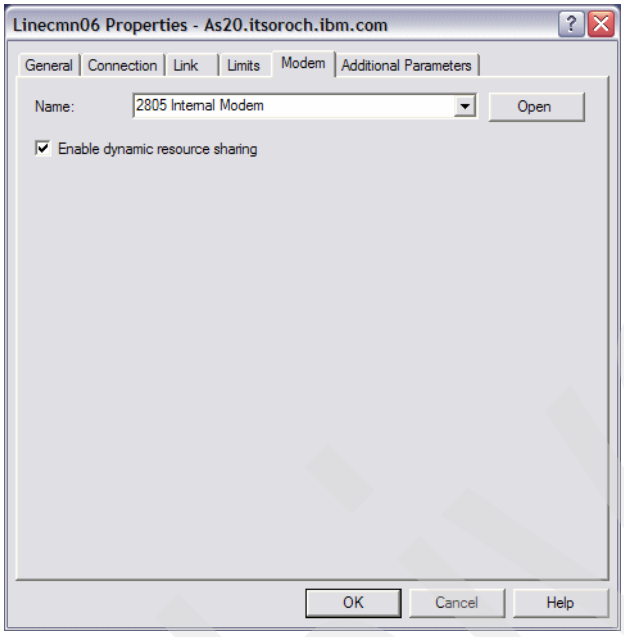


Figure 17-88 LINECMN06 Properties window

Tip: Your Originator profile CMN06ORG and your Receiver profile CMN06RCV share the single link configuration LINECMN06. You have now changed the link configuration LINECMN06 to enable dynamic resource sharing, so nothing further must be done with the Receiver profile CMN06RCV.

To demonstrate this, we continue and edit the properties of the Receiver profile CMN06RCV in the following steps.

5. In your Receiver profile window, click the **Connection** tab. Click **Open** in the Link configuration box, as shown in Figure 17-89. In this example, the line resource to be shared between the Originator profile and the Receiver profile is LINECMN06.

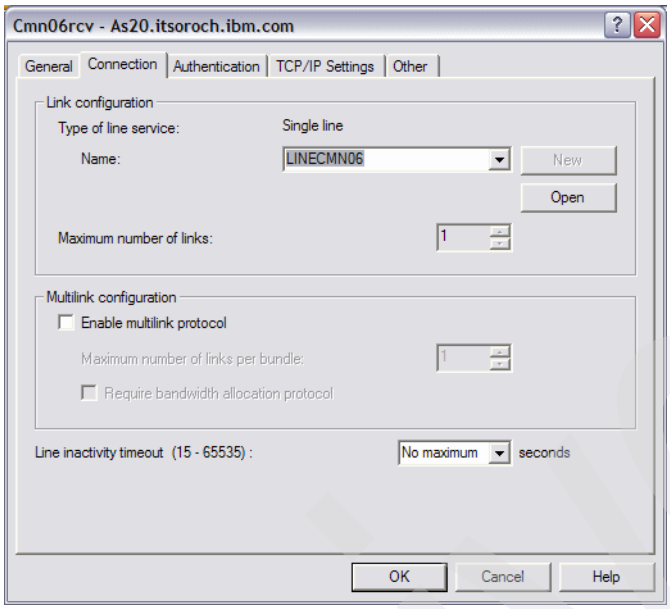


Figure 17-89 Receiver profile window

6. In the Linecmn06 Properties window, make sure that your hardware resource is selected in the Hardware resource box, as shown in Figure 17-90. In this example, CMN06 is selected. Click the **Modem** tab.

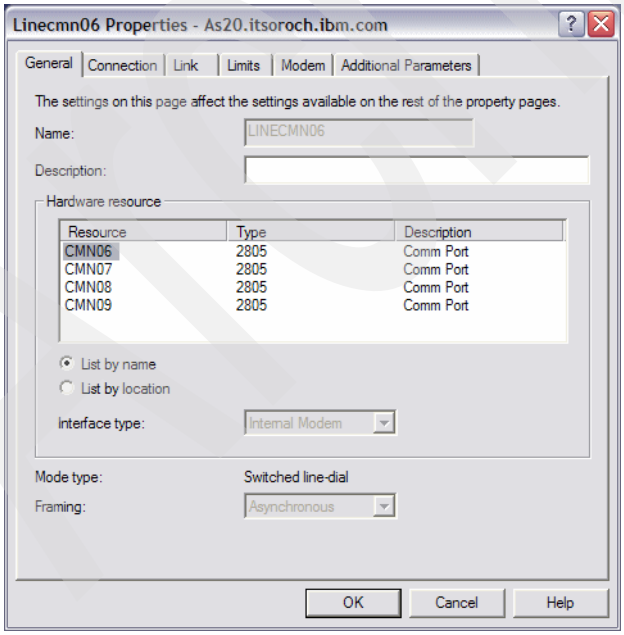


Figure 17-90 LINECMN06 Properties window: General tab

7. On the Modem tag, “Enable dynamic resource sharing” has already been selected. Your Originator profile CMN06ORG and your Receiver profile CMN06RCV share this single link configuration, LINECMN06. You have already changed this link configuration to enable

dynamic resource sharing, so nothing further must be done with the Receiver profile CMN06RCV. Click **OK** to save the configuration.

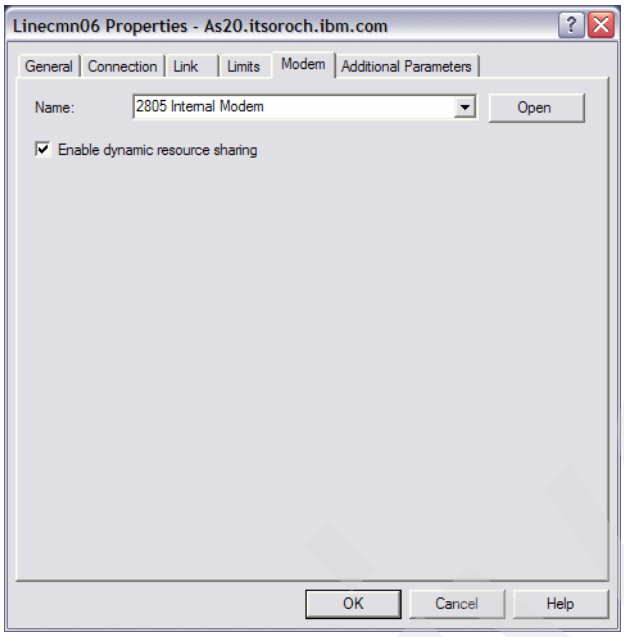


Figure 17-91 LINECMN06 properties window: Modem tab

8. In your Receiver profile window, click **OK** to save the configuration.

Testing the dynamic resource sharing scenario

To do this:

1. In the iSeries Navigator window, right-click your Receiver profile that shares the line resource with your Originator profile, and choose **Start**. In this example, start the Receiver profile CMN06RCV, as shown in Figure 17-92. The status of CMN10RCV is now `Waiting for incoming call`.

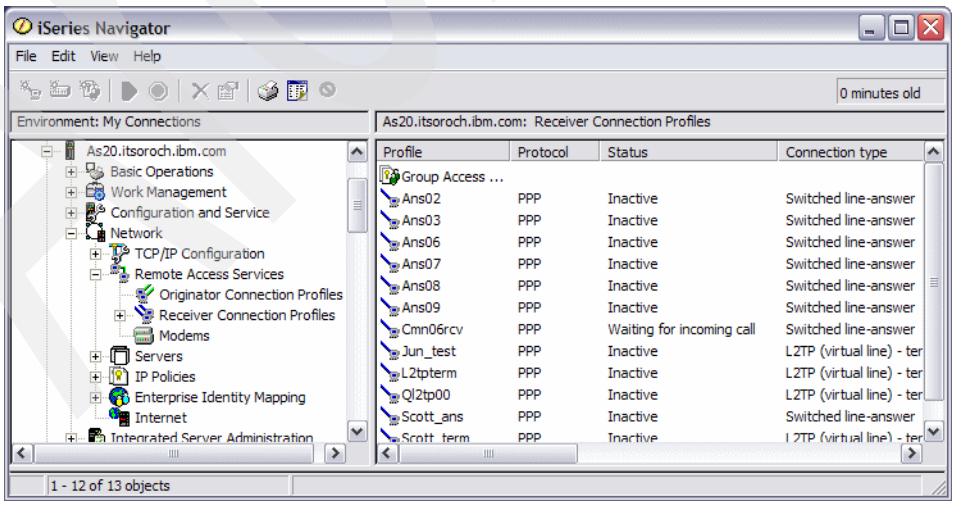


Figure 17-92 iSeries Navigator window: Cmn06rcv: Waiting for incoming call

2. In the iSeries Navigator window, start your Originator profile that shares the line resource with your Receiver profile. In this example, start the Originator profile CMN06ORG, as shown in Figure 17-93. The status of CMN06ORG is now `Calling remote system`.

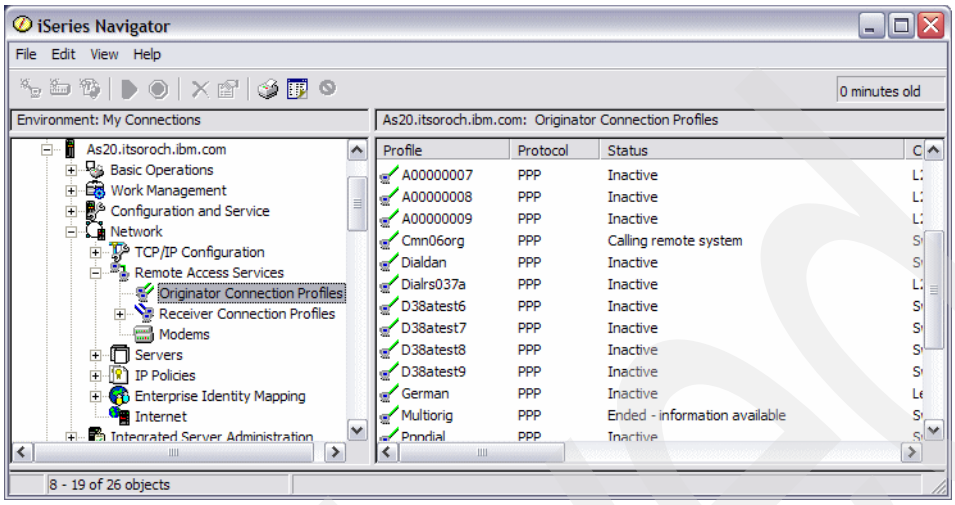


Figure 17-93 iSeries Navigator window: Cmn06org: Calling remote system

3. While the Originator profile is initiating a call, watch the Receiver profile status. In this example, look at the Receiver profile CMN06RCV, as shown in Figure 17-94. The status of CMN06ORG is now `Resource sharing - waiting for line to become active`. This represents that the Receiver profile CMN06RCV is waiting for a line resource ready, which is currently used by Originator profile CMN06ORG.

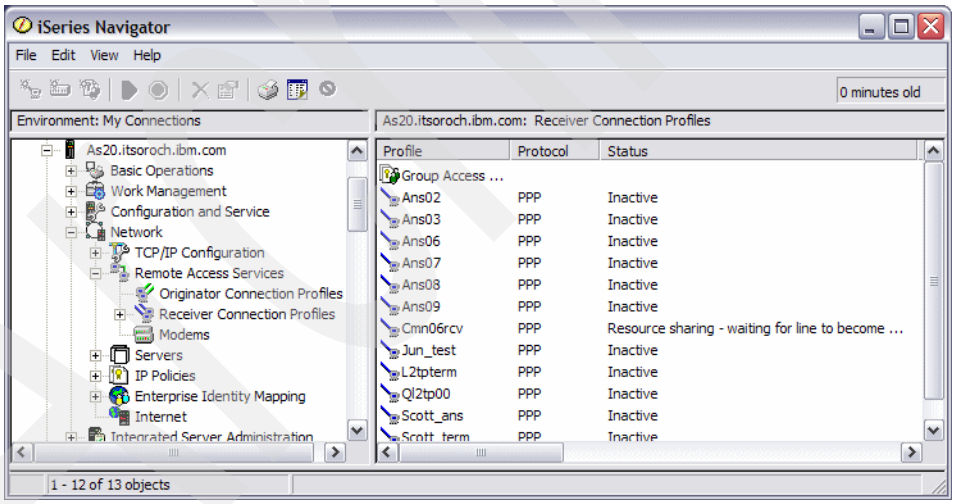


Figure 17-94 iSeries Navigator window: Cmn06rcv is now Resource sharing

Tip: Right-click the active **Originator or Receiver** profile to open the context menu. Select **Connections**. This Connections window can give you a lot of very useful information in just one place that can be used to identify the status of the profile and enables you to initiate problem determination.

Use the **Refresh** button to watch the profile step through a series of states as both the Receiver and Originator profile share the single and common link resource. To observe the state messages for both the Receiver and Originator profile, we suggest that you continuously click the **Refresh** button on a 5250 panel with a list of objects that are always changing state.

Selecting the job under the Primary Status column enables you to select either the Details or Jobs button in the right pane. Clicking **Jobs** displays the jobs that are associated with this profile. Selecting **Details** presents a list of session attributes.

Starting in V5R4, profile sessions run as threads within job QTPPPCTL instead of as separate session jobs. This changes the information available from the Connections window. After selecting a Connection, clicking **Jobs** displays the controlling job. Selecting **Details** presents a list of session attributes. Selecting **Call Log** presents details of the attempt to initiate the session. Selecting **Message Log** presents the messages associated with the details of the attempt to initiate the session.

17.3 Dial-on-demand with unnumbered PPP connection

In this scenario, we show how to configure dial-on-demand with unnumbered PPP connection.

Tip: Why do we call this an unnumbered network? To learn more, see “Tip: An unnumbered network” on page 89.

17.3.1 Scenario overview

You might choose this scenario if these conditions apply:

- If you need to connect two System i’s with a switched line (telephone line) at a reasonable cost.
- You want a server to dial up to the remote system automatically if there is IP traffic demand to get access to remote system. You also want server to disconnect the PPP connection after a specified idle time.
- You want to configure a PPP connection without specifying another IP subnet; you just want to assign an existing IP interface for local and remote PPP IP address. The tremendous advantage of this is that your routing table is updated automatically as the PPP connection is started and stopped. No manual intervention is required.

Sample network configuration

Figure 17-95 shows the sample network configuration of this scenario.

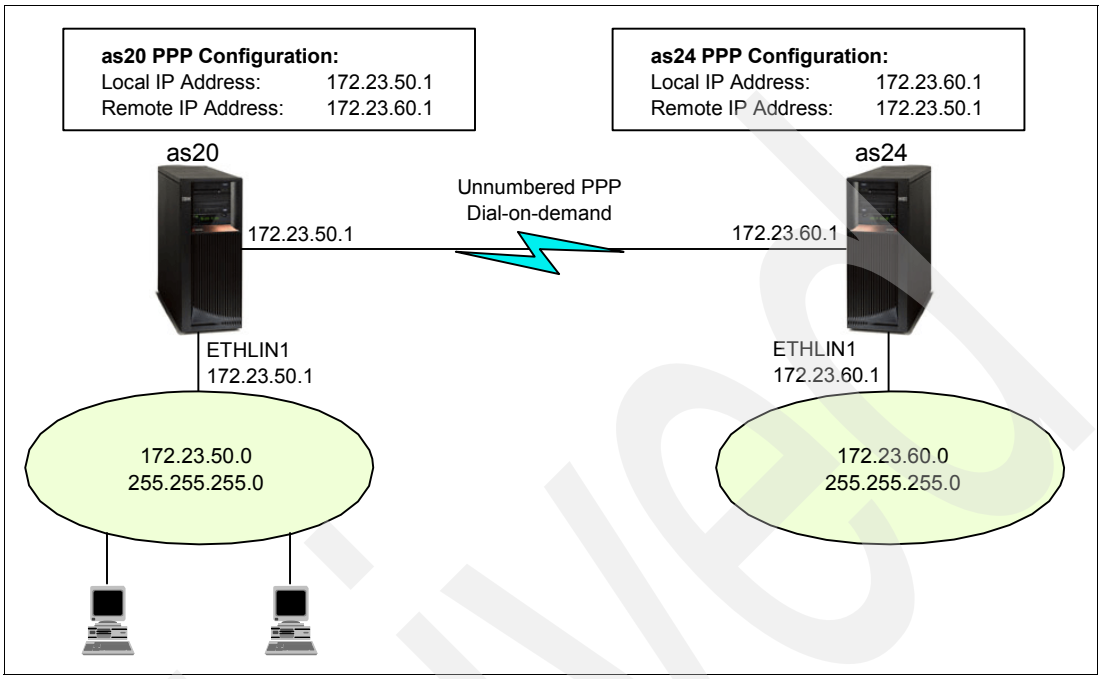


Figure 17-95 Sample network configuration: dial-on-demand with unnumbered PPP connection

17.3.2 Planning worksheet for dial-on-demand with unnumbered PPP connection

Table 17-2 shows the planning worksheet for preparing the required parameters to configure the scenario for dial-on-demand with the unnumbered PPP connection. We have filled in our answers for each question in the adjacent Scenario answers column.

Table 17-2 Planning worksheet for dial-on-demand with unnumbered PPP connection scenario

No	Questions to create dial-on-demand with unnumbered PPP connection scenario	Scenario answers
1	What is the Originator profile name on AS20?	DEMAND20
2	What is the Line name for Originator profile DEMAND20?	PPPLINE
3	What is the hardware resource name for line PPPLINE?	CMN06
4	What is the phone number to reach to the AS24 system?	123-4567
5	What is Line inactivity time-out value for Originator profile DEMAND20?	60 seconds
6	What is the user ID and password to get access to AS24 system with PAP authentication?	user ID = makoto password = password
7	What is the Local IP address on the AS20 side? This is also used as the Remote IP address on the AS24 side of the network.	172.23.50.1

No	Questions to create dial-on-demand with unnumbered PPP connection scenario	Scenario answers
8	What is the Local IP address on the AS24 side? This is also used as the Remote IP address on the AS20 side of the network.	172.23.60.1
9	What is the Receiver profile name on AS24?	DEMAND24
10	What is the Line name for Receiver profile DEMAND24?	PPPLINE
11	What is the hardware resource name for line PPPLINE?	CMN05
12	What is the validation list name on Receiver profile AS24?	PPP

17.3.3 Configuring dial-on-demand with unnumbered PPP connection

In this scenario, you will create dial-on-demand with unnumbered PPP connection scenario in the following steps:

- Step 1: Create Originator profile on AS20
- Step 2: Create Receiver profile on AS24
- Step 3: Test the dial-on-demand PPP connection

Step 1: Create Originator profile on AS20

In this step, we create an Originator profile on AS20:

1. In the iSeries Navigator window, expand **Network** → **Remote Access Services**. Right-click **Originator Connection Profiles** and choose **New Profile** (Figure 17-96).

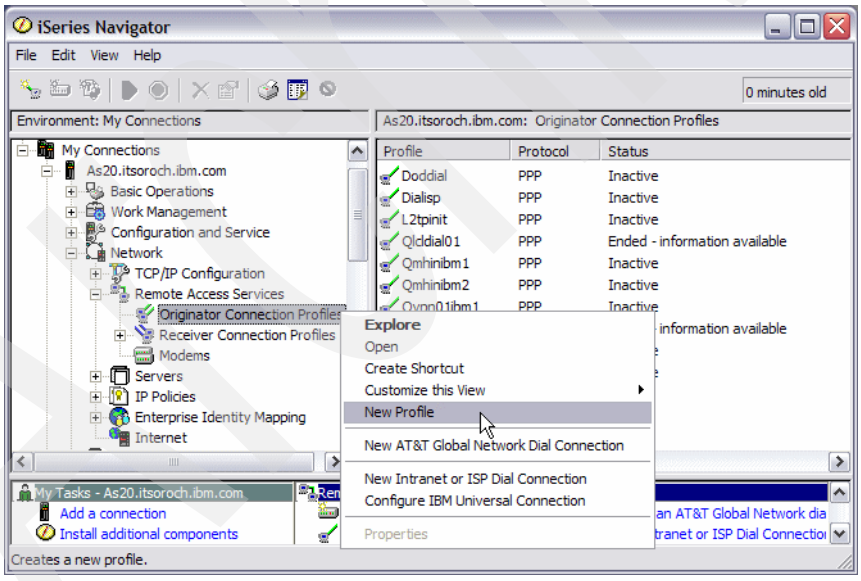


Figure 17-96 iSeries Navigator window: creating a new Originator profile

2. In the New Point-to-Point Connection Profile Setup window, select **PPP** under Protocol type and **Switched line** for Connection type. Choose **dial-on-demand (dial only)** as the Operating mode, as shown in Figure 17-97. Click **OK** to continue.

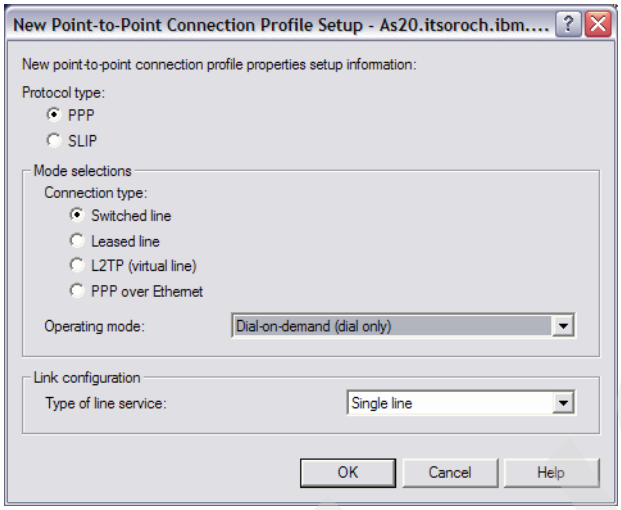


Figure 17-97 New Point-to-Point Connection Profile Setup window

3. In the New Point-to-Point Profile Properties window, enter `DEMAND20` (answer 1 in Table 17-2 on page 575) in the Name field, as shown in Figure 17-98. Click the **Connection** tab to continue.

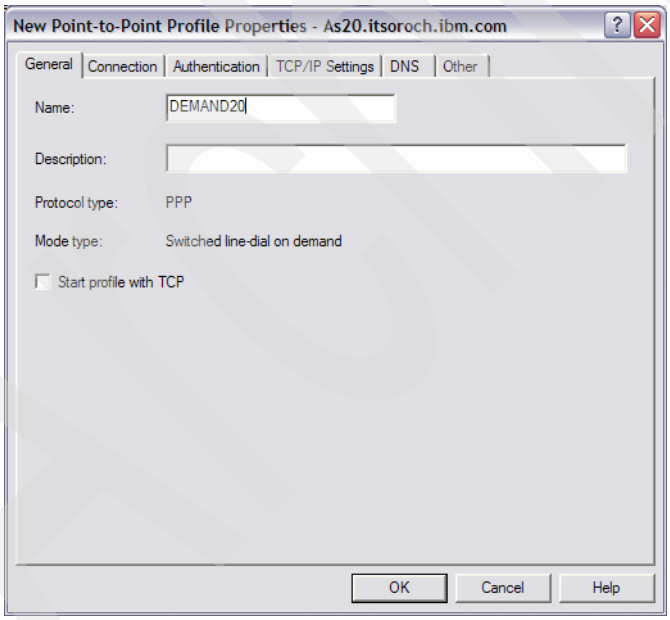


Figure 17-98 New Point-to-Point Profile Properties window

4. On the Connection tab, enter `PPPLINE` (answer 2 in Table 17-2 on page 575) in the Name field, as shown in Figure 17-99. Click **New** to continue.

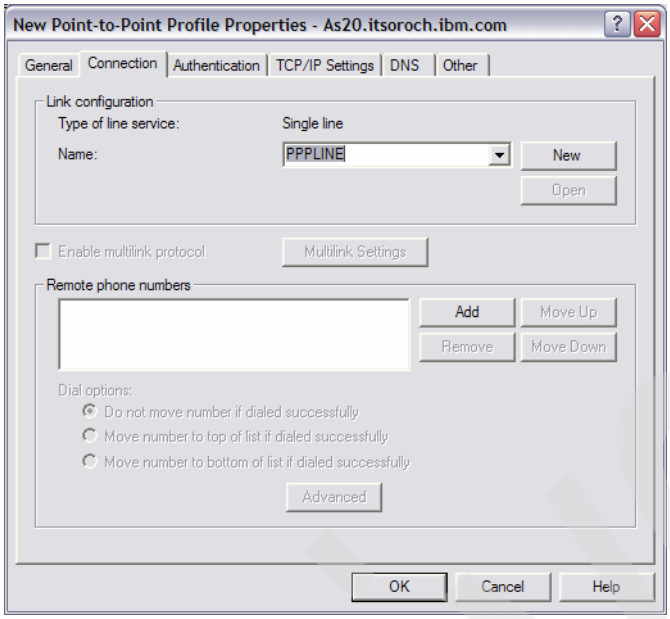


Figure 17-99 New Point-to-Point Profile Properties window: Connection tab

5. In the New Line Properties window, choose **CMN06** (answer 3 in Table 17-2 on page 575) as the hardware resource, as shown in Figure 17-100. In this scenario, we keep the defaults for all other parameters for PPPLINE. Click **OK** to continue.

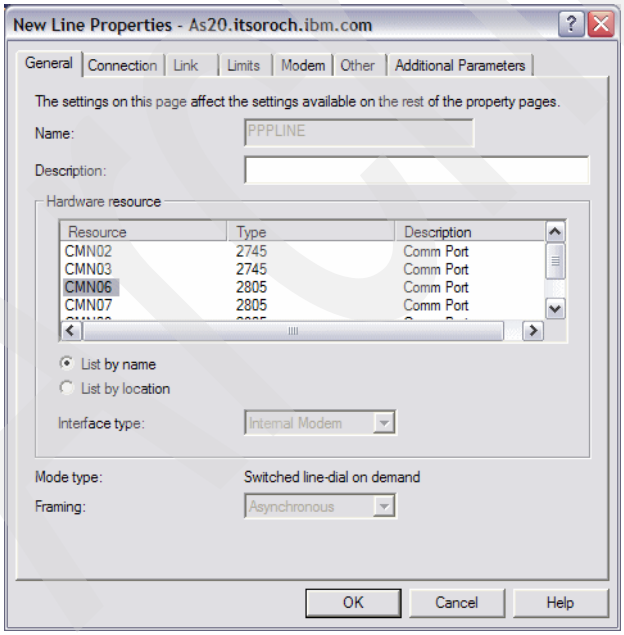


Figure 17-100 New Line Properties window

6. In the New Point-to-Point Profile Properties window (Figure 17-101), click **Add** and enter `123-4567` in the Remote phone numbers field (answer 4 in Table 17-2 on page 575), and press Enter. Click **Advanced**.

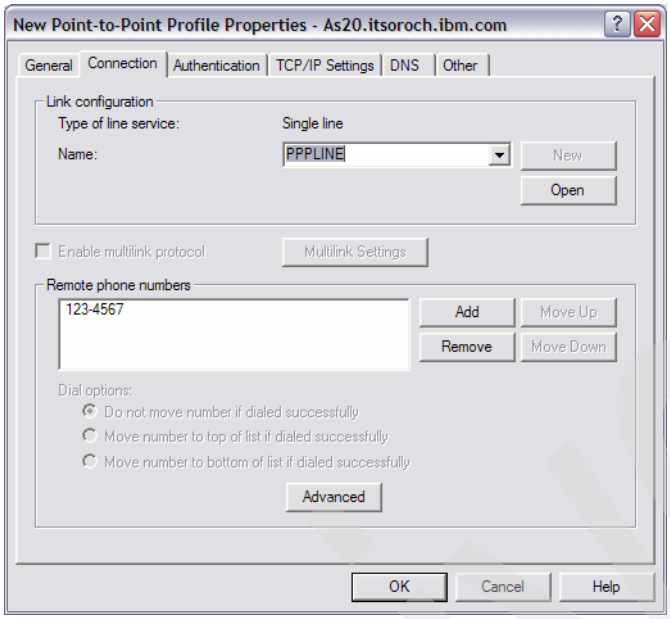


Figure 17-101 New Point-to-Point Profile Properties window

7. In the Advanced Dial Parameters window, enter `60` (answer 5 in Table 17-2 on page 575) for Line inactivity time-out, as shown in Figure 17-102. Click **OK** to continue.

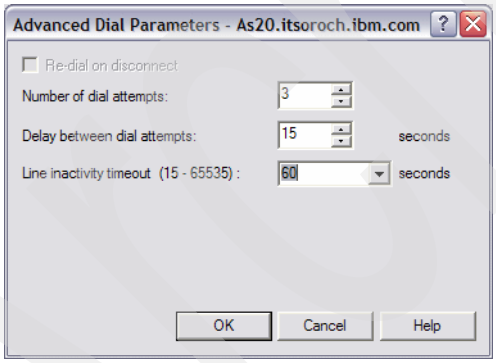


Figure 17-102 Advanced Dial Parameters window

8. In the New Point-to-Point Profile window, click the **Authentication** tab.

9. On the Authentication tab (Figure 17-103), check **Allow the remote system to verify the identity of this iSeries server**. Select **Require unencrypted password (PAP)** and enter `makoto` in the User name field. Enter `password` in the Password field (answer 6 in Table 17-2 on page 575).

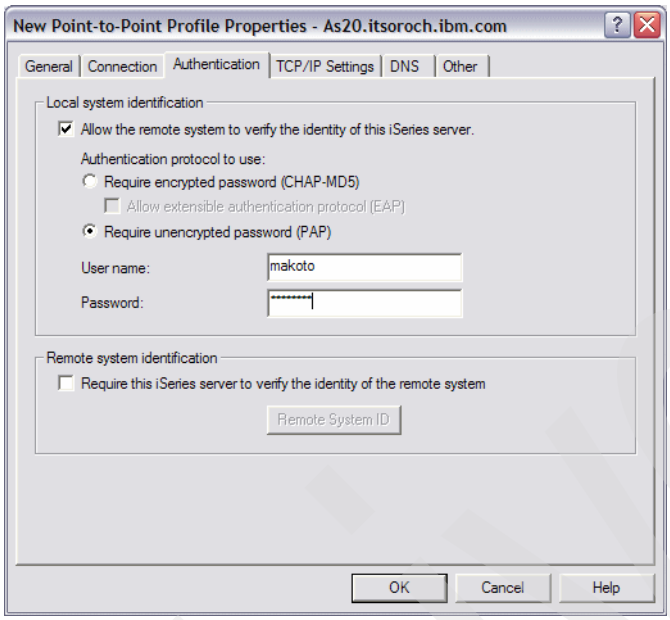


Figure 17-103 New Point-to-Point Profile Properties window: Authentication tab

10. Click the **TCP/IP Settings** tab. In the Password confirmation window that pops up, enter `password` in the input field. Select **Use fixed IP address** for Local IP address and choose **172.23.50.1** (answer 7 in Table 17-2 on page 575). For Remote IP address, select **Use fixed IP address** and enter `172.23.60.1` (answer 8 in Table 17-2 on page 575), as shown in Figure 17-104. Click **OK** to continue.

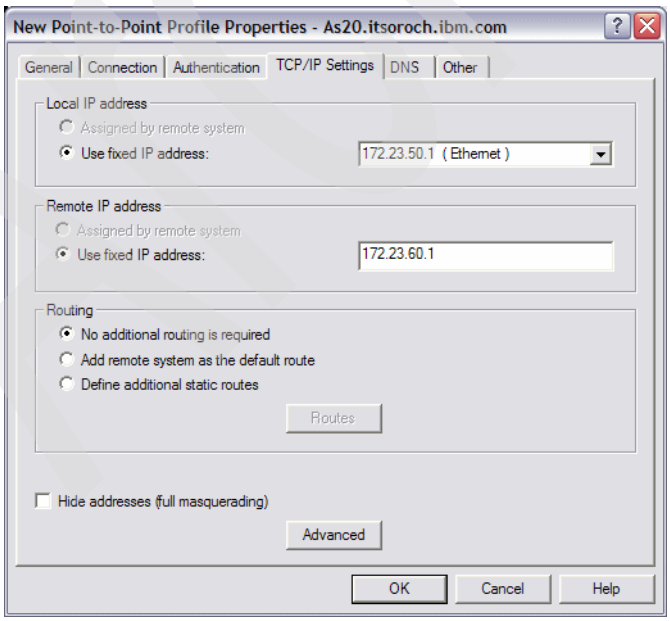


Figure 17-104 New Point-to-Point Profile window

Step 2: Create Receiver profile on AS24

In this step, we create a Receiver profile on AS24:

1. In the iSeries Navigator window (Figure 17-105), expand **Network** → **Remote Access Services**. Right-click **Receiver Connection Profiles** and choose **New Profile**.

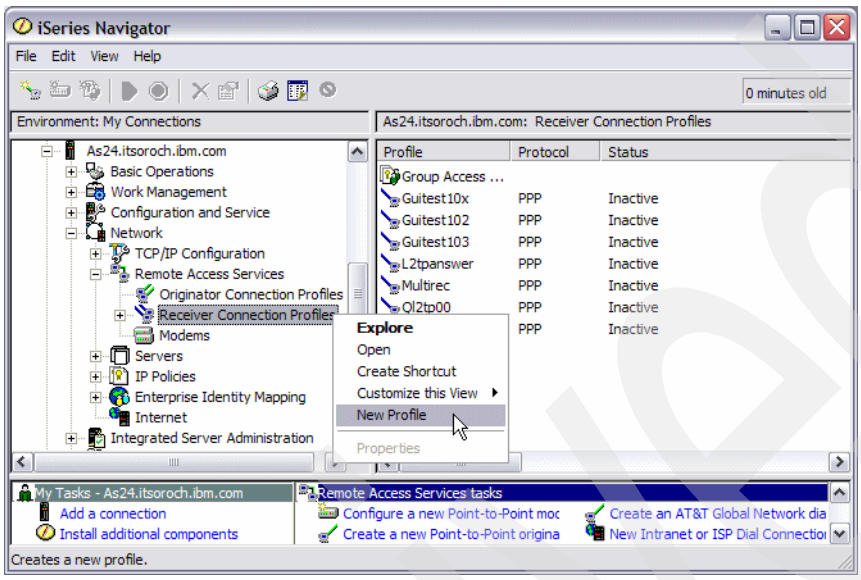


Figure 17-105 iSeries Navigator window

2. In the New Point-to-Point Connection Profile Setup window, select **PPP** for Protocol type. For Connection type, select **Switched line**. Choose **Answer** for Operating mode, as shown in Figure 17-106. Click **OK** to continue.

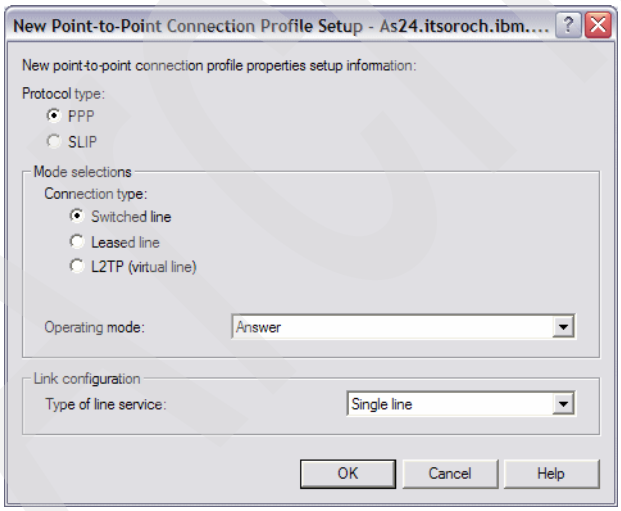


Figure 17-106 New Point-to-Point Connection Profile Setup window

3. In the New Point-to-Point Profile Properties window, enter `DEMAND24` (answer 9 in Table 17-2 on page 575) in the Name field, as shown in Figure 17-107. If desired, select to **Start Profile with TCP**. Click the **Connection** tab to continue.

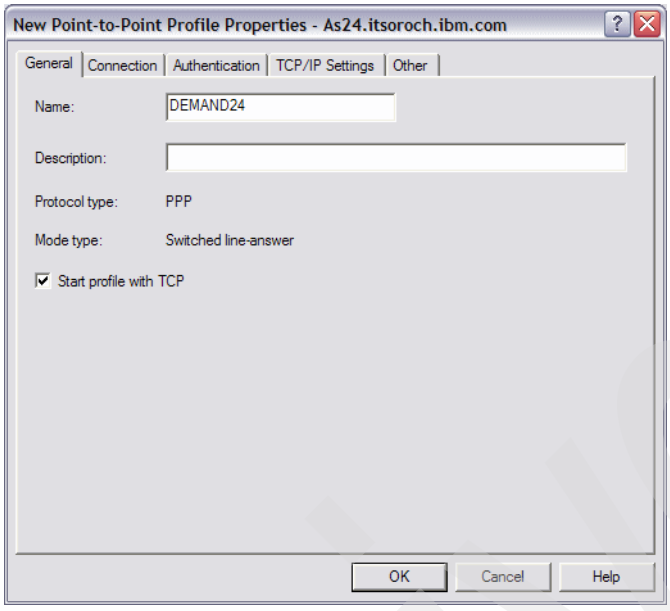


Figure 17-107 New Point-to-Point Profile window

4. On the Connection tab, enter `PPPLINE` (answer 10 in Table 17-2 on page 575) in the Name field, as shown in Figure 17-108. Click **New** to continue.

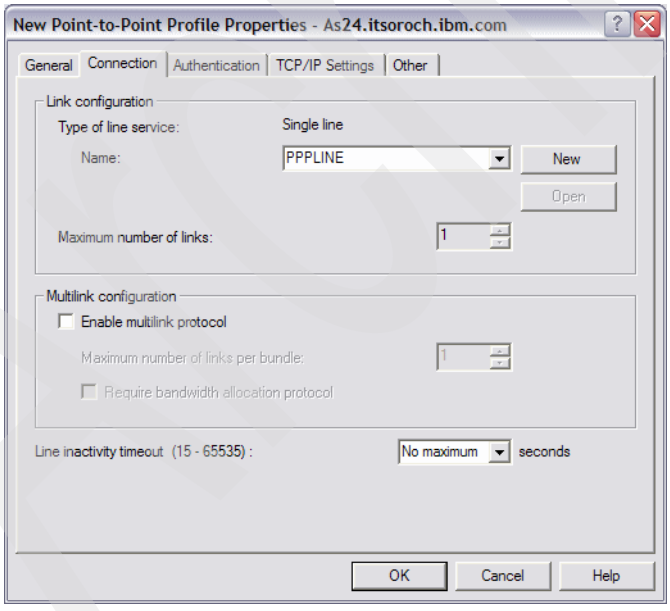


Figure 17-108 New Point-to-Point Profile window

5. In the New Line Properties window, choose **CMN05** (answer 11 in Table 1-2) for Hardware resource, as shown in Figure 17-109. In this scenario, we keep the defaults for all other parameters for PPPLINE. Click **OK** to continue.

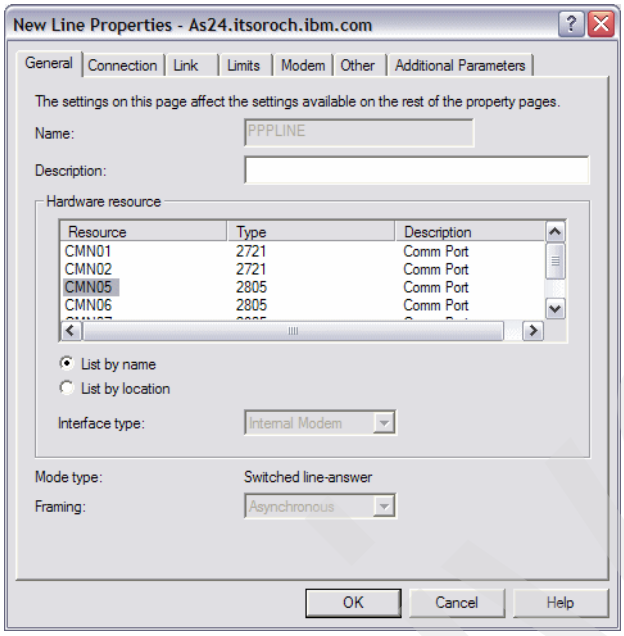


Figure 17-109 New Line Properties window

6. In the New Point-to-Point Profile Properties window, click the **Authentication** tab. Check **Requires the i5/OS server to verify the identity of the remote system**. Choose **PPP** (answer 12 in Table 1-2) in the Validation list name field. Check **Allow unencrypted password (PAP)**, as shown in Figure 17-110. Click **New**.

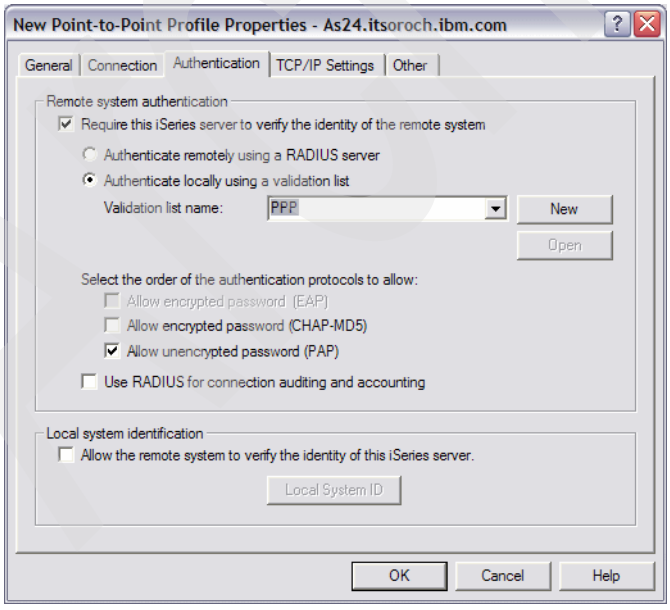


Figure 17-110 New Point-to-Point Profile Properties window: Authentication tab

7. In the New Validation List window click **Add**, as shown in Figure 17-111.

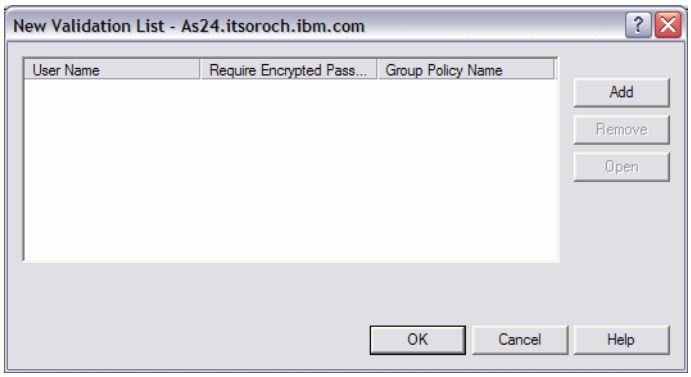


Figure 17-111 New validation list window

8. In the Add PPP user window, select **Require unencrypted password (PAP)**. Enter `makoto` for the User name and `password` in the Password field (answer 6 in Table 1-2), as shown in Figure 17-112. Click **OK** to continue.

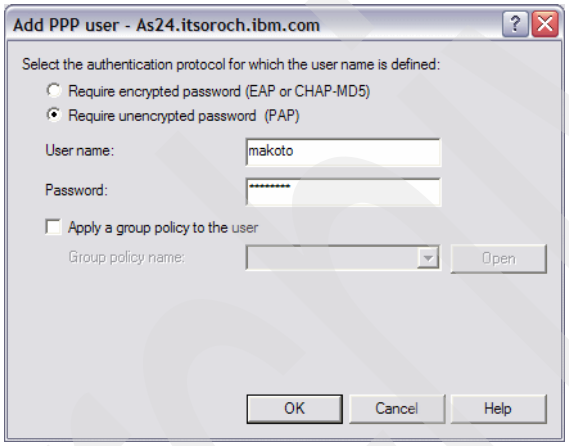


Figure 17-112 Add PPP user window

9. In the password confirmation window that pops up, enter `password` again and click **OK**. Returning to the New Validation List window, click **OK**.

10. In the New Point-to-Point Profile Properties window, click the **TCP/IP Settings** tab. Choose **172.23.60.1** (answer 8 in Table 1-2) in Local IP address field. Under Remote IP address, choose **Fixed IP Address** as the IP address assignment method and enter `172.23.50.1` (answer 7 in Table 1-2) as the Starting IP address. Check **Allow remote system to assign its own IP address**, as shown in Figure 17-113. Click **OK**.

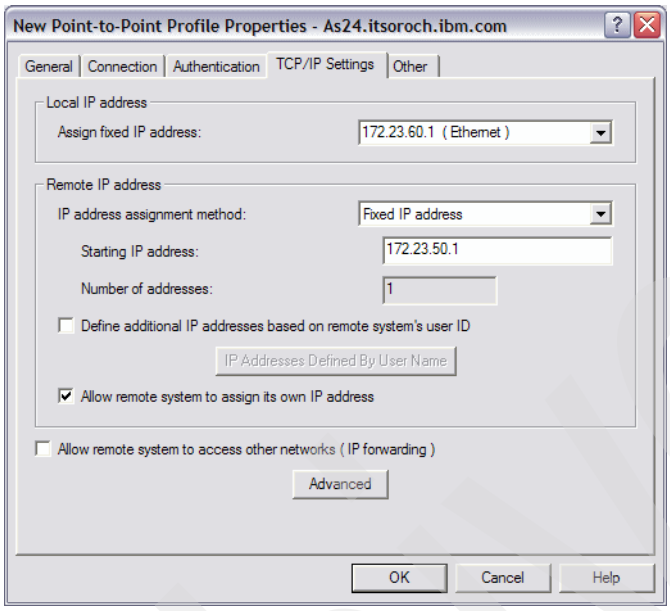


Figure 17-113 New Point-to-Point Profile Properties window: TCP/IP Settings tab

Step 3: Test the dial-on-demand PPP connection

In this step, we run a test on the dial-on-demand PPP connection:

1. In the iSeries Navigator window, right-click **Demand20** → **Start**. Wait until its Status becomes **Waiting for dial - Switched line-dial on demand**, as shown in Figure 17-114.

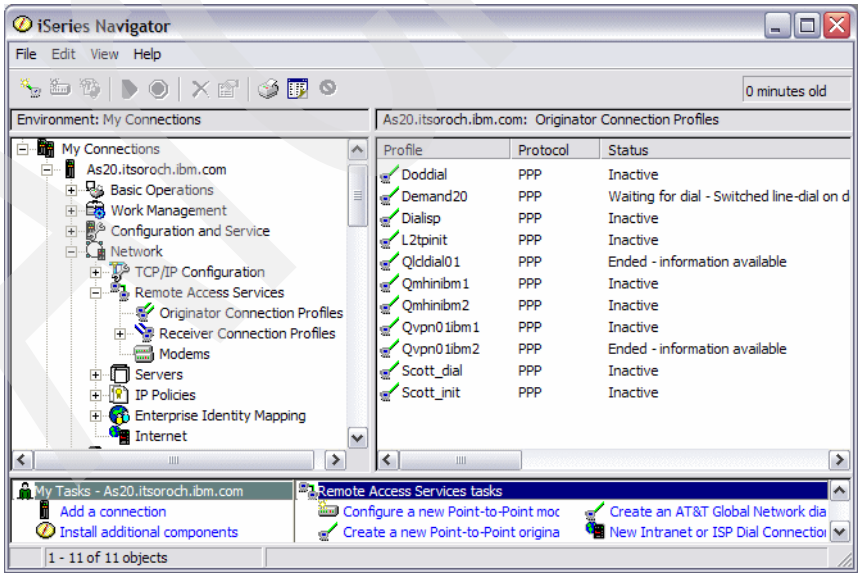


Figure 17-114 iSeries Navigator window

2. In the iSeries Navigator window, right-click **Demand24** → **Start**. Wait until its Status becomes Waiting for incoming call. This will be similar to the graphic shown in Figure 17-114 on page 585.
3. On the Command Entry panel on AS20, enter `FTP RMTSYS('172.23.60.1')` to initiate a traffic demand to the AS24 system, as shown in Figure 17-115.

```
 Command Entry                          AS20
                                                          Request level:   4
 Previous commands and messages:

   (No previous commands or messages)


Bottom
 Type command, press Enter.
 ===> FTP RMTSYS('172.23.60.1')
```

Figure 17-115 Command Entry panel for AS20: FTP to remote system AS24

4. In the iSeries Navigator window, click **Originator profile** on AS20 and press F5 (the refresh key). The Status of Demand20 is now Calling remote system, as shown in Figure 17-116.

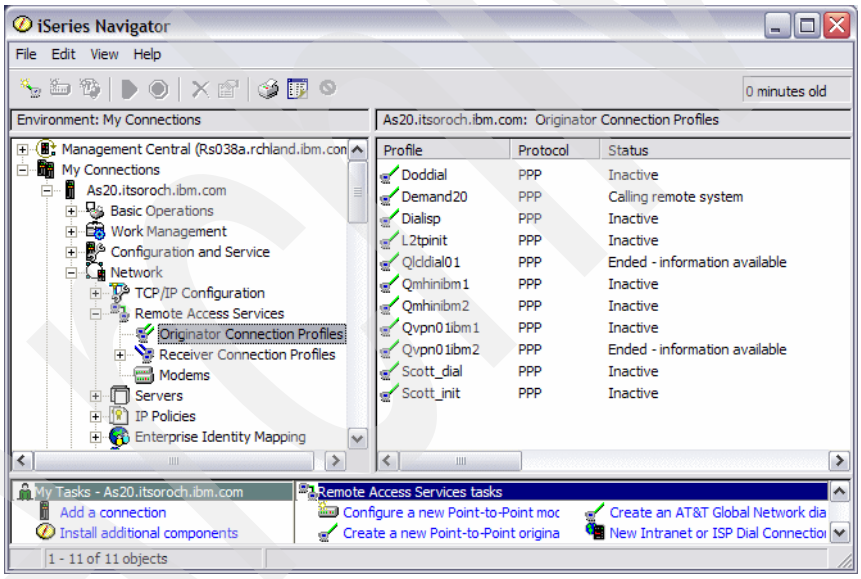


Figure 17-116 iSeries Navigator window: Status: Calling remote system AS24

If you continue to press F5. Demand20 Status will become Active. After the FTP application is ended, its Status becomes Waiting for dial - Switched line-dial on demand after 60 seconds of idle time.

17.4 System i RADIUS NAS

In this scenario, we show how to configure a System i RADIUS NAS server working with a remote RADIUS server. We also demonstrate how to configure a Windows XP client for the

PPP connection for this scenario. This scenario includes the procedure for setting up two different PC-based RADIUS servers.

17.4.1 Scenario overview

You might choose this scenario if these conditions apply:

- If you have a large network with many System i servers distributed in many branch offices around the country (or world) and you want to centrally administer the user IDs and passwords in one location. In this way you do not have to have people administering user IDs and passwords at every branch office.
- If you have people who move about the country (or world) and need to use your private network for access to secure applications. Of course, the individual can make long-distance toll calls back to the home-city System i, but that is not cost effective.
- If you have security or business reasons to track authentication, authorization, and accounting in a network that is distribute through a large geographic area.

Sample network configuration

Figure 17-117 shows the sample network configuration of this scenario.

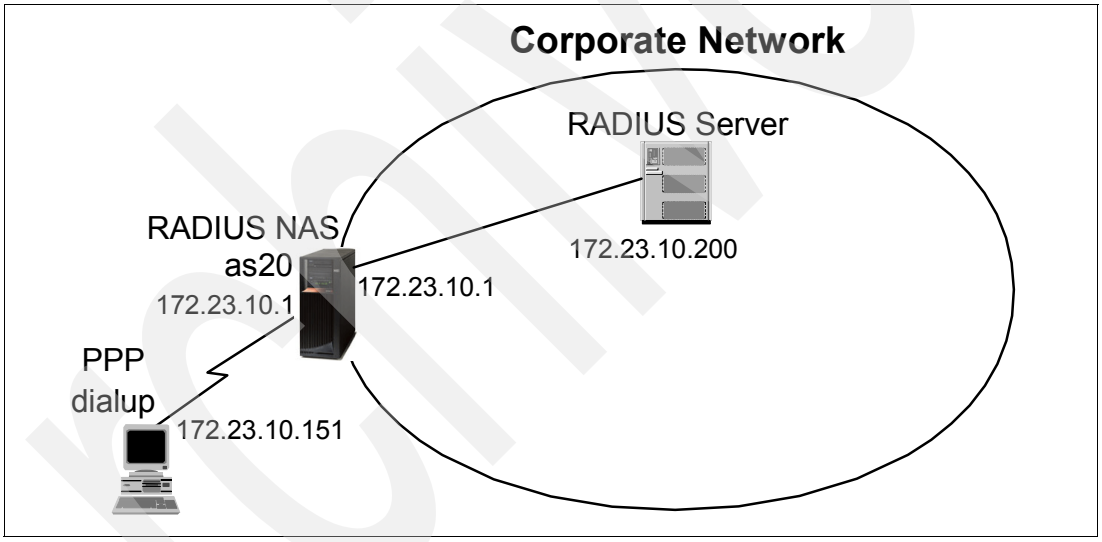


Figure 17-117 Sample network configuration: System i RADIUS NAS server with RADIUS server

17.4.2 Planning worksheet for System i RADIUS NAS with RADIUS server

Table 17-3 shows the planning worksheet for preparing the required parameters to configure a System i NAS working with a RADIUS server scenario. We have filled in our answers for each question in the adjacent Scenario answers column.

Table 17-3 Planning worksheet for i5/OS NAS server working with RADIUS server scenario

No.	Questions to create i5/OS NAS server working with RADIUS server scenario	Scenario answers
1	What is the IP address of the System i RADIUS NAS server on the LAN? What IP address will also be used for the unnumbered local IP address for the Receiver profile on the System i?	172.23.10.1

No.	Questions to create i5/OS NAS server working with RADIUS server scenario	Scenario answers
2	What is the IP address of RADIUS server?	172.23.10.200
3	What is the IP address assigned to the remote PPP client by the RADIUS server?	172.23.10.151
4	What is the password for the shared secret used between the System i RADIUS NAS and the RADIUS server?	secret Note: You may want to select something more difficult to guess than this.
5	What is the remote PPP client user ID administered by the RADIUS server?	alachmann
6	What is the remote PPP client password administered by the RADIUS server?	passv5r4
7	What is the name of the System i communication resource that will be supported by a Receiver profile?	CMN12
8	What is the port number for RADIUS authentication?	UDP 1645
9	What is the port number for RADIUS accounting?	UDP 1646

Note: By convention in newer RADIUS implementations UDP port 1812 is used for authentication and UDP port 1813 is used for accounting. In this scenario we used UDP 1645 for authentication and UDP 1646 for accounting.

17.4.3 Configuring the System i RADIUS NAS with RADIUS server

In this section we create the scenario of a System i RADIUS NAS working with a RADIUS server using the following steps:

- Step 1: Setting up a RADIUS server on your network
- Step 2: Installation and configuration of Media Online Italia's RADTAC
- Step 3: Create Network Access Server (NAS) on AS20.
- Step 4: Create Receiver profile for PPP connection on AS20.
- Step 5: Create a PPP connection profile on a Windows XP client.
- Step 6: Test a PPP connection.

Step 1: Setting up a RADIUS server on your network

To facilitate a quick deployment and ease of setup, we limit our RADIUS server scenarios to two Windows-based products: RADTAC and BSAC. Both products are publicly available on the Internet as 30-day trial versions. Both setup scenarios assume the information listed in Table 17-4.

Table 17-4 RADIUS scenario prerequisite

Item	Prerequisite
PC System minimum requirements	Pentium® III or later, 200 MB disk space available, 256 MB RAM
Operating system	Windows XP Professional with Service Pack 2 installed

Item	Prerequisite
Network attach	Physically attached to the same network as your System i NAS, only one network interface installed on the PC
IP setup	Static configuration for one IP address only, preferably on the same subnet as your System i NAS, identical name resolution available to System i and the PC

Follow the procedure in “Step 2: Installation and configuration of Media Online Italia’s RADTAC” on page 589. The choice of this RADIUS server or some other server is yours to make.

Step 2: Installation and configuration of Media Online Italia’s RADTAC

RADTAC is a shareware RADIUS server from Media Online Italia. Its 30-day trial version is available from http://www.radtac.com. Look for the install program radtac756.exe (or the most current Evaluation version), which is about 12.3 MB. This RADIUS server runs on Windows 2000, 2003, XP, 98, and ME.

To start the installation, log on to Windows with admin privileges, download radtac756.exe (or the most current Evaluation version), and start the self-extracting installation file.

After the self-extracting archive has unpacked to a temporary directory on your disk, the RADTAC initial installation window opens. Click **Next** to follow a standard Windows installation process.

Depending on your system speed, the installation can take several minutes.

With successful installation, RADTAC adds two shortcut icons to your desktop: RadTac 2000 Server Radius and RadTac 2000 Server Administrator. To configure the RADTAC server, follow these steps (using Table 17-3 on page 587 for many of the values used in these steps):

1. To start and configure RADTAC, double-click the RadTac 2000 Server Radius icon on your desktop.
2. After the RADTAC server successfully starts, double-click the RadTac 2000 Server Administrator icon on your desktop and select **Options** → **General Options**.

3. This opens the RADTAC general Options window, as shown in Figure 17-118.
   a. Enter:

      Local Host Address 172.23.10.200
      IP Address for Log display 172.23.10.200

   b. Select a readable font for your console.
   c. Select **Crypted** to activate password encryption.

   Click the **Validation Mode** tab to continue.

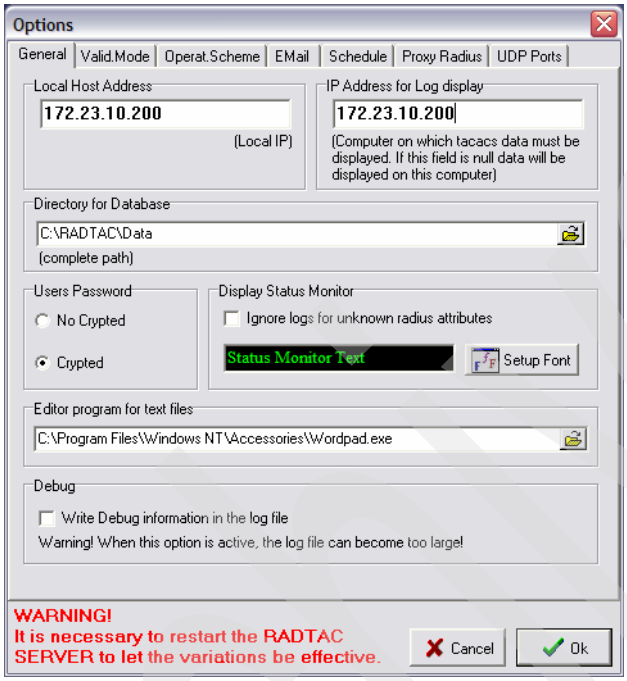


Figure 17-118 RADTAC general Options setup window 1

4. On the Validation Mode tab (Figure 17-119), select **Windows 2000 - Windows XP Active Directory®** (to keep things simple in our scenario). Enter `activedir` for the Active Directory Domain Name. Optionally, you may use Windows NT 4 Users Database or Internal Users Database (Windows 9x) if applicable to your network. Click the **Operating Scheme** tab.

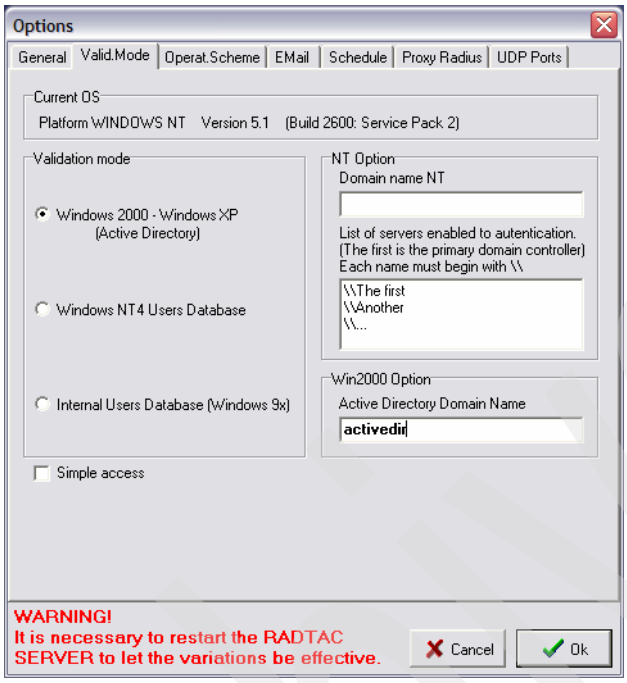


Figure 17-119 RADTAC general options setup window 2

5. Select **One RadTac server connected to all the router of the network**, as shown in Figure 17-120. Click **OK** to continue.

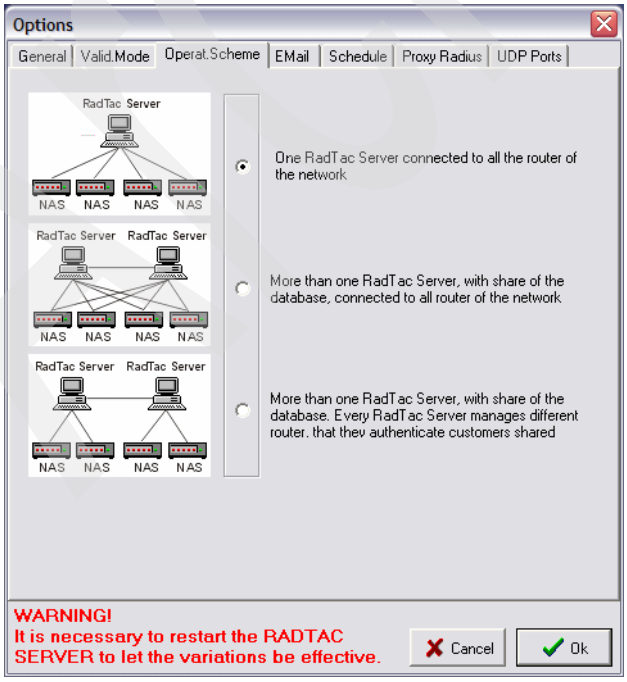


Figure 17-120 RADTAC general options setup window 3

6. Switch your window to the RADTAC Server and click the **Stop** and **Start** buttons to restart RADTAC.
7. You may now add users to your server. To do this, switch back to the RADTAC Administrator. Click the plus sign (**+**) at the top of the window to add an initial user, as shown in Figure 17-121.

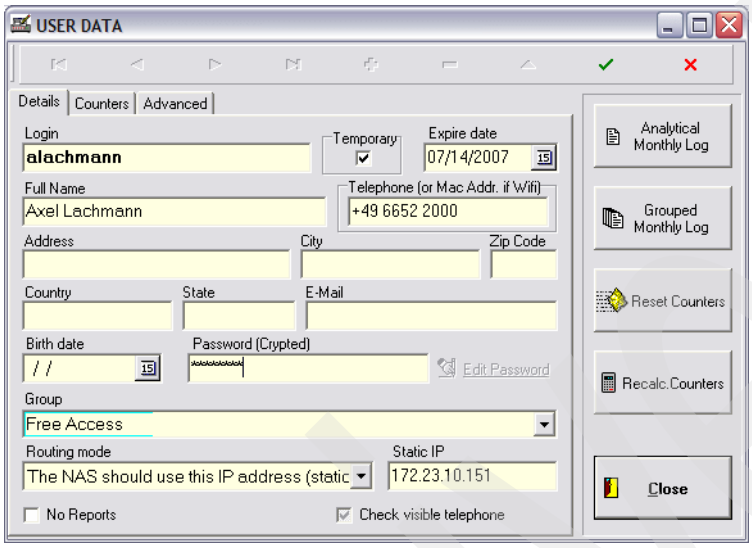


Figure 17-121 RADTAC user creation

Follow these steps to add a new user to the RADTAC database:

a. Enter your desired user ID (we used `alachmann`) as your User Login name. Optionally, add additional information about this user.

b. Choose **Free Access** in the Group field.

c. In the Routing mode field, select **Static IP** to set up an IP address per user for your System i server to assign. Enter an IP address that is part of your local System i IP address range. In this case we specified `172.23.10.151`.

   In this scenario we assume that the RADIUS server, not the System i RADIUS NAS, will assign the IP addresses.

> **Tip:** The RADTAC RADIUS server gives you three options for assigning IP addresses to the remote PPP client at this point:
>
> ```
> The NAS should select an address for the user
> The NAS should allow the user to select an address
> The NAS should use this IP address (static IP)
> ```
>
> Most likely your choice would be for the System i RADIUS NAS to select the proper address for the PPP client, as this is dependant on the IP addresses that are locally administered.
>
> As an example, if a user dials into Rochester today, you would want an IP address that is routable in Rochester. If tomorrow the same user dials into Makuhari, Japan, you would most likely want a different locally routable IP address assigned. This is best done by the RADIUS NAS (your System i) in most situations.

d. Enter the password `passv5r4` (see Table 17-3 on page 587 item 6) in the Password (Crypted) field. This field is case sensitive, so you should use lowercase letters. You must click **Edit Password** to enable the editing of the Password (Crypted) field.

e. Select the check mark icon at the top-right of your user entry window when you are finished. You can repeat this process for as many users as you like, but at least one user is required for RADTAC to function properly.

f. When you are finished adding users, click **Close**.

8. The final step in our RADIUS setup is to define your System i RADIUS NAS as a router to RADTAC. To do this, select **File** → **Routers and Proxy Radius** from the RADTAC administration menu, as seen in Figure 17-122.

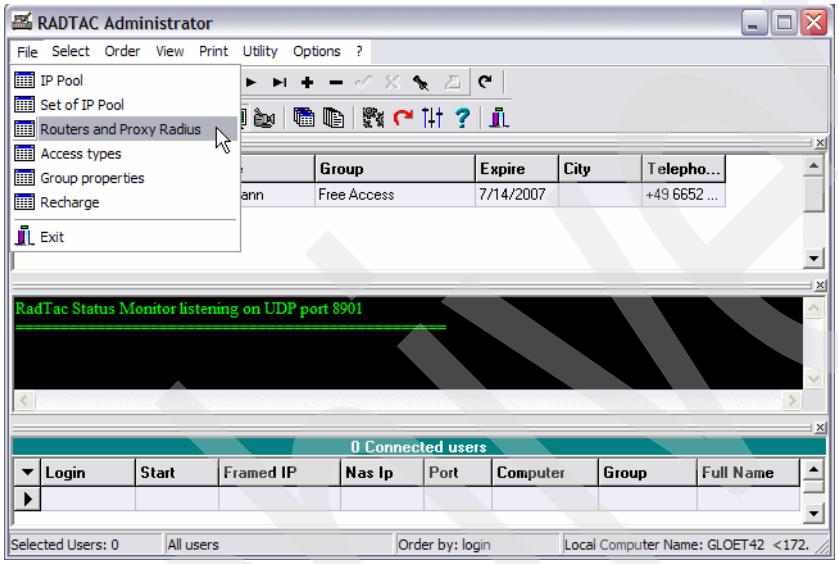


Figure 17-122 RADTAC add router

9. This opens the ROUTERS and PROXY RADIUS configuration window, as seen in Figure 17-123. Follow these steps to add a router record for the i5/OS RADIUS NAS:
   a. Click the plus sign (**+**) at the top of the window.
   b. Enter the IP address of your i5/OS RADIUS NAS, `172.23.10.1`, into the IP Address field.
   c. Enter your case-sensitive shared secret (see Table 17-3 on page 587 item 4) of `secret`.
   d. Enter as a Description the name of your System i platform, such as `AS20`.
   e. To save your changes, click the check mark at the top-right, then click **Close**.

Note: You may repeat this process for as many System i servers that you need to define as an NAS in your network.

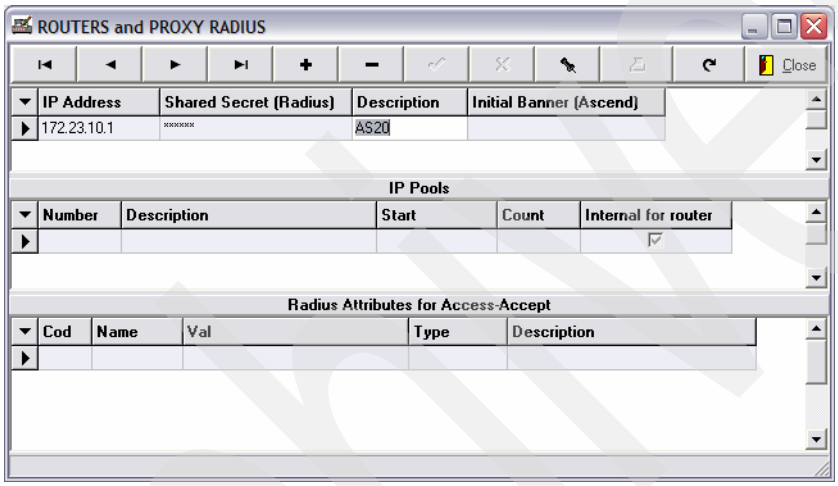


Figure 17-123 RADTAC remote RADIUS NAS IP range setup

Step 3: Create Network Access Server (NAS) on AS20

Remote Access Services must be enabled for RADIUS in order for PPP or L2TP to use it. You have the option of enabling some or all of the RADIUS functions. If the RADIUS NAS server has not been configured, a configuration wizard is initiated when Enable RADIUS Network Access Server connection is selected. Separate RADIUS servers can be used for authentication and accounting.

In our scenario, we have only one RADIUS server at IP address 172.23.10.200 for both authentication and accounting, as is depicted in Figure 17-117 on page 587.

1. In the iSeries Navigator window, expand **Network**. Right-click **Remote Access Services** and choose **Services**, as shown in Figure 17-124.

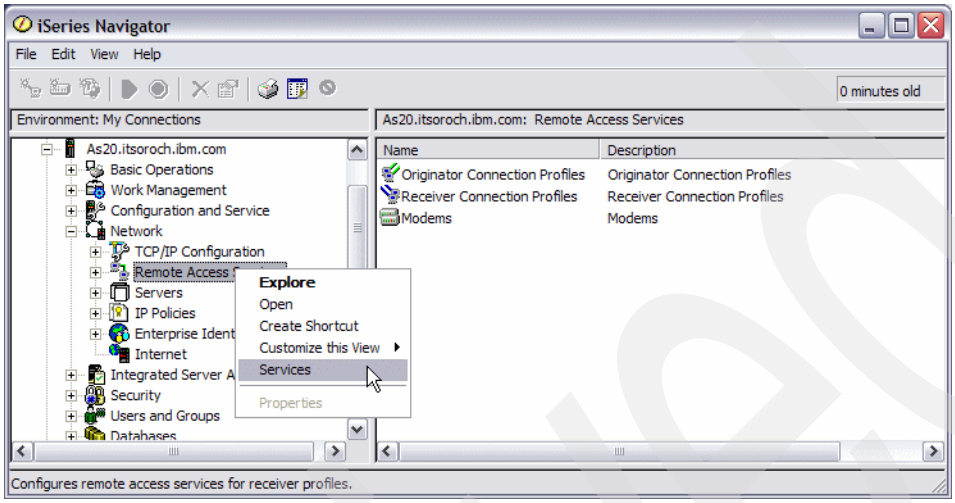


Figure 17-124 NAS RADIUS enablement

2. In the Remote Access Server for Receiver Profile window, check all four check boxes, as shown in Figure 17-125.

> **Note:** If RADIUS NAS settings have not been previously defined, when you check **Enable RADIUS Network Access Server connection**, the RADIUS NAS Settings panel will automatically be displayed. If this occurs, when you have completed your configuration you will need to select the remaining check boxes, as shown in Figure 17-125.

   - Enable RADIUS Network Access Server connection.
   - Enable RADIUS for connection accounting.
   - Enable RADIUS for authentication.
   - Enable RADIUS for TCP/IP address configuration.

   Click the **RADIUS NAS Settings** button to continue.

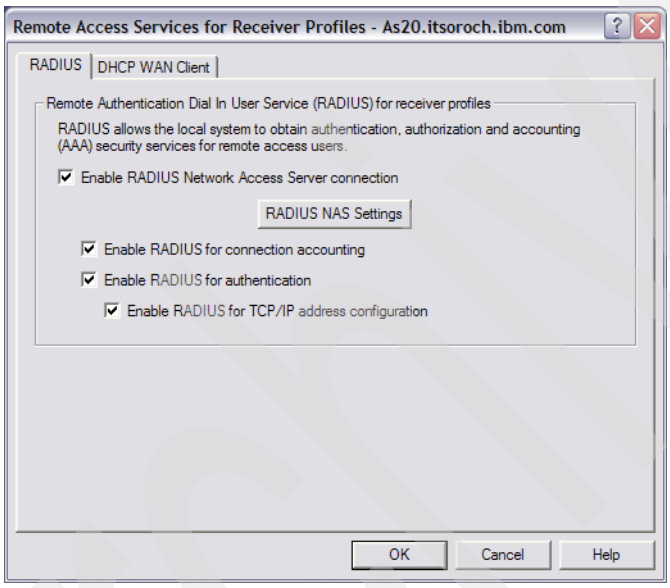


Figure 17-125 Remote Access Server for Receiver Profile window

3. In the RADIUS NAS Properties window, enter a description if needed. In our scenario, we keep the default values, as shown in Figure 17-126. Click the **Authentication Server** tab.

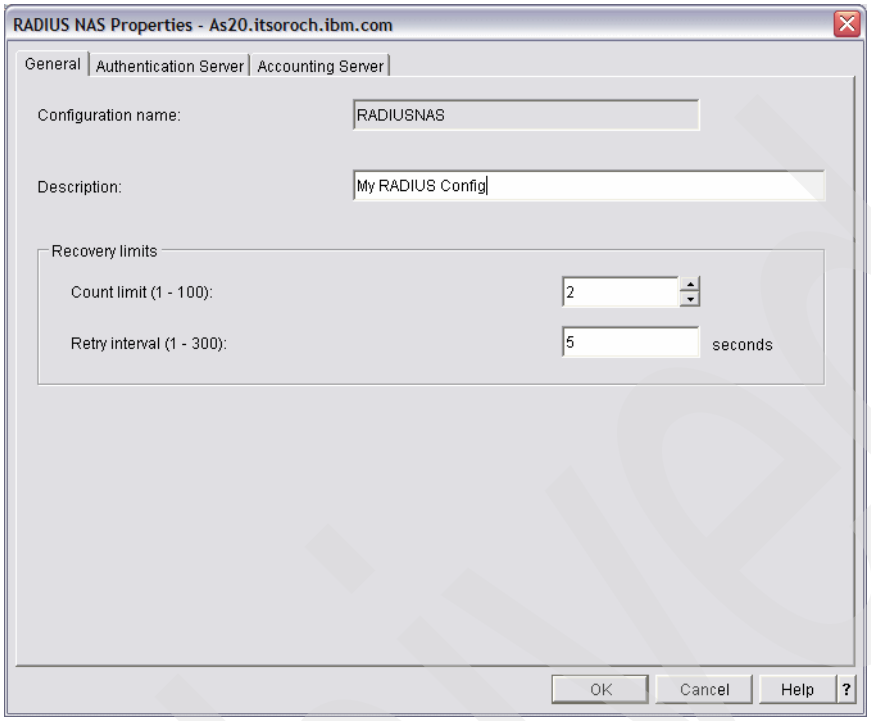


Figure 17-126 RADIUS NAS Properties window

4. In the RADIUS NAS Properties window, click **Add**. Use the values found in Table 17-3 on page 587 to complete the RADIUS Authentication Server Configuration window, as shown in Figure 17-127:

   Local IP address 172.23.10.1
   Server IP address 172.23.10.200
   Password secret
   Port 1645

   Click **OK** to continue.

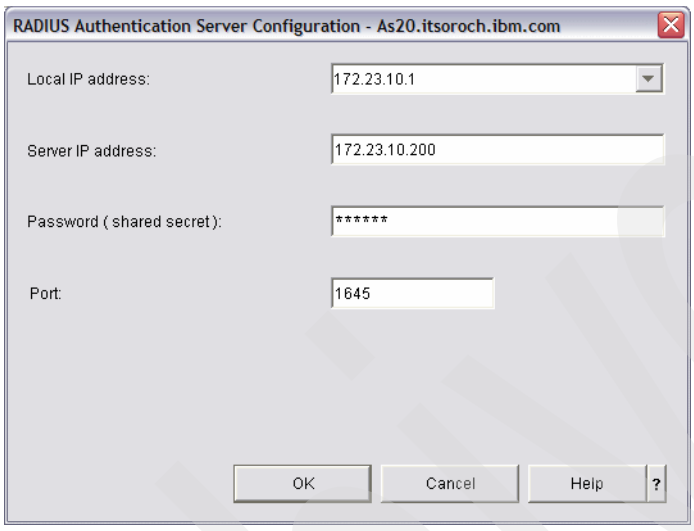


Figure 17-127 RADIUS Authentication Server Configuration window

5. In the RADIUS NAS Properties window, click the **Accounting server** tab. Click **Add**.

6. In the RADIUS Accounting Server Configuration window (Figure 17-128), enter:

   Local IP address 172.23.10.1
   Server IP address 172.23.10.200
   Password secret
   Port 1646

   Click **OK** to continue.

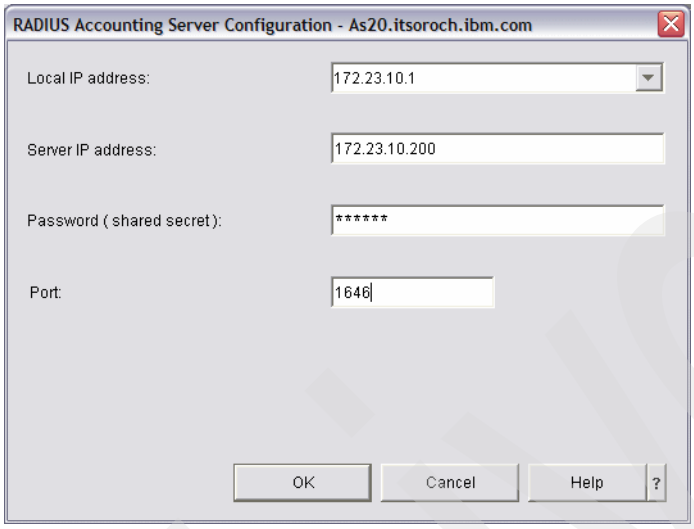


Figure 17-128 Adding an accounting server

7. In the RADIUS NAS Properties window, click **OK**.

Now you can use a remote RADIUS server to provide authentication and accounting type services on your System i when using PPP or L2TP.

Step 4: Create Receiver profile for PPP connection on AS20

Now that you have set up RADIUS and NAS, the last step is to set up a PPP connection to use these services. In our scenario, we create a dial-up connection from a remote client (for example, a PC) to your System i. To be able to do that, you must create a PPP Receiver profile on System i AS20 with iSeries Navigator:

1. In the iSeries Navigator window, expand **Network** → **Remote Access Services** then right-click **Receiver Connection Profiles** and select **New Profile**, as shown in Figure 17-129.

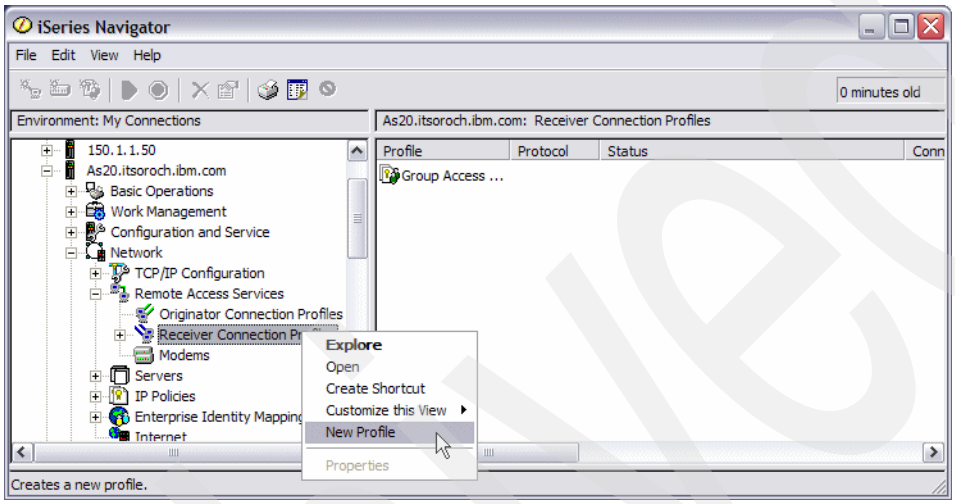


Figure 17-129 iSeries Navigator window

2. In the New Point-to-Point Connection Profile Setup window (Figure 17-130) choose:

   Protocol type selection PPP
   Connection type Switched line
   Operation mode Answer

   Click **OK** to continue.

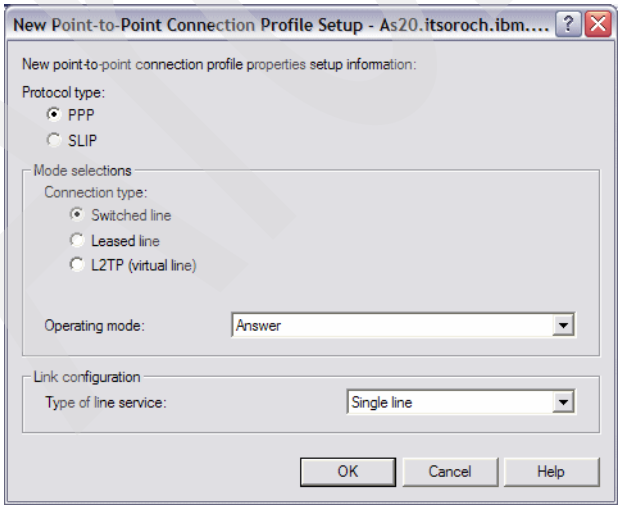


Figure 17-130 PPP profile initial setup

3. In the New Point-to-Point Profile Properties window, enter `Dialin` in the Name field, as shown in Figure 17-131. If you want the profile to start automatically with TCP then check **Start profile with TCP**. Click the **Connection** tab.

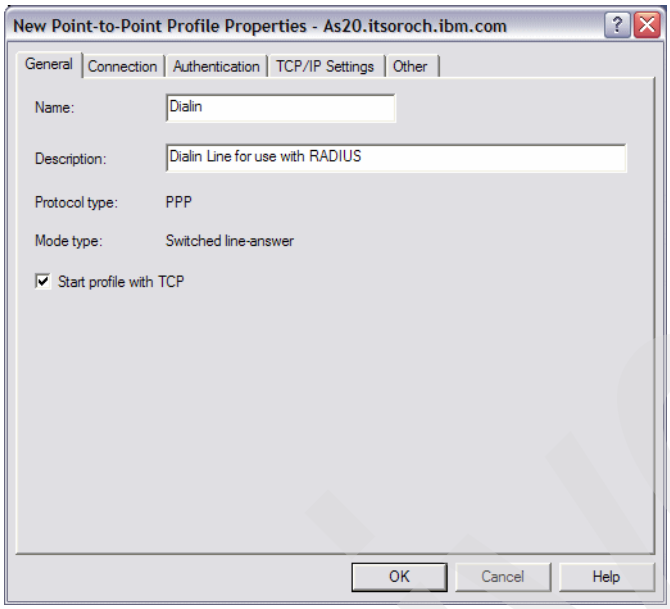


Figure 17-131 PPP general properties

4. Choose **dialrad** in the Link configuration Name field, as shown in Figure 17-132. Click **New** to continue.

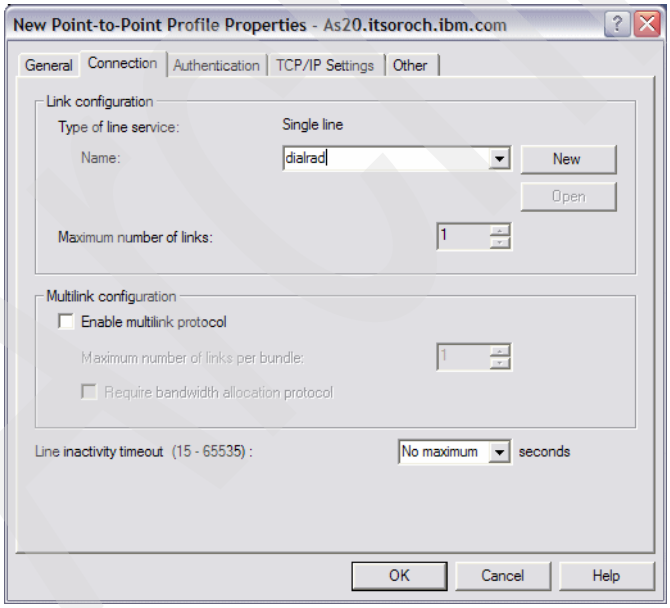


Figure 17-132 New Point-to-Point Profile Properties window

5. This opens the New Line Properties window. Select **CMN12** in the Hardware resource field, as shown in Figure 17-133. Click the **Modem** tab.

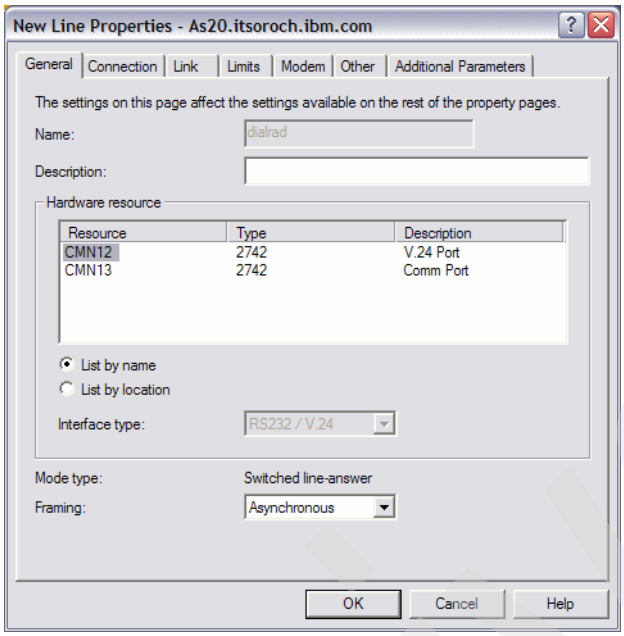


Figure 17-133 Dialrad properties window

6. Select the appropriate external modem from the pull-down menu, as shown in Figure 17-134. Click **OK** to continue.

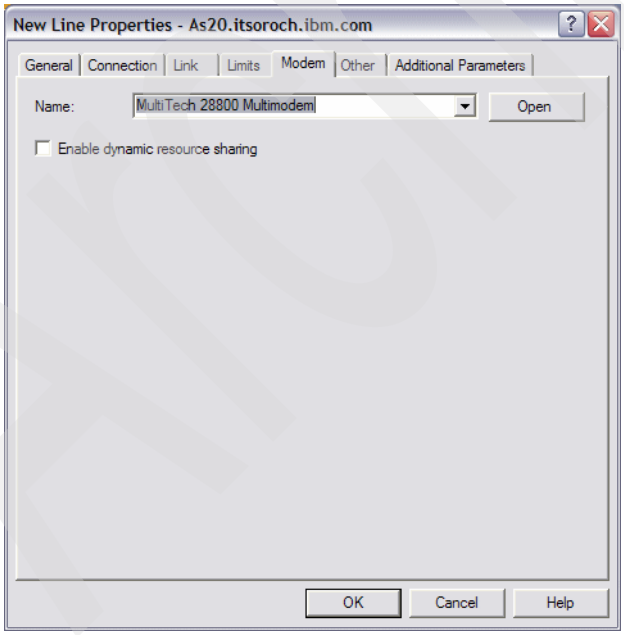


Figure 17-134

7. In the New Point-to-Point Profile Properties window, click the **Authentication** tab. Check **Require this iSeries server to verify the identity of the remote system**. Select **Authenticate remotely using a RADIUS server** and check **Use RADIUS for connection auditing and accounting** to enable these two functions, as shown in Figure 17-135. Click the **TCP/IP Settings** tab.

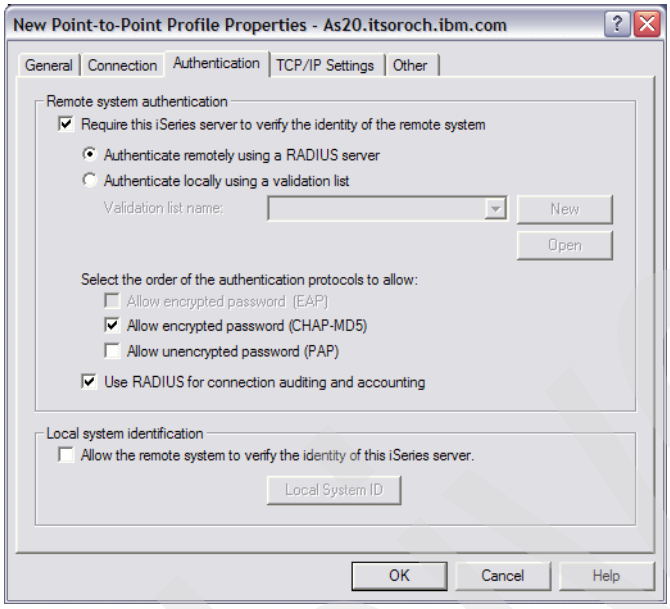


Figure 17-135 New Point-to-Point Profile Properties window

8. Choose **172.23.10.1** in the Local IP address field and **RADIUS** for IP Address assignment method, as shown in Figure 17-136. Click **OK**.

> **Tip:** In a larger and more complex network you might not want the RADIUS server to be dictating the IP address of the PPP client. Depending on how your IP network is configured it might be better for the IP address of the PPP client to be determined automatically by the System i RADIUS NAS. The reason? Your local *branch office System* i can assign IP addresses that are allocated out of the local subnet to ensure that the address is routable not only within the branch office but to the rest of the corporate network.
>
> Look again at Figure 17-136 and see that your local System i RADIUS NAS could also be assigning IP addresses from a fixed IP address pool or via the DHCP WAN client service.
>
> Most likely in this scenario you would not choose to assign an IP address based on the remote system's user ID, as this information is kept on the RADIUS server and not on your System i RADIUS NAS.

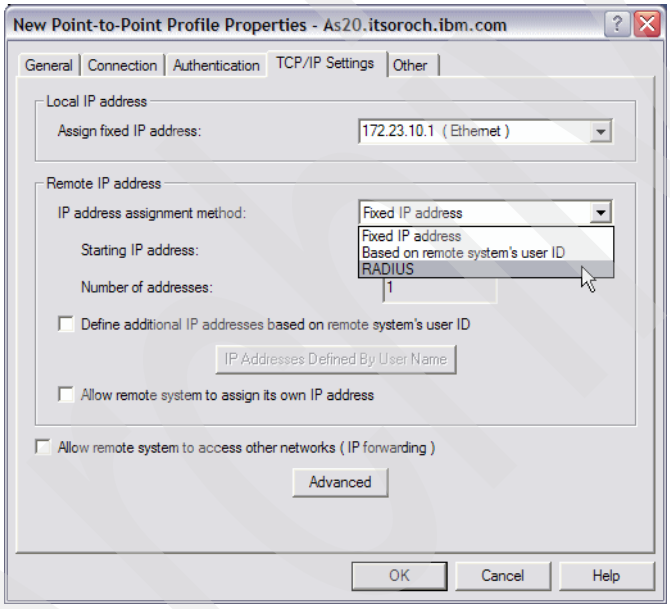


Figure 17-136 New Point-to-Point Profile Properties window

Step 5: Create a PPP connection profile on a Windows XP client

In this step, we create a PPP connection scenario on the Windows XP side:

1. Select **Start** → **Control Panel**. In the Control Panel window, double-click **Network connections**.
2. In the Network connections window under the Network Tasks pane, select **Create a new connection**.
3. In the Welcome to the New Connection Wizard, click **Next**.
4. In the New Connection Wizard - Network Connection Type window, check **Connect to the network at my workplace** and click **Next**.
5. In the New Connection Wizard - Network Connection window, check **Dial-up connection** and click **Next**.

6. In the New Connection Wizard - Connection Name window, enter the name of the system you will be dialing into and click **Next**.
7. In the New Connection Wizard - Phone Number to Dial window enter your dial-up phone number in the phone number field and click **Next**.
8. In the New Connection Wizard - Connection Availability window choose either **Anyone's use** or **My use only** as applies, and click **Next**.
9. In the New Connection Wizard - Completing the Network Connection Wizard, select to **Add a shortcut to this connection to my desktop** and click **Finish**.
10. The new Connection should appear. Click **Properties**.
11. In the (connection name) Profiles window, click the **Security** tab. In the Security window, select **Advanced** and click **Settings**.
12. In this scenario, **CHAP** must be chosen, as shown in Figure 17-137, because the RADIUS server is configured to accept CHAP authentication.

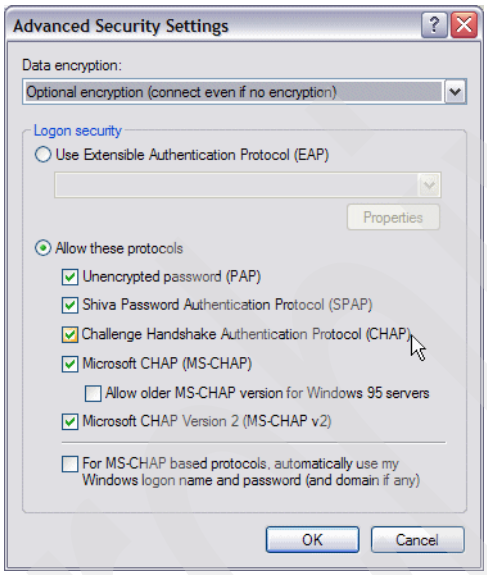


Figure 17-137 Windows XP client: Advanced Security Settings window

Step 6: Test a PPP connection

In this step, we explain how to test the PPP connection:

1. Double-click the **RadTac 2000 Server RADIUS** icon and select **start server** to bring up the RADIUS server.

2. In the iSeries Navigator window, right-click the **Dialin** Receiver profile and choose **Start**, as shown in Figure 17-138.

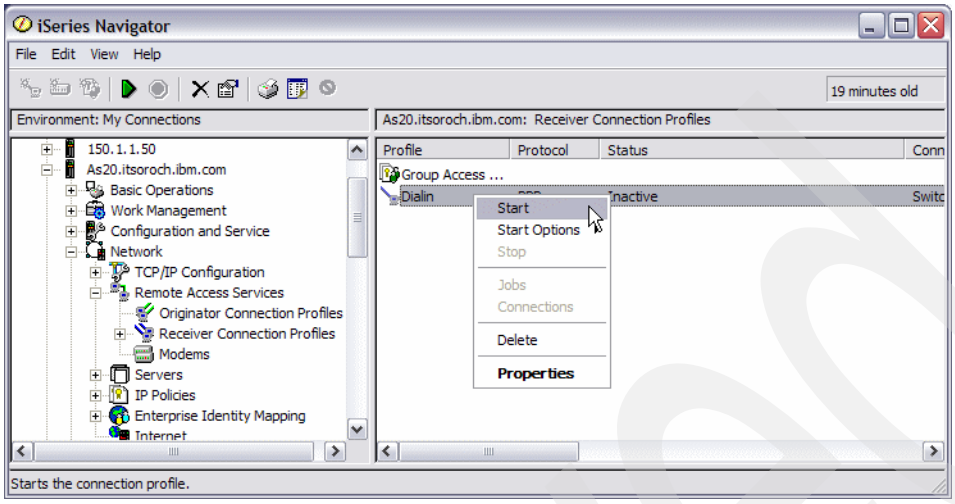


Figure 17-138 iSeries Navigator window

3. On the Windows XP client, double-click the Dialup Connection icon that you created in "Step 5: Create a PPP connection profile on a Windows XP client" on page 604. Enter the appropriate user name and password and click **Dial**. Wait until authentication is completed.

4. If a PPP connection is established, you see the messages on RADTAC server window shown in Figure 17-139. This window contains the PPP user information.

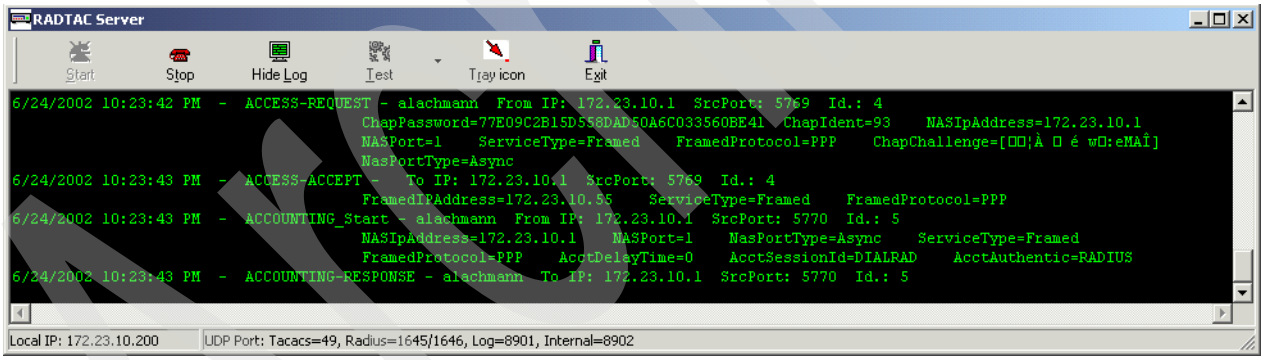


Figure 17-139 RADTAC server window

5. To see the difference between a straight PPP dialup and one using RADIUS, open the user data window in your RADTAC administrator after a few successful dial-ups, as shown in Figure 17-140.

   In the yellow field at the bottom of the window you now see all of the users' accounting data in one central location on your RADIUS server, no matter what RADIUS NAS the user has used to dial up to.

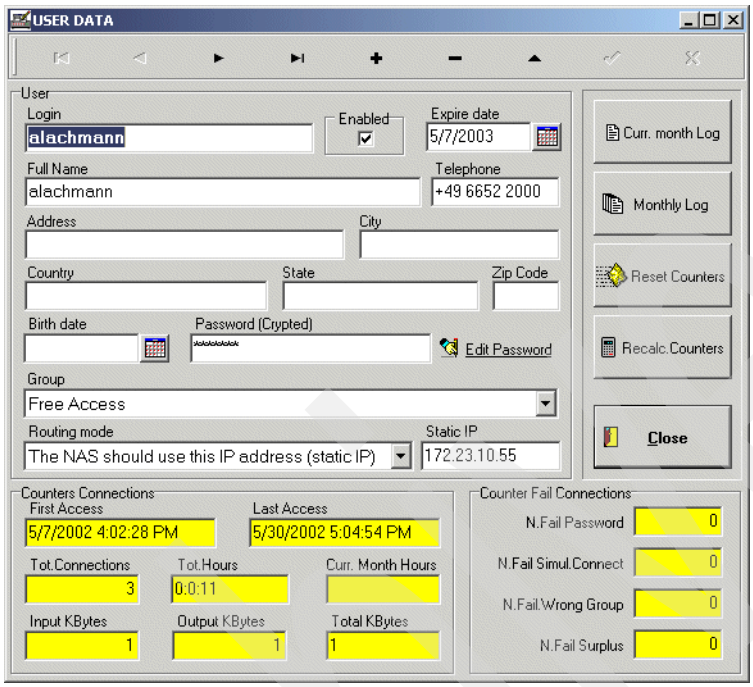


Figure 17-140 RADTAC: the results

Problem determination

A great tool is available via System i Remote Access Services: the CLOG.

If the PPP connection cannot be established, use the following procedure to view the job information for a PPP connection.

In the unlikely case that something is not working quite the way it should, a great tool is available via the System i Remote Access Services: the Communications Log (CLOG). To get to this spool file, right-click your PPP profile in iSeries Navigator and select **Connections**, as seen in Figure 17-141.

Tip: By default, the Communication Log (CLOG) will be generated only if the connection fails. That is, the default behavior is to log messages only if there are errors. If you want to force the CLOG to generate a file in all cases, right-click the **Originator profile** and select **Start Options** → **Log Messages** from the context menu. This should be done prior to starting the Originator profile.

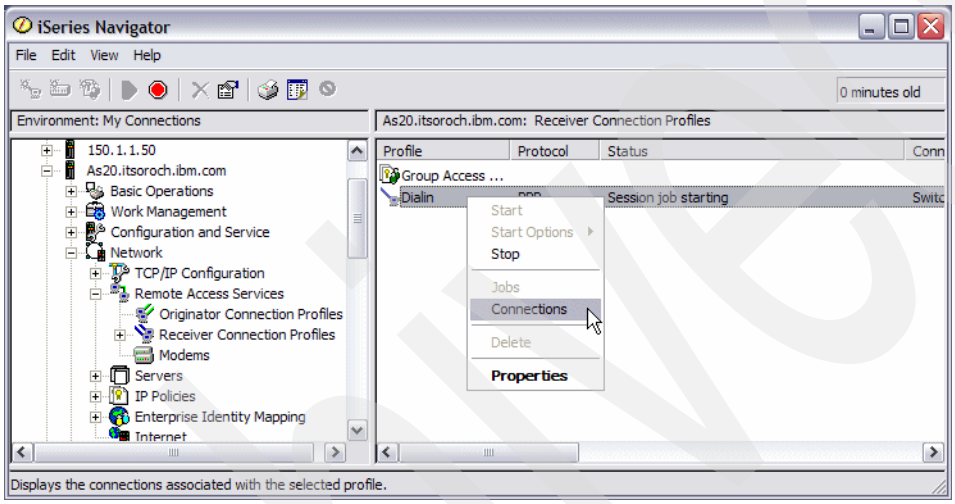


Figure 17-141 PPP connections

This opens the Connections window shown in Figure 17-142.

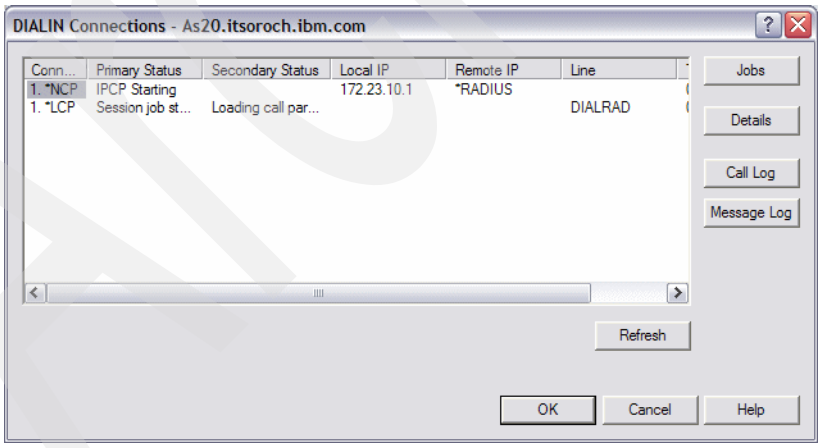


Figure 17-142 PPP connections selection

Select the connection that you want to look at and click **Call Log** to bring up the CLOG for that connection, as seen in Figure 17-143.

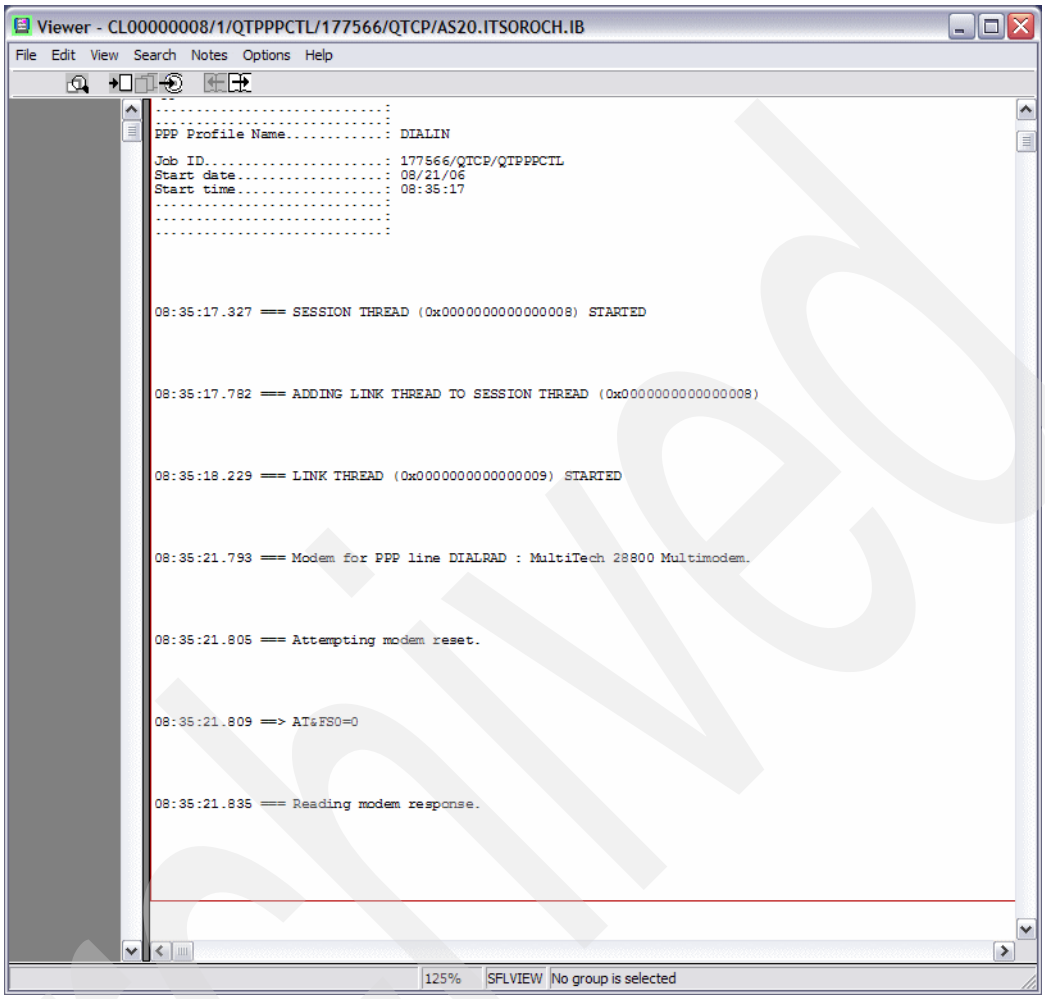


Figure 17-143 PPP CLOG sample

The CLOG displays a detailed view of all communications involved in a PPP job, so no matter what your PPP problem might be, the chances are always very high that you can find useful information in this log.

Starting in i5/OS V5R4, an additional tool is available via the System i Remote Access Services: the Message Log (MLOG). Select the connection that you want to look at and click **Message Log** to bring up the MLOG for that connection, as seen in Figure 17-144.

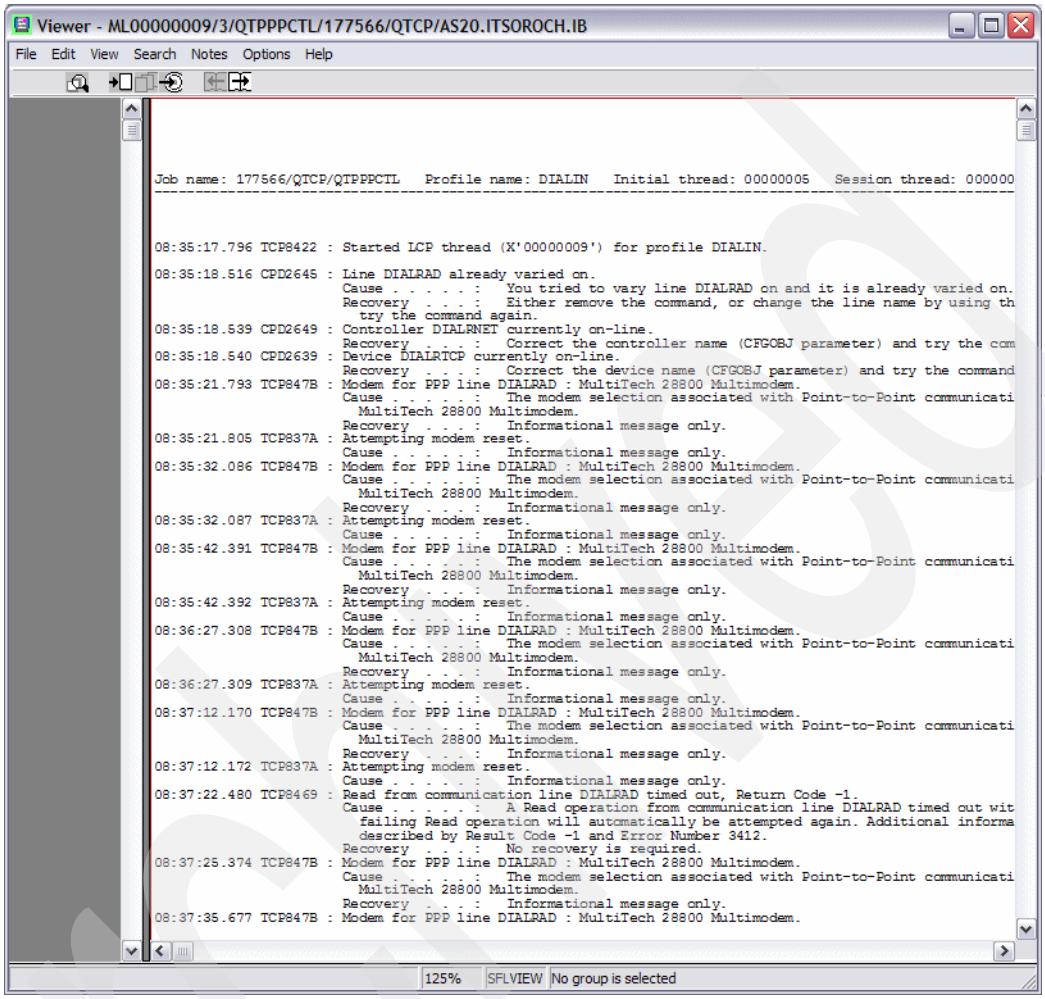


Figure 17-144 PPP MLOG example

17.5 Assigning an IP address to PPP client from DHCP server

A new feature of the DHCP server on System i, introduced in V5R1, is DHCP WAN Client support. The DHCP WAN Client acts as a DHCP client, requesting IP addresses from the DHCP server address pool on behalf of the application that called it.

An example of an application that can use the DHCP WAN Client is a Point-to-Point Protocol (PPP) Receiver profile.

PPP is a protocol that provides a standard method for transporting multiprotocol datagrams over point-to-point links. PPP hosts can obtain an IP address by connecting to a System i that has a defined PPP Receiver profile.

Starting with OS/400 V5R1 and i5/OS you can define a PPP Receiver connection profile that will assign an IP address to clients using the DHCP server address pool. This gives the administrator greater flexibility in managing a precious resource: the IP addresses.

Note: Before OS/400 V5R1 and i5/OS the system administrator defined within the PPP profile a single IP address or a range of IP addresses to be assigned to PPP hosts. Similarly, LAN-connected hosts could obtain an IP address from a System i via DHCP.

Thus, an administrator had to separately define address pools to serve both PPP and LAN connected hosts. There was no mechanism for sharing free addresses from either pool.

17.5.1 Scenario overview

You might choose this scenario if these conditions apply:

- When the number of remote clients was small, you found it easier to hard-code the assignment of IP addresses by remote user ID. Now that your network is bigger you are looking for a way to automate the assignment of the remote client IP address.
- You had been assigning remote client IP addresses out of a separate (from the DHCP server) pool of IP addresses. This did not allow you to share IP addresses within a single pool leased from the System i DHCP server.

Sample network configuration

Figure 17-145 shows the sample network configuration of this scenario.

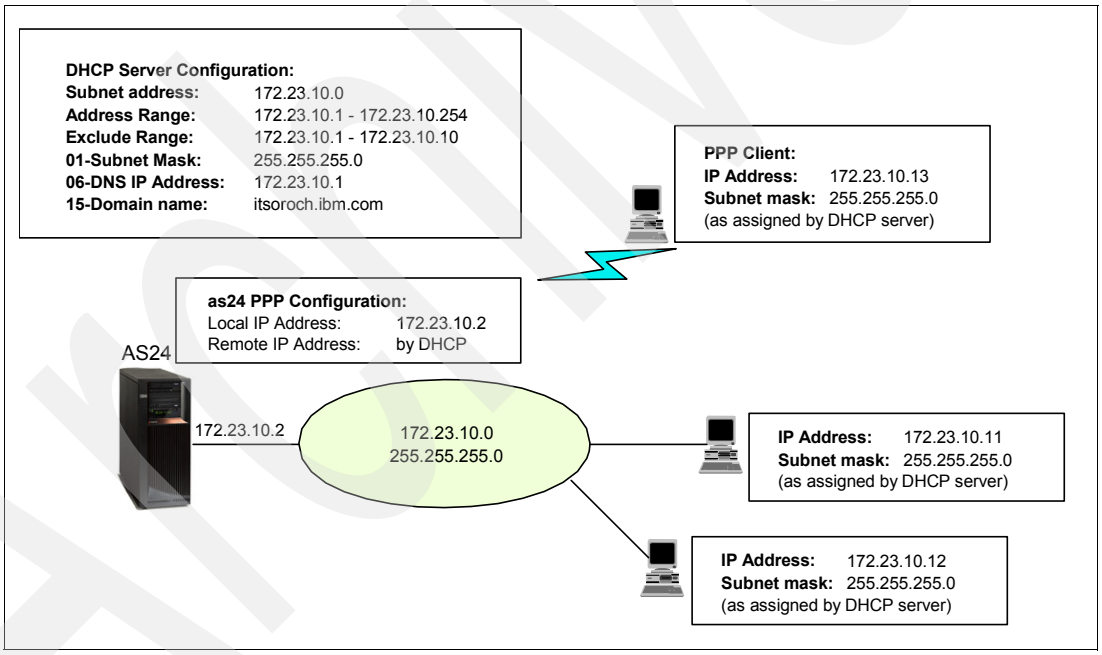


Figure 17-145 PPP Receiver profile requests IP address via DHCP WAN client to DHCP server

Assumptions

The network used in this scenario has the following characteristics:

- There is a single physical subnet.
- The System i has only one network adapter on the LAN, with only one IP interface.
- The subnet IP address is 172.23.10.0, the netmask is 255.255.255.0. The subnet mask enables the DHCP server to service 254 clients. The range of IP addresses 172.23.10.1 to 172.23.10.10 is reserved for other servers and will be excluded from the addressing pool.
- There is a single System i DHCP server that will allocate the IP addresses in the network.

- There are no routers or bridges in this network.
- The System i and its DHCP server have already been configured using the steps outlined in 15.1, “DHCP: One physical network, one logical network, one DHCP server” on page 270. That is, we assume that we have a working DHCP server already configured and started that is leasing IP addresses for LAN clients.
- The System i already has a PPP Receiver profile created and configured. We will modify that PPP Receiver profile to assign remote IP address to the client using the DHCP server.

17.5.2 How-to

In this scenario we create and enable our System i to receive a call from a PPP client, causing the DHCP WAN client to request a lease from the System i DHCP server, by using the following steps:

- Step 1: Enable Remote Access Services for DHCP on your System i
- Step 2: Modify your System i PPP Receiver profile
- Step 3: Test your configuration

Step 1: Enable Remote Access Services for DHCP on your System i

To configure the DHCP proxy client on your System i, perform the following steps:

1. Start iSeries Navigator.
2. Expand your System i connection. You may be asked to enter your user ID and password.
3. Expand **Network**.
4. Right-click **Remote Access Services** and choose **Services** (Figure 17-146).

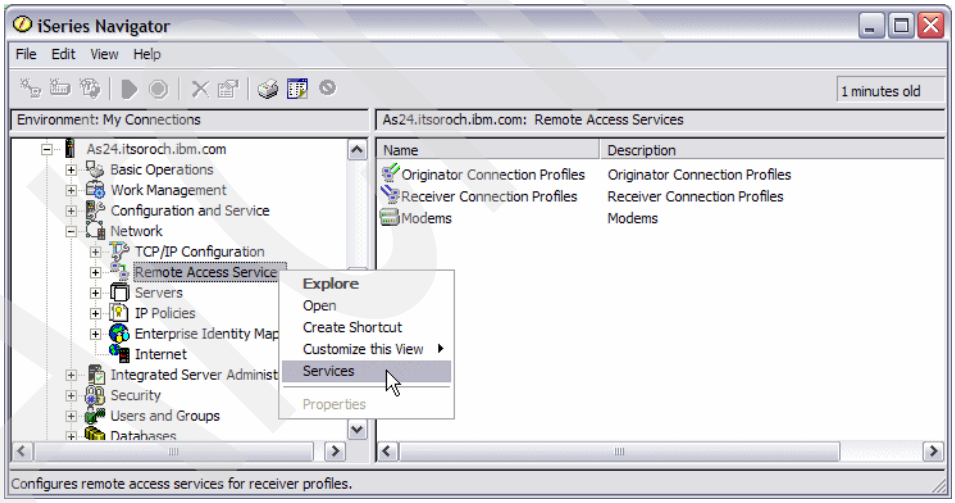


Figure 17-146 Remote Access Services: Services

5. The Remote Access Services for Receiver Profiles is displayed. Select the **DHCP WAN Client** tab as shown in Figure 17-147. Check the **Enable DHCP WAN client** check box. Click **Add**.

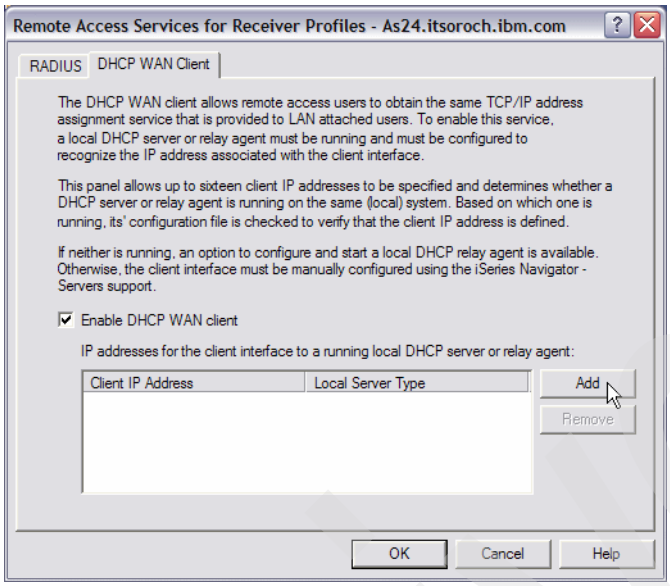


Figure 17-147 Remote Access Services for Receiver Profiles: DHCP WAN Client tab

6. The DHCP WAN Client Local Interface Definition window opens (see Figure 17-148). Select the local IP address that will be used by the DHCP proxy client. Click **OK** to close the DHCP WAN Client Local Interface Definition window.

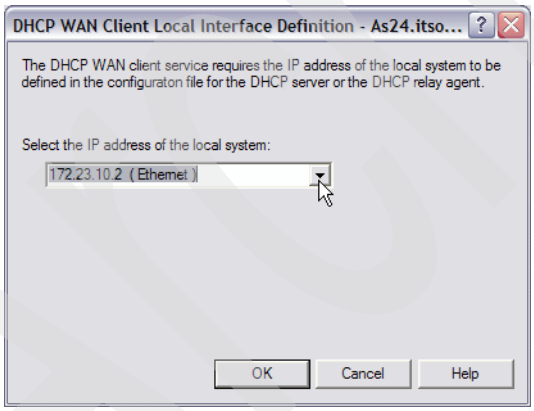


Figure 17-148 DHCP WAN Client Local Interface Definition

Note: The DHCP proxy client places this IP interface into the RELAY AGENT field (giaddr field) of the DHCP packet that will be sent to the DHCP server to request an IP address. Thus, the server has a clue to what address pool to select an IP address from to assign to the client.

Note: The same IP address will be used in the Local Address field of the PPP Receiver Connection profile.

7. Click **OK** in Remote Access Services for Receiver Profiles to save the configuration.

Step 2: Modify your System i PPP Receiver profile

Now modify the PPP Receiver profile's TCP/IP Settings tab to assign the remote IP address using the DHCP server:

1. Expand **Network** → **Remote Access Services** and then select **Receiver Connection Profiles**.
2. Make sure that the status of the PPP Receiver profile is Inactive. (If not, right-click the **PPP** Receiver profile and choose **Stop** from the context menu.) Do not continue until a refresh of the display shows that the Receiver profile is Inactive.
3. Right-click the **PPP** Receiver profile and choose **Properties**.
4. Select the **TCP/IP Settings** tab, as seen in Figure 17-149.
   a. Under Local IP address (local to the iSeries) choose **172.23.10.2** for Assign fixed IP address.

   > **Tip:** Make sure that the IP address you select here was already defined in the DHCP client proxy configuration (Figure 17-148 on page 613). Also, by specifying the IP address of a local LAN interface for the local System i side of a PPP connection, we are creating what is called an unnumbered network. See "Tip: An unnumbered network" on page 89 for more information.

   b. Under Remote IP address (remote from the System i) choose **DHCP** as the IP address assignment method.
5. Click **OK** to save the configuration.

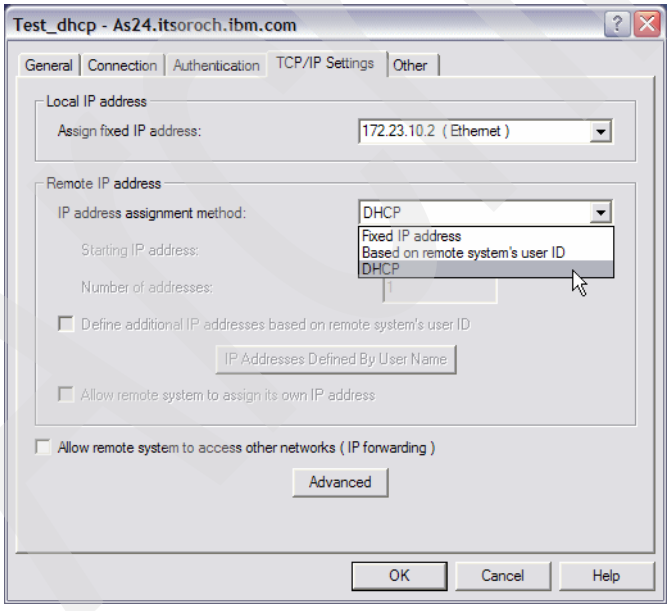


Figure 17-149 Receiver Connection Profile properties: TCP/IP Settings tab

Step 3: Test your configuration

You are now ready to make the call and test your connection:

1. Start your PPP Receiver profile on the System i. Right-click the profile name and select **Start** from the context menu. Ensure that the Status turns to Waiting for incoming calls.

2. Make the call from your Windows client by double-clicking the connection document.
3. Using iSeries Navigator, monitor the connection as seen in Figure 17-150. You should see the status of the Receiver profile going through a series of states ending with Active.

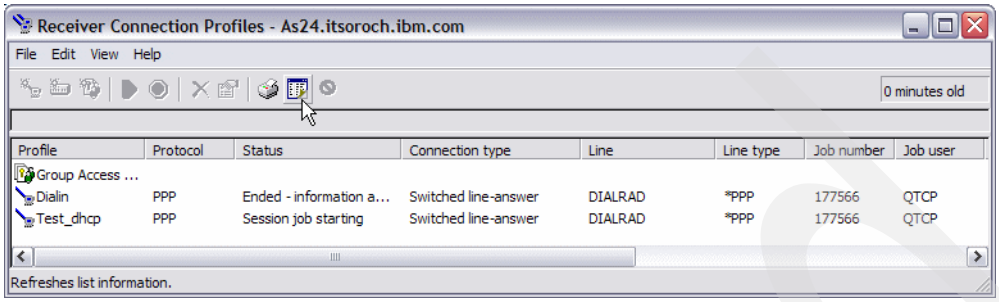


Figure 17-150 iSeries Navigator: Receiver Connection Profiles: Monitor test_dhcp using refresh

4. Your Windows client has just connected to the System i via a PPP connection. PING your System i from a command entry window, as seen in Figure 17-151.

```
C:\>ping 172.23.10.2

Pinging 172.23.10.2 with 32 bytes of data:

Reply from 172.23.10.2: bytes=32 time=160ms TTL=64
Reply from 172.23.10.2: bytes=32 time=160ms TTL=64
Reply from 172.23.10.2: bytes=32 time=160ms TTL=64
Reply from 172.23.10.2: bytes=32 time=150ms TTL=64

Ping statistics for 172.23.10.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 150ms, Maximum =  160ms, Average =  157ms
```

Figure 17-151 PING is successful from the Windows client to the System i IP address 172.23.10.2

5. On your Windows client, find the IP address that has been dynamically assigned by the System i DHCP server using the **ipconfig** command, as shown in Figure 17-152.

```
C:\>ipconfig /all

PPP adapter to103:

        Connection-specific DNS Suffix  . :
        Description . . . . . . . . . . . : WAN (PPP/SLIP) Interface
        Physical Address. . . . . . . . . : 00-53-45-00-00-00
        Dhcp Enabled. . . . . . . . . . . : No
        IP Address. . . . . . . . . . . . : 172.23.10.11
        Subnet Mask . . . . . . . . . . . : 255.255.255.255
        Default Gateway . . . . . . . . . : 172.23.10.11
        DNS Servers . . . . . . . . . . . : 172.23.10.1
```

Figure 17-152 ipconfig displays the IP address of the Windows client as 172.23.10.11

6. In iSeries Navigator, use the DHCP monitor to identify the lease for your Windows client. To do this, expand **Network** → **Servers** and then select **TCP/IP**. Right-click **DHCP** and choose **Monitor**. The DHCP Monitor will be displayed, as seen in Figure 17-153.

   You should be able to find your lease issued from the System i DHCP server based on the IP address.

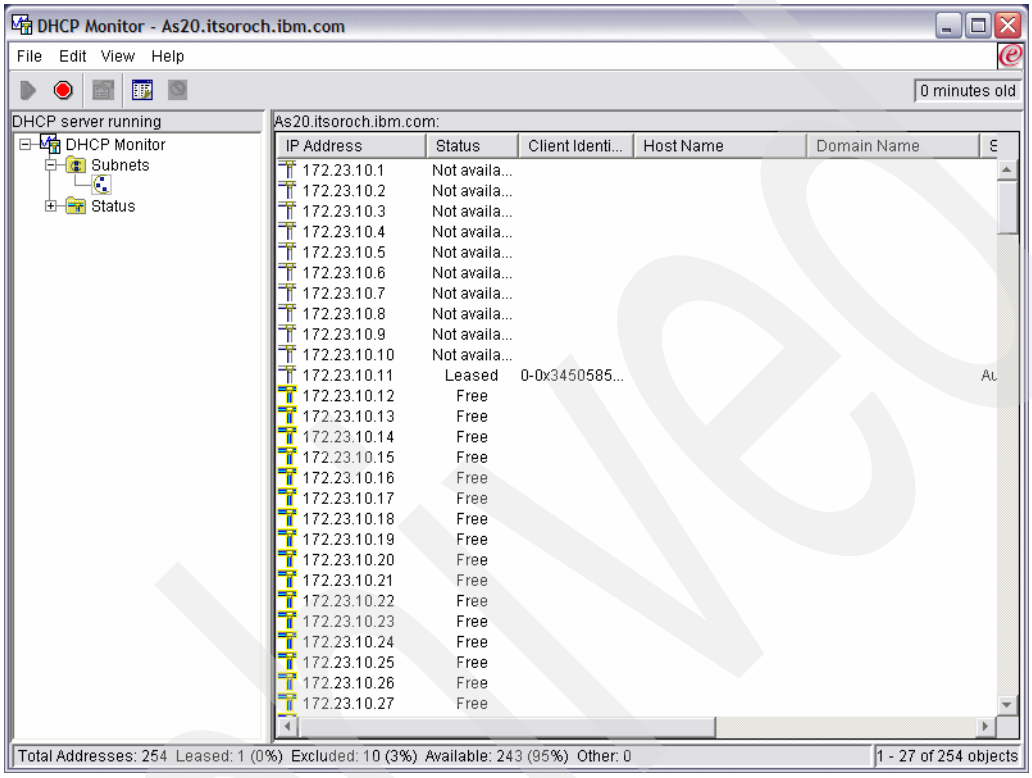


Figure 17-153 DHCP Monitor: PPP WAN client is leased the IP address 172.23.10.11 from DHCP

18

QoS scenarios

This chapter contains various QoS scenarios. Each sample scenario has four sections:

- The scenario overview. This includes the conditions in which you would choose the scenario and a sample network configuration.
- A planning worksheet. This worksheet helps you prepare the required parameters that you will need to configure the sample configuration.
- Step-by-step guide to configuring your sample configuration.
- An explanation of how to test the sample configuration.

This chapter contains the following sample configurations.

18.1 QoS: Inbound admissions policy: Connection rate

This scenario describes the configuration of the i5/OS QoS server for controlling the inbound HTTP traffic.

Problem definition

For this scenario, your Web server's resources are being overloaded by the client requests entering your network from a specific groups of users. You have to slow the incoming HTTP traffic to your Web server on the local interface 10.1.1.1 coming from these users.

The Figure 18-1 shows the network diagram for the problem. This QoS can control traffic flow only in one direction.

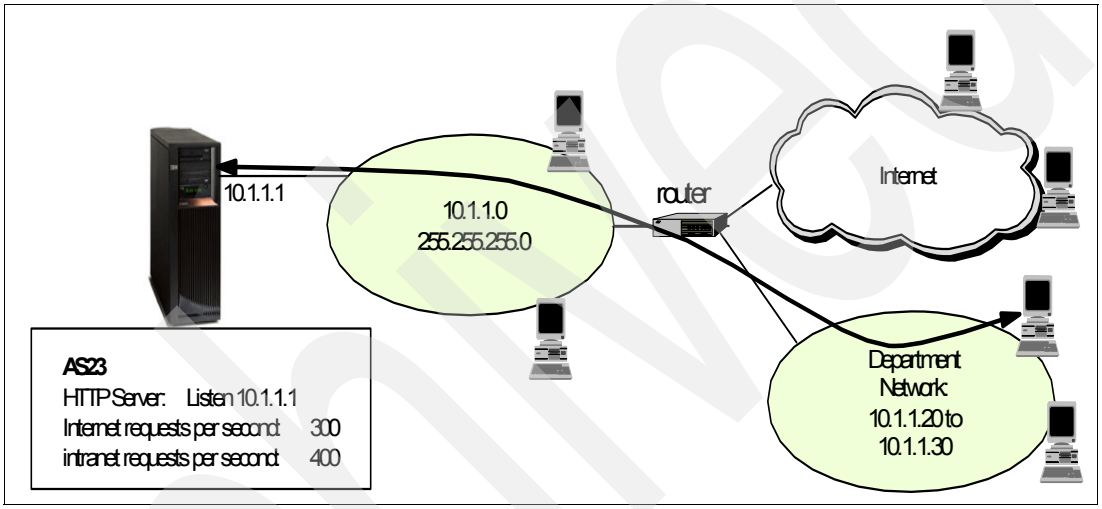


Figure 18-1 Traffic from intranet department is flooding the critical HTTP server

Solution definition

QoS can help us restrict the accepted inbound connection attempts, based on connection attributes (for example, IP address) to the server. To achieve this, we decide to implement an inbound admission policy that restricts the number of accepted inbound connections.

To configure an inbound policy, you must decide whether you are restricting traffic to a local interface or to a specific application, and whether you are restricting it from a particular client. In this case, we want to create a policy that restricts connection attempts from a set of clients going to port 80 (HTTP protocol) on our local IP interface 10.1.1.1. Because we are defining this restriction by IP address, we should create a connection rate policy.

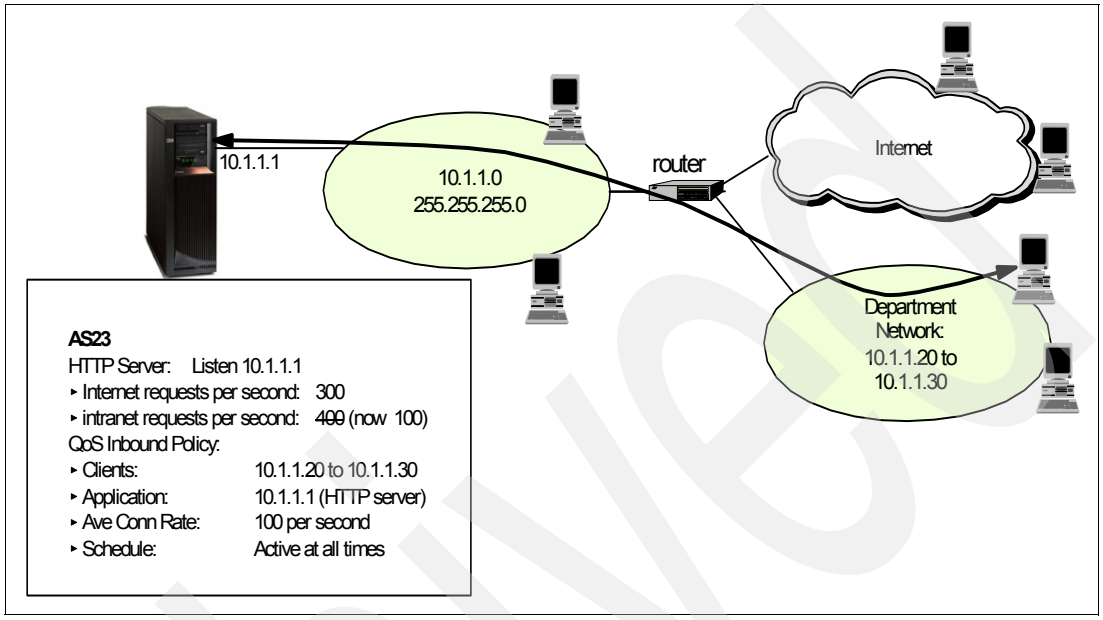


Figure 18-2 QoS: Limiting inbound connection rate: solution

Assumptions

The assumptions are:

- The System i is on V5R2 or later and the TCP/IP is installed and configured.
- iSeries Access is configured on a PC with the latest service pack downloaded from:

  `http://www.ibm.com/servers/eserver/iseries/access/casp.html`
- The IP address of our System i is 10.1.1.1. The IP address of the clients is in the range of 10.1.1.20 to 10.1.1.30.

How-to

To configure the connection rate inbound policy of QoS, we perform the following tasks:

- Step 1: Configure QoS on System i.
- Step 2: Configure the QoS inbound connection rate policy.
- Step 3: Test the configuration.

Step 1: Configure QoS on System i

In this scenario we assume that you have not yet configured your System i with QoS. To configure QoS, perform the following steps:

1. As shown in Figure 18-3, in iSeries Navigator expand the System i Name → **Network** → **IP Policies**. Right-click **Quality of Service** and select **Configuration** from the context menu.

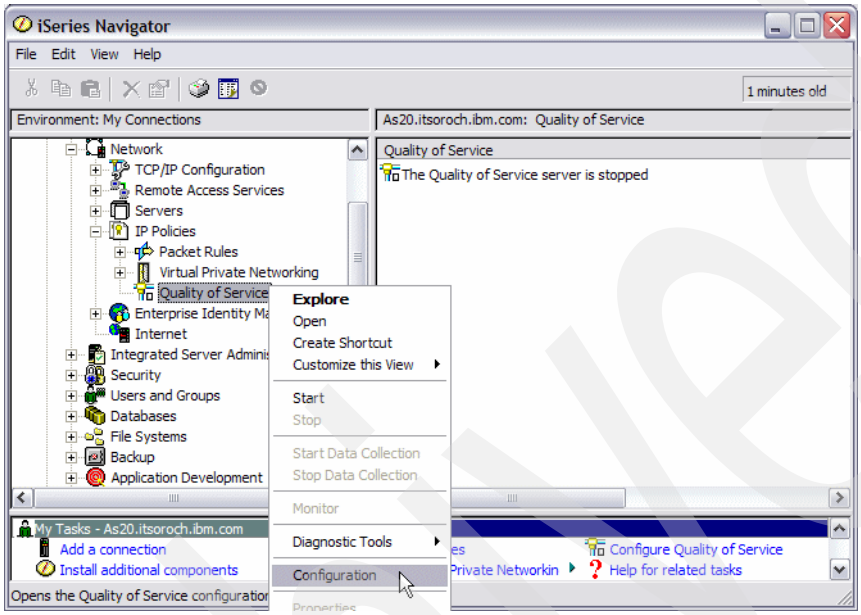


Figure 18-3 Configuration for QoS

2. The initial configuration wizard appears. Click **Next** to continue.

> **Restriction:** The wizard will not appear if QoS has been configured previously. If you would like to use the wizard but it will not start, you may circumvent by using the following steps:
>
> 1. End the QoS server (either via iSeries Navigator or green screen).
> 2. Delete the file /QIBM/UserData/OS400/QOS/ETC/POLICYD.CONF.
> 3. Start the QoS server.
> 4. Return to step 1.

5. As shown in Figure 18-4, select the Start Server configuration parameters, such as **autostart QoS server with TCP/IP** and whether to start the QoS server now. Make your selections and click **Next** to continue.

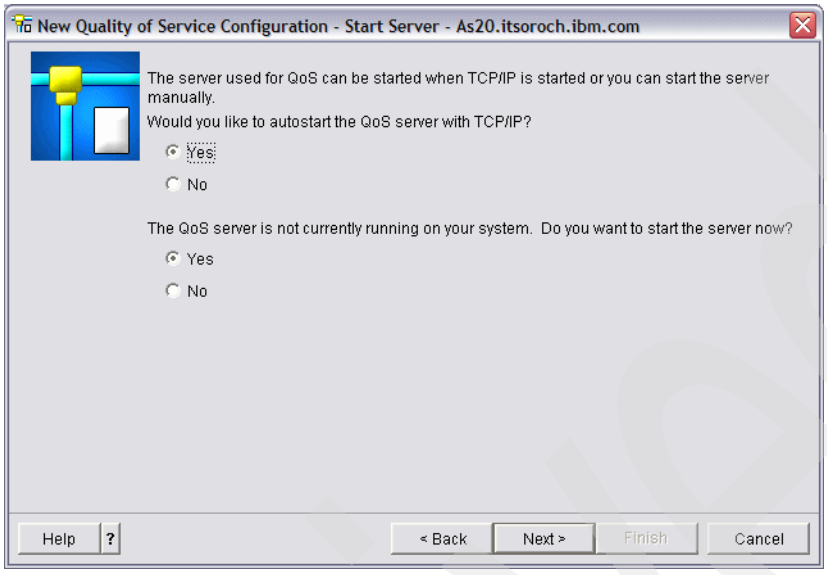


Figure 18-4 New QoS configuration: Start parameters

6. The next window asks whether you want to record and view performance data. We have elected not to collect this information, so for our configuration we selected **No**. Click **Next**.
7. You are prompted to choose whether to journal QoS server activity. Again, we selected **No**. Click **Next**.
8. The next window asks whether you want to export your QoS policies to a Directory Server. We selected **No**. Click **Next.**
9. This opens the summary of the settings. Review them to make sure that they are accurate, and click **Finish** to save the configuration.

10. When the QoS server is configured it shows the QoS Server Configuration window (Figure 18-5).

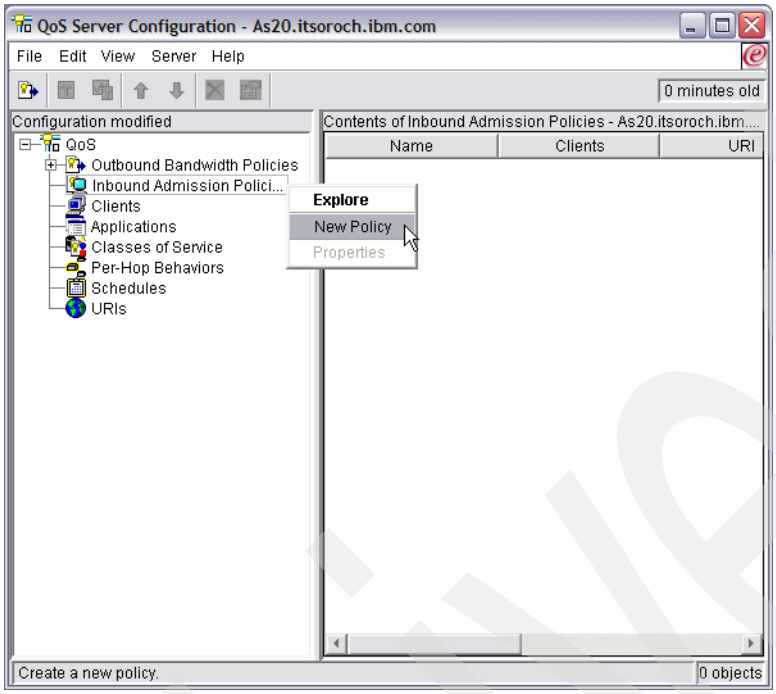


Figure 18-5 QoS Server Configuration: New inbound connection rate policy

Step 2: Configure the QoS inbound connection rate policy

To do this:

1. In the QoS Server Configuration window right-click **Inbound Admission Policies** and choose **New Policy**, as shown in Figure 18-5.
2. This opens a wizard for configuring the inbound connection rate policy. Click **Next** to start the wizard.
3. Give a meaningful name to the policy. We specify `Restrict_HTTP`. Click **Next** to continue.

4. As shown in Figure 18-6, the wizard asks whether this policy applies to all clients or to a specific group of clients. As our problem definition we designate a range of IP addresses by selecting **Specific address or addresses** and clicking **New**.

   In the New Client window, specify the Name of this new group of clients. We specify `http_restrict`.

   Select IP address range as `10.1.1.20` to `10.1.1.30`, which represents the department network in Figure 18-2 on page 619. Click **OK** and **Next** to continue.

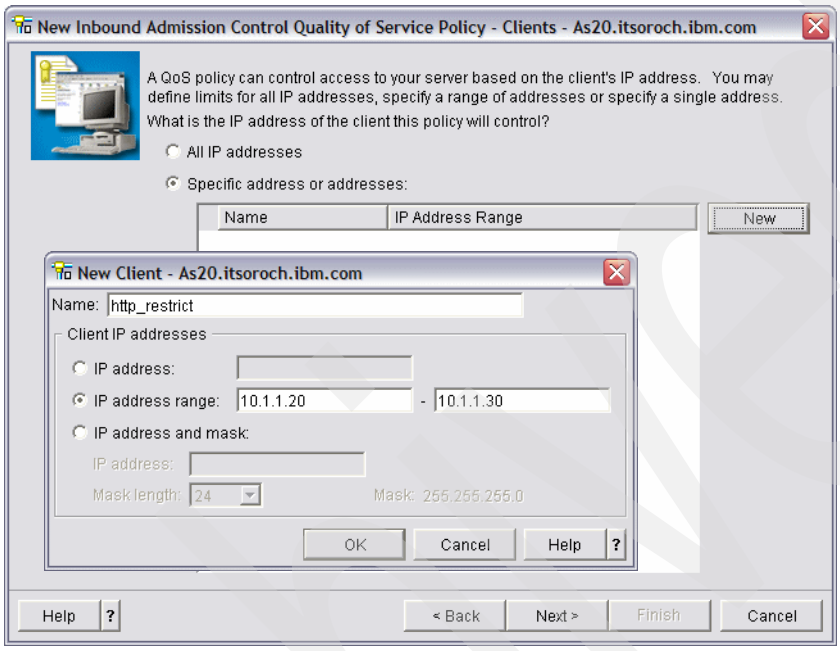


Figure 18-6 QoS Inbound Connection Rate Policy: Select client IP range

5. In the next window, you can select whether you want this policy to act on any URIs or specific URIs. Click **Any URI** and then **Next** to continue.
6. In the next window, add the port that you want this policy to act on. Select **Specific port**, **range of ports or server type**, and click **New**, then name the application and select the server type HTTP as the traffic is coming for the HTTP server. Click **OK** and then **Next** to continue.
7. In the next window, you may control access for all codepoints or a specific codepoint. Select **All codepoints** and click **Next** to continue.
8. In the next window provide the Local IP address of the HTTP server. It can be specified as all or, if the system has multiple IPs and NICs, as the specific IP on which the HTTP traffic is coming. We select a single IP address and supply the `10.1.1.1` interface. Click **Next** to continue.
9. In the next window you will specify the class of service for this policy. Click **New**.
10. This will launch the **Class of Service Wizard**. Click **Next** to continue.
11. In the next window, provide a name and description for the class of service and click **Next** to continue.
12. In the next window, select to have this class of service for **Inbound only** and click **Next** to continue.

13. As shown in Figure 18-7, the average connection rate and the connection burst limit are to be entered, and the priority of these connections can be set. Table 18-1 shows these parameters and the default settings.

Table 18-1 QoS inbound policy: rate control parameters

Parameter	Description	Setting
Average connection rate	The average connection rate specifies the limit of new, established connections or the rate of accepted URI requests allowed into a server. If a request would cause the server to exceed the limits you set, the server denies the request. The average connection request limit is measured in connections per second.	100
Connection burst limit	The burst limit size determines the buffer capacity, which holds bursts of connections. Connection bursts may enter the server at a faster rate than it can handle or than you may want to allow. If the number of connections in a burst exceeds the connection burst rate you set, then the additional connections are discarded.	100
Priority	Specify the priority of all connections within this policy. This determines which accepted connections are handled first.	Medium

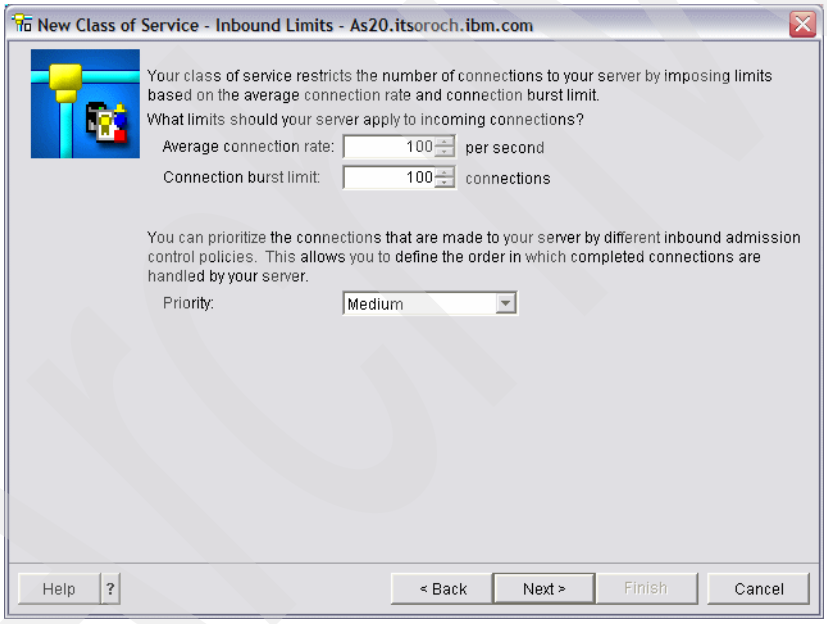


Figure 18-7 Inbound Connection Rate QoS Policy: Limits

These values must be fine-tuned based on the load conditions and business requirements. Click **Next** to continue.

14. The summary panel is shown for the class of service. Click **Finish**. Click **Next** to continue.

15. In Figure 18-8, we define the schedule for which this policy will be active. We select **Active at all times**. Click **Next** to continue.

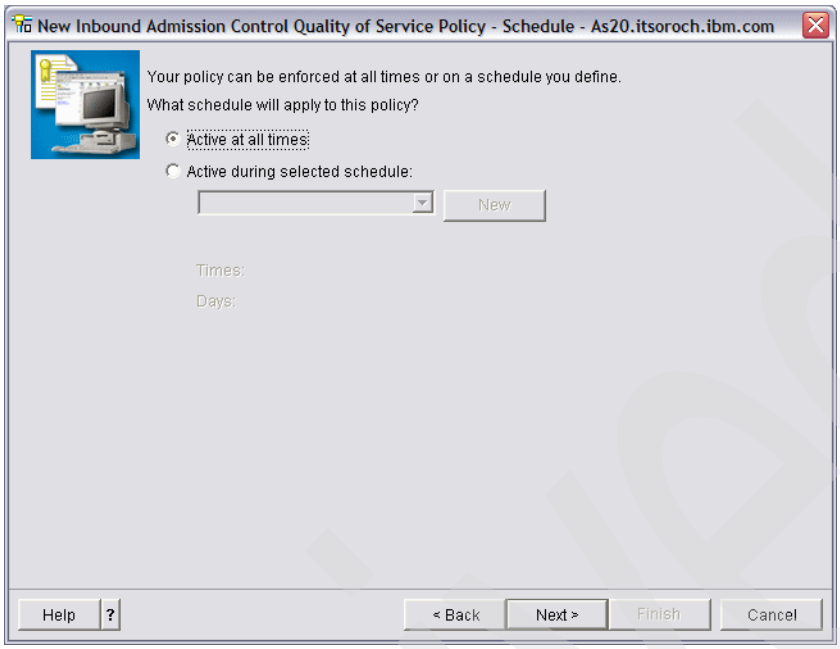


Figure 18-8 Inbound Connection Rate QoS Policy: Schedule

16. Figure 18-9 shows the summary of the policy. Click **Finish** to save the policy.

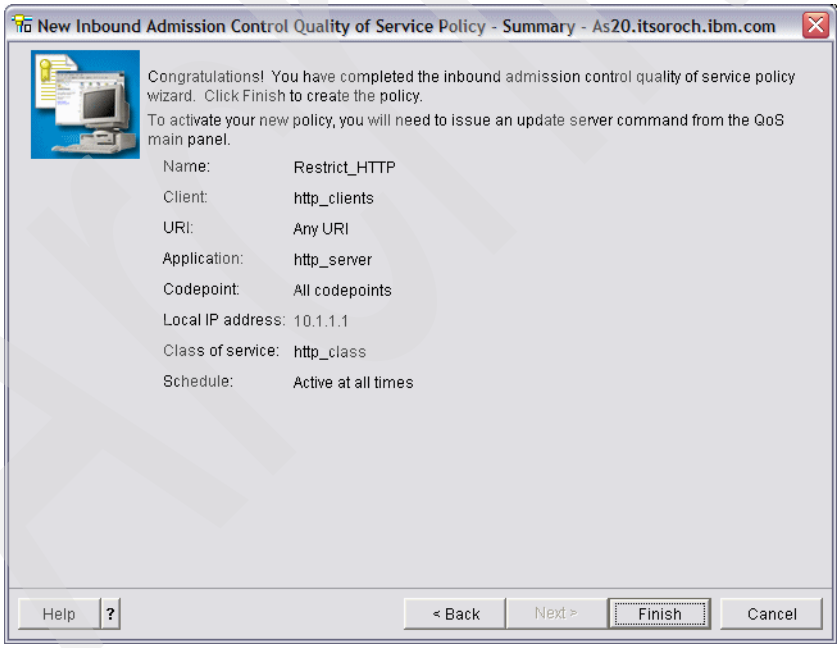


Figure 18-9 Inbound Connection Rate QoS Policy: Summary

Step 3: Test the configuration

To verify that the policy is working as it is supposed to, start the QoS Monitor.

1. In iSeries Navigator right-click **Quality of Service** and choose **Monitor**.

2. In the QoS Monitor window, select **File** → **Start QoS Data Collection** (Figure 18-10).

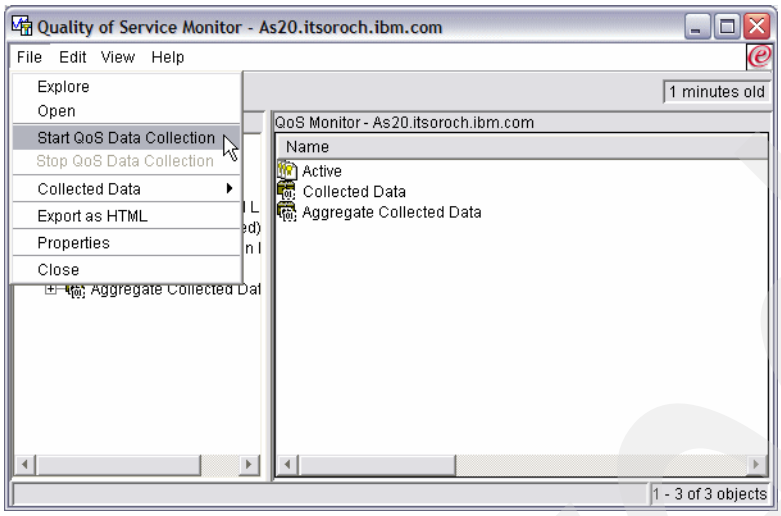


Figure 18-10 Quality of Service Monitor

3. Select the policy that you want to monitor, then refresh the window to see the updates.
4. Check all measured fields, such as accepted requests, dropped requests, total requests, and connection rate. Dropped requests indicate when traffic exceeds the configured policy values. Accepted requests indicate the number of bits controlled by this policy (from the time the packet was started to the present monitor output). Based on these values you can fine-tune the policy further.

18.2 QoS: Inbound admissions policy: limiting connection rate based on HTTP URI

This scenario describes the configuration of the i5/OS QoS server for controlling inbound HTTP traffic based on the URI request rate.

Unlike connection rate policies, URI policies have more control because they examine content, not just the packet headers. The content they examine could include URI name or other application-specific information. For i5/OS, the relative URI name is used to define the policy (for example, /products/clothing).

Problem definition

Your Web server is getting choked. Figure 18-11 shows that at peak times you see about 300 connections per second from both your own department and the users on the Internet for very large files that are available for download from a /Downloads directory. This peak activity is flooding your network and ISP and as such leads to other performance problems with other critical e-business applications.

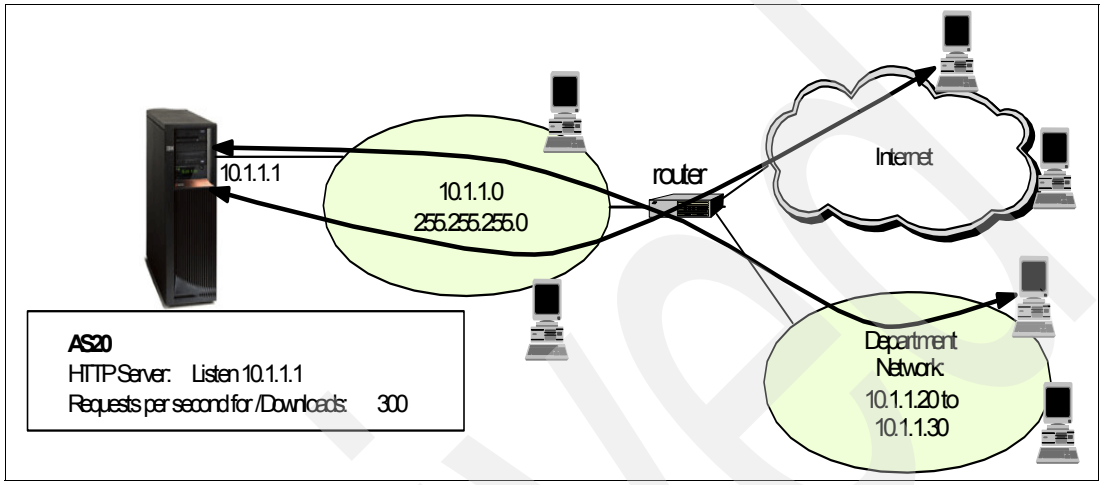


Figure 18-11 Problem: Incoming requests for large files in /Downloads directory: too many at 300/s

Solution definition

The traffic to the Web site can be restricted by using the QoS URI rate policy (Figure 18-12).

This type of policy applies admission controls, based on application level information, to limit the URI requests accepted by the server. This is also referred to as header-based connection request control, which uses URIs to set priorities.

Specifically, if any client is accessing a file in the /Downloads directory, the QoS server on i5/OS will allow only 100 such connections per second.

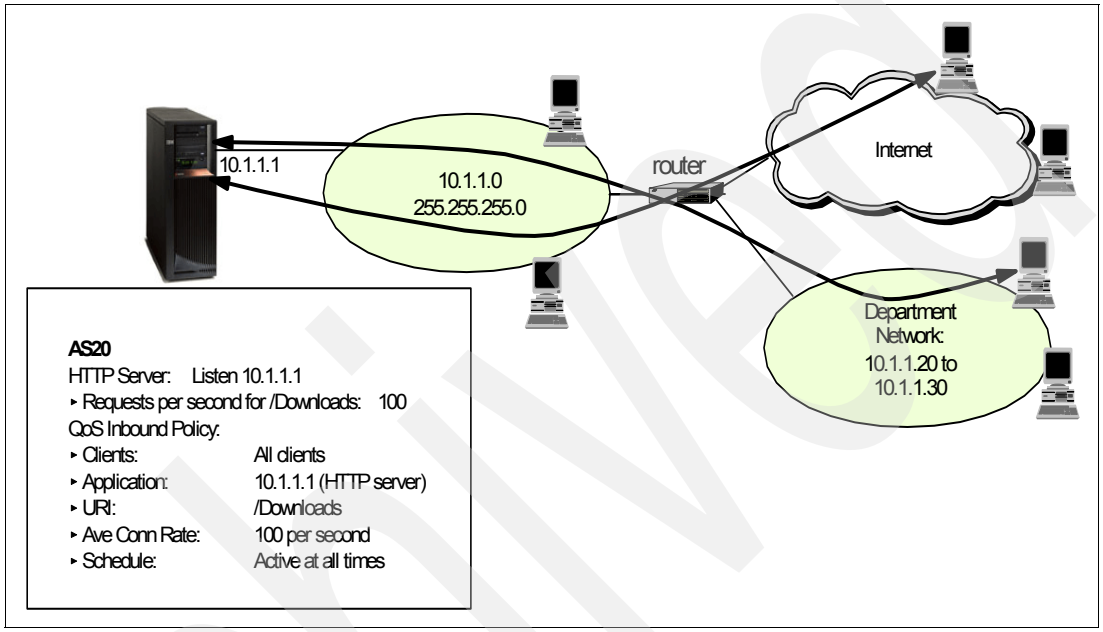


Figure 18-12 Solution: restrict incoming requests for the /Downloads directory to 100/s

Note: The application port assigned in the new URI policies must match the Listen directive enabled for FRCA in the Apache Web server configuration. If the port values do not match, the QoS URI policy will not function as expected.

Assumptions

The assumptions are:

- The System i is on V5R2 or later and the TCP/IP is installed and configured.
- iSeries Access is configured on a PC with the latest service pack from the Web site:

  `http://www.ibm.com/servers/eserver/iseries/access/casp.html`
- The QoS is configured on the system.
- The Web site address is as20.itsoroch.ibm.com and the port is 8000. The download traffic is coming to http://as20.itsoroch.ibm.com:8000/Downloads
- FRCA is enabled for the HTTP server instance.

How-to

To configure the URI Rate based Inbound Policy of QoS, we perform the following tasks:

To configure QoS for the first time, refer to “Step 1: Configure QoS on System i” on page 620 as part of scenario 18.1, “QoS: Inbound admissions policy: Connection rate” on page 618.

- Step 1: Configure the QoS URI request rate inbound policy.
- Step 2: Test the configuration.

Step 1: Configure the QoS URI request rate inbound policy

To do this:

1. In the QoS Server Configuration, expand **QoS** → **Inbound Admission Policies** and select **New Policy** from the context menu, as shown in Figure 18-13.

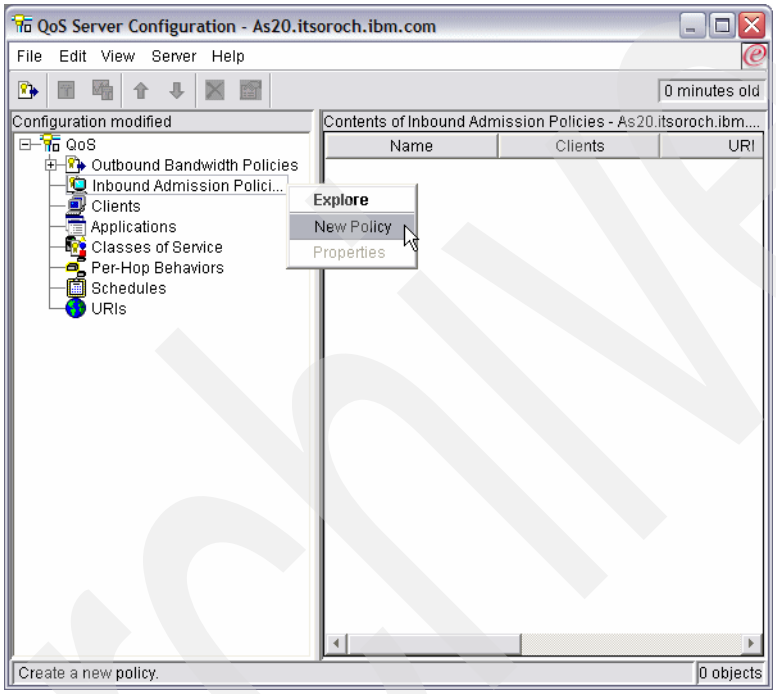


Figure 18-13 QoS Server Configuration: new inbound URI rate policy

2. This opens a wizard for configuring the Inbound Admission Control Quality of Service Policy. Click **Next**.
3. Give a meaningful name to the policy. We specified `URIExample`. Click **Next** to continue.
4. In the next window you have the option for this policy to control all IP addresses or specific IP addresses. Select **All IP addresses** and click **Next**.

5. In the next window you have the option for this policy to control any URI or a specific URI. Select **Specific URI** and click **New**.
6. As shown in Figure 18-14, enter a name for the URI object that you are creating and specify the relative path as `/Downloads/`. Click **OK** and **Next** to continue.

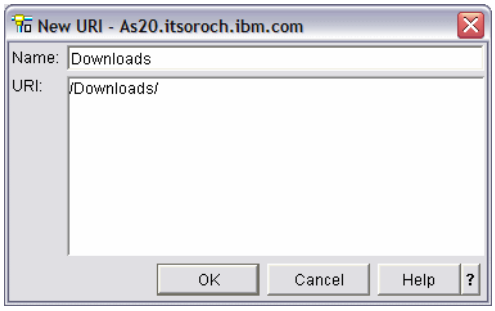


Figure 18-14 Inbound Server Request URI QoS Policy: New URI for /Downloads

7. In the next window, add the port that you want this policy to act on. Select **Specific port, range of ports or server type** and click **New**, then name the application, select **Port**, and specify port 8000, as the traffic is coming in on this port according to our solution definition. Click **OK** and then **Next** to continue.
8. In the next window, you can grant or restrict access based on a DiffServ codepoint. We select **All codepoints**. Click **Next** to continue.
9. Here we can specify the local IP address. It can be specified as All or, if the system has multiple IPs and NICs, as the specific IP on which the HTTP traffic is coming. We select **All IP addresses**. Click **Next** to continue.
10. In the next window, we are asked for the Class of Service. Click **New**. This brings up the New Class of Service wizard. Click **Next** to continue.
11. Enter a descriptive name and description for this policy. Click **Next** to continue.
12. In the next window, we can define whether the class of service affects inbound and outbound activity. Select **Inbound only** and click **Next** to continue.
13. As shown in Figure 18-15 on page 631, the average connection rate and the connection burst limits are entered and the priority of these connections can be set. Table 18-2 defines these parameters.

Table 18-2 Rate parameters

Parameter	Description	Setting
Average connection rate	The average connection rate specifies the limit of new, established connections or the rate of accepted URI requests allowed into a server. If a request would cause the server to exceed the limits you set, the server denies the request. The average connection request limit is measured in connections per second.	100
Connection burst limit	The burst limit size determines the buffer capacity, which holds bursts of connections. Connection bursts may enter the server at a faster rate than it can handle or than you may want to allow. If the number of connections in a burst exceeds the connection burst rate you set, then the additional connections are discarded.	5
Priority	Specify the priority of all connections within this policy. This determines which accepted connections are handled first.	Low

These values must be fine-tuned based on load conditions and business requirements. Click **Next** to continue.

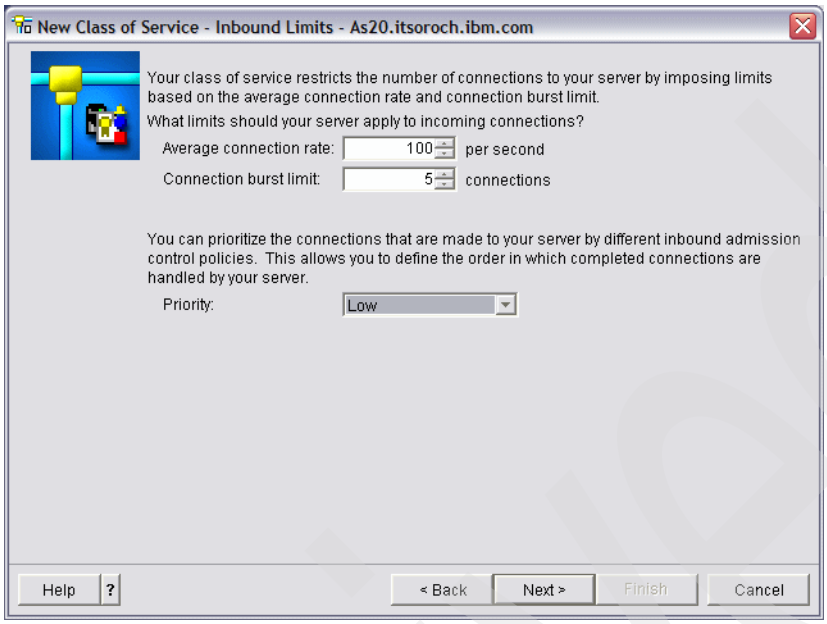


Figure 18-15 Inbound server request URI QoS Policy: Limits

14. The next window shows the summary panel for the Class of Service wizard. Click **Finish** then click **Next** to continue.
15. In the next window, define the schedule for this policy. Select **Active at all times**. Click **Next** to continue.
16. The last window is the summary. Click **Finish** to save this policy.

Step 2: Test the configuration

To verify that the policy is working as it is supposed to, start the QoS monitor.

1. In iSeries Navigator, right-click **Quality of Service** and choose **Monitor**.
2. In the Monitor window, select **File** → **Start QoS Data Collection** (Figure 18-16).

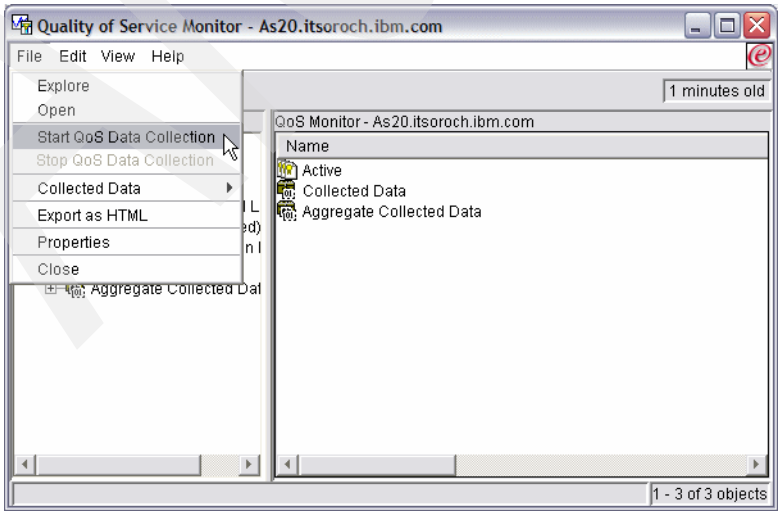


Figure 18-16 Quality of Service Monitor: Start collection of data

3. In the monitor window select the policy you want to monitor, and refresh the window to see the updates as shown in Figure 18-17.

   Make sure to check measured fields such as accepted requests, dropped requests, total requests, and connection rate. Dropped requests indicate when traffic exceeds the configured policy values. Accepted requests indicate the number of bits controlled by this policy (from the time the packet was started to the present monitor output). Based on these values you can further fine-tune the policy.

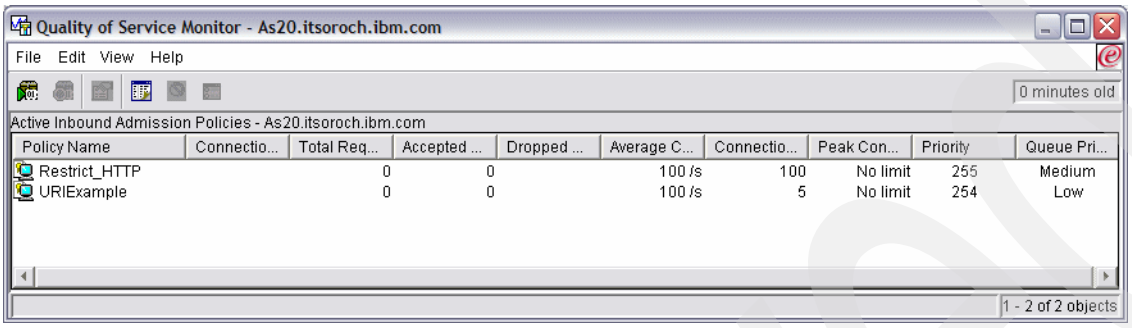


Figure 18-17 QoS monitor: inbound rate control for a URI-based policy

4. Close the monitor. From the QoS Server Configuration Inbound Admissions Policies Contents, right-click the policy name and select **Properties** from the context menu in order to edit the policy's properties. This is also where you edit the schedule, client, applications, traffic management, and so on.

18.3 QoS: outbound bandwidth policies: differentiated services

This scenario describes the configuration of the i5/OS QoS server for controlling traffic from the Web server using the differentiated service policy.

A differentiated service policy divides your traffic into classes. All traffic within this policy is assigned a codepoint that tells routers how to treat the traffic. In this scenario, the policy would be assigned a low codepoint value to affect how the network prioritizes browser traffic.

Note: The classes currently do not have specific bandwidth values assigned to them. Therefore it is necessary for the customer and the service provider to agree on what bandwidth corresponds to what class.

Problem definition

Your company has been experiencing high levels of browser traffic from a specific group of users on Fridays. This traffic has been interfering with the accounting department, which also requires good network performance for their accounting applications on Fridays.

Solution definition

To limit browser traffic out of your network, create a differentiated service policy that divides the traffic into classes. All traffic within this policy is assigned a codepoint. This codepoint tells routers how to treat the traffic. In this scenario, the scheduled policy would be assigned a low codepoint value on Friday that will affect how the System i and any QoS-aware routers treat this traffic.

Assumptions

The assumptions are:

- The System i is on V5R2 or later, and the TCP/IP is installed and configured.
- iSeries Access is configured on a PC with the latest service pack from the Web site:

  `http://www.ibm.com/servers/eserver/iseries/access/casp.html`
- The QoS is configured on the system.
- The Web site address is as20.itsoroch.ibm.com and the port is 8001. The client subnet is 10.10.10.0.

How-to

To configure the differentiated classes of service Policy of QoS, we perform these tasks:

To configure QoS for the first time refer to “Step 1: Configure QoS on System i” on page 620 as part of scenario 18.1, “QoS: Inbound admissions policy: Connection rate” on page 618.

- Step 1: Configure the outbound QoS differentiated class of service policy.
- Step 2: Test the configuration.

Step 1: Configure the outbound QoS differentiated class of service policy

To do this:

1. In the QoS Server Configuration (Figure 18-18), expand **QoS** → **Outbound Network Policies**. Right-click **DiffServ** and select **New Policy** from the context menu.

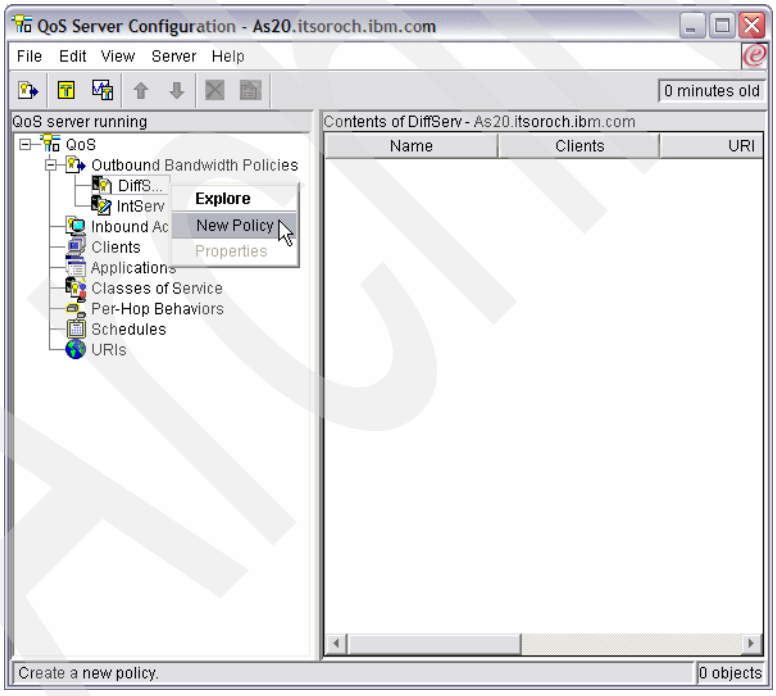


Figure 18-18 Differentiated policy: new policy

2. This opens a wizard for configuring a new differentiated services policy. Click **Next** to start the wizard, and provide a meaningful name to the policy. For our example, we specified `Restrict_user_group`. Click **Next** to continue.

3. As shown in Figure 18-19, enter the IP address of the clients or the group of clients that are to be controlled by this policy. We enter `10.10.10.1` for the IP address and a Mask length of `24` (the same as specifying a subnet mask of 255.255.255.0) for restricting the whole client subnet as per our problem definition.

   Click **OK** and **Next** to continue.

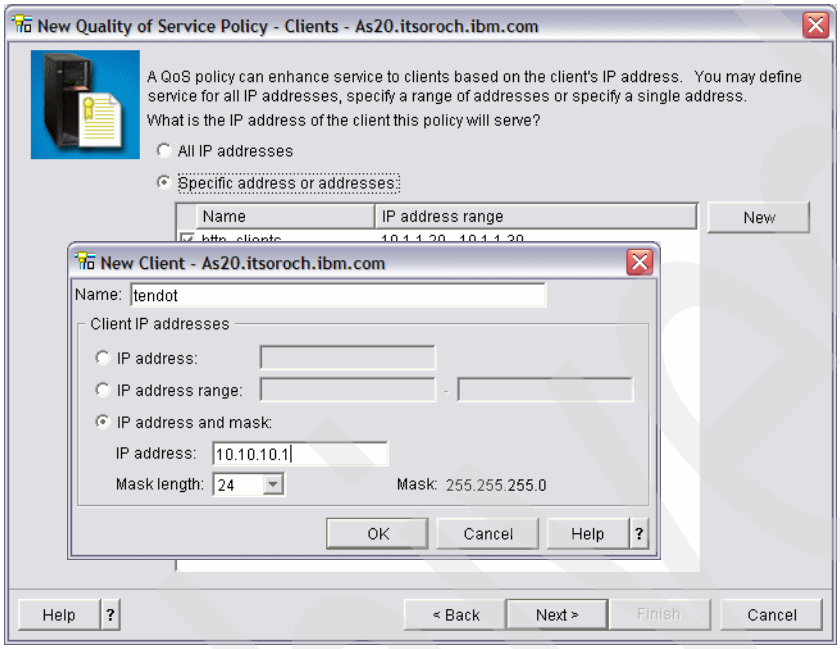


Figure 18-19 QoS differentiated Service Policy: New Client: restrict client IP

4. The next panel allows you to filter on the token and priority in the request. Select **Any token** and **All priorities** and click **Next** to continue.
5. Add the port that you want this policy to act on and also the protocol. Select **Specific port, range of ports or server type** and click **New**, then name the application, select **Port**, and specify port 8001, as the traffic is coming in on this port according to our solution definition. Click **OK**. We also select **All** protocols. Click **Next** to continue.
6. You are asked for the Local IP. It can be specified as All or, if the system has multiple IPs and NICs, as the specific IP on which the HTTP traffic is coming. We select **All IP addresses**. Click **Next** to continue.
7. In the next window we define the class of service that is to be used by this policy. As we do not have the class of service defined, we click **New**, which opens a wizard for creating the differentiated class of service. Click **Next** to continue with this wizard.
   a. Enter a Name. We named our new differentiated class of service `LOW`. Click **Next**.
   b. In the next panel we specify that this class of service will only apply to outbound connections. Click **Next**.

c. As shown in Figure 18-20, the wizard asks for the per-hop behavior (PHB) to be assigned to this policy. Diff serv on System i uses classes to determine what type of per-hop treatment is to be given. The classes are built by using the PHB that is described in the RFCs 2474, 2475, 2597, and 2598. The PHB describes what kind of delay/though put/loss characteristics are desired for the packet of the data.

   The classes currently do not have specific bandwidth values assigned to them. Therefore it is necessary for a customer and a service provider of routers to agree on what bandwidth corresponds to a class.

   We select `Class 2`. This sets, by default, a Differentiated Services Code Point (DSCP) of binary '010000'. Click **Next** to continue.

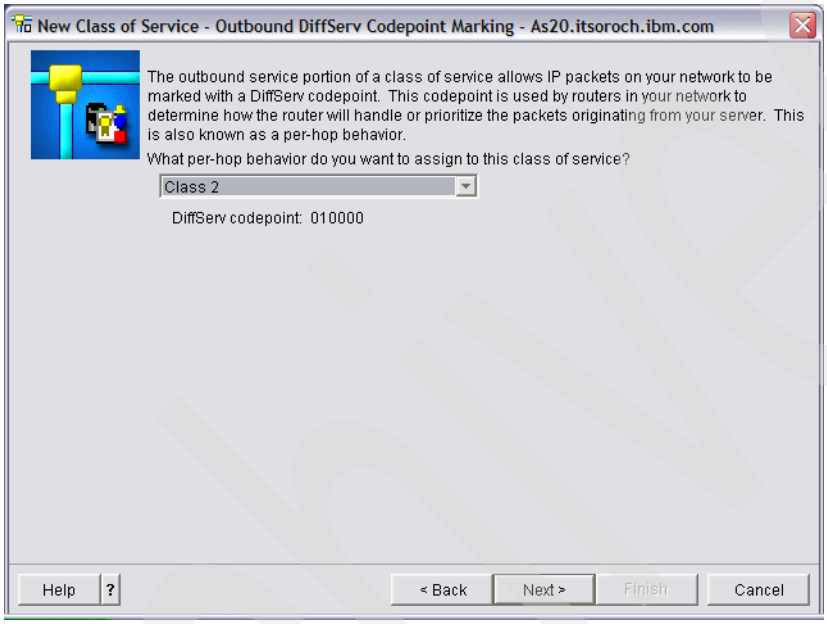


Figure 18-20 Class of service: per-hop behavior

d. Next, you are asked whether you want to perform server traffic metering.

   Traffic metering imposes limits and measures if the traffic profile for a QoS policy is actually being implemented on the data packets. If the forwarding treatment a packet receives does not correspond to the policy it was assigned, it is considered out-of-profile. The meter passes this information to other conditioning functions. The other functions (such as marking, shaping, dropping) then trigger actions that make sure that the data packets comply with the policies.

   Even though i5/OS is not a QoS-aware router and cannot make routing decisions based on how the Differentiated Services Code Point (DSCP) is set in the IP packet, it can restrict the flow of application data into the network, as defined by an outbound bandwidth policy.

   We select **Yes** and **Next** to continue.

e. Next the wizard prompts for the parameters of rate control limits that are described in Table 18-3. After filling in the values for the parameters, click **Next** to continue.

Table 18-3 Rate control limits for QoS Diff Serv Policy

Parameter	Description	Settings
Token Bucket Size	This determines the buffer capacity, which holds bursts of data. Burst data is information that an application gives the server to send out, at a faster rate than it can exit. If an application quickly sends enough burst data to the server, the buffer fills up. When data leaves the server as fast as it enters the server, then the token bucket size remains unchanged. When the buffer is filled, QoS treats additional data packets as out-of-profile. In this policy we can determine how QoS handles out-of-profile traffic.	16 KB
Average Rate Limit	Specifies the average rate that this class of service allows for the sum of all connections going to the token bucket. This average rate can be a unit from 10 Kbps to 1 Gbps. The average rate limit must be less than the peak rate limit, so you do not use the entire interface.	8 Mbps
Peak Rate Limit	Specifies the maximum data rate entering the token bucket that this class of service allows. This peak rate can be a unit from 10 Kbps to 1 Gbps. The default rate is *do not limit*. When you assign do not limit to the rate, you are making the available resources the limit.	10 Mbps

f. Next, you can specify how out-of-profile packets are handled. We select Drop UDP packets or reduce TCP congestion window. Reducing the TCP congestion window reduces the number of packets that can be sent into the network at any given time. Click **Next** and **Finish** to save the differentiated class of service.

8. With one wizard closed you now return to your first wizard, as seen in Figure 18-21. Click **Next** to continue.

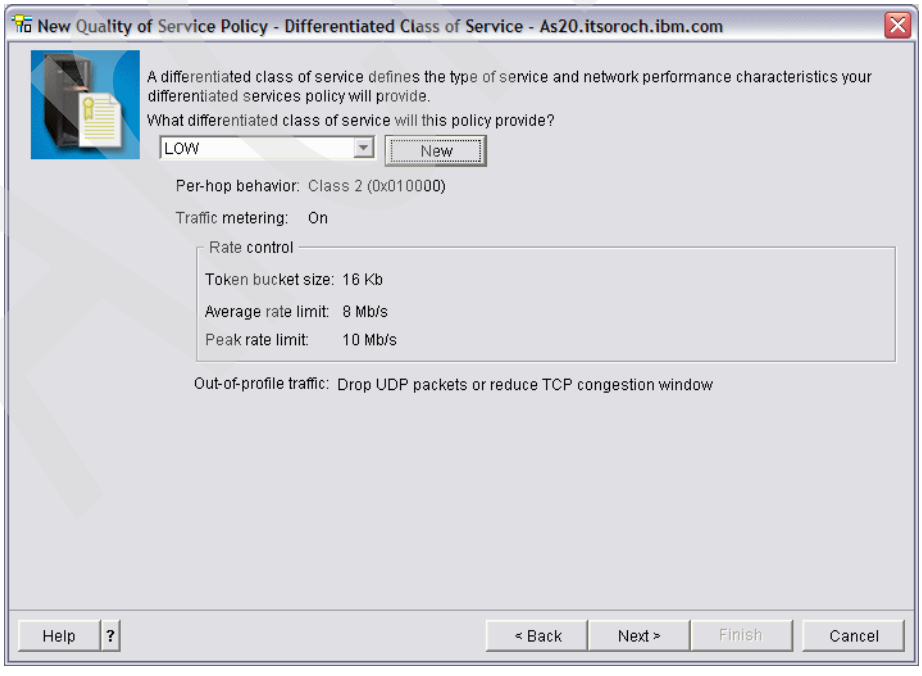


Figure 18-21 New diff class of service

9. As shown in Figure 18-22, specify the date and time the policy is to be active. We select Fridays and 9:00 am to 7:00 pm as per our problem definition. Click **OK** and then **Next** to continue.

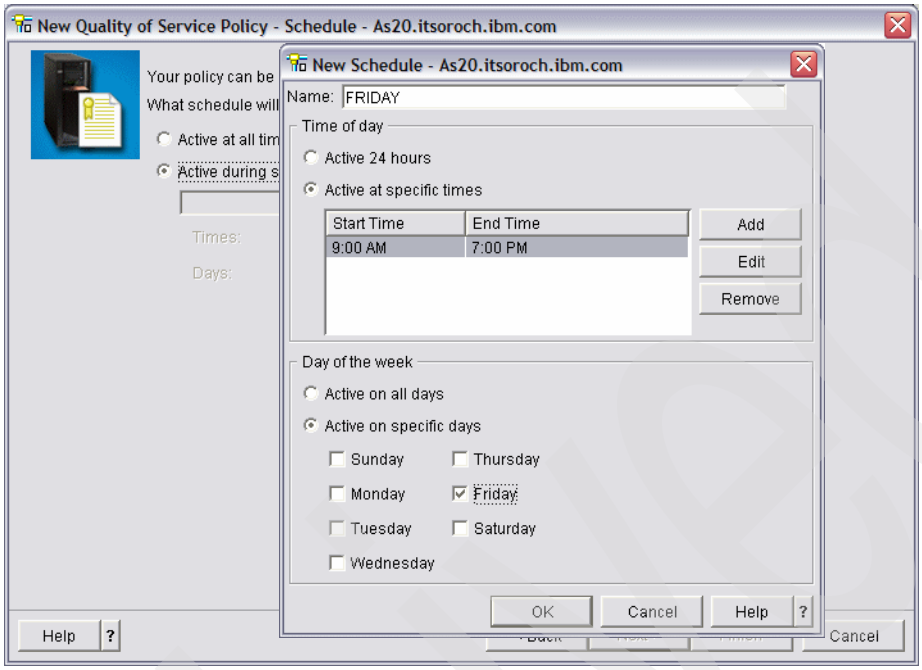


Figure 18-22 QoS Diff Serv Policy: Schedule

10. This opens the Summary window for the policy. Click **Finish** to save the policy.

11. In the QoS Server Configuration window, click **Server** → **Update** (Figure 18-23) to update the active server with the current configuration.

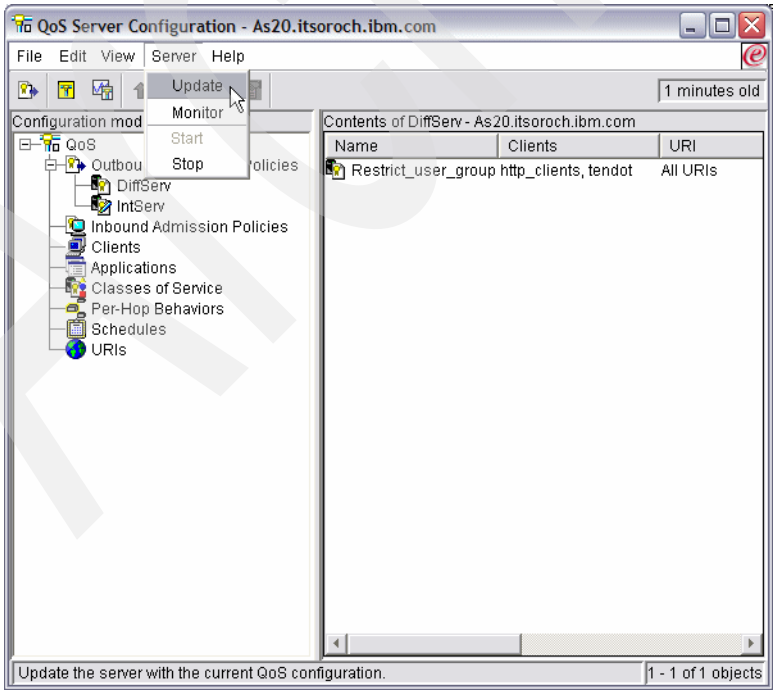


Figure 18-23 QoS Configuration: update active configuration

Step 2: Test the configuration

To verify that the policy is working as it is supposed to, start the QoS monitor:

1. In iSeries Navigator, right-click **Quality of Service** and choose **Monitor** from the context menu.
2. In the QoS Monitor window, click **File** → **Start QoS Data Collection** (Figure 18-24).

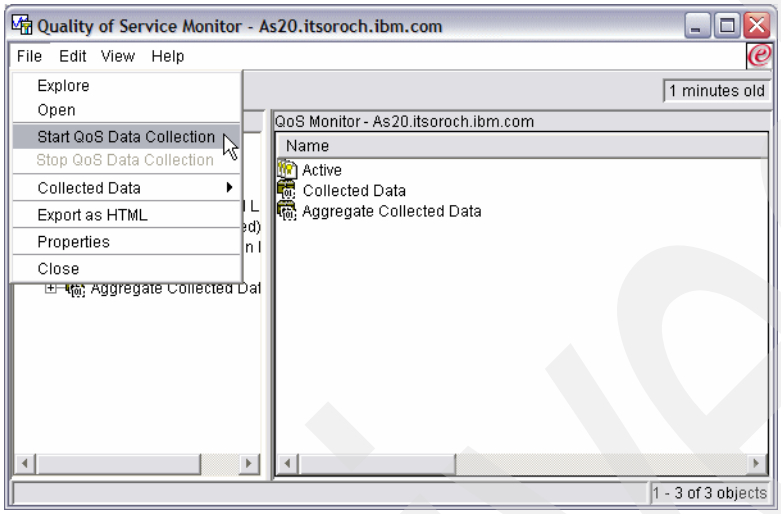


Figure 18-24 QoS monitor: Start QoS Data Collection

3. Select the policy that you want to monitor and refresh the window to see the updates.
4. Make sure to check all measured fields, such as Bit Rate, Packets In-Profile, and Bits Out-Of-Profile. Based on these values, you can further fine-tune the policy.

Tip: Remember that the Active monitor will only show the policy while it is active. In our scenario, this is on Fridays from 9:00 a.m. till 7:00 p.m.

18.4 QoS: dedicated delivery: integrated services policy

This scenario describes the configuration of the i5/OS QoS server for giving the assured bandwidth to a sensitive application using integrated service policy of QoS.

Integrated services reserve resources for a particular policy before the data is sent. The routers are signaled before data transfer and the network actually agrees to and manages (end-to-end) data transfer based on a policy. The bandwidth request comes in a reservation from the client. If all routers in the path agree to the requirements coming from the requesting client, the request gets to the server and intserv policy. If the request falls within the limits defined by the policy, the QoS server grants permission for the RSVP connection and sets aside the bandwidth for the application.

Problem definition

The chief executive officer (CEO) of your company is going to give a live broadcast to a client across the country between 1:00 PM and 2:00 PM. You must guarantee that IP telephony will have specific bandwidth to prevent interruptions during the broadcast. In this scenario, the application resides on the System i.

Solution definition

Extremely sensitive applications require a guaranteed connection. Because the application your CEO is using requires a smooth, uninterrupted transfer, you will have to use a guaranteed integrated service policy. Guaranteed service controls the maximum queuing delay, so that packets will not be delayed over a designated amount of time.

Integrated service policies require RSVP-enabled applications, so ensure that your server has RSVP-enabled applications, using the Resource Reservation Setup Protocol (RAPI) API or qtoq() QoS socket APIs.

Integrated service policies also require that the routers along the traffic's path are RSVP-enabled.

Assumptions

The assumptions are:

- The System i is on V5R2 or later and the TCP/IP is installed and configured.
- iSeries Access is configured on a PC with the latest service pack from the Web site:

  http://www.ibm.com/servers/eserver/iseries/access/casp.html
- The QoS is configured on the system.
- The Web address is as20.itsoroch.ibm.com, the port is 2427, and the client address is 190.86.23.1.

How-to

To configure the integrated service policy of QoS, perform the following tasks:

To configure QoS for the first time, refer to "Step 1: Configure QoS on System i" on page 620 as part of scenario 18.1, "QoS: Inbound admissions policy: Connection rate" on page 618.

- Step 1: Configure the QoS integrated service policy.
- Step 2: Test the configuration.

Step 1: Configure the QoS integrated service policy

To do this:

1. In the QoS Server Configuration, expand **QoS** → **Outbound Bandwidth Policies**. Right-click **Intserv** and choose **New Policy**, as shown in Figure 18-25.

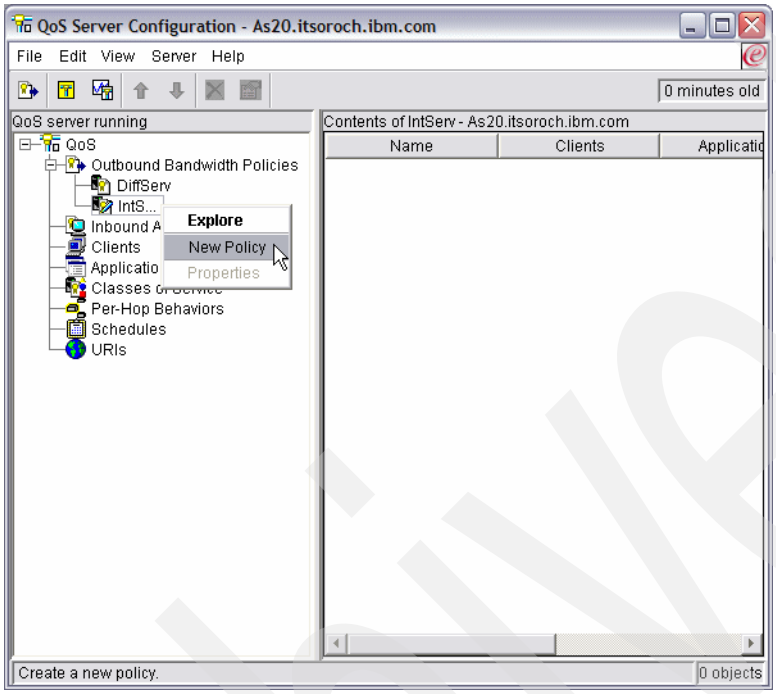


Figure 18-25 Integrated service: New Policy

2. Click **Next** to start the wizard that is used to configure the new integrated service policy. Enter a meaningful name for the new policy. We named ours `IP_Telephony`. Click **Next**.

3. Input the IP address of the clients or the group of clients that are to be controlled by this policy. Select **Specific address or addresses** and click the **New** button. We enter the IP address of the specific client as `190.86.23.1` as per our problem definition (Figure 18-26). Click **OK** and then click **Next** when you return to the wizard.

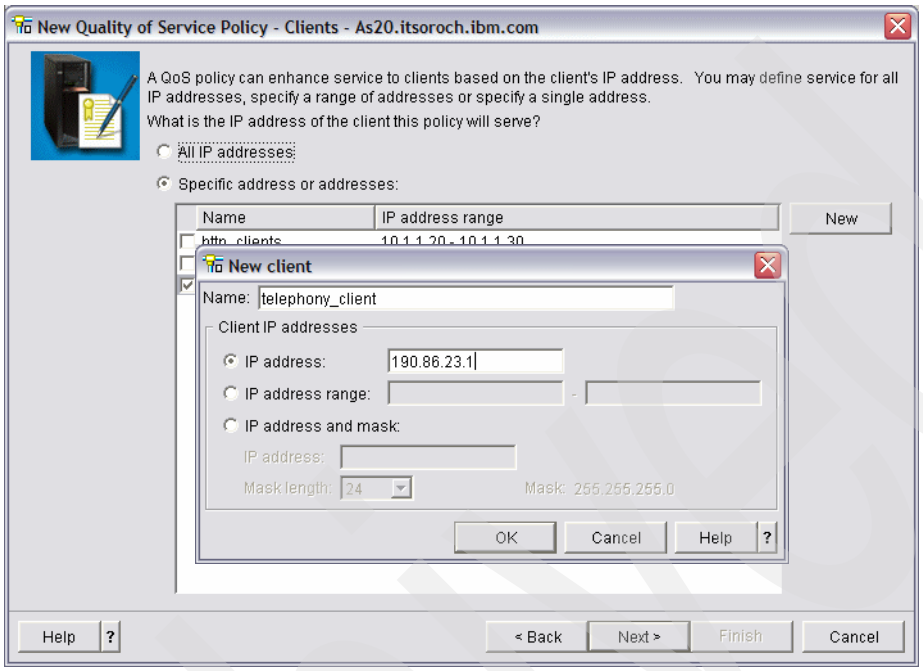


Figure 18-26 Int Service Policy: Client IP

4. In the next window add the port that you want this policy to act on and the protocol. Select **Specific port or range of ports** and click the **New** button. Supply a meaningful name and `2427`, as the traffic will be using this port according to our problem definition. Click **OK**. Select all protocols and click **Next**.
5. Specify the Local IP. It can be specified as all or, if the system has multiple IPs and NICs, as the specific IP on which the traffic is coming. We select **All** IP addresses. Click **Next** to continue.
6. In the next window we define the class of service and per-hop behavior (PHB) to be used by integrated services policy. For our example, we do not use PHB. Click **Next**.
7. Enter the parameters for bandwidth sizing as defined in Table 18-4. Fill in the parameters that are assigned in the Value column and click **Next** to continue.

Table 18-4 Int Serv: Performance Limits

Parameter	Description	Value
Maximum No. Flows	Specify the maximum number of connections (flows) this policy allows at one time. It can be between 1 to 65,535 and do not limit.	1
Aggregate Token Rate Limit	The rate limit specifies the number of bits per second allowed into a network. Any client requesting RSVP from the server will ask for a specific amount of bandwidth (flow limit). The QoS policy looks at the requested bandwidth and compares it with the rate and flow limits for this policy. If the request would cause the server to exceed its limits, the server denies the request.	11 Mbps

Parameter	Description	Value
Token Bucket Size	The token bucket size determines the buffer capacity, which holds bursts of data. Burst data is information that an application gives the server to send out, at a faster rate than it can exit. If an application quickly sends enough burst data to the server, the buffer fills up. If the application sends information slower than it can exit the server, the buffer empties. When data is leaving the server as fast as it is entering the server, then the token bucket size remains unchanged. When the buffer is filled, QoS treats additional data packets as out-of-profile.	16 Kbps
Token Rate Limit	This is the same as the Aggregate token rate limit. It is individual to each flow.	10 Mbps

8. Next we enter the schedule for the policy to be active. Select **Active during selected schedule** and click **New**. Give your new schedule a meaningful name and click **Add** to specify the time span you want the policy to be enforced. Enter the values to be active. We enter `1:00 pm` to `2:00 pm` for all days as per our problem (Figure 18-27). Click **OK**, click **OK** again, and click **Next** to continue.

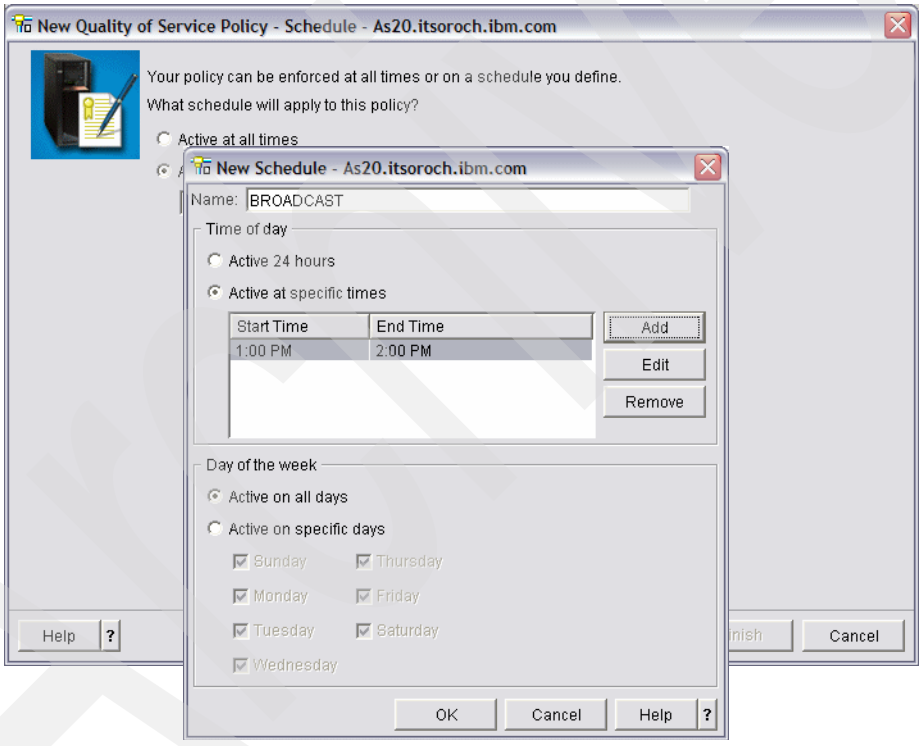


Figure 18-27 Int service: Schedule

9. The Summary window for the policy opens. Click **Finish** to save the policy.

10. In the QoS Server Configuration window, click **Server** → **Update** (Figure 18-28) to update the active server with the current configuration.

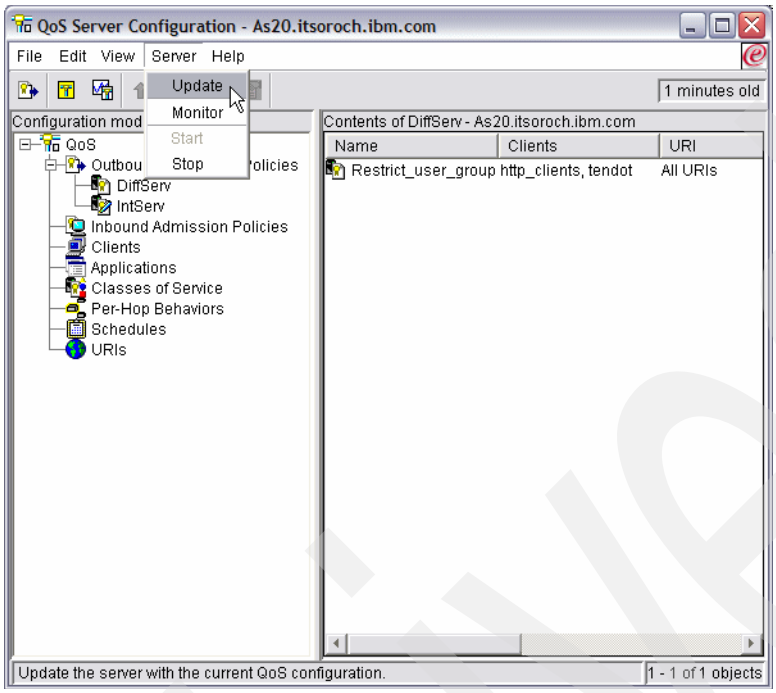


Figure 18-28 QoS Configuration: Update Config

Step 2: Test the configuration

To verify that the policy is working as it is supposed to, start the QoS monitor:

1. In iSeries Navigator, right-click **Quality of Service** and choose **Monitor**.
2. In the Monitor window, select **File** → **Start QoS Data Collection** (Figure 18-29).

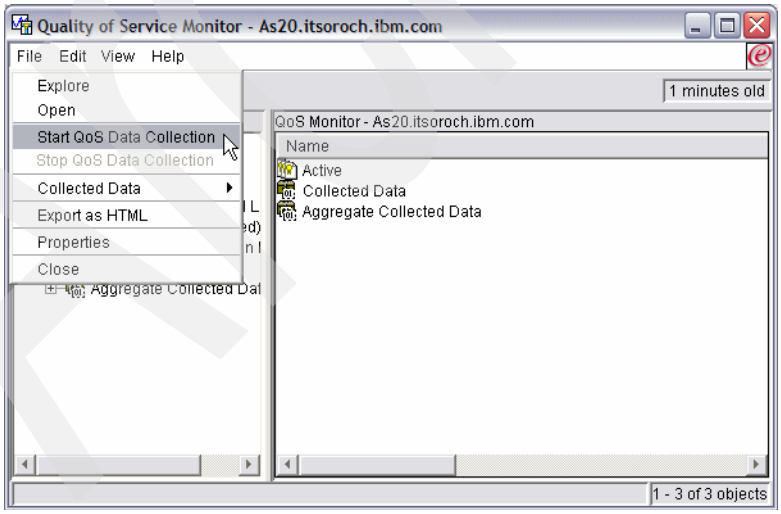


Figure 18-29 Monitor: Start QoS Data Collection

3. Select the policy that you want to monitor and refresh the window to see the updates.
4. Make sure to check measured fields such as bit rate, packet total, bits total, and bits non-conformant. Based on these values, you can further fine-tune the policy.

Tip: Remember that the Active monitor will only show the policy while it is active. In our scenario, this is from 1:00 p.m. till 2:00 p.m.

5. Close the monitor. From the QoS Server Configuration Outbound Bandwidth Policies Contents, right-click the IntServ policy name and select **Properties**. These windows enable you to edit the policy's properties as well as schedule, client, applications, and traffic management, and so on.

Review, conclusions, and references

After the policies for QoS are defined, analyze the data in the in monitor. The most interesting fields are those that obtain their data from your traffic. Make sure to check all relevant fields and fine-tune the policy based on this data until you get your desired result. In V5R2 and later releases, the data from the monitor can also be exported and stored for further reference.

For further information about the QoS, read RFCs 2474, 2475, 2597, and 2598.

To help you plan the QoS in your environment, a step-by-step planning advisor can be found in the iSeries Information Center. To access the Planning Advisor, visit:

`http://publib.boulder.ibm.com/infocenter/iseries/v5r4/topic/rzall/rzalladvqos00.htm`

Part 3

Advanced administration

Now that we have detailed a lot of the functions that you will need to help you automate your TCP/IP network, and shown you how to apply these functions step-by-step to solve some common networking problems, we finish by taking things one step further.

In this part of the book, we give you tips and techniques for optimizing performance in your dynamic network. We also cover important considerations when starting and ending TCP/IP. In addition, we introduce some of the many ways to programmatically interface with the TCP/IP stack and various applications. Finally, we wrap up with a discussion of what to do if something goes wrong (not that it ever should with System i).

19

Optimizing performance in a TCP/IP network

Although your TCP/IP network may be up and working fine, your network connectivity may not be optimized to deliver the best possible performance. If each of the client workstations can access all of the needed applications on the server, why would you risk altering the network configuration? For example, if it takes more than six hours to replicate a given database over the network, what if this replication time could be cut in half?

There is no guarantee that any of the items listed in this chapter could result in that significant of a performance boost, but using a combination of the items recommended here will help to improve your network configuration and enable you to obtain optimal performance.

19.1 Network/Line Description settings

When considering “optimizing throughput between two systems,” what is often overlooked is the word “between.” Many IT folks concentrate solely on the two endpoint systems and do not focus on the infrastructure between these two systems, which could very well be the cause of their throughput headaches.

The first thing to consider is the network bandwidth available for your application. You must consider all “hops” in the communications path to the remote system. At each network segment there might be different bandwidth levels available to this traffic. For instance, it probably does you no good to have 1 Gb adapters at each endpoint system when your high-throughput traffic flows over a 56 Kb link.

Also consider the networking hardware and software used between the two endpoint systems. It is possible that a firewall or other packet filtering device is affecting throughput, although most likely these devices would affect connectivity. The use of proxy servers could also introduce delays. A DNS server that is down can also cause long initial connectivity delays, appearing as throughput delays, in many cases.

19.1.1 Line Description configuration

There are two values in the configuration of an i5/OS line description that should be checked to make sure that they match the hub/switch in your network:

- Line Speed: This setting specifies the speed of the adapter in bits/second. The line speed choices vary depending on the type of adapter being used. Some adapters support multiple line speeds. Obviously the higher line speeds will result in higher performance, but the i5/OS line description line speed has to match the hub/switch to which the adapter is connected. For the fewest problems and optimal performance, do not use *AUTO if the port on the switch/hub is configured for a fixed line speed.
- Duplex: This setting specifies whether the adapter can send and receive data simultaneously. In half-duplex mode, the hardware must alternate between sending data and receiving data. In full duplex mode, one wire in the cable is dedicated to send data, and another wire in the cable is dedicated to receive data, therefore allowing data to be sent and received simultaneously. For optimal performance full duplex should be used, but this setting must match what was configured on the hub/switch.

The top communications throughput problem from an IBM service perspective in terms of frequency and severity is an incorrectly set line speed or duplex parameter (or both) in the line description. These two values should match the settings on the port of the hub/switch this network adapter plugs in to**.**

19.1.2 Maximum Frame Size and Maximum Transmission Unit (MTU)

Another value in the i5/OS line description configuration that has a large impact on performance is the Maximum Frame Size. This setting specifies the largest frame size that can be transmitted and received on this line description. The possible values for this setting are dependent on the type of adapter being used and which protocols are being used on that adapter.

The maximum frame size is typically set to the largest size that is supported by that particular adapter. However, the frame size specified for a given line description/adapter may not be consistent with the frame sizes being used by the rest of the network. To solve this issue, the i5/OS line description’s Maximum Frame Size can be overridden by specifying a lower value in the Maximum Transmission Unit (MTU) field of the TCP/IP Interface configuration. After

this is done, any traffic that uses this particular TCP/IP interface will then use this value for its MTU, regardless of what the line description was configured for. The default value for this parameter, *LIND, indicates that the MTU value should be based on the frame size that was specified in the line description.

In addition to allowing the line description's frame size value to be overridden, the TCP/IP Interface's MTU value can also be overridden. This is done by specifying a lower MTU value in the TCP/IP Route configuration. If this is done, any traffic that uses this route will use the route specified value for its MTU, regardless of what the MTU is in the TCP/IP interface or the frame size is in the line description. The default value for the route MTU parameter, *IFC, indicates that the MTU value specified in the TCP/IP Interface (that is bound to this route) should be used.

So in summary, maximum frame size on the line description can be overridden by the Maximum Transmission Unit (MTU) setting in the TCP/IP interface defined on that line description, which in turn can be overridden by the MTU setting in the TCP/IP routes that are defined. Figure 19-1 shows this relationship.

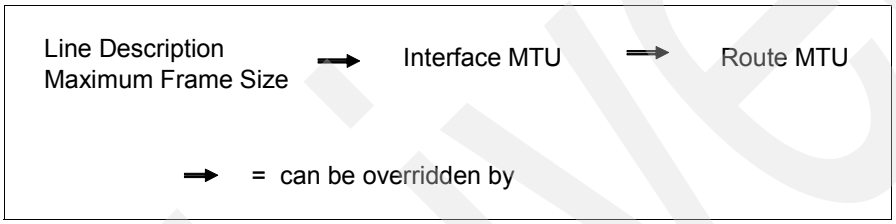


Figure 19-1 Overriding MTU values

Figure 19-2 illustrates these concepts.

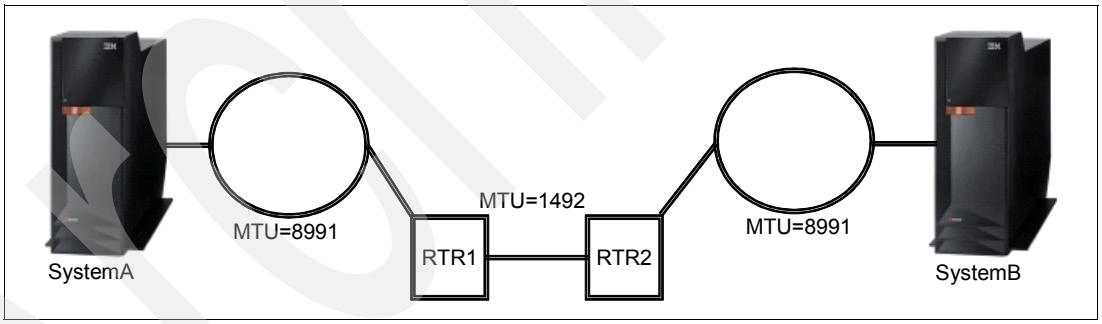


Figure 19-2 Overriding MTU values illustration

In this example, both System A and B have 1 Gb Ethernet adapters and are using Jumbo Frames. Unfortunately, there is a link between the two systems that has an MTU setting of 1492. In order to allow System A and B to communicate efficiently, a route should be defined to override the line description's maximum frame size. This route should force traffic that is sent between the two systems to use an MTU size of 1492. A route would have to be defined on each system because both have 1 Gb Ethernet adapters.

19.2 TCP/IP send and receive buffers

The TCP send and receive buffers are the buffers that reside between the application and the TCP layer in the TCP/IP stack, which is illustrated in Figure 19-3.

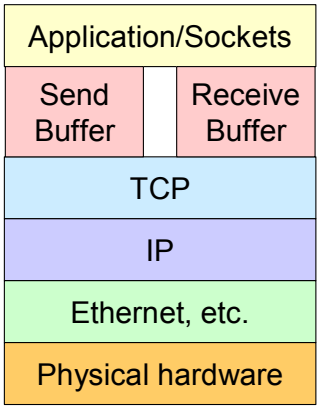


Figure 19-3 TCP/IP stack

Both of these buffers default to 8 K (8192) bytes in size. This 8 K buffer space for the sends and receives is taken dynamically as needed for an individual connection. If this amount is not needed for a given connection, then it is not allocated. These 8 K buffer sizes are loosely enforced. i5/OS will buffer slightly more than these configured amounts during data transfers. The TCP receive window size for the connection is based on the receive buffer value.

The TCP receive buffer is the amount of buffer space that is set aside to hold incoming application data that has not yet been received by the local application. When the receive buffer space starts to fill up, TCP will *advertise* a smaller and smaller window size on the TCP/IP connection. The TCP receive buffer can be viewed in a communications trace by viewing the `Window:` portion of the trace. This value *advertises* the maximum amount of data, in bytes, that the system has room for in its receive buffer. After the TCP receive buffer has been completely filled, the remote system will be prevented from sending any additional data until the local application receives some of the data out of this buffer.

If the remote system's TCP receive buffer has been completely filled, the TCP Window size will be advertised as 0 and the sending system will then start buffering outgoing data in the TCP send buffer space. If the TCP send buffer is completely filled, the application will be prevented from sending any additional data.

The default sizes of the TCP/IP send and receive buffers are set through the CHGTCPA command, as illustrated in Figure 19-4.

```
                    Change TCP/IP Attributes (CHGTCPA)

Type choices, press Enter.

TCP keep alive . . . . . .    320          1-40320, *SAME, *DFT
TCP urgent pointer . . . . .  *BSD          *SAME, *BSD, *RFC
TCP receive buffer size  . .  8192         512-8388608, *SAME, *DFT
TCP send buffer size . . . .  8192         512-8388608, *SAME, *DFT
TCP R1 retransmission count   3            1-15, *SAME, *DFT
TCP R2 retransmission count   16           2-16, *SAME, *DFT
TCP minimum retransmit time   250          100-1000, *SAME, *DFT
TCP time-wait timeout  . . .  120          0-14400, *SAME, *DFT
TCP close connection message  *THRESHOLD   *SAME, *THRESHOLD, *ALL...
UDP checksum . . . . . .      *YES         *SAME, *YES, *NO
Path MTU discovery:
  Enablement . . . . . .      *YES         *SAME, *DFT, *NO, *YES
  Interval . . . . . .        10           5-40320, *ONCE
IP datagram forwarding . . .  *YES         *SAME, *YES, *NO
IP source routing  . . . . .  *YES         *SAME, *YES, *NO
IP reassembly time-out . . .  10           5-120, *SAME, *DFT
                                                                More...
F3=Exit   F4=Prompt   F5=Refresh   F10=Additional parameters   F12=Cancel
F13=How to use this display        F24=More keys
```

Figure 19-4 Using CHGTCPA command to set buffers

The settings in CHGTCPA are system-wide, meaning that all TCP/IP connections across the system will use these send and receive buffer size settings by default. This default 8K setting in CHGTCPA was fine for Telnet and simple interactive applications. However, 8K buffer sizes are often much too small for applications that send and receive large amounts of data.

No application that sends and receives large amounts of data should rely on the system-wide CHGTCPA send and receive settings, but rather update the buffer sizes itself on a per-connection basis. The SO_SNDBUF and SO_RCVBUF socket options enable an application to set these send and receive buffers for a given connection. All applications that are performance sensitive should be using the SO_SNDBUF and SO_RCVBUF socket options to update the buffer sizes to more appropriate values.

If 8K is too small for most applications, what should the system-wide settings be for CHGTCPA and what sizes should be used for SO_SNDBUF and SO_RCVBUF? The answer depends on several factors such as:

- How much bandwidth is available on this interface/network?
- Is the bandwidth dedicated to this system/application?
- What is the quality of the network? Packet loss? Latency?
- How much data traffic is being generated by the system/application?

If the network path has dedicated bandwidth and allows for little to no packet loss, then it is okay to have your send/receive (depending on the role your i5/OS is playing) buffers increased from the default value of 8K. However, large buffer sizes used over a poor network could worsen already poor throughput.

The only real way to determine the optimal buffer sizes for both CHGTCPA as well as the socket options is through trial-and-error. The settings that are ideal for some applications may not be appropriate for others. Some possible starting values to consider using if you are experiencing performance/throughput concerns are 64K for the CHGTCPA (system-wide) settings and 1 MB for the SO_SNDBUF and SO_RCVBUF settings for those applications that deal with bulk data throughput.

Example 19-1 shows an example of how to set the SO_SNDBUF and SO_RCVBUF socket options.

Example 19-1 How to set the SO_SNDBUF and SO_RCVBUF socket options

```
int sd, rc, bufsize;
bufsize = 64 * 1024;

rc = setsockopt(sd, SOL_SOCKET, SO_SNDBUF,
                (char *)&bufsize, sizeof(bufsize));
if (rc == -1)
{
   perror("setsockopt failed");
   exit(-1);
}

rc = setsockopt(sd, SOL_SOCKET, SO_RCVBUF,
                (char *)&bufsize, sizeof(bufsize));
if (rc == -1)
{
   perror("setsockopt failed");
   exit(-1);
}
```

19.3 Sockets programming tips and techniques

Although you may have optimally tuned your network configuration with 1 Gb Ethernet adapters/switches, Jumbo frames, and large send/receive buffers; a poorly written TCP/IP application will still perform poorly. In this section we discuss simple programming tips and techniques for optimal performance from your socket applications.

Additional information about each of the socket concepts covered in this section can be found in the V5R4 iSeries Information Center at:

http://publib.boulder.ibm.com/infocenter/iseries/v5r4/index.jsp

19.3.1 IFS versus Sockets APIs

There are two different sets of APIs that can be used to transmit data over a network:

- read() and write() IFS APIs
- send() and recv() socket APIs

The IFS read() and write() are generic APIs that can be used to read/write data to files as well as sockets. These APIs incur additional overhead in order to allow them to work against both files and sockets.

The send() and recv() APIs, on the other hand, can only be used with sockets. These APIs have been optimized to bypass much of the generic descriptor management logic that is required when using read() and write(). This optimization was implemented in V4R5 and is unique to i5/OS.

Both the IFS and the socket APIs can be used to transfer data over the network, but the socket APIs require less CPU cost and fewer instructions to send and receive the same amount of data on both the sending and the receiving systems every time that an API call is made.

19.3.2 Nagle algorithm and TCP_NODELAY

By default, TCP/IP may introduce a small delay in the sending of data due to logic in the TCP/IP stack referred to as the Nagle algorithm (RFC 896). This algorithm works by buffering several small outgoing messages and sending them all at once. Specifically, as long as there is a sent packet for which the sender has received no acknowledgement, the sender should keep buffering its output until it has a full packet's worth of output. Part of the original design for Nagle's algorithm was to efficiently handle telnet connections. The authors did not want every key stroke on a telnet session to be sent over the network in a separate TCP/IP packet.

Most applications do not transfer data one byte at a time, but are request/response-based and do not want any delays in the processing of their requests. All TCP/IP sockets-based applications that send data (in chunks larger than 1 byte at a time) should enable the TCP_NODELAY socket option on each TCP/IP connection to prevent this send-side delay from occurring.

This Nagle algorithm also interacts badly with TCP/IP delayed acknowledgements. If both TCP/IP delayed acknowledgements and Nagle are being used, large performance delays can occur.

The TCP_NODELAY option is enabled on a given TCP/IP connection using the setsockopt() socket API, as illustrated in Example 19-2.

Example 19-2 Using setsockopt() socket API

```
int sd, rc, flag = 1;

rc = setsockopt(sd, IPPROTO_TCP, TCP_NODELAY,
                (char *)&flag, sizeof(flag));
if (rc == -1)
{
   perror("setsockopt failed");
   exit(-1);
}
```

19.3.3 Sending multiple data buffers efficiently

In many cases an application may have multiple data buffers that it needs to transmit over to the remote system. For example, it may have a fixed header followed by one or more variable-length buffers. In order to send multiple blocks of data, the sockets send() API could be invoked multiple times, once for each buffer that has to be sent. Alternatively, a more efficient method to send multiple data buffers is with the sendmsg() API, which allows up to 16 data buffers to be sent at one time.

Using the sendmsg() call to send multiple buffers instead of calling the send() API multiple times will result in less processing on both the sending and receiving systems. On the sending system there will be fewer API calls being made, fewer calls down to the TCP/IP stack, fewer calls down to the adapter, and also most likely fewer TCP/IP packets sent out over the network because multiple application data buffers will be combined. On the receiving system, if there are fewer TCP/IP packets arriving (due to the remote application's send data buffers being combined), then less processing will be required to receive them. Hence the sendmsg() API will allow processing to be done more efficiently on both the sending and the receiving systems.

Like the sendmsg() API on the sending system, the recvmsg() API can be used on the receiving system to receive incoming data into multiple different data buffers. The benefit provided by recvmsg() over multiple recv() calls is much less than the combining of send() calls on the sending side.

19.3.4 Receiving data with MSG_WAITALL and SO_RCVLOWAT

When transmitting data over a TCP/IP connection, typically the receiving application can make no guarantee how much data will be returned on a given receive operation. The amount of data returned by a given recv() call will vary based on a number of conditions such as:

- Size of the buffer specified on the recv() call
- Size of the buffers being sent by the remote application
- Timing issues between when recv() is called versus when the data was transmitted
- Routers and switches in the network
- MTU size of each of the route segments between the two systems
- Amount of data already queued up to be received on the receiving system
- Amount of data this is currently flowing over the network
- Amount of data queued up to be sent on the sending system
- The values of various socket options and flags

Because the amount of incoming data returned by recv() varies (it could be as little as 1 byte), it is common for an application to call recv() over and over again until all of the incoming data has been received. Calling the recv() API repetitively in such a loop is inefficient. There are two alternative methods that an application can use to control how much data is returned on a given receive operation: the MSG_WAITALL flag and the SO_RCVLOWAT socket option.

MSG_WAITALL is a message flag that specifies that the data buffer specified on this given recv() call should be completely filled by i5/OS before control is returned to the application. If there is not enough data queued up in the TCP receive buffer to completely fill the specified data buffer, i5/OS will block waiting for additional incoming data to arrive. Support for this message flag was added in V5R2.

MSG_WAITALL is useful if the application is using fixed buffer lengths or if the application has a fixed header on the front of the data that is being transmitted. In the case of a fixed header, the application can guarantee that the entire fixed header is received at one time by using MSG_WAITALL on the recv() call. After the header has been received, all of the rest of the incoming data can then be received using a second recv() call with another MSG_WAITALL flag specified.

For example, assume a recv() with MSG_WAITALL is specified and the data buffer specified on that recv() call was 64 bytes. If remote system sends only 30 bytes, the recv() call will block waiting for another 34 bytes to arrive before it completes.

The SO_RCVLOWAT socket option is similar to the MSG_WAITALL flag except that this socket option specifies a minimum amount of data that has to be returned on all recv() calls that are done on this TCP/IP connection. Support for this socket option was added in V4R3.

If more than the minimum is sent while blocked on a recv() call, i5/OS will return the entire amount received when the minimum requirement is met. For example, if SO_RCVLOWAT was set to a value of 64 bytes and a recv() was done that specified a 512-byte buffer, when 30 bytes arrive from the remote system, the recv() call would remain blocked waiting for at least another 34 bytes to arrive. If the sending system then sent over a 200-byte block of data, then recv() would complete with 230 bytes (as long as the 200 bytes did not get fragmented by the switches and routers on the network).

19.3.5 Waiting for incoming data – SO_RCVTIMEO

Another useful socket option that is not well known is the SO_RCVTIMEO. This option allows an application to specify how long each recv() call should block waiting for incoming data to arrive from the remote system. When this option is set, all subsequent recv() calls will block only the length of time that was specified in this option. In order to use this socket option, the _XOPEN_SOURCE constant must be defined at the top of your program to a value of 520 or greater before any header file is included, as illustrated in Example 19-3.

Example 19-3 Defining _XOPEN_SOURCE constant

```
#define _XOPEN_SOURCE 520

#include <sys/socket.h>
#include <netinet/in.h>
```

The SO_RCVTIMEO socket option can be combined with the MSG_WAITALL flag or the SO_RCVLOWAT option that was mentioned earlier.

19.3.6 Inheritance of socket options from listening socket

Throughout this section we have discussed several useful socket options such as:

- SO_RCVBUF
- SO_SNDBUF
- TCP_NODELAY
- SO_RCVLOWAT
- SO_RCVTIMEO

Calling the setsockopt() API to set each of these options on a given connection takes a measurable amount of overhead and processing time. This processing time can be significant for short-lived request/response-type connections. This overhead processing time can be eliminated for server applications by setting each of these options on the listening socket before the incoming connection is accepted. Although most of these options do not make much sense on a listening socket, the settings will be saved. All socket options, including the non-blocking option, will be inherited by all new incoming connections that arrive on this listening socket.

Using inheritance of socket options from the listening socket may not seem that useful, but it can have a significant performance impact if the server application handles hundreds of connections per second.

19.3.7 Asynchronous I/O APIs on i5/OS

When a socket application has to handle multiple connections simultaneously, several design decisions should be considered. In some cases, multiple threads or jobs may be considered. The use of the select() API may also be used to allow a single thread to monitor activity on a number of sockets. If you have a socket application that has to handle multiple connections simultaneously, is multithreaded, or uses the select() API, then you should consider modifying the application to use the i5/OS Asynchronous I/O socket APIs. These APIs are:

- Asynchronous Socket APIs:
  - QsoStartAccept()
  - QsoStartRecv()
  - QsoStartSend()

- Asynchronous Secure Socket APIs:
  - gsk_secure_soc_startRecv()
  - gsk_secure_soc_startSend()
  - gsk_secure_soc_startInit()
- I/O Completion Port APIs:
  - QsoCreateIOCompletionPort()
  - QsoDestroyIOCompletionPort()
  - QsoWaitForIOCompletion()
  - QsoPortIOCompletion()

The use of these APIs allows more efficient use of system resources, improved scalability, and improved throughput and performance. Many existing i5/OS servers have already switched over to use these APIs and they have observed large increases in performance as a result. Those servers that make use of these i5/OS Asynchronous socket APIs include: HTTP Server (Powered by Apache), Domino, FTP, and NetServer.

In order to obtain the highest possible performance for your server application, it is recommended that the Asynchronous I/O APIs listed above be used.

20

Considerations for starting and ending TCP/IP

This chapter discusses the different ways that TCP/IP may be started. It also discusses potential problems that may occur if starting TCP/IP is not carefully planned.

The discussion includes the following topics:

- Starting TCP/IP: IPL attributes versus start-up program
- Starting TCP/IP on systems with a 3494 Tape Library
- Restricted state
- Ending TCP/IP
- Starting and ending TCP/IP references

20.1 Introduction

TCP/IP may be started in a number of ways. It is important for a system administrator to understand the methods for starting TCP/IP in order to use the best possible way for a specific system. Race conditions, unnecessary start-up requests by multiple jobs, or possible start-up errors may be reduced or eliminated by understanding the concepts that are presented in this chapter.

20.2 Starting TCP/IP: IPL attributes versus start-up program

There is an IPL attribute (DSPIPLA/CHGIPLA parameter STRTCP) that can be set to control whether TCP/IP will be started as part of IPL processing or when the iSeries is brought out of restricted state. If the IPL attribute STRTCP is set to *YES then TCP/IP will be started with whatever command defaults are set on the system for an IPL or coming out of restricted state. If the IPL attribute STRTCP is set to *NO, then TCP/IP will not be started for the previously mentioned cases.

TCP/IP can be started in the following ways:

- As part of IPL processing. Set IPL attribute STRTCP(*YES). TCP/IP will be started as part of IPL processing using the STRTCP command defaults.
- As part of a “start-up” program. Set IPL attribute STRTCP(*NO). TCP/IP will not be started as part of normal IPL processing. In this case you must have a STRTCP command coded in your start-up program to have TCP/IP started.
- As part of vary on of a 3494 Tape Media Library device. Refer to 20.3, “Starting TCP/IP on systems with a 3494 Tape Library” on page 659, for information about this subject.
- As part of coming out of normal iSeries processing when coming out of restricted state. Set IPL attribute STRTCP(*YES). TCP/IP will be started as part of normal iSeries processing coming out of restricted state using the STRTCP command defaults.
- By issuing the STRTCP CL command from a command line.
- By issuing a STRTCP CL command from a program.
- Vary on of an NWSD. The vary on processing will start TCP/IP.

You should ensure that TCP/IP is started in only one place on the system. We recommend using the IPL attribute STRTCP set to *YES to start TCP/IP as part of normal IPL processing.

A few customers have encountered problems with getting all of the interfaces and servers that are configured to be autostarted with using the IPL attribute STRTCP set to *YES. Because of the workload that your system may have, during IPL you may experience delays in the vary on of communication lines that your IP addresses are configured with. In this case we recommend that you move the starting of TCP/IP to your startup program. See 20.7, “Starting and ending TCP/IP references” on page 663, for sources of additional information about this topic.

As part of i5/OS V5R4 start TCP/IP processing, the TCP/IP servers and PPP profiles that are configured to be autostarted will be started as a submit job from the QSYSWRK/QTCPIP job. The submit job to start the TCP/IP servers will be done after the interfaces that are configured to be autostarted have been processed. At this point the QTCPSTSVRS job will start the TCP/IP servers. You should keep in mind that the starting of the TCP/IP interfaces is not the same as an interface being “active.” If a TCP/IP server is dependant on an interface being active, it is normal for the server to have logic to retry for successful operations with that interface for a short period of time until the interface becomes active. There have been cases

when a user-defined server is configured to be started as part of start TCP/IP processing but does not have the logic to retry its operation with an interface and fails to start because the interface has been delayed becoming active. In this case either the server must be modified or its configuration changed to retry during its initialization or it should not be configured to be started when TCP/IP starts in order to avoid these type of failures.

You can view the logging of what happened during the starting of TCP/IP as part of the IPL process in the QSTRTCP job (user profile QPGMR). You can view the results of starting the TCP/IP servers by viewing the QTCPSTSVRS job log (user profile QTCP). After TCP/IP has started, the persistent jobs QSYSWRK/QTCPIP and QSYSWRK/QTCPMONITR will be present until TCP/IP has been ended.

The QSYSWRK/QTCPIP job processes requests to start and end TCP/IP interfaces as part of STRTCPIFC and ENDTCPIFC processing. This job is also responsible for performing the starting of interfaces that are configured to be autostarted as part of start TCP/IP processing. This job's job log contains information about the processing of starting or ending interfaces, interfaces becoming active or inactive, and important information about processing it is doing. The work done by the QSYSWRK/QTCPIP job is beyond the scope of this paper and therefore its operation will not be described here. This job is important for the operation of TCP/IP and in general processes events that are posted to it by the TCP/IP SLIC protocol stack and passes them onto the QSYSWRK/QTCPIP job for logging. Events such as interfaces becoming "active" or "inactive" are handled by the QSYSWRK/QTCPMONITR job.

When TCP/IP is started there is a remote possibility that you can receive a TCP1A1B (`Not able to determine if job NNNNNN/QTCP/QTCPIP started.`) message in response to the start TCP/IP request (STRTCP). The STRTCP command is waiting for a response to be posted back to it from the QSYSWRK/QTCPIP job. As mentioned previously, part of start-up processing for start TCP/IP is to start the QSYSWRK/QTCPIP job. If the system is busy the starting of this job may be delayed and as a result you might see this message posted to the job log. If this occurs, you should check that the QSYSWRK/QTCPIP did start. You can use the WRKACTJOB CL command to determine whether the job started.

20.3 Starting TCP/IP on systems with a 3494 Tape Library

With the release of i5/OS V5R2, support was added to allow attaching a 3494 Tape Library Device using TCP/IP. The 3494 will request that TCP/IP be started and start an IPv4 address (specified by the LCLINTNETA parameter on the CRTDEVMLB, CHGDEVMLB, or CFGDEVMLB command). The request to start TCP/IP is a result of a 3494 Tape Library device operation. This request is valid for both normal operating mode and restricted state.

When a start TCP/IP is requested by the 3494 Tape Library device driver, it is started programmatically as STRTCP STRSVR(*NO) STRIFC(*NO) STRPTPPRF(*NO) STRIP6(*NO). This means that the TCP/IP protocol stack is started, but none of the TCP/IP servers, interfaces, PPP profiles, or IPv6 will be started. However, the IPv4 address configured for the 3494 Tape Library is started.

When TCP/IP is started in the restricted state because of a request by the 3494 Tape Library device driver, TCP/IP is started with STRTCP STRSVR(*NO) STRIFC(*NO) STRPTPPRF(*NO) STRIP6(*NO), and the IPv4 address configured for the 3494 Tape Library device is started. If the iSeries is then brought out of restricted state by starting the subsystems (STRSBS command) and the IPL attribute is set to start TCP/IP (STRTCP *YES) then, assuming that the normal defaults are set for the STRTCP command, the TCP/IP servers, IP interfaces, and PPP profiles that are configured as automatically started will be started.

If TCP/IP has been ended and the iSeries is in normal operating mode and the 3494 Tape Library device operation requests that TCP/IP be started, TCP/IP will also be started with STRTCP STRSVR(*NO) STRIFC(*NO) STRPTPPRF(*NO) STRIP6(*NO). To start the TCP/IP servers, interfaces, and PPP profiles in this scenario, you have basically two options. One is to end TCP/IP, then issue a STRTCP CL command to get the servers, interfaces, IPv6 and PPP profiles started. The other option is to issue individual CL commands to first start the TCP/IP interfaces (STRTCPIFC INTNETADR(*AUTOSTART)), then start the TCP/IP servers (STRTCPSVR SERVER(*AUTOSTART)) and the host servers (STRHOSTSVR SERVER(*ALL)) and the PPP profiles (STRTCPPTP CFGPRF(*AUTOSTART)). If you need to start the IPv6 portion of the TCP/IP protocol stack, you have to end TCP/IP and restart it as there is not an option to start IPv6 separate from start TCP/IP (STRTCP).

When TCP/IP is started as a result of a STRTCP CL command being issued assuming typical command defaults STRTCP STRSVR(*YES) STRIFC(*YES) STRPTPPRF(*YES) STRIP6(*YES), the TCP/IP server, interfaces (IPv6 and IPv6), and PPP profiles will be started. This would be the case when TCP/IP is started as a result of starting TCP/IP at IPL time by setting the IPL attribute (STRTCP *YES) to start TCP/IP. If TCP/IP is started in restricted state by a user issuing at STRTCP STRSVR(*NO) STRIFC(*NO) STRPTPPRF(*NO) STRIP6(*NO), and then the system is brought out of restricted state by starting the subsystems, the TCP/IP servers, IP interfaces, and PPP profiles will not be started. We recommend that you end TCP/IP before starting the subsystems (STRSBS).

To avoid potential problems with starting TCP/IP, with a System i platform that has a 3494 Tape Library configured to use TCP/IP, the following information should be taken into consideration.

A 3494 Tape Library can be configured to be varied on at IPL time. If the iSeries IPL attributes are configured to have TCP/IP started at IPL time (CHGIPLA STRTCP(*YES)), it is strongly recommended that the 3494 Tape Library be configured to not be varied on at IPL time.

Depending on the mix of the workload that the iSeries has during an IPL, there can be situations in which starting TCP/IP will not be completely successful. If TCP/IP is configured to start at IPL and the 3494 Tape Library is configured to vary on at IPL, then TCP/IP servers that are configured to be started automatically may not be started. The situation arises as a result of a race condition to start TCP/IP as a result of the vary on processing of the 3494 Tape Library occurring before the request to start TCP/IP as specified by the IPL attributes processing to start TCP/IP as part of IPL occurs.

To avoid potential problems associated with starting TCP/IP, it is recommended that you configure your 3494 Tape Library (CRTDEVMLB ONLINE(*NO)) to not be varied on at IPL if TCP/IP is configured to be started as part of the IPL (IPLA STRTCP *YES). The 3494 Tape Library, however, can be varied on using your system start-up program after verifying that TCP/IP is fully active. See Chapter 21, “Checking TCP/IP status programmatically” on page 665, for more information.

20.4 Restricted state

Attention: This section focuses on restricted state operation only.

The design for being able to start and end TCP/IP in restricted state is to support a limited set of capabilities that save and restore operations require. Typically an application running in the Controlling Subsystem requires that one IPv4 interface be active so it can communicate using the communications network.

Refer to 20.3, “Starting TCP/IP on systems with a 3494 Tape Library” on page 659, for details of restricted state operation and the specifics of the 3494 Tape Media Library support. There is a specific set of operation capabilities that pertain only to the 3494 and not the general topic of TCP/IP operation in restricted state.

When the iSeries is in restricted state, it is possible to start and end TCP/IP and to start and end IPv4 interfaces. There are some iSeries limitations as to what can and cannot be done when operating in restricted state as it relates to subsystems and jobs. See 20.7, “Starting and ending TCP/IP references” on page 663, for sources of additional information.

TCP/IP can only be started with all of the part STRTCP parameters specified as *NO; specifically, STRTCP STRSVR(*NO) STRIFC(*NO) STRPTPPRF(*NO) STRIP6(*NO). Any other parameter specification is not valid. TCP/IP can be ended in restricted state by issuing the ENDTCP CL command.

IPv4 addresses can be started and ended by issuing the STRTCPIFC and ENDTCPIFC CL commands.

While you are operating in restricted state you can use the NETSTAT CL command and the options to check on the status of TCP/IP. In addition, the QTCP and QSYSOPR message queues can be displayed for additional information about such things as interfaces being “active” (TCP8A3E message) when you have requested to start them or “inactive” (TCP8A40 message) when you have requested to end them.

Starting TCP/IP servers, PPP profiles, and the IPv6 portion of the protocol stack are not permitted because the subsystems that would be required are not active.

If you start TCP/IP while in restricted state, it is recommended that you end TCP/IP before you bring the iSeries out of restricted state. You should verify that TCP/IP has ended (not active) before exiting restricted state. The results of this command inform you whether TCP/IP is active. It is also possible to obtain this information programmatically as mentioned previously.

When the iSeries is being brought into restricted state and TCP/IP is running, you should verify that TCP/IP has ended before you try to start it in restricted state. If you try to start TCP/IP before it has ended, you will receive a TCP1A04 message (TCP/IP currently active).

20.5 Ending TCP/IP

Attention: This section discusses ending TCP/IP in *normal* operating mode.

The recommended way to end TCP/IP processing is to issue the ENDTCP CL command., which is the complement of the STRTCP CL command. The processing that is performed at start TCP/IP time is taken into consideration at end TCP/IP time so as to comprehensively end the TCP/IP servers, interfaces, PPP profiles, and the protocol stack. Other mechanisms could be employed to perform end TCP/IP processing in aggregate, but we recommend that you issue the ENDTCP CL command.

You can determine whether TCP/IP has ended by issuing the NETSTAT CL command. When TCP/IP has ended, you will receive a TCP2670 message (`Not able to complete request. TCP/IP services are not available.`). You can also use the Retrieve TCP/IP Attributes (QtocRtcTCPA) API to programmatically retrieve information about the status of TCP/IP. See Chapter 21, “Checking TCP/IP status programmatically” on page 665 for more information. See 20.4, “Restricted state” on page 660, for information about ending TCP/IP in restricted state.

The following list briefly describes the processing order when you end TCP/IP:

1. End TCP/IP servers.
2. End TCP/IP interfaces and release the lock on the device (QTCPIP) for the Line Description.
3. End the QSYSWRK/QTCPIP and QSYSWRK/QTCPMONITR jobs.
4. End the SLIC TCP/IP protocol stack.
5. End TCP/IP processing.

When TCP/IP has ended you will receive a TCP1A01 message (`ENDTCP completed successfully`) message.

The processing for ending the TCP/IP servers involves making a program call to each of the configured TCP/IP server management programs. Each server management program is called in turn and is expected to do a minimum of processing with this in order to cause the end server processing in the appropriate server job or jobs. For additional considerations relating to user-defined servers, see 20.6.2, “User-defined servers” on page 663.

The processing for ending the TCP/IP interfaces is carried out by the QSYSWRK/QTCPIP job. A request is sent to the job to end each of the interfaces. The IPv6 interfaces and Stateless Address Configuration end processing is performed and then the IPv4 interfaces are ended. In general the end processing for interfaces should proceed relatively quickly. If there were to be any problems the information will be logged on the QTCPIP job log. You can view the job log after TCP/IP has been ended if you feel the need to do so.

The QSYSWRK/QTCPIP and QSYSWRK/QTCPMONITR jobs will be ended after all of the interfaces have been ended. In turn the SLIC TCP/IP protocol stack will be ended after these two jobs are ended. At this point the only work to be done is a bit of cleanup, and then the ENDTCP CL command processing program will return. End TCP/IP processing is complete.

Information about the processing that took place is logged on the QTCPIP job log and the QTCP and QSYSOPR message queues.

We recommend that you do not cancel the QSYSWRK/QTCPIP job. To end TCP/IP, you should issue the ENDTCP CL command. If you were to cancel the QSYSWRK/QTCPIP job, comprehensive end TCP/IP processing would not be performed. In this case the IP interfaces, SLIC TCP/IP protocol stack, and the QSYSWRK/QTCPMONITR jobs would end. The TCP/IP servers would not be ended. The servers might subsequently end because they encounter errors with not having the SLIC TCP/IP protocol stack active and they may not depending on what processing if any they might be performing.

20.6 Other considerations

This section covers other topics related to starting and ending TCP/IP.

20.6.1 Network servers

TCP/IP and TCP/IP interfaces may be started at the time a network server is varied on. There are several things that a user may be able to do in order to get better performance when varying on network servers after an IPL:

- Start TCP/IP prior to any vary on of network server descriptions via the IPL attributes or the start-up program, and have your network server descriptions configured so they do not automatically vary on at IPL.

- Ensure that interfaces associated with the network server descriptions do not specify AUTOSTART(*YES). This can improve the performance of starting TCP/IP because the QTCPIP job will not be issuing vary on requests for the network servers, which in turn could cause delays in processing other TCP/IP-related requests.
- When TCP/IP has been started, vary on the network servers. The VRYCFG command allows certain objects such as network servers to be varied on simultaneously by submitting multiple batch jobs (SBMMLTJOB parameter), so you should consider using this option if you have multiple network servers.

20.6.2 User-defined servers

Starting with i5/OS V5R2, users have the ability to define their own TCP/IP servers, which can be started via the Start TCP/IP (STRTCP) and Start TCP/IP Server (STRTCPSVR) commands or ended via the End TCP/IP (ENDTCP) and End TCP/IP Server (ENDTCPSVR) commands. When processing a user-defined server, these commands will call a user-written server management program. When designing a server management program, keep in mind:

- The server management program should not perform any long-running operations. It should leave long-running operations to the server job or jobs themselves. A long-running operation or a poorly written program could prevent other servers from starting or the start or end TCP/IP operation from completing normally.
- Starting TCP/IP should not be attempted by either the server management program or the server jobs. TCP/IP status as well as the status of resources such as TCP/IP interfaces may be checked programmatically.
- Implement retry logic in the event that a needed resource such as a TCP/IP interface is not available.

20.7 Starting and ending TCP/IP references

Additional information is available in the iSeries Information Center at:

`http://publib.boulder.ibm.com/infocenter/iseries/v5r4/index.jsp`

Information about restricted state is available in the iSeries Information Center under these topics:

- **Systems management → Basic system operations → i5/OS concepts → i5/OS restricted state**
- **Systems management → Work management → Manage work → Manage subsystems → Place the system in restricted state**
- **Systems management → Work management → Experience reports → Restricted state**

Information about starting TCP/IP and system startup programs is available in the iSeries Information Center under the topics:

- **Networking → TCP/IP troubleshooting → Troubleshooting tools and techniques → Troubleshooting tips → Verify system startup considerations for networking → Timing considerations**
- **Systems management → Basic system operations → Start and stop the server → Start the server → Change the IPL startup program**

21

Checking TCP/IP status programmatically

This chapter discusses the reasons why it is important for a TCP/IP application to check the status of TCP/IP in a proper way. It also provides an example of a program that checks TCP/IP status.

21.1 Considerations for checking TCP/IP status

Many applications rely on TCP/IP, so it is important that they check the status of TCP/IP in a proper manner. In addition, it may also be necessary for an application to determine whether the system is in a restricted state. Proper checking by an application can reduce the possibility of application errors involving the lack of resources or timing situations.

If TCP/IP is in the process of starting or ending, or if the system is in a restricted state, the necessary TCP/IP support for your application may not be available. In addition, when the system is in a restricted state, other resources such as subsystems may also be unavailable. In restricted state, only the controlling subsystem and a single user job are available. TCP/IP may be active in restricted state, and it is possible to start TCP/IP interfaces.

IBM i5/OS provides an Application Programming Interface (API) called Retrieve TCP/IP Attributes (QtocRtvTCPA) that can be used to determine the status of TCP/IP and whether the system is in restricted state. Only a single call to this API is needed to determine both of these conditions, as well as a lot of other information that may be useful to an application. Status information for both IPv4 and IPv6 is provided.

21.2 CL programming example for checking TCP/IP status

The following CL program uses the Retrieve TCP/IP Attributes (QtocRtvTCPA) API to check whether TCP/IP is active and whether the system is in restricted state.

Important: This example uses new CL support, which is available in i5/OS Version 5 Release 4 Modification 0 (V5R4M0).

1. Create a source file member for the CL program. This step assumes that a library MYLIB and source file MYFILE have already been created.
   a. Start an editing session using the following command:

      ```
      STRSEU SRCFILE(MYLIB/MYFILE) SRCMBR(RTVTCPSTS) TYPE(CLLE)
      ```

      Note: The file member type must be CLLE because you are creating an Integrated Language Environment® (ILE) CL program.

   b. Enter the program source statements, as shown in Example 21-1.

Example 21-1 CL program for retrieving TCP/IP status

```
PGM
             DCL        VAR(&RCVR) TYPE(*CHAR) LEN(140)
             DCL        VAR(&RCVRLEN) TYPE(*INT) LEN(4) VALUE(140)
             DCL        VAR(&FORMAT) TYPE(*CHAR) LEN(8) +
                          VALUE('TCPA0100')
             DCL        VAR(&ERRCODE) TYPE(*CHAR) LEN(8) +
                          VALUE(X'0000000000000000')
             DCL        VAR(&STSPTR) TYPE(*PTR) ADDRESS(&RCVR 8)
             DCL        VAR(&STSVAL) TYPE(*INT) STG(*BASED) LEN(4) +
                          BASPTR(&STSPTR)
             DCL        VAR(&LMTPTR) TYPE(*PTR) ADDRESS(&RCVR 136)
             DCL        VAR(&LMTVAL) TYPE(*INT) STG(*BASED) LEN(4) +
                          BASPTR(&LMTPTR)
```

```
            DCL        VAR(&ACTIVE) TYPE(*INT) LEN(4) VALUE(1)
            DCL        VAR(&SYSRSTD) TYPE(*INT) LEN(4) VALUE(1)
            /* Call the QtocRtvTCPA API to retrieve TCP/IP status.  */
            CALLPRC    PRC('QtocRtvTCPA') PARM((&RCVR) (&RCVRLEN) +
                         (&FORMAT) (&ERRCODE))
            /* Determine whether TCP/IP is fully active.            */
            IF         COND(&STSVAL *EQ &ACTIVE) THEN(SNDPGMMSG +
                         MSG('IPv4 TCP/IP is ACTIVE.'))
            ELSE       CMD(SNDPGMMSG MSG('IPv4 TCP/IP is not ACTIVE.'))
            /* Determine whether the system is in restricted state. */
            IF         COND(&LMTVAL *EQ &SYSRSTD) THEN(SNDPGMMSG +
                         MSG('The system is in restricted state.'))
            ELSE       CMD(SNDPGMMSG MSG('The system is NOT in +
                         restricted state.'))
ENDPGM
```

2. Leave the source editing session and save your changes.
3. Create the CL module object.

   ```
   CRTCLMOD MODULE(MYLIB/RTVTCPSTS) SRCFILE(MYLIB/MYFILE)
   ```

4. Create the program object.

   ```
   CRTPGM PGM(MYLIB/RTCPS) MODULE(MYLIB/RTVTCPSTS) BNDSRVPGM(QSYS/QTOCNETSTS)
   ```

   Note: This program binds to the service program QTOCNETSTS because it provides the QtocRtvTCPA API.

5. Run the program:

   ```
   CALL MYLIB/RTCPS
   ```

In this example, if IPv4 TCP/IP is fully active and if the system is not in restricted state, the following messages are sent to the job log:

```
IPv4 TCP/IP is ACTIVE.
The system is NOT in restricted state.
```

In order to check the status of IPv6 TCP/IP, the program in Example 21-1 on page 666 would have to change to specify the value TCPA1100 for the variable &FORMAT. The field used to check for restricted state is only available in format TCPA0100. Coincidentally, the fields for the status of IPv4 and IPv6 TCP/IP are at the same offset in the data returned by both the TCPA0100 and TCPA1100 formats.

21.2.1 References

Additional information is available in the iSeries Information Center at:

http://publib.boulder.ibm.com/infocenter/iseries/v5r4/index.jsp

Documentation for the Retrieve TCP/IP Attributes (QtocRtvTCPA) API is available in the iSeries Information Center under the topic **Programming** → **APIs** → **APIs by category** → **Communications** → **TCP/IP Management APIs**.

Information about CL programming is also available in the iSeries Information Center under the topic **Programming** → **CL** → **CL Programming**.

22

Using alias names and setting proxy ARP and preferred interface lists programmatically

Several new TCP/IP networking features and functions are being introduced in Version 5 Release 4 Modification 0 of i5/OS. These include the ability to assign a name to a TCP/IP interface that can be used in the place of an IP address in programs. Also, a new feature called the preferred interface list is being introduced to assist with handling adapter failures.

This chapter provides information about these new features in the following topics:

- Using interface alias names
- Proxy ARP and the preferred interface list
- Putting it all together
- References

22.1 Using interface alias names

Interface alias names provide a way to assign a name to a TCP/IP interface, which can simplify the design of programs that manage TCP/IP interfaces. Rather than hard-coding TCP/IP interface IP addresses, programs can be written to use alias names. An alias name can be assigned to an interface either through CL commands or iSeries Navigator. This includes the following set of CL commands:

- Add TCP/IP Interface (ADDTCPIFC)
- Change TCP/IP Interface (CHGTCPIFC)
- End TCP/IP Interface (ENDTCPIFC)
- Remove TCP/IP Interface (RMVTCPIFC)
- Start TCP/IP Interface (STRTCPIFC)

In addition to CL command support, the Convert Interface ID (QtocCvtIfcID) API is available to convert between alias names and IP addresses. This API supports both IPv4 and IPv6 addresses. The following example shows use of alias names and the QtocCvtIfcID API.

Important: The examples in this chapter use new CL support that is available in i5/OS Version 5 Release 4 Modification 0 (V5R4M0).

1. Define a TCP/IP interface with an alias name. This step assumes that an Ethernet line description ETHLINE already exists.

```
ADDTCPIFC INTNETADR('10.1.1.1') LIND(ETHLINE1) SUBNETMASK('255.255.255.255')
ALIASNAME(ETHLINE1)
```

2. Create a source file member for a CL module. This step assumes that a library MYLIB and source file MYFILE have already been created.
3. Start an editing session:

```
STRSEU SRCFILE(MYLIB/MYFILE) SRCMBR(NAME2IPA) TYPE(CLLE)
```

Note: The file member type must be CLLE because you are creating an Integrated Language Environment (ILE) CL program.

4. Enter the source statements.

Example 22-1 CL source for converting from an alias name to an IP address

```
PGM        PARM(&IFCNAME &IFCADDR)
             DCL        VAR(&IFCNAME) TYPE(*CHAR) LEN(50)
             DCL        VAR(&IFCADDR) TYPE(*CHAR) LEN(15)
             DCL        VAR(&RCVR) TYPE(*CHAR) LEN(72)
             DCL        VAR(&RCVRLEN) TYPE(*INT) LEN(4) VALUE(72)
             DCL        VAR(&FORMAT) TYPE(*CHAR) LEN(8) +
                          VALUE('NCII0100')
             DCL        VAR(&REQCCSID) TYPE(*INT) LEN(4) VALUE(0)
             DCL        VAR(&ERRCODE) TYPE(*CHAR) LEN(8) +
                          VALUE(X'0000000000000000')
             DCL        VAR(&IPADDRPTR) TYPE(*PTR) ADDRESS(&RCVR 8)
             DCL        VAR(&IPADDR) TYPE(*CHAR) STG(*BASED) LEN(15) +
                          BASPTR(&IPADDRPTR)
             CALLPRC    PRC('QtocCvtIfcID') PARM((&RCVR) (&RCVRLEN) +
                          (&FORMAT) (&IFCNAME) (&REQCCSID) (&ERRCODE))
             CHGVAR     VAR(&IFCADDR) VALUE(&IPADDR)

ENDPGM
```

5. Leave the editing session and save your changes.
6. Create the CL module object:

```
CRTCLMOD MODULE(MYLIB/NAME2IPA) SRCFILE(MYLIB/MYFILE)
```

7. Create a second source file member for a CL module.
   a. Start an editing session:

```
STRSEU SRCFILE(MYLIB/MYFILE) SRCMBR(CVT2IPA) TYPE(CLLE)
```

   b. Enter the source statements.

Example 22-2 CL source for calling the NAME2IPA module

```
PGM        PARM(&NAME)
             DCL        VAR(&NAME) TYPE(*CHAR) LEN(25)
             DCL        VAR(&IPADDR) TYPE(*CHAR) LEN(15)
             CALLPRC    PRC(NAME2IPA) PARM((&NAME) (&IPADDR))
             SNDPGMMSG  MSG(&IPADDR)

ENDPGM
```

8. Create the CL module object:

```
CRTCLMOD MODULE(MYLIB/CVT2IPA) SRCFILE(MYLIB/MYFILE)
```

9. Create the program object:

```
CRTPGM PGM(MYLIB/CVT2IPA) MODULE(MYLIB/CVT2IPA MYLIB/NAME2IPA)
BNDSRVPGM(QSYS/QTOCNETSTS)
```

Note: This program binds to the service program QTOCNETSTS because it provides the QtocCvtIfcID API.

10. Run the program:

```
CALL MYLIB/CVT2IPA PARM('ETHLINE1')
```

In this example, the following message is sent to the job log:

```
10.1.1.1
```

22.2 Proxy ARP and the preferred interface list

Proxy ARP enables a physical interface to answer Address Resolution Protocol (ARP) requests on behalf of a virtual IP or virtual Ethernet address. The physical interface in this case is referred to as an agent interface because it answers APR requests on behalf of the virtual address. This enables the virtual address to be known to the network; otherwise, remote systems must have a route defined to the virtual IP address.

Virtual IP addresses can be used to provide both inbound and outbound load balancing, as well as fault tolerance in the event of an adapter failure. Fault tolerance capability is achieved through the specification of the same virtual IP address as the associated local interface for more than one physical interface. In releases prior to i5/OS V5R4, the operating system would select the agent based on either the highest-speed interface available (i5/OS V5R3) or the first interface activated (i5/OS V5R2).

Starting with i5/OS V5R4, you can manually select which adapters and IP addresses are to be the preferred interface for VIPA proxy ARP agent selection. You can select which interface to use by creating a preferred interface list for use in the event of an adapter failure. A preferred interface list is an ordered list of the interface addresses that will take over for the failed adapters. You can use either iSeries Navigator (via the interface properties) or the Change TCP/IP IPv4 Interface (QTOCC4IF) API to configure a preferred interface list. The preferred interface list is also configurable for both virtual Ethernet and virtual IP address interfaces.

22.3 Putting it all together

To put it all together:

1. Define several TCP/IP interfaces with an alias names. This step assumes that Ethernet line descriptions ETHLINE1 and ETHLINE2 already exist.

```
ADDTCPIFC INTNETADR('10.1.1.1') LIND(ETHLINE1) SUBNETMASK('255.255.255.255')
ALIASNAME(ETHLINE1)
ADDTCPIFC INTNETADR('10.1.1.2') LIND(ETHLINE2) SUBNETMASK('255.255.255.255')
ALIASNAME(ETHLINE2)
ADDTCPIFC INTNETADR('10.1.1.3') LIND(*VIRTUALIP)
SUBNETMASK('255.255.255.255')
ALIASNAME(MYVIPA)
```

2. Create a source file member for a CL module. This step assumes that a library MYLIB and source file MYFILE have already been created.

   a. Start an editing session:

      ```
      STRSEU SRCFILE(MYLIB/MYFILE) SRCMBR(CHGVIPA) TYPE(CLLE)
      ```

   b. Enter the source statements.

Example 22-3 CL source for setting proxy ARP and the preferred interface list

```
PGM
             /* Declare the parameters passed into this module.       */
             DCL        VAR(&VIPAIFC) TYPE(*CHAR) LEN(25) +
                          VALUE('MYVIPA                  ')
             DCL        VAR(&PRFIFC1) TYPE(*CHAR) LEN(25) +
                          VALUE('ETHLINE1                ')
             DCL        VAR(&PRFIFC2) TYPE(*CHAR) LEN(25) +
                          VALUE('ETHLINE2                ')
             /* Declare the variables for calling QTOCC4IF.           */
             DCL        VAR(&IFCINF) TYPE(*CHAR) LEN(196)
             DCL        VAR(&FORMAT) TYPE(*CHAR) LEN(8) +
                          VALUE('IFCH0100')
             DCL        VAR(&ERRCODE) TYPE(*CHAR) LEN(8) +
                          VALUE(X'0000000000000000')
             DCL        VAR(&IFCINFLEN) TYPE(*INT) STG(*DEFINED) +
                          LEN(4) DEFVAR(&IFCINF)
             DCL        VAR(&IFCINFIPA) TYPE(*CHAR) STG(*DEFINED) +
                          LEN(15) DEFVAR(&IFCINF 5)
             DCL        VAR(&RESERVED) TYPE(*CHAR) STG(*DEFINED) +
                          LEN(1) DEFVAR(&IFCINF 20)
             DCL        VAR(&PROXYARP) TYPE(*INT) STG(*DEFINED) +
                          LEN(4) DEFVAR(&IFCINF 21)
             DCL        VAR(&OFSPRFIFC) TYPE(*INT) STG(*DEFINED) +
                          LEN(4) DEFVAR(&IFCINF 25)
             DCL        VAR(&NUMPRFIFC) TYPE(*INT) STG(*DEFINED) +
                          LEN(4) DEFVAR(&IFCINF 29)
             DCL        VAR(&LENPRFIFC) TYPE(*INT) STG(*DEFINED) +
                          LEN(4) DEFVAR(&IFCINF 33)
             DCL        VAR(&PRFIFCPTR) TYPE(*PTR) ADDRESS(&IFCINF 0)
             DCL        VAR(&PRFIFC) TYPE(*CHAR) STG(*BASED) LEN(15) +
                          BASPTR(&PRFIFCPTR)
             DCL        VAR(&IFCNAME) TYPE(*CHAR) STG(*DEFINED) +
                          LEN(24) DEFVAR(&IFCINF 37)
             /* Initialize the interface information parameter fields  */
             /* for calling the QTOCC4IF API.                          */
             CHGVAR     VAR(&IFCINFLEN) VALUE(196)
             CHGVAR     VAR(&RESERVED) VALUE(X'00')
             CHGVAR     VAR(&PROXYARP) VALUE(1)
             CHGVAR     VAR(&OFSPRFIFC) VALUE(60)
             CHGVAR     VAR(&NUMPRFIFC) VALUE(2)
             CHGVAR     VAR(&LENPRFIFC) VALUE(16)
             CHGVAR     VAR(&IFCNAME) VALUE('*SAME                    ')
             /* Convert the virtual IP address name to an IP address.  */
             CALLPRC    PRC(NAME2IPA) PARM((&VIPAIFC) (&IFCINFIPA))
             /* Address the beginning of the preferred interface list. */
             CHGVAR     VAR(%OFS(&PRFIFCPTR)) VALUE(%OFS(&PRFIFCPTR) +
                          + &OFSPRFIFC)
```

```
            /* Convert the first proxy agent name to an IP address    */
            /* and store it in the preferred interface list.          */
            CALLPRC    PRC(NAME2IPA) PARM((&PRFIFC1) (&PRFIFC))
            /* Address the next element in the preferred interface    */
            /* list.                                                  */
            CHGVAR     VAR(%OFS(&PRFIFCPTR)) VALUE(%OFS(&PRFIFCPTR) +
                         + &LENPRFIFC)
            /* Convert the second proxy agent name to an IP address   */
            /* and store it in the preferred interface list.          */
            CALLPRC    PRC(NAME2IPA) PARM((&PRFIFC2) (&PRFIFC))
            /* Change the virtual IP interface to enable proxy ARP    */
            /* and to set the preferred interface list.               */
            CALL       PGM(QSYS/QTOCC4IF) PARM(&IFCINF &FORMAT +
                         &ERRCODE)
ENDPGM
```

3. Create the CL module object:

```
CRTCLMOD MODULE(MYLIB/CHGVIPA) SRCFILE(MYLIB/MYFILE)
```

4. Create the program object:

```
CRTPGM PGM(MYLIB/CHGVIPA) MODULE(MYLIB/CHGVIPA MYLIB/NAME2IPA)
BNDSRVPGM(QSYS/QTOCNETSTS)
```

Note: This step assumes that the NAME2IPA module from 22.1, “Using interface alias names” on page 670 has already been created.

5. Run the program:

```
CALL MYLIB/CHGVIPA
```

Using iSeries Navigator, the preferred interface list and proxy ARP setting for interface 10.1.1.3 can now be verified by viewing the Advanced tab of the interface properties as shown in Figure 22-1.

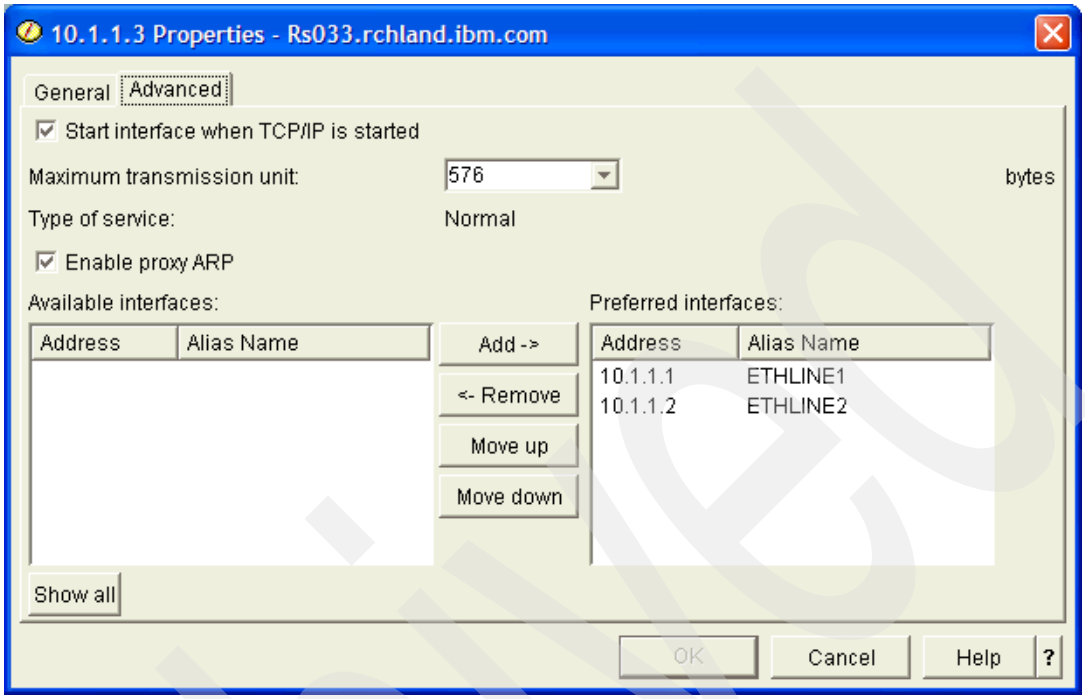


Figure 22-1 Interface properties showing preferred interface list for 10.1.1.3

22.4 References

Additional information is available in the iSeries Information Center at:

`http://publib.boulder.ibm.com/infocenter/iseries/v5r4/index.jsp`

Documentation for the Change TCP/IP IPv4 Interface (QTOCC4IF) and Convert interface ID (QtocCvtIfcID) APIs is available in the iSeries Information Center under the topic **Programming → APIs → APIs by category → Communications → TCP/IP Management APIs**.

Information about CL programming is also available in the iSeries Information Center under the topic **Programming → CL → CL Programming**.

Information about Proxy ARP and workload balancing is available in the iSeries Information Center under the topic **Networking → TCP/IP applications, protocols, and services → TCP/IP routing and workload balancing**.

23

Using exit programs

This chapter describes what you can do with the exit points that are defined for IBM-supplied TCP/IP applications.

An exit point is a defined interface that enables you to write a program that can change the behavior of i5/OS or IBM-supplied products in specific instances. The following i5/OS TCP/IP applications have exit points defined:

- Telnet
- File Transfer Protocol (FTP)
- Trivial File Transfer Protocol (TFTP) server
- Remote Execution (REXEC) server
- Dynamic Host Configuration Protocol (DHCP) server

The formal programming interface for each exit point is defined in the System i5 Information Center, and is not repeated here. Use that information to actually write any exit programs.

23.1 Basic exit program information

An exit point is a defined programming interface for writing a program called an *exit program*, which is called by IBM-supplied code in specific instances. The parameters passed to your exit program and the information that your exit program can return are defined by an *exit point* format. Each exit point has at least one associated exit point format.

Different exit points that perform similar functions may share an exit point format. This enables you to write a single exit program that you can use to process multiple exit points. For example, several of the TCP/IP applications have a “Request Validation” exit point, and all of these exit points share the same exit point format. As a result, you can write a single exit program that validates requests from any (or all) of these applications.

Some exit points have more than one exit point format. This usually happens when features are added to an application that requires a change to the parameters passed to an exit program. By defining a new exit point format, existing exit programs can still be used, but may not be able to utilize newly added features or functions. When you are designing an exit program for an exit point with more than one exit point format, make sure to look at all of the formats and choose the one that best meets your needs. (In most cases, the exit point format name with the highest numeric value is the most current and has the most features available.)

An exit program should always leave the state of the job unchanged. For example, if an exit program opens a file, it should close the file before returning. Likewise, if the exit program must change a job attribute to operate properly, it should save the value of that attribute before making the change and then restore it to the original (saved) value before ending. Remember that the exit program is called within the job where a specific exit point condition occurs, and the design of the program should not assume that calls made to it will occur in any specific order. Also note that some i5/OS TCP/IP applications “reuse” jobs for different sessions, so even the calls to your exit program within a single job may not be related. The time spent running the exit program delays processing in the TCP/IP application, so exit programs should be designed to be as efficient as possible.

After your exit program is written, you must add it to the exit point (or points) that are to call it and specify the exit point format you chose for your program. See an example of how to do this in the System i5 Information Center at:

`http://publib.boulder.ibm.com/infocenter/iseries/v5r4/topic/rzaiq/rzaiqinstep.htm`

If you do not want to write your own exit programs, you might investigate other software providers’ products. Several products are available from independent software vendors that use the TCP/IP application exit points, and one of these might meet your needs.

23.2 Request Validation exits

Several IBM-supplied TCP/IP applications implement a Request Validation exit point. This exit point enables you to increase the security of your system and gives you the opportunity to reject a specific request even if it would otherwise be allowed by normal i5/OS object security. (However, this exit point cannot be used to circumvent object security; if the user making the request does not have the required authority to access an object on the system, the Request Validation exit point *cannot* be used to make a request for that object “work.”)

As an example, XYZ Company has a call center for employees with questions about their pay and benefits. This center is staffed by people in the Human Resources department. They are given read access to the company’s master payroll file so that they can answer questions from employees about payroll issues. However, because the payroll file contains very

sensitive information, the manager of the HR department wants to prevent the call center employees from being able to send the file to another system. By writing an exit program for the Request Validation exit point, you could prevent these employees from using FTP to access the master payroll file.

The applications that implement the Request Validation exit point are:

- FTP Client

  Exit point name: QIBM_QTMF_CLIENT_REQ
- FTP Server

  Exit point name: QIBM_QTMF_SERVER_REQ
- REXEC Server

  Exit point name: QIBM_QTMX_SERVER_REQ
- TFTP Server

  Exit point name: QIBM_QTOD_SERVER_REQ

All of these exit points share a single exit point format: VLRQ0100.

23.2.1 Capabilities of a Request Validation exit program

An exit program added to one of the Request Validation exit points is called whenever that application is about to process a request made by a user. The information passed to the exit program is from the standpoint of the application calling it; for example, if the exit program is called by the FTP client application, any file or directory name is for the “local” system where the user is typing FTP subcommands (not the “remote” system where the FTP server is running).

A Request Validation exit program receives the following information when called:

- An identifier that defines which application is calling the exit program. This parameter (Application identifier) allows a single exit program to process requests from different applications.
- An identifier for the type of operation requested. This parameter (Operation identifier) tells the exit program the category of the request, such as initiating a new session, sending a file to another system, receiving a file from another system, or running a CL command).
- The user profile making the request. (For client applications, this is the user profile of the job where the client is running; for server applications, this is the user profile used to log on to the server.)
- The Internet Protocol (IP) address of the *remote* system (from the standpoint of the application calling the exit program). For client applications, this is the IP address of the server to which the client is connected; for server applications, this is the IP address of the system where the client is running. The IP address information is passed in two parameters: IP address and Length of IP address.
- Operation-specific information, which provides specific information about the request. For example, if the operation identifier specifies that a file is being sent to another system, the operation-specific information contains the full path name of the file to which is to be sent. This information is also passed in two parameters: Operation-specific information and Length of operation-specific information.

You design your exit program to make a determination about whether the request should be allowed. (However, as noted before, even a request allowed by your exit program still might fail due to i5/OS object security. For example, if the user attempts to run a CL command but

does not have the authority to do so, that attempt will fail even if your exit program *allows* it. A Request Validation exit program can only be used to further restrict an operation beyond that enforced by i5/OS object security.)

The Request Validation exit program must set an output parameter (Allow operation) that tells the TCP/IP application how the request should be handled. In the simple case, the exit program sets this parameter to either 0 (Reject operation) or 1 (Allow operation).

Two other values can be specified for the Allow operation parameter: 2 (Always allow operation) and -1 (Always reject operation). These tell the calling application that any operation with the same operation identifier requested in the same session should be treated as though the exit program allowed (2) or rejected (-1) the operation (without actually calling the exit program again). Using these values enables your exit program to prevent unnecessary overhead if the requested operation type will always result in the same response from your exit program.

Note: Setting an operation identifier to -1 (Always reject) can result in situations that initially might seem to be incorrect or confusing. For example, if an FTP server Request Validation exit program returns -1 (Always reject) in response to a CWD (change directory) FTP subcommand. the FTP server will reject any file transfer request for a file not in the “current directory” without even calling the exit program. Although this might seem to be incorrect behavior, it is not, because this operation is functionally equivalent to the user first performing a change directory subcommand followed by the request to transfer the file.

Your exit program can also perform additional functions with the information passed to it. For example, you might want to use a Request Validation exit program to keep a log of files transferred with FTP, along with the user and IP address information. Such a log might be useful for auditing, tracking intrusions into your system, and so on. Another example is that a Request Validation exit program (along with a Server Logon exit program) together can implement “anonymous” FTP; that is, the ability to make certain files on your system “publicly” available without a user needing a user profile on the FTP server system. (Anonymous FTP is covered in the FTP Security topic in the System i5 Information Center.)

23.3 Server Logon exits

The i5/OS FTP and REXEC servers implement the Server Logon exit point. The purpose of this exit point is to give you additional control over how users log on to the server. Using this exit point, you can specify the user profile for logging on a user, perform your own password and authentication processing, and change some of the initial characteristics of the session, in addition to just allowing or denying a specific logon.

The Server Logon exit points are implemented in the following TCP/IP Applications:

FTP Server Exit point name: QIBM_QTMF_SVR_LOGON

REXEC Server Exit point name: QIBM_QTMX_SVR_LOGON

23.3.1 Capabilities of a Server Logon exit program

An exit program added to one of the Server Logon exit points is called whenever that application is about to authenticate and log on a user. The Server Logon exit point has 3 defined exit point formats; the exact capabilities of your exit program depend on which exit point format you choose.

TCPL0100 format

The TCPL0100 format does *not* support the ability to return a password longer than 10 characters. If your system is running with the QPWDLVL system value set to 2 or 3 (that is, 128-character password support) and your exit program needs to return passwords to the server application, you cannot use this exit point format.

A Server Logon exit program for the TCPL0100 exit point format receives the following information when called:

- An identifier for which application is calling the exit program. This parameter (Application identifier) allows a single exit program to process requests from different applications.
- The user identifier supplied to the server, along with the length of this identifier.
- The authentication string (password) supplied to the server, along with the length of this authentication string.
- The Internet Protocol (IP) address (in dotted-decimal format) of the system where the client application is running, along with the length of this address string.

Your exit program uses this information to determine whether the logon should be allowed, and if so, how the logon should be performed. Note that the Server Logon exit program can be used to bypass the authentication that is normally performed by i5/OS during logon, so you should design your exit program carefully and test it thoroughly.

There are four output parameters that your exit program can return:

- A return code, which specifies whether the logon should be allowed at all, and if so, how the logon should be performed. Your exit program must always set a return value for this parameter. Depending on the value set for this parameter by your exit program, your exit program may also need to return values in one or more of these parameters.
- A user profile returned by your exit program. Depending on the value set by your exit program for the return code, the server will use either the user identifier passed by the client during logon processing, or a user profile name returned by your exit program in this parameter. Using this parameter, your exit program can allow someone to log on to the server who does not have a profile on your server; this capability is useful to implement anonymous FTP.
- A password string returned by your exit program. Depending on the value set by your exit program for the return code, the server will use one of the following options for supplying a password to i5/OS for authentication:
  - The authentication string received from the client application
  - The string returned in this parameter by your exit program
  - Not require a password at all (that is, your exit program can force a successful logon without requiring a password)

  Important: You should *never* store passwords in a program. This parameter is provided so that your exit program can perform algorithmic password processing (for example, secure hash functions or encryption/decryption).

- A library name returned by your program. Depending on the value set by your exit program for the return code, the server will set the current library to either the value specified by your exit program in this parameter or the current library field of the profile used to log on to the server.

TCPL0200 format

The TCPL0200 format is available only for the FTP server (it is not implemented for the REXEC server), does not support longer passwords, and has no advantages over the TCPL0300 format. It is no longer recommended for new development so is not covered here.

TCPL0300 format

The TCPL0300 exit point format uses three types of parameters:

Input	Parameters passed to your exit program
Output	Parameters returned by your exit program
Input/Output	Parameters passed to your exit program that you can modify to change how server logon options will be processed

Note: In the descriptions that follow, Input/Output parameters are marked in *italics.*

A Server Logon exit program for the TCPL0300 exit point format receives the following information when called:

- An identifier for which application is calling the exit program. This parameter (Application identifier) allows a single exit program to process requests from different applications.
- The user identifier supplied to the server, along with the length of this identifier.
- The authentication string (i.e., password) supplied to the server, along with the length of this authentication string and the CCSID (coded character set identifier) of the authentication string.
- The Internet Protocol (IP) address (in dotted-decimal format) of the system where the client application is running, along with the length of this address string.
- *The initial current library. When your exit program is called, this parameter is set to the special value *CURLIB, which means to use the current library specified in the user profile used to log on to the server. If you want a different library to be used as the initial current library, your exit program must set this parameter to the name of that library.*
- *Length of the initial home directory. When your exit program is called, this parameter is set to 0, meaning that the home directory is set to that specified in the user profile used to log on to the server. If you want a different directory to be used, your exit program must set the initial home directory parameter to the fully qualified path name of the directory, and set this parameter to the length of that fully qualified directory name.*
- *CCSID of initial home directory. When your exit program is called, this parameter is set to 0, meaning that any home directory string is returned in the CCSID of the job. If your exit program returns a home directory path name in a different CCSID, you must set this parameter to that CCSID.*
- *Application-specific information* and the length of the application-specific information. When the application identifier is 2 (REXEC server program) these parameters are not used and the length of application-specific information parameter is zero. When the application identifier is 1 (FTP server program), *the application-specific information parameter contains the following fields: (those in italics can be changed by your exit program):*
  - *Initial setting of name format*
  - *Initial current working directory (current library or home directory)*
  - *Initial file listing format (i5/OS or UNIX)*
  - Control connection security mechanism

- *Data connection encryption option*
- Control connection cipher suite
- *Data connection cipher suite*

Your exit program uses this information to determine whether the logon should be allowed, and if so, how the logon should be performed. Note that the Server Logon exit program can be used to bypass the authentication that is normally performed by i5/OS during logon, so you should design your exit program carefully and test it thoroughly.

There are six output parameters that your exit program can return (*in addition to the input/output parameters already described):*

- A return code (Allow logon) that specifies whether the logon should be allowed at all, and if so, how the logon should be performed. Your exit program must always set a return value for this parameter. Depending on the value set for this parameter by your exit program, your exit program may also need to return values in one or more of these parameters.
- A user profile returned by your exit program. Depending on the value set by your exit program for the Allow logon parameter, the server will use either the user identifier passed by the client during logon processing, or a user profile name returned by your exit program in this parameter. Using this parameter, your exit program can allow someone to log on to the server who does not have a profile on your server; this capability is useful to implement anonymous FTP.
- A password string returned by your exit program, along with parameters for the length and CCSID of this password. Depending on the value set by your exit program for the return code, the server will use one of the following options for supplying a password to i5/OS for authentication:
  - The authentication string received from the client application
  - The string returned in this parameter by your exit program
  - Not require a password at all (that is, your exit program can force a successful logon without requiring a password)

Important: You should *never* store passwords in a program. This parameter is provided so that your exit program can perform algorithmic password processing (for example, secure hash functions or encryption/decryption).

23.4 REXEC Server Command Processing Selection exit

By default, the i5/OS REXEC server processes commands passed to it as Control Language (CL) commands. The REXEC Server Command Processing Selection exit point (exit point name: QIBM_QTMX_SVR_SELECT) enables you to specify an alternate command processor to interpret and run the command (or commands) for a specific session.

23.4.1 REXEC Server Command Processing Selection exit program capabilities

An exit program added to the REXEC Server Command Processing Selection exit point is called whenever the REXEC server is about to process a command received from an REXEC client. The exit program receives the following information when called:

- The user profile making the request.

- The Internet Protocol (IP) address of the system where the REXEC client is running. The IP address information is passed in two parameters: Remote IP address and Length of Remote IP address.
- The command string to be run and the length of the command string.

  You design your exit program to make a determination about which command processor the command should be passed to and whether the REXEC server needs to perform EBCDIC - ASCII character conversion for the command and the returned data.

  The exit program must set an output parameter (Command processor identifier) that tells the REXEC server how the command string should be processed (as a CL command, as a Qshell command, or as a fully qualified file name specifying an executable shell script or program). If your exit program specifies that the command is to be treated as anything other than an i5/OS CL command, your program must also set the output parameter specifying whether character conversion is performed on the data.
- Your exit program can also perform additional functions with the information passed to it. For example, you might want to use an exit program for this exit point to keep a log of all commands that are processed by the REXEC server.

23.5 Telnet exits

The i5/OS Telnet server operates by using virtual devices, which are i5/OS objects that simulate physical 5250 architecture terminals. This architecture enables applications on i5/OS to use the same programming interfaces for Telnet-connected sessions as with physical terminals. However, there are some limitations related to the way that Telnet allocates and uses virtual devices. The Telnet server provides two exit points that enable you to write exit programs that provide more flexibility in setting up and managing virtual devices.

23.5.1 Telnet Device Initialization exit point

The purpose of the Telnet Device Initialization exit point is to give you more flexibility in how a virtual device for a new Telnet session is created, and the ability to automatically sign on a user to the interactive job attached to the virtual device. The exit point name and exit point format for the Telnet Device Initialization exit point are QIBM_QTG_DEVINIT and INIT0100 (respectively).

An exit program added to the Telnet Device Initialization exit point is called after the Telnet server has received a request for a new session from a Telnet client and determined the attributes and capabilities of that client. The parameters passed between the Telnet server and the exit program fall into several categories:

- Connection information and emulated terminal capabilities. The data in these parameters are determined by the connection attributes and the programmed capabilities of the Telnet client, so the exit program cannot change these characteristics. They are provided to enable your exit program to make decisions about how the Telnet session and the associated virtual device will operate, and are contained in two multi-field parameters (and an associated length parameter).
  - The Connection description information parameter contains fields that provide the exit program information about the IP addresses of the client and server, the type of workstation requested by the client, whether a valid password (or password equivalent) was provided by the client, whether the session is encrypted using the Transport Layer Security/Secure Sockets Layer (TLS/SSL) protocols (and if so, further information about whether the client provided a valid certificate to allow client authentication as part of the TLS/SSL session initialization).

  - The Environment options parameter enables your exit program to determine the specific features that the Telnet client supports. These features are determined through a negotiation process specified by the Telnet protocol. This buffer contains the Telnet-protocol defined options negotiated by the client exactly as sent; the length of this data is specified by the Length of environment options parameter. Your exit program can parse this buffer if you need to determine specific information about the capabilities of the Telnet client program.

- Device description information. This parameter is used to communicate the attributes about the virtual device to be created and is filled in by your exit program (if you do not want to use the Telnet defaults). Attributes you can specify include the virtual device name, keyboard identifier, code page, and character set for the virtual device.

  Note: You can choose a convention for your virtual device names and then use i5/OS workstation entries to route the jobs associated with specific virtual devices to specific subsystems. To accomplish this, choose a “root” name for all virtual devices that are to go to a specific subsystem, and create two workstation entries: one that assigns any virtual device with that root to the QINTER subsystem when the virtual device is initialized, and another to assign the virtual device to the desired target subsystem at signon. Then have your exit program assign a unique identifier for each virtual device to be routed to that subsystem and append it to the chosen root. When the user logs on to the Telnet session, the associated interactive job will be routed to the desired subsystem. By using different root values, your exit program can route Telnet sessions to different subsystems as desired.

- Automatic sign-on information. If you want to allow automatic sign-on for this session, your exit program must set a value of 1 for the Allow auto-signon output parameter. Your exit program sets values for fields in the User description information parameter to set the characteristics for the auto-signon operation; the user profile field is required to be set, and your exit program can optionally set the Current library, Initial menu, and Initial program to call fields. The Telnet server will process the signon as though the user had typed these values on the default i5/OS signon panel. (If your exit program does not set the Allow auto-signon parameter to 1, the contents of the User information parameter are completely ignored by the Telnet server.)
- The Allow connection parameter. Your exit program must return a value of 1 in this parameter for the Telnet session to be allowed; otherwise, the session will be rejected and the connection with the Telnet client will be closed.

23.5.2 Telnet Device Termination exit point

The purpose of the Telnet Device Termination exit point is to give you the ability to take some action at the end of a Telnet session. The exit point name and exit point format for the Telnet Device Termination exit point are QIBM_QTG_DEVTERM and TERM0100 (respectively).

An exit program added to the Telnet Device Termination exit point is called when the Telnet client ends the Telnet session. The Telnet server passes a single parameter to the exit program, which is the name of the virtual device that was associated with the Telnet session. There are no return parameters.

Your exit program can use this information to audit or log the end of the session.

23.6 DHCP exits

The Dynamic Host Configuration Protocol (DHCP) server enables you to centrally administer your network and automatically assign configuration information (including IP addresses) to devices on your network. This capability avoids the need to manually configure each device. The i5/OS DHCP server has three exit points defined to enable you to meet auditing, logging, and additional security requirements you might have.

23.6.1 DHCP Address Binding Notify exit

The purpose of the DHCP Address Binding Notify exit point is to give you the ability to take some action whenever the DHCP server successfully leases (assigns) an IP address to a specific device on your network. The exit point name and exit point format for the DHCP Address Binding Notify exit point are QIBM_QTOD_DHCP_ABND and DHCA0100 (respectively).

An exit program added to the DHCP Address Binding Notify exit point is called when the DHCP server assigns an IP address to a network device. The following parameters are passed to the exit program:

Request type Specifies whether the network device requesting an address is using the Bootstrap Protocol (BOOTP) or the DHCP protocol.

Client IP address The IP address assigned to the device by the DHCP server.

Client identifier The unique identifier of the client to which the IP address has been assigned. (This field usually contains the network interface hardware address of the client machine).

Lease duration Identifies the time period for which the client can use the IP address (in seconds), or a special value of all bits set to 1 if the lease duration is infinite.

Response packet The actual BOOTP or DHCP packet that completed the address assignment as transmitted to the requesting client machine.

The Client IP address, Client identifier, and Response packet parameters each have an associated length parameter that contains the length of the data passed to your exit program for that parameter.

There are no return parameters. An exit program for this exit point usually performs auditing and logging of assigned network addresses for problem determination, accounting, or network security monitoring.

23.6.2 DHCP Address Release Notify exit

The purpose of the DHCP Address Release Notify exit point is to give you the ability to take some action whenever the DHCP server releases (de-allocates) an IP address (which was assigned to a specific device on your network). The exit point name and exit point format for the DHCP Address Binding Notify exit point are QIBM_QTOD_DHCP_ARLS and DHCR0100 (respectively).

An exit program added to the DHCP Address Release Notify exit point is called when the DHCP server releases the IP address that has been assigned to a network device. The following parameters are passed to the exit program:

- Reason for release: specifies why the IP address was released:
  - The client machine requested release of the address.

- The lease duration expired without the client requesting a renewal.
- A DHCP administrator explicitly released the address.

- Client IP address: the IP address assigned to the device by the DHCP server that is now being released.
- Client identifier: the unique identifier of the client to which the IP address was assigned. (This field usually contains the network interface hardware address of the client machine.)

The Client IP address and Client identifier parameters each have an associated length parameter that contains the length of the data passed to your exit program for that parameter.

There are no return parameters. An exit program for this exit point usually performs auditing and logging of network addresses for problem determination, accounting, or network security monitoring.

23.6.3 DHCP Request Packet Validation exit

The purpose of the DHCP Request Packet Validation exit point is to give you the ability to further restrict which packets will be processed by the DHCP server beyond the validation tests that the server already performs. The exit point name and exit point format for the DHCP Address Binding Notify exit point are QIBM_QTOD_DHCP_REQ and DHCV0100 (respectively).

An exit program added to the DHCP Request Packet Validation exit point is called each time the DHCP server receives an incoming BOOTP or DHCP request packet but before any processing of the packet has taken place. The following parameters are passed to the exit program:

Request packet An exact copy of the request packet received by the DHCP server exactly as it was received from the network.

Length of request packet The length (in bytes) of the received packet.

Your exit program should inspect the request packet and perform any tests or validation you require of the information contained in the packet. When your exit program has made a determination as to whether the packet should be processed, it sets the Allow operation output parameter to:

- 0 if the packet should be rejected; the DHCP server will not perform any additional processing of the packet.
- 1 if the request packet should be allowed to continue. The DHCP server will continue processing the packet as normal. (If the packet fails the DHCP server's validation tests, the packet will still be rejected.)

24

Problem determination: where to start when things do not work

This chapter provides basic problem determination steps starting with Program Temporary Fixes (PTFs) and going as far as collecting traces. One extremely important tip in performing problem determination is taking one step at a time. A methodical process is required so that you actually know what fixed your problem.

24.1 Preface: what you need to know before you start

The most common source of network connectivity problems is an incorrect or incomplete TCP/IP configuration. The complexity of the network adds another range of potential failure points, such as gateways or routers, DNS servers, firewalls, and ISPs. The next most common cause for failure is missing Program Temporary Fixes (PTFs). There are now group PTFs for TCP/IP - SF99313 for V5R2, SF99314 for V5R3, and SF99315 for V5R4. We recommend that you periodically download and install the appropriate Group PTF so that you stay current with available TCP/IP fixes. In addition, the Support Center maintains a publicly available software knowledge base document that provides information about the TCP/IP Group PTFs for the supported releases and links to documents covering older releases.

The KB document number is 22960332 and it is available from the Support Center home page at:

`http://www.ibm.com/servers/eserver/support/iseries/index.html`

Select **Technical databases** → **Software Knowledge Base**, then simply issue a search on keyword 22960332

The Support Center also maintains a current list of PTFs ("Recommended for specific products or functions"), which includes categories such as: TCP/IP, Telnet, SMTP, Netserver, Management Central, Operations Console, and much more. The topics vary by release. The address is:

`http://www-912.ibm.com/s_dir/slkbase.nsf/recommendedfixes`

A basic understanding of your network topology is essential for a complete analysis of connectivity problems:

- What is your IP address and subnet mask? Do you have multiple IP interfaces (multiple local addresses) assigned on the source system?
- What is the IP address you are attempting to connect to and its subnet mask?
- Is the target IP address in the same subnet as your IP address? For example, say the IP address of your system is 109.5.92.28 with a subnet mask of 255.255.255.224 and the target system has an address of 109.5.92.3. A subnet calculator will show you that this subnet has maximum of 30 host addresses in the range of 109.5.92.1 - 109.5.92.30. Also note, the first and last address in each subnet range are reserved for multicast and broadcast.

  Tip: There are many freeware subnet calculators available on the Internet to help you make this determination. Simply go to the search engine of your choice and searched on the keywords `subnet calculator`.

- If your addresses are not in the same subnet, do you have a gateway or router configured?

  Note: To cross subnets you must have a route or gateway configured. When configuring routes on the System i the term *next hop* is used to specify the address of the router, on the local subnet, that will be used to route the packet.

- Are you using a DNS server or multiple DNS servers? Does it reside on the System i or is it an external server? (You may need to make sure that the DNS server is active.)
- Is there a firewall or a router between your system and the target system? This could come into play if certain ports are being filtered out, so some ports may not be allowed to

pass through, or specific IP addresses may be blocked, depending on the specific TCP/IP application. The System i also has the ability to set up filter rules.

24.2 Basic TCP/IP connectivity verification

The most common problem is the inability for one system to establish a TCP/IP connection to a target system. Generally, the root cause behind this can be isolated very quickly by following a few basic procedures. This flowchart (Table 24-1) depicts a logical sequence used to isolate the cause of a connection failure and the suggested steps to resolve the problem or redirect the focus to the suspected network component. As the complexity of your network grows, the potential points of failure increase.

If you already have the ability to PING the remote system and you are experiencing a problem with the behavior of a specific TCP/IP application, skip over Table 24-1 and proceed to 24.3, “Application specific problem scenarios” on page 694. This section highlights some of the common problem symptoms and likely solutions. Section 24.4, “Tools of the trade” on page 696, explains some of the most common tools used for problem determination, as well as how and when they could be most useful.

Table 24-1 Problem determination for connectivity

Flow chart step number	Next step or process
1) Check whether TCP/IP is active.	ACTION: Issue the command NETSTAT ► If a menu is displayed, then TCP/IP is active. Go to step 4. ► If message TCP2670 is received, TCP/IP must be started (step 2).
2) Start TCP/IP.	ACTION: Issue the command STRTCP. Specify *YES for the STRIFC and STRSVR parameters. If you are using IPv6, specify *YES for the STRIP6 parameter. If you are using PPP, specify *YES for the STRPTPPRF parameter. ► This should start all of the TCP/IP applications that have the autostart configuration value set to *YES and all TCP/IP interfaces that are configured with autostart *YES. If this was successful, go to step 4. ► Message TCP1A04 will be displayed if TCP is already active. Attempt to use NETSTAT again as described in step 1. If this is successful, go to step 4. ► If TCP/IP failed to start, go to step 3.
3) STRTCP failed.	ACTION: ► Locate joblog for job QTCPIP - `WRKSPLF QTCP`, look for this JOB name with matching date and time. ► Look for messages sent to system operator - `DSPMSG QSYSOPR`. ► Look for messages in History log - `DSPLOG` - Use F4 to select time/date window. ► Check for ‘recommended PTF’ (see chapter preface). ► `CHKPRDOPT PRDID(5722TC1)` - will verify any IBM-supplied components that are missing or damaged. If this fails, reinstall TCP/IP (product is 5722TC1 for V5 or 5769TC1 for V4) then reinstall cume PTF package, then repeat verification test.
4) Can you the PING remote host by name?	► Yes, go to step 7. ► ACTION: If this fails, attempt to PING by remote IP address (step 5).

Flow chart step number	Next step or process
5)PING by remote IP address.	► Yes, then we need to examine name resolution process. Go to step 6. ► No, then we need to determine whether there is a TCP Configuration error. ► ACTION: – Issue the command `NETSTAT *IFC` and verify that you have an active interface, to support your *DIRECT or *DEFAULT route. (Use the command `NETSTAT *RTE` or option 2 on the menu to check route status.) – Is the target IP address on a direct attached subnet? (Is it in the same subnet as the System i? See 24.1, “Preface: what you need to know before you start” on page 690, for subnet calculator discussion.) • If the interface is not active, issue the command `STRTCPICF ‘`*x.x.x.x*`’` or CFGTCP option 1, F11 to see the status, then option 9 to start (where *x.x.x.x* is the locally assigned IP address for the System i). • If the interface is active and the associated route is available, check the target system to ensure that TCP/IP is active there. • Ping the local IP address or loopback. If this works we know that the local system’s TCP stack is operational. • Check whether there are any filter rules that would prevent ICMP traffic, or are blocking your local IP address, or are blocking replies from the target IP address. These filters could be set on a router or a firewall; check with your network administrator. (Filter rules could also be set in the local System i IP Packet Filters on an interface level. To check for any local packet filter rules, go to iSeries Navigator. Select your System i → **Network** → **IP Policies** → **Packet Rules**.)

Flow chart step number	Next step or process
6) Check host name resolution process.	► NOTE: Try the command `NSLOOKUP` *`hostname`*. (See 24.4.1, “Commonly used commands and utilities” on page 696 for more details.) This is a quick test of whether your system can contact a DNS server. This tool does not use the local host table. If unable to contact the default *CFG for the server, the message: *Cannot find server name for address x.x.x.x: No response from server* will be issued. Then this utility will determine whether you have a DNS server configured on the local System i and will return the message: Server: *ASJ4L1.RCHLAND.IBM.COM Address: 0.0.0.0*, if there is no DNS configured on this System i. ► ACTION: – Issue command CFGTCP then option 12. The parameter HOSTSCHPTY will determine whether the process searches the local host table first and then any remote DNS server. This is also where you will configure the DNS server addresses. • NOTE: If *LOCAL is searched first and no matching entry is found, the process will then check for remote DNS services. If a matching entry is found in the local host table, it will not check the remote DNS. If *REMOTE is searched first, and no reply is received (times out), the process will be repeated several times for each DNS server and will then default to search the local host table. – HOSTSCHPTY *LOCAL: Check the local host table for a matching entry in CFGTCP option 10: • If there is a matching entry, confirm that the IP address is correct. • If there is no matching entry, add a host table entry (`ADDTCPHTE`) or use option 1 on the host table display. You can define up to four names for each host IP address. Typically this might be a short name and a fully qualified host.domain name. Command format is: `ADDTCPHTE INTNETADR('1.1.1.1') HOSTNAME((RMTHOST) (RMTHOST.MYDOMAIN.COM)) TEXT('description of remote system')`. – HOSTSCHPTY *REMOTE: Verify that IP address for the remote DNS has been configured - CFGTCP option 12: • Is the DNS address correct? If not, get the correct address and update this field. • Can you ping the address of the DNS? If yes, is the DNS running on the local System i? Whether DNS is on the local System i or a remote system, verify that the target host has been added to the DNS and that the DNS has been started. In iSeries Navigator, select **Network** → **Servers** → **DNS**.
7) If PING by name was successful.	► This means that you have proven that there is network connectivity, the correct interface and routes are available, and the host name resolution process worked. ► ACTION: – Verify that the correct server job is active on the target system. For example, if attempting a Telnet session to a target System i, verify server is active on target. iSeries Navigator: **Network** → **Servers** → **TCP/IP**: • Issue the STRTCPSVR *TELNET command. • Verify that the server job is in LISTEN state, issue the command `NETSTAT *CNN`. You can sort this display by local PORT (F14); Telnet will be port 23. • Check job logs - see table at end of this section for server job names.

24.3 Application specific problem scenarios

The following sections contain common troubleshooting techniques for the DHCP and PPP applications on the System i.

24.3.1 DHCP problem scenarios

Table 24-2 describes several symptoms that may be present when DHCP is not functioning properly. The symptoms are listed in the left column, and the possible causes, fixes, and documentation that may be needed to analyze the problem are listed to the right of the symptom description.

Table 24-2 Malfunctioning DHCP issues

Problem symptom	Possible causes, corrective action, documentation collection
DHCP was working, added a new IP interface using the same physical resource, now it no longer serves IP addresses to the DHCP clients. DHCP Server Not Giving Correct Subnet.	This a common problem, you can still do what was working previously, with a slight DHCP configuration change.For example, say your original IP subnet was 111.111.11.0,with a mask of 255.255.255.0 and you have added a new System i interface with an address of 222.222.22.0 ► Add another subnet in the DHCP server's configuration. The new subnet should have a subnet address of 222.222.22.0 (assuming a mask of 255.255.255.0). Add a single address, such as 222.222.22.1, to the address pool. Then EXCLUDE the 222.222.22.1 address from the pool to prevent the DHCP server from handing that address out. Create a SUBNET GROUP in the DHCP configuration. Add the original 111.111.11.0 subnet and the 222.222.22.0 subnet. This informs the server that these two LOGICAL subnets are on the same PHYSICAL subnet. ► Another option is to add a new DHCP pool of addresses in the new subnet. ► Incorrect subnet: This could occur if the Subnet Mask option is not included in the DHCP subnet configuration. Add the correct subnet mask for the clients to use this subnet mask. The default is 255.0.0.0.
DHCP Relay Agent Not Forwarding DHCP Requests.	See whether the System i has two or more interfaces defined for the same subnet. Does the DHCP relay configuration have all of those interfaces defined? If more than one interface has been defined, the appropriate interfaces must be defined in the DHCP relay configuration.
DHCP server is not starting. Port 67 is in use.	BOOTP is most likely binding to port 67 before DHCP can. CL command CFGTCP, Option 20. Work with the BOOTP server and ensure that it is set AUTOSTART *NO. Ensure that DHCP is set AUTOSTART *YES**.** Note: If the `STRTCPSVR *ALL` command is run, BOOTP will be started before DHCP and will cause the problem explained above. To resolve the problem, run the `ENDTCPSVR *BOOTP` command followed by the `STRTCPSVR *DHCP` command.

Problem symptom	Possible causes, corrective action, documentation collection
The DHCP server job QTODDHCPS will end immediately with message.	If the message in the DHCP server job log is TCP5736. This message is posted because the user set up the IP address and subnet mask indicating that the DHCP server will be handling a very large number of addresses. For example: If the IP address is 99.0.0.0 with a subnet mask of 255.255.0.0, this has no range limitations and will result in message TCP5736. To correct this configuration issue, do one of the following: ► Set the subnet mask to limit the maximum number of IP addresses the server will be administering to the number that is really needed. ► Set the range to limit the number of IP addresses that the server will be administering to the number that is really needed.

24.3.2 PPP problem

Table 24-3 describes several symptoms that may be present when PPP is not functioning properly. The symptoms are listed in the left column, and the possible causes, fixes, and documentation that may be needed to analyze the problem are listed to the right of the symptom description.

Table 24-3 Symptoms of improperly functioning PPP

Problem symptom	Possible causes, corrective action, documentation collection
PPP dial connection profile will not go active No carrier	This is most commonly a configuration error for the necessary modem initialization. Use the WRKTCPPTP command and check the job log and CLOGS for this profile. The CLOG log file is a record of the dial-in or dial-out session, including all AT commands and phone numbers dialed. To view the CLOG: WRKTCPPTP, then option 14 next to the desired profile. The CLOG will be one of the spooled files for the job, with a name such as CLOG123456 or CL12345678 (depending upon your release). Use option 5 to view it. Additionally starting in V5R4, a matching Message Log (MLOG) with a name such as ML12345678 should be viewed for error messages associated with the failure. Prior to V5R4, these messages would be found in the QPJOBLOG. Verify that the modem initialization string is correct for your modem.
PPP dial connection profile will not go active No dial tone	This is most commonly a configuration error for the phone number or a phone line cabling problem. Use the WRKTCPPTP command and check the job log and CLOGS for this profile. The CLOG log file is a record of the dial-in or dial-out session, including all AT commands and phone numbers dialed. To view the CLOG, WRKTCPPTP, then option 14 next to the desired profile. The CLOG will be one of the spooled files for the job, with a name such as CLOG123456 or CL12345678 (depending upon your release). Use option 5 to view it. Additionally starting in V5R4, a matching MLOG with a name such as ML12345678 should be viewed for error messages associated with the failure. Prior to V5R4, these messages would be found in the QPJOBLOG. Verify that the phone number being dialed is the correct format. For example, verify that you are using a prefix of 9 if required in order to get an outside line and verify that you are using commas (which insert a 1-second delay) when needed with your PBX. Also verify that the phone line is correctly plugged in and that with an analog phone attached you can get a dial tone.

24.4 Tools of the trade

There any many commands, logs, and utilities that can be used to help further isolate problems. This section outlines a wide selection of these, describes what information they can provide, and offers suggestions about when and why they may be most useful.

24.4.1 Commonly used commands and utilities

The following section contains commands that are used to display TCP/IP configurations and debug connectivity problems. These functions are part of the 5722SS1 and 5722TC1 licensed program products.

Configure TCP/IP (CFGTCP)

From the green screen, this command provides the basic menu interface for the most common setup requirements. Note that all of these functions and more can be accomplished using iSeries Navigator as well. Figure 24-1 shows the menu options.

```
CFGTCP                         Configure TCP/IP
                                                          System:   AS20
Select one of the following:

     1. Work with TCP/IP interfaces
     2. Work with TCP/IP routes
     3. Change TCP/IP attributes
     4. Work with TCP/IP port restrictions
     5. Work with TCP/IP remote system information

    10. Work with TCP/IP host table entries
    11. Merge TCP/IP host table
    12. Change TCP/IP domain information

    20. Configure TCP/IP applications
    21. Configure related tables
    22. Configure point-to-point TCP/IP
```

Figure 24-1 Sample of green screen interface for configuring and managing TCP/IP

The most commonly used options are 1, 2, 3, 10, 12, 20, 21, and 22:

- Option 1: This option enables you to check the status of your interfaces, stop and start them, create new interfaces, and change the properties of an existing interface.
- Option 2: Gives you the opportunity to view, change, and create your manually defined routing entries. Most configurations require at least a *DEFAULT route entry.
- Option 3: This is where you would view or change the TCP/IP global attributes such as the Keep Alive timer, Retransmission retry limits, whether IP forwarding is allowed, and much more.
- Option 10: The host table on the System i is used to map IP addresses to specific host names. You can define the local system's name and address in this table to make applications that require a reverse lookup respond more quickly. (This is required for some applications such as the HTTP Admin server at earlier releases.) This table is also used by the System i for name resolution.
- Option 12: This is where you configure the fully qualified host.domain name for this specific System i. Other parameters include the host name search priority (which can be set to *LOCAL or *REMOTE), and the IP address (up to three) for your DNS servers.
- Option 21: This option presents another menu with three elements: 1. Work with service table entries, 2. Work with protocol table entries, 3. Work with network table entries. A common requirement, if using iSeries Navigator option *Configuration and Services*, is that you enter the necessary service table entry, which is: `as-sts (service name) 3000 (port) TCP (the protocol)`. Note that this is case sensitive.
- Option 22: Here you can delete existing profiles and create, change, and monitor PPP connection profiles, modem options, PPP line descriptions, and stop and start profiles.

Work with TCP/IP Network Status (NETSTAT)

The NETSTAT command and WRKTCPSTS (native i5/OS command format) are identical. These commands provide a menu with six options (Figure 24-2) that enable you to view interface information, route information (including routes that are automatically or dynamically generated by the system), and the status of connections. This function is also available through iSeries Navigator: **Network** → **TCP/IP Configuration** → **IPv4** or **IPv6**.

```
Work with TCP/IP Network Status
                                                          System: MYSYSTEM
Select one of the following:

     1. Work with TCP/IP interface status
     2. Display TCP/IP route information
     3. Work with TCP/IP connection status
     4. Work with IPv6 interface status
     5. Display IPv6 route information
     6. Work with IPv6 connection status


Selection or command
===>

F3=Exit   F4=Prompt   F9=Retrieve   F12=Cancel
```

Figure 24-2 NETSTAT options

NETSTAT is a very good way to perform quick checks of the TCP/IP configuration and status:

- *Interfaces* (options 1 and 4) shows the IP addresses you have configured for this system, the network address, the associated line description, MTU size, and the current status.
- *Routes* (options 2 and 5) shows any routes that were manually configured, the *DIRECT routes that are automatically configured for you, any dynamically generated routes, whether the route is available, and the route MTU size.
- *Connections* (options 3 and 6) reports which server ports are active by name, and F14 shows the active ports by number. This option also gives the state of each port (for example, if the port is ready and waiting for connections it will have a LISTEN status). This display includes the IP address for any remote connections to this system, remote IP address, and the remote port used.

```
Work with TCP/IP Connection Status
                                                          System:
Type options, press Enter.
  3=Enable debug   4=End   5=Display details   6=Disable debug
  8=Display jobs

     Remote           Remote     Local
Opt  Address          Port       Port       Idle Time  State
     *                *          as-vrtp >  134:00:09  Listen
     *                *          hprip-ctl  134:02:37  *UDP
     *                *          hprip-n >  134:02:37  *UDP
     *                *          hprip-h >  134:02:37  *UDP
     *                *          hprip-med  134:02:37  *UDP
     *                *          hprip-low  134:02:37  *UDP
     9.10.108.167     1332       telnet     000:02:03  Established
     9.10.109.45      1639       telnet     000:00:00  Established
     127.0.0.1        as-svrmap  55903      133:59:22  Close-wait




                                                                  Bottom
F3=Exit   F5=Refresh   F9=Command line   F11=Display byte counts   F12=Cancel
F20=Work with IPv6 connections   F22=Display entire field   F24=More keys
```

Figure 24-3 NETSTAT, sample of Connections display

PING

This is the most commonly used command to assist with verification of connectivity. PING uses the ICMP protocol to broadcast an ECHO request to the network. The command options enable you to broadcast by host name or by host IP address. You can specify IP version (IPv4 or IPv6), the size of the test packet to be sent, the number of packets to send, wait time for a response from the target host, and the ability to select a local interface IP address as the source address. This function is supported by all platforms generally in the lower layers of the TCP stack, so the individual TCP/IP applications do not have to be active.

In addition to checking for connectivity, PING can be useful if there is a suspected network data integrity problem. The recipient of the ECHO request must respond with a mirror image of the data packet it receives. The larger the packet size, the higher the probability that if data bytes are being lost, PING may help in detecting this.

This is also available using iSeries Navigator: **Network** → **TCP/IP Configuration** → **Utilities** → **Ping**.

TRACEROUTE

The Trace TCP/IP Route (TRCTCPRTE in native i5/OS command format) traces the route of IP packets to a user-specified destination system. The route can involve many different systems along the way. The route is traced by sending packets (called probes; actually these are ECHO requests) to the destination system. Each probe contains an upper limit (called Time To Live or TTL) on the number of hop systems the probe can pass through.

A route is traced by successively incrementing the TTL of the probe packets by one hop. The trace ends when either a probe response is received from the destination system or the probe Time To Live value equals the maximum allowed.

This command can be very useful to help isolate communications performance problems by checking to see how many hops are between the source and the target system and whether any delays are being incurred between hops.

A useful feature of TRACEROUTE is the ability to use UDP probes instead of ICMP ECHO requests. This can be helpful if firewalls are set to block ICMP requests but allow UDP. Note that the default is to use ICMP.

Another useful feature of TRACEROUTE is the ability to control the Allow Fragmentation bit within the IP header of the probes. By setting Allow fragmentation to *NO and increasing the probes packet lengths, you can determine whether fragmentation is occurring between your System i and the target destination. This can help you determine the optimal MTU setting to use on your TCP/IP routes or Interfaces.

The command supports both host name or IP address for the destination system. A simpler form of this utility is also available using iSeries Navigator but selecting **Network → TCP/IP Configuration → Utilities → Trace Route**.

NSLOOKUP

NSLOOKUP or Start DNS Query (STRDNSQRY in native i5/OS format), starts the Name Server Lookup tool. This utility can be used to verify that a DNS server is responding correctly before you configure your system to use it. You can also retrieve DNS information about hosts, domains, and DNS servers. There are two parameters for this command: HOSTNAME for specifying a host name or IP address, and DMNNAMSVR, the DNS server that it will use. You can point it to any valid DNS server that can be reached from this System i. The default is *CFG, which attempts to set one of the following as its default DNS server for the session:

- The DNS server your system is configured to use, based on CFGTCP option 12
- The DNS server that is running on your local system

A simpler form of this utility is also available using iSeries Navigator: **Network → TCP/IP Configuration → Utilities → Look Up Host**

Work with Problems (WRKPRB)

The Work with Problems display shows descriptions of both system-detected and user-perceived system problems. You can work with a problem, work with alerts, or work with the text you added to the problem record. This utility is part of the autonomic system design and works in conjunction Electronic Service Agent™ to automatically or manually report system-detected errors. Use this command to check for any errors that the system has logged. These can be reported directly to IBM. IBM will search for any known problems and if a matching symptom is found, the fix can be automatically downloaded to your system.

Product Activity Log (PAL)

PAL™ and the service action log are accessed through System Service Tools (STRSST). These are helpful in determining whether there are any hardware or LIC-level errors being logged. Specifically relating to communications, the PAL can help isolate a hardware interface problem (LAN IOA, cable, connection to a switch, and so on) or a modem problem. To access the PAL: **STRSST → 1 - Start a service tool → 1 - Product activity log 1 - Analyze log → 5 - Communications** (you can select a date and time window to review) → **1 - Display Analysis**. From here look for any entries associated with the IO resource in question. A reference code that can be used to further isolate the failure may be displayed.

The Service Action Log is more commonly used for reporting hardware errors. This log covers all hardware including communications hardware. It is accessed via **STRSST → 1 - Start a service tool → 7 - Hardware service manager → 6 - Work with service action log**.

24.4.2 Advanced tracing utilities

These utilities include the communication trace, a very flexible tool that is usable by most of the technical community. This section also includes utilities that would not be directly useful to the general user community but are essential tools used by support service personnel.

Currently, advanced tracing utilities are available only through a 5250 green screen interface.

Communications trace status commands

STRCMNTRC, ENDCMNTRC, PRTCMNTRC, DLTCMNTRC, CHKCMNTRC, and DMPCMNTRC are the CL commands for starting, stopping, printing, deleting, and checking the status of a communications trace. System Service Tools has an equivalent capability: In STRSST, select **1 - Start a Service Tool → 3 - Work with communications trace**.

The communications trace is probably the most frequently used tool for assisting in the analysis of communications problems. The trace collection occurs at the IOP level so that it is as close to the network interface as possible. This makes it very similar to a sniffer trace. However, the System i can capture only data that is addressed to it, or any broadcast data. It can be used to capture all communications protocols, such as TCP/IP, PPP, SNA, SDLC, Async, and Bisync. In this chapter, we focus on just the TCP/IP area.

All of the communication trace commands have the same first two parameters. New in V5R2 is the DMPCMNTRC command, which provides the ability to move the trace data from the internal buffer to an IFS stream file. PRTCMNTRC lists an optional parameter of IFS Path as the second parameter. The first parameter is the Configuration Object. This is the name of the line description associated with the TCP/IP interface. The next parameter is TYPE, for this topic this will always be *LIN.

STRCMNTRC

There are numerous options on the STRCMNTRC command that enable you to specify the buffer size reserved to hold the trace data, whether the trace should stop if this buffer fills up or if it should wrap, or whether to capture all or just the first portion of the actual data in each packet. You can activate certain filters, such as remote IP address or IP protocol number. After the trace is stopped you can format and print the data as many times or in as many ways as you need. There are many different situations, so there is never a single answer for how to collect the data. However, we recommend not using any filters while capturing the data to enable you to adjust the filters at print time and know that the trace records that you need to see are not filtered out during the capture.

For example:

```
===> STRCMNTRC CFGOBJ(MYLINENAME) CFGTYPE(*LIN) MAXSTG(*MAX) DTADIR(*BOTH)
TRCFULL(*WRAP) USRDTA(*MAX) TEXT('sample cmd format')
```

In this example, notice that MAXSTG is set at *MAX, which at V5R2 and later attempts to allocate 1 GB of storage. USRDTA at *MAX is specified to see the entire user data in each packet.

CHKCMNTRC

The Check Communications Trace command returns the communications trace status. The status is returned through message CPF39A9, and status can be STARTING, ACTIVE, WAITING, ENDING, ENDED, or ERROR.

ENDCMNTRC

The End Communications Trace command ends the trace running on the specified line, network interface, or network server description.

DMPCMNTRC

The Dump Communications Trace command copies the unformatted trace data to a user-specified stream file. The data in the stream file can be formatted at a later time on either the current system or a different system, by using the Print Communications Trace (PRTCMNTRC) command and specifying the FROMSTMF parameter.

The advantage of using DMPCMNTRC to move the raw data to an IFS stream file is that you can then delete the trace (DLTCMNTRC) and start another collection. The raw data, now in the stream file, can be printed and formatted multiple times concurrently while collecting new trace data.

PRTCMNTRC

The Print Communications Trace command transfers the communications trace data to a spooled file or can be routed to an output file. The spool file has the name QPCSMPRT and will be routed to the outq associated with the user who issued the command. This command can also be used to format communications trace data that was previously dumped to a stream file using the Dump Communications Trace (DMPCMNTRC) command.

This command has a variety of optional parameters. Because this utility is designed to support all communications protocols, we focus only on the parameters that are most pertinent to TCP/IP.

This utility attempts to translate the user data from hex to printable characters, and this is displayed to the far right side of the printed output, commonly referred to as the *eye-catcher*. Most TCP/IP applications are ASCII-based and the default for the CODE parameter is *CALC. This assumes ASCII for any TCP/IP-formatted data. However, Client Access, Telnet 5250 is EBCDIC, so keep this in mind if you want to see the eye-catcher for a Client Access Telnet session.

The most important parameter to set for a TCP/IP communication trace is the FMTTCP option. Set this to *YES. Depending on the trace data you wish to filter by, set the TCPIPADR parameter to the IP address of the remote session partner. You can narrow this down further by selecting the TCP/IP port of either the server or the client. To see the ICMP traffic on an

Ethernet line, be sure to code FMTETH(*YES). Finally, to remove the clutter of broadcast messages, such as ARP requests, select FMTBCD(*NO). For example, tracing a Client Access Telnet session to the System i, where the CA client has an IP address of 9.9.9.9:

```
PRTCMNTRC CFGOBJ(line-name) CFGTYPE(*LIN) CODE(EBCDIC) FMTTCP(*YES) TCPIPADR('9.9.9.9')
SLTPORT(23) FMTETH(*YES) FMTBCD(*NO)
```

Note: For Windows-based clients, to see the TCP/IP configuration, go to a DOS window and enter the command IPCONFIG /ALL.

DLTCMNTRC

The Delete Communications Trace command deletes the communications trace and releases the trace buffer back to the system. Although you can configure more than one communications trace on a resource, only one trace can be running at a time. Prior to starting a new trace, the previous trace must be ended and deleted.

SAMPLE communication traces

The comm trace tool also breaks down the important fields in UDP, TCP, and ICMP headers. You may need to refer to the IBM Redbooks publication *TCP/IP Tutorial and Technical Overview*, GG24-3376, or the RFCs for guidance in breaking these frames down further.

Sample of Address Resolution Protocol (ARP)

Address Resolution Protocol is not IP, but a special protocol used to match IP addresses with LAN hardware addresses. The requester sends a packet containing the IP address of the remote system. If the remote hears the ARP request, it replies with a frame containing its own hardware address and route information (if any bridges were crossed). The Communications trace tool also formats the ARP protocol.

In order to see ARPs in your traces, you must format the trace for Broadcast Data.

```
76   S       33  1136.8                    FFFFFFFFFFFF  C00027023AF3   LLC     UI                        OFF     AA   A
                 Routing Information . . . . . . . . . :  0270
                 Frame Type :  ARP          Src Addr: 9.5.83.179       Dest Addr: 9.5.83.146       Operation: REQUEST
                 ARP Header :  000608000604000140002 7023AF3090553B3000000000000009055392
77   R       33  1136.8                      400027023AF3  8004AC339B5D   LLC     UI                        OFF     AA   A
                 Routing Information . . . . . . . . . :  02F0
                 Frame Type :  ARP          Src Addr: 9.5.83.146       Dest Addr: 9.5.83.179       Operation: RESPONSE
                 ARP Header :  000608000604000200 4AC339B5D090553924000 27023AF3090553B3
```

Figure 24-4 Sample ARP request and response

DNS request and response

Applications wishing to access another system by name will issue a DNS request. The DNS server responds with the IP address of the remote system or, for SMTP, an MX record listing. This is an example of a reverse DNS query, where the application knows an IP address but wants to retrieve the host and domain name of the remote system. Note that this is a UDP frame, not TCP.

```
DNS REQUEST
    334   S       7b698.2                        40000FDAB083  40007A01F67E   LLC     UI                          OFF     AA   AA
                      Frame Type :  IP            TOS: NORMAL            Length:    68  Protocol: UDP           Datagram ID: D0DC
                                    Src Addr: 9.5.83.31        Dest Addr: 9.5.100.76       Fragment Flags: MAY  ,LAST
                      SNAP Header:  0000000800
                      IP Header  :  45000044D0DC00004011E0570905531F0905644C
                      IP Options :  NONE
                      UDP  . . . :  Src Port:  1347,Unassigned    Dest Port:    53,DOMAIN        Message Length:        48
                      UDP Header :  0543003500304B1E
                      Data . . . :  000101000001000000000000232360237330135013907494E2D41444452   *.............26.73.8.2.IN-ADDR*
                                    044152504100000C0001                                          *.ARPA.....                    *


DNS RESPONSE
    335   R      204  1698.2                           40007A01F67E  40000FCAA083   LLC     UI                          OFF     AA   AA
                      Frame Type :  IP            TOS: NORMAL            Length:   199  Protocol: UDP           Datagram ID: 2ABC
                                    Src Addr: 9.5.100.76       Dest Addr: 9.5.83.31        Fragment Flags: MAY  ,LAST
                      SNAP Header:  0000000800
                      IP Header  :  450000C72ABC00001D11A8F50905644C0905531F
                      IP Options :  NONE
                      UDP  . . . :  Src Port:    53,DOMAIN        Dest Port:  1347,Unassigned    Message Length:       179
                      UDP Header :  0035054300B3B67B
                      Data . . . :  000185800001000100020002023236023733013501390749 4E2D41444452   *..**.........26.73.8.2.IN-ADDR*
                                    044152504100000C0001C00C000C000100015180001907703833393035 34   *.ARPA.....*.......Q*...P839054*
                                    077263686C616E640369626D03636F6D000135013907494E2D4144445204   *.RCHLAND.IBM.COM..8.2.IN-ADDR.*
                                    4152504100000200010001518000 0A077263686E616D65C03CC04D000200   *ARPA.......Q*...RCHNAME*<*M...*
                                    0100015180000C097263686E616D652D62C03CC0690001000100015180 00   *...Q*...RCHNAME-B*<*I......Q*.*
                                    040905644CC07F000100010001518000040905644B                     *...DL*.......Q*....DK         *
```

Figure 24-5 Sample DNS request and response

For more information about Domain Name Services request and response, consult *TCP/IP Tutorial and Technical Overview*, GG24-3376, for frame details.

TCP Connection Establishment

SYN, SYN ACK, and ACK: *The Three-Way Handshake*

This is the normal way that TCP connections are established. The client sends a SYN to request the opening of a port. The server responds with a SYN ACK, and the client then sends an ACK (see the 'code bits:'). Note that the initial Sequence and ACK numbers are set at this time. If the server refuses the request, it will usually send an RST to close the connection immediately. (Some will send a FIN instead.)

```
275   R       49  4366.0                        40007A01F67E  40000FCAA083   LLC     UI                          OFF     AA   AA
                  Frame Type :  IP            TOS: NORMAL           Length:    44  Protocol: TCP           Datagram ID: 49B4
                                Src Addr: 9.5.73.26        Dest Addr: 9.5.83.31        Fragment Flags: MAY  ,LAST
                  SNAP Header:  0000000800
                  IP Header  :  4500002C49B400003E0684D50905491A0905531F
                  IP Options :  NONE
                  TCP  . . . :  Src Port:  1114,Unassigned    Dest Port:    80,Unassigned
                                SEQ Number:   492232705 ('1D56E001'X)  ACK Number:          0 ('00000000'X)
                                Code Bits: SYN                  Window:  4096  TCP Option: MSS= 1460
                  TCP Header :  045A00501D56E00100000000600210000D7E10000020405B4
  276   S       49  4366.0                          40000FDAB083  40007A01F67E   LLC     UI                          OFF     AA   AA
                  Frame Type :  IP            TOS: NORMAL           Length:    44  Protocol: TCP           Datagram ID: 3454
                                Src Addr: 9.5.83.31        Dest Addr: 9.5.73.26        Fragment Flags: MAY  ,LAST
                  SNAP Header:  0000000800
                  IP Header  :  4500002C345400004006983509055 31F0905491A
                  IP Options :  NONE
                  TCP  . . . :  Src Port:    80,Unassigned    Dest Port:  1114,Unassigned
                                SEQ Number:  1263304112 ('4B4C7DB0'X)  ACK Number:  492232706 ('1D56E002'X)
                                Code Bits: SYN ACK              Window:  8192  TCP Option: MSS=  536
                  TCP Header :  0050045A4B4C7DB01D56E00260122000027000000204 0218
  277   R       45  4366.0                          40007A01F67E  40000FCAA083   LLC     UI                          OFF     AA   AA
                  Frame Type :  IP            TOS: NORMAL           Length:    40  Protocol: TCP           Datagram ID: 49B5
                                Src Addr: 9.5.73.26        Dest Addr: 9.5.83.31        Fragment Flags: MAY  ,LAST
                  SNAP Header:  0000000800
                  IP Header  :  4500002849B500003E0684D80905491A0905531F
                  IP Options :  NONE
                  TCP  . . . :  Src Port:  1114,Unassigned    Dest Port:    80,Unassigned
                                SEQ Number:   492232706 ('1D56E002'X)  ACK Number: 1263304113 ('4B4C7DB1'X)
                                Code Bits: ACK                  Window:  4096  TCP Option: NONE
                  TCP Header :  045A00501D56E0024B4C7DB150101000269 10000
```

Figure 24-6 Sample of normal TCP three-way handshake

TRCTCPAPP

The Trace TCP/IP Application (TRCTCPAPP) command is used by service personnel when trace information must be captured for one of the following TCP/IP applications: FTP, SMTP server, SMTP client, Post Office Protocol (POP), TELNET/VTAPI, host servers, Distributed Data Management (DDM), Virtual Private Network (VPN), Layer Two Tunneling Protocol (L2TP), certificate services, PPP, QoS, simple Network Time Protocol (NTP), directory services, HTTP server powered by Apache, or packet rules.

This trace is essentially a log of the internal logic and event handling within the application being trace. Built within these IBM-supplied products are internal trace points that the developers can use to understand how the processing events are being handled. Typically, this is not a trace that the average user would employ; however, most of these traces provide comment text with each trace point that can be fairly legible even for the inexperienced.

Sample TRCTCPAPP: tracing SMTP client, host name resolution.
This example was generated using the CL command SNDDST to generate outbound Internet mail. Both the SMTP server and client traces were started (TRCTCPAPP *SMTPSVR and TRCTCPAPP *SMTPCLT). Three files were generated when the trace was ended, and all three files have the same name: QTMSSTRC. In this example, if you follow the text, you see the statement `Cannot resolve IP address with LHT`. This indicates that the local host table was searched unsuccessfully. By following the progression, you can see the preparations to send a query to the DNS servers. The comment `Cannot resolve IP address with LHT` indicates that the local host table was searched first, then configured DNS servers.

```
090103 212554 1 qtmsmxdc.h   630 At least 1 recipient requires resolution
090103 212554 1 qtmsmxdc.h   642 build_MX_unique() called
090103 212554 1 qtmsmxdc.h   665 Processing unresolved recipient:1
090103 212554 1 qtmsmxdc.h   665 Address resolution required for recipient:1
090103 212554 1 qtmsmxdc.h   642 handle_route() called
.
.
090103 212554 1 qtmsmxdc.h   642 Searching LHT for domain name us.ibm.com
090103 212554 1 qtmspace.C   490 free_space() will destroy QTMSGPS15 from lib QTEMP
090103 212554 1 qtmsmxdc.h   642 qtmsrlht() called
090103 212554 1 qtmsclnt.C  9273 get_trim_str: to_str length 10,us.ibm.com
090103 212554 1 qtmsmxdc.h   642 name us.ibm.com not found
090103 212554 1 qtmsclnt.C  9273 get_trim_str: to_str length 10,us.ibm.com
090103 212554 1 qtmsmxdc.h   630 Cannot resolve IP address with LHT
090103 212554 1 qtmsmxdc.h   679 process_from_lht returned, rc: 1
090103 212554 1 qtmsmxdc.h   642 qtmsbmxq() called
090103 212554 1 qtmsclnt.C  9273 get_trim_str: to_str length 10,us.ibm.com
090103 212554 1 qtmsmxdc.h   658 MX-Type query built for domain us.ibm.com.
090103 212554 1 qtmsbmxs.C  2687 Now Entering GetDNSes() @A9A
090103 212555 1 qtmsbmxs.C  2705 DNS 0 =     99.99.99.100
090103 212555 1 qtmsbmxs.C  2705 DNS 1 =     99.99.99.200
090103 212555 1 qtmsbmxs.C  2705 DNS 2 = 99.99.99.75
090103 212555 1 qtmsbmxs.C  2754 Now Exiting GetDNSes() fRC = 1. @A9A
090103 212555 1 qtmsmxdc.h   642 Using DNS Server: 99.99.99.100
090103 212555 1 qtmsmxdc.h   642 dns1.rchland.ibm.com
```

Figure 24-7 Sample TRCTCPAPP of SMTP host name resolution process

STRTRC
The Start Trace command starts traces of original program model (OPM) programs, Integrated Language Environment (ILE) procedures, and Java™ (compiled and JIT). Tracing can be done for multiple jobs using this command. Any number of trace sessions can be started, but active trace session identifiers must be unique across the system. This command can trace call-return flow, data returned by trace points defined in the operating system, component trace information, or all three.

The trace session continues until ended with the End Trace (ENDTRC) command. A trace session can be ended from the same job or a different job.

This has recently become the most frequently requested trace when the problem analysis process has progressed to the development level. The trace format is very similar to a TRCJOB; however, this command enables multiple jobs to be traced within a single process. This command was introduced at V4R5 and significantly enhanced at V5R1. By default, the output is place in a library of your choice, or the spool file name is QPSRVTRCJ. Sample command:

```
STRTRC SSNID(*GEN) JOB((QTCP/QTSMTP*))
```

This example lets the system generate a session ID, which is used later for ending the trace and tracking the data. It will trace all jobs that begin with the characters QTSMTP*. Message CPC3921 should be logged showing the system generated session ID.

```
21:49:09.919734 0000003E *UNMR RETURN   QDMCLOSE QSYS       x------ x0003EF     7     .000134    0    0    0    0    0
21:49:09.919379 0000003E        RETURN   QC2UTIL1  QSYS       x0003BB x0000FC    11     .000008    0    0    0    0    0
21:49:09.919779 0000003E        RETURN   QTMSCLCP  QTCP        000753  000954     5     .000020    0    0    0    0    0
                        MODULE   QTMSSRVS
                PROC    process_from_lht__FP9MX_work_s
                EXIT    PROGRAM  QC2UTIL1  QSYS               000954
                        MODULE   QC2DATM   QBUILDTC1
                PROC    _C_leapadj
```

Figure 24-8 Sample STRTRC output

TRCINT

The Trace Internal command is the command interface to the Trace Licensed Internal Code service tool and is used for problem analysis. Prior to V4R5, these trace utilities were only available through the SST (Start Service Tool). Specific types of traces are started and stopped by using this command. This tool normally would be used only at the direction of service personnel and typically requires an IBM developer to interpret the trace data. It has dozens of options that can be used to trace TCP data, UDP, ICMP, and dozens of other filters. The output created by the trace is placed in a trace table. The records from the trace table can be written to a spool file, tape, or optical media.

Sample format tracing with just TCP data on a PPP connection with no other filters set:

```
TRCINT SET(*ON) TRCTYPE(*TCPIP) TCPDTA(*TCP () () *N *N PPPLINE *PPP)
```

TRCCNN

The Trace Connection command allows the tracing of encrypted data flowing over IP and SSL connections. Specific types of traces are started and stopped by using this command. This Trace Internal (TRCINT) command collects the trace records and generates an intermediate spooled file named QPCSMPRT. The QPCSMPRT spooled file data is used to generate a spooled file named QSYSPRT. The user data for the QSYSPRT file is TRCCNN. You can also use TRCCNN with a QPCSMPRT spooled file generated by using TRCINT directly. TRCCNN can extract and format the IP and SSL connection–related trace records. This enables you to use TRCINT to collect many types of trace records and then use TRCCNN to format the subset of trace records related to IP or SSL connections. A sample command format is:

```
TRCCNN SET(*ON) TRCTYPE(*SSL) TCPDTA(*TCP (23) () *N '99.99.99.99')
*TRCTYPE *SSL: Trace SSL (Secure Sockets Layer) connection data
```

Table 24-4 TCP/IP server jobs - excepts from Knowledge Base document 18931401

Job name	Job description	Subsystem
QTCPIP	TCP/IP Interface daemon	QSYSWRK
QTCPMONTR	TCP Event monitor	QSYSWRK
QTVTELNET	Telnet server	QSYSWRK
QTVDEVICE	Telnet device manager	QSYSWRK
QTFTPnnnnn	FTP Server	QSYSWRK
QTODDHCPR	BootP DHCP Relay Agent	QSYSWRK
QTODDHCPS	QTODDHCPS	QSYSWRK
QTOQSRVR	QOS server	QSYSWRK

Job name	Job description	Subsystem
QTBOOTP	BootP Server	QSYSWRK
QTRTDnnnnn	RouteD Server	QSYSWRK
QTOBNDS	Domain Name Server	QSYSWRK
QTOBXFER	DNS (Secondary) Zone Transfer	QSYSWRK
QTOBXMIT	DNS (Primary) Zone Transfer	QSYSWRK
QTPOPnnnnn	POP Server	QSYSWRK
QTSMTPSRVR	SMTP Server	QSYSWRK
QTMSSRCD	SMTP Server (PJ in V4R4 and later)	QSYSWRK
QTSMTPCLNT	SMTP Client	QSYSWRK
QTMSCLCP	SMTP Client (PJ in V4R4 and later)	QSYSWRK
QTSMTPBRCL	SMTP Bridge Client	QSYSWRK
QTSMTPBRSR	SMTP Bridge Server	QSYSWRK
QTPPPCTL	PPP Control Job	QSYSWRK
QTPPDIALnn	Point-to-Point Session Dial	QSYSWRK
QTPPANSnnn	Point-to-Point Session Answer	QSYSWRK
QTPPPL2TP	L2TP Server (V4R4 and later)	QUSRWRK
QTPPPL2SSN	L2TP Session (V4R4 and later)	QUSRWRK

24.5 Security tips and comments

It is important to understand that several of the advanced tracing utilities will capture the data that is flowing between systems. The communication trace is a prime example. Because the data can often be confidential in nature, be aware of the need to maintain tight control over access to the commands and utilities. When using the internal virtual Ethernet LAN between partitions, this data is completely secure from any PC-based LAN tracing tool; however, the standard i5/OS communication trace can still be used and the data, if unencrypted, will still flow in the clear and can be captured with this utility.

Although encrypted data is fairly secure and certainly unreadable as normal text, the TRCINT and TRCCNN utilities have the ability to capture this data before encryption on the transmitting side, or after decryption on the receiving side.

The default FTP welcome panel from the System i includes the fully qualified host name and IP address by default. For security reasons it may be appropriate to change this default.

To change the text of this 220 message, do the following:

1. Enter `WRKMSGF MSGF(QTCP/QTCPMSGF)`.
2. Choose option **12, Work with message descriptions**.
3. Locate message ID TCP120D.
4. Choose option **12, change this message**. Edit the first-level message text. The default is:

   ```
   '220- &1 at &2.'
   ```

Here, variable &1 is the subsystem QTCP and &2 is the host name.

5. Type your text in place of this, and save it. This will take effect at the next FTP signon.

24.6 For more information

One good resource is the System i Support Center. This has a wealth of information such as Problem Solving, and from here you can search a number of different Technical databases, which include APARs, Preventive Service Planning, PTF Cover Letters, and the Knowledge Base, just to name a few:

http://www.ibm.com/servers/eserver/support/iseries/index.html

Another good resource is the iSeries Information Center. This site includes a great deal of information pertinent to problem analysis, links to online manuals, and much more:

http://publib.boulder.ibm.com/iseries/

Choose an appropriate language and version.

Two categories in the left-side navigation bar apply directly to this topic.

The first is within the main topic of **Networking** → **Network Communications** → **Get started with communications** → **Troubleshoot.**

The second category is more generic: **Troubleshooting**.

Part 4

Appendixes

The following topics are found in the appendixes:

- Appendix A, “Additional material” on page 713, is where we describe the step-by-step instructions for the setup of the scenario as described this Redbooks publication.
- Appendix B, “IPv6 reference information” on page 715, gives you more details for the IPv6 implementation on the iSeries.

A

Additional material

This Redbooks publication refers to additional material that can be downloaded from the Internet as described below.

Locating the Web material

The Web material associated with this Redbooks publication is available in softcopy on the Internet from the IBM Redbooks publications Web server. Point your Web browser to:

ftp://www.redbooks.ibm.com/redbooks/SG246718

Alternatively, you can go to the IBM Redbooks publications Web site at:

ibm.com/redbooks

Select **Additional materials** and open the directory that corresponds with the Redbooks publication form number, SG246718.

Using the Web material

The additional Web material that accompanies this Redbooks publication includes the following files.

A Web application for testing features of the HTTP Server powered by Apache

Download these files if you wish to have a ready-made Web application for testing various features of the V5R2 HTTP Server (powered by Apache). Scenario 18.2, “QoS: Inbound admissions policy: limiting connection rate based on HTTP URI” on page 626, uses an HTTP server to test QoS functions.

Tip: The library and directory source included in the files below originally came from the Redbooks publication *IBM HTTP Server (powered by Apache): An Integrated Solution for IBM eServer iSeries Servers*, SG24-6716. For more setup and configuration details, refer to this book.

File name	*Description*
ReadMe-01.txt	Contains directions on how to handle the files after you download them from the Internet.
tcp52dmast.zip	This is a zipped directory of /tcp52dmast and all of its subdirectories. Here, all of the ITSOco Web site and configuration files can be found.
tcp52lmast.savf	This is an i5/OS Save File (*SAVF) object that contains library TCP52LMAST and other System i specific objects to support some examples in this IBM Redbooks publication. It was saved with a target release of V5R2.

Support for an application that writes all interactive jobs and their corresponding IP addresses to a file

If you want to download and make modifications to an application that writes all interactive jobs and their corresponding IP addresses to a file, use these files:

File name	*Description*
ReadmeIPTool.txt	Step-by-step instructions for the setup of an application that writes in a file all interactive jobs and the corresponding IP addresses.
iptool.zip	This contains the save file with the interactive job-to-IP address application.

B

IPv6 reference information

This appendix provides reference information for the IPv6 support on the V5R4 release of i5/OS. IBM is implementing IPv6 in i5/OS over several software releases. Although IBM has yet to update many of the system applications that run on i5/OS to use IPv6, the IPv6 support that is currently available with the V5R4 release is more than adequate for developing and testing most IPv6 applications.

This appendix contains a table that compares IPv4 concepts and services with the V5R4 implementation of IPv6 on i5/OS. This appendix also provides instructions for using the communications trace tool, because the procedure is different than an IPv4 trace.

Comparison: IPv4 to IPv6

The following table provides information about various IP concepts, objects, and applications in the context of IPv4 and IPv6 as implemented on the V5R4 release of i5/OS.

Table 24-5 IPv4 to IPv6 comparison

	IPv4	IPv6
Address	32 bits long (4 bytes). Address is composed of a network and a host portion that depend on address class. Various address classes are defined: A, B, C, D, or E depending on initial few bits. The total number of IPv4 addresses is 4 294 967 296. The text form of the IPv4 address is nnn.nnn.nnn.nnn, where 0<=nnn<=255, and each n is a decimal digit. Leading zeros may be omitted. Maximum number of print characters is 15, not counting a mask.	128 bits long (16 bytes). Basic architecture is 64 bits for the network number and 64 bits for the host number. Often, the host portion of an IPv6 address (or part of it) will be a MAC address or other interface identifier. Addresses are assigned to interfaces, not nodes. Depending on the subnet prefix, IPv6 has a more complicated architecture than IPv4. The number of IPv6 addresses is 10^{28} (79 228 162 514 264 337 593 543 950 336) times larger than the number of IPv4 addresses. The text form of the IPv6 address is xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx, where each x is a hexadecimal digit representing 4 bits. Leading zeros may be omitted. The double colon (::) may be used once in the text form of an address to designate any number of 0 bits. For example, ::ffff:10.120.78.40 is an IPv6 IPv4-mapped address. See RFC4291 for details.
Address mask	Used to designate network from host portion.	Not used (see “Address prefix” on page 716)
Address prefix	Sometimes used to designate network from host portion. Sometimes written as /nn suffix on presentation form of address.	Used to designate the subnet prefix of an address. Written as /nnn (up to 3 decimal digits, 0 <= nnn <= 128) suffix after the print form. An example is fe80::982:2a5c/10, where the first 10 bits comprise the address prefix.
Loopback address	An interface with address of 127.*.*.* (typically 127.0.0.1) that can only be used by a node to send packets to itself.	The concept is the same as in IPv4, and the IPv6 loopback address is 0000:0000:0000:0000:0000:0000:0000:0001 or ::1 (shortened version).
Unspecified address	Apparently, not defined, as such. Socket programming uses 0.0.0.0 as INADDR_ANY.	Defined as ::/128 (128 0 bits). It is used as the source IP in some neighbor discovery packets and various other contexts, such as sockets. Socket programming uses ::/128 as in6addr_any.

	IPv4	IPv6
Address types	Unicast, multicast, and broadcast.	Unicast, multicast, and anycast. Anycast is new for IPv6: a packet is sent to the nearest node supporting a given anycast address. Multicast is more important and fundamental to IPv6 than to IPv4, because IPv6 neighbor discovery is based on multicast. Multicasts begin with 0xff to distinguish them from unicast. Anycast addresses are drawn from the unicast address space and are not distinguished by any particular prefix. Anycast addresses are not supported in V5R4.
address lifetime	Generally, not an applicable concept, except for addresses assigned using DHCP.	IPv6 addresses have two lifetimes: preferred and valid, with the preferred lifetime always <= valid. After the *preferred* lifetime expires, the address is not to be used as a source IP address. After the *valid* lifetime expires, the address is not used (recognized) as a valid destination IP address for incoming packets. Some IPv6 addresses have, by definition, infinite preferred and valid lifetimes; for example, link-local (see Table 24-5 and “address scope” on page 717).
address scope	For unicast addresses, the concept does not apply. There are designated private address ranges and loopback. Outside of that, addresses are assumed to be global.	In IPv6, address scope is part of the architecture. Unicast addresses have two defined scopes, link-local and global. Multicast addresses have 14 scopes. Default address selection for both source and destination takes scope into account. A *scope zone* is an instance of a scope in a particular network. As a consequence, IPv6 addresses sometimes have to be entered or associated with a zone ID. The syntax is %zid where zid is a number (usually small) or a name. The zone ID is written after the address and before the prefix. For example, 2ba::1:2:14e:9a9b:c%3/48.

	IPv4	IPv6
public and private addresses	All IPv4 addresses are public, except for three address ranges that have been designated as private by IETF RFC1918: 10.*.*.* (10/8), 172.16.0.0 through 172.31.255.255 (172.16/12), and 192.168.*.* (192.168/16). Private address domains are commonly used within organizations. Private addresses cannot be routed across the Internet.	IPv6 has an analogous concept, but with important differences. Addresses are public or temporary, previously termed anonymous. See RFC3041. Unlike IPv4 private addresses, temporary addresses can be globally routed. The motivation is also different; IPv6 temporary addresses are meant to shield the identity of a client when it initiates communication (a privacy concern). Temporary addresses have a limited lifetime and often do not contain an interface identifier that is a link (MAC) address. They are generally indistinguishable from public addresses. IPv6 has the notion of limited address scope using its architected scope designations (see Table 24-5 and “address scope” on page 717).
address allocation	Originally, addresses were allocated by network class. As address space is depleted, smaller allocations using Classless Inter-Domain Routing (CIDR) are made. Allocation has not been balanced among institutions and nations.	Allocation is in the earliest stages. The Internet Engineering Task Force (IETF) and Internet Architecture Board (IAB) have recommended that essentially every organization, home, or entity be allocated a /48 subnet prefix length (see RFC3177). This would leave 16 bits for the organization to do subnetting. The address space is large enough to give every person in the world their own /48 subnet prefix length.
interface	The conceptual or logical entity used by TCP/IP to send and receive packets; always closely associated with an IPv4 address, if not named with an IPv4 address. Sometimes referred to as a *logical interface*. Can be started and stopped independently of each other and independently of TCP/IP using STRTCPIFC and ENDTCPIFC commands and using iSeries Navigator.	Same concept as IPv4. Can be started and stopped independently of each other and independently of TCP/IP using iSeries Navigator only. There are no CL commands to start, stop, or configure IPv6 interfaces in V5R4.
LAN connection	Used by an IP interface to get to the physical network. Many types exist, for example, PPP, token-ring, and Ethernet. Sometimes referred to as the physical interface, link, or line.	IPv6 has the same concept. Only Ethernet lines are supported in the V5R4 release of i5/OS.

	IPv4	IPv6
Address Resolution Protocol (ARP)	Address Resolution Protocol is used by IPv4 to find a physical address, such as the MAC or link address, associated with an IPv4 address.	IPv6 embeds these functions within IP itself as part of the algorithms for stateless address autoconfiguration and neighbor discovery using Internet Control Message Protocol version 6 (ICMPv6). There is no such thing as ARP6.
IP header	Variable length of 20-60 bytes, depending on IP options present.	Fixed length of 40 bytes. There are no IP header options. Generally, the IPv6 header is simpler than the IPv4 header.
IP header Type of Service (TOS) byte	Used by QoS and Differentiated Services to designate a traffic class.	Designates the IPv6 traffic class, similarly to IPv4. Uses different codes. The V5R4 release of i5/OS does not support IPv6 QoS.
IP header protocol byte	The protocol code of the transport layer or packet payload; for example, ICMP.	The type of header immediately following the IPv6 header. Uses the same values as the IPv4 protocol field, but the architectural effect is to allow a currently defined range of next headers, and is easily extended. The next header will be a transport header, an extension header, or ICMPv6.
source address selection	An application may designate a source IP (typically, using the sockets bind() API). If it binds to INADDR_ANY, a source IP is chosen based on the route.	As with IPv4, an application may designate a source IPv6 address using the bind() API. Similar to IPv4, it can let the system choose an IPv6 source address by using in6addr_any. But as IPv6 lines have many IPv6 addresses, the internal method of choosing a source IP is different.
IP header options	Various options that may accompany an IP header (before any transport header).	The IPv6 header has no options. Instead, IPv6 adds additional (optional) extension headers. The extension headers are AH and ESP (unchanged from IPv4), hop-by-hop, routing, fragment, and destination. The V5R4 release of i5/OS supports the fragmentation header. Support for additional extension headers will be added in future releases.
Maximum Transmission Unit (MTU)	Maximum transmission unit of a link is the maximum number of bytes that a particular link type, such as Ethernet or modem, supports. For IPv4, 576 is the typical minimum.	IPv6 has an architected lower bound on MTU of 1280 bytes. That is, IPv6 will not fragment packets below this limit. To send IPv6 over a link with less than 1280 MTU, the link-layer must transparently fragment and defragment the IPv6 packets.

	IPv4	IPv6
Fragments	When a packet is too big for the next link over which it is to travel, it can be fragmented by the sender (host or router).	For IPv6, fragmentation can only occur at the source node, and reassembly is only done at the destination node.
Internet Control Message Protocol (ICMP)	ICMP is used by IPv4 to communicate network information.	Used similarly for IPv6; however, Internet Control Message Protocol version 6 (ICMPv6) provides some new attributes. Basic error types remain, such as destination unreachable, echo request, and reply. New types and codes are added to support neighbor discovery and related functions.
Internet Group Management Protocol (IGMP)	IGMP is used by IPv4 routers to find hosts that want traffic for a particular multicast group, and used by IPv4 hosts to inform IPv4 routers of existing multicast group listeners (on the host).	Replaced by MLD (multicast listener discovery) protocol for IPv6. Does essentially what IGMP does for IPv4, but uses ICMPv6 by adding a few MLD-specific ICMPv6 type values.
Transport layers	TCP, UDP, and RAW.	Same three transports exist and are functionally unchanged for IPv6.
Ports	TCP and UDP have separate port spaces, each identified by port numbers in the range 1-65535.	For IPv6, ports work the same as IPv4. Because IPv6 is a new address family, there are now four separate port spaces. For example, there are two TCP port 80 spaces to which an application can bind, one in AF_INET and one in AF_INET6.
Port restrictions	iSeries Navigator enables you to configure selected port numbers or port number ranges for TCP or UDP so that they are only available for a specific profile.	Configured port restrictions apply to both IPv4 and IPv6 in the V5R4 release of i5/OS.

	IPv4	IPv6
Sockets API	The socket APIs define the industry standard interface for an application to send/receive data over TCP/IP.	The AF_INET6 address family was added to i5/OS in V5R2. The AF_INET6 address family allows applications to use IPv6. These enhancements have been designed so that existing IPv4 applications are completely unaffected by IPv6 and IPv6 API changes. Applications that want to support concurrent IPv4 and IPv6 traffic, or IPv6-only traffic, are easily accommodated using IPv4-mapped IPv6 addresses of the form ::ffff:a.b.c.d, where a.b.c.d is the IPv4 address of the client. The new APIs also include support for converting IPv6 addresses from text to binary and from binary to text. See 3.3.2, “Sockets enhancements” on page 67, for more information.
Communications trace	A tool that is used to collect a detailed trace of TCP/IP packets that enter and leave i5/OS.	Communications trace was enhanced in V5R2 to support IPv6.
Route	Logically, a mapping of a set of IP addresses (may contain only 1) to a physical interface and a single next-hop IP address. IP packets whose destination address is defined as part of the set are forwarded to the next hop using the line. IPv4 routes are associated with an IPv4 interface, hence, an IPv4 address. The default route is *DFTROUTE.	Conceptually, the same as IPv4. One important difference: IPv6 routes are associated (bound) to a physical interface rather than an interface. There are various reasons for this. One reason is that source address selection functions differently for IPv6 than for IPv4. See “source address selection” on page 719.
Packet forwarding	The TCP/IP stack on i5/OS can be configured to forward IP packets it receives for non-local IP addresses. Typically, the inbound interface and outbound interface are connected to different LANs.	IPv6 packets are not forwarded by the V5R4 release of i5/OS.
Starting and stopping TCP/IP	Use STRTCP and ENDTCP to start or end TCP/IP.	Same as IPv4. IPv4 and IPv6 are not started or stopped independently of one another or independently of TCP/IP. That is, you start and stop all of TCP/IP, not just IPv4 or IPv6. IPv6 interfaces are automatically started if the AUTOSTART parameter = *YES (the default). IPv6 cannot be used or configured without IPv4. IPv6 loopback interface, ::1, is automatically configured and enabled in V5R4.

	IPv4	IPv6
Configuration	Configuration must be done on a newly installed system before it can communicate (that is, IP addresses and routes must be assigned).	Configuration is optional, depending on functions required. iSeries Navigator must be used to configure IPv6. IPv6 can be configured over any Ethernet interface. Stateless address autoconfiguration can be enabled on any Ethernet line. So, the system will be able to communicate with other IPv6 systems that are local and remote, depending on the type of network and whether an IPv6 router exists.
iSeries Navigator support	iSeries Navigator provides a full configuration function for TCP/IP.	iSeries Navigator is required to configure IPv6.
Renumbering	Done by manual re-configuration, with the possible exception of DHCP. Generally, for a site or organization, a difficult and troublesome process to avoid if possible.	Is an important architectural element of IPv6, and is supposed to be largely automatic, especially within the /48 prefix.
Domain Name System (DNS)	Applications accept host names and then use DNS to get an IP address, using socket API gethostbyname(). Applications also accept IP addresses and then use DNS to get host names using gethostbyaddr(). For IPv4, the domain for reverse lookups is in-addr.arpa.	Same for IPv6. Support for IPv6 exists using AAAA (quad A) record type and reverse lookup (IP-to-name). An application may elect to accept IPv6 addresses from DNS (or not) and then use IPv6 to communicate (or not). The socket API gethostbyname() is unchanged for IPv6 and the getaddrinfo() API can be used to obtain (at application choice) IPv6 only, or IPv4 and IPv6 addresses. For IPv6, the domain used for reverse nibble lookups is ip6.arpa, and if not found then ip6.int. See 3.3.3, “i5/OS DNS support for IPv6” on page 68 for more information.
Netstat	A tool to look at status of TCP/IP connections, interfaces, or routes. Available using iSeries Navigator and 5250.	Same for IPv6. IPv6 is supported for both 5250 and iSeries Navigator.
PING	Basic TCP/IP tool to test your ability to reach a remote host. Available using iSeries Navigator and 5250.	Same for IPv6. IPv6 is supported for both 5250 and iSeries Navigator.
Trace Route	Basic TCP/IP tool to do path determination. Available using iSeries Navigator and 5250.	Same for IPv6. IPv6 is supported for both 5250 and iSeries Navigator.

	IPv4	IPv6
Node info query	Does not exist.	A simple and convenient network tool that should work like PING, except with content: an IPv6 node may query another IPv6 node for the target's DNS name, IPv6 unicast address, or IPv4 address. Not supported in the V5R4 release of i5/OS.
Packet tunneling	In IPv4, tunneling occurs in VPN for tunnel-mode VPN connections (IPv4 tunneled in IPv4) and in L2TP.	IPv6 tunneling in not supported in the V5R4 release of i5/OS.
File Transfer Protocol (FTP)	File Transfer Protocol enables you to send and receive files across networks.	FTP on i5/OS has not been updated to support IPv6.
Telnet	Telnet enables you to log on and use a remote computer as though you were connected to it directly.	Telnet on i5/OS has not been updated to support IPv6.
Routing Information Protocol (RIP)	RIP is a routing protocol supported by the routed daemon.	RIPng (RIPv6) is not supported in the V5R4 release of i5/OS. IPv6 routing uses static routes.
Simple Network Management Protocol (SNMP)	SNMP is a protocol for system management.	SNMP on i5/OS has not been updated to support IPv6.
virtual private networking (VPN)	Virtual private networking (using IPsec) enables you to extend a secure, private network over an existing public network.	VPN support on i5/OS has not been updated to support IPv6.
Quality of service (QoS)	Quality of service enables you to request packet priority and bandwidth for TCP/IP applications.	Quality of Service on i5/OS has not been updated to support IPv6.
Packet filtering	Basic firewall functions integrated into TCP/IP, configured using iSeries Navigator.	Packet filtering on i5/OS has not been updated to support IPv6.
Network Address Translation (NAT)	Basic firewall functions integrated into TCP/IP, configured using iSeries Navigator.	NAT on i5/OS has not been updated to support IPv6. More generally, IPv6 does not require NAT. The expanded address space of IPv6 eliminates the address-shortage problem and enables easier renumbering.
Dynamic Host Configuration Protocol (DHCP)	Used to dynamically obtain an IP address and other configuration information.	The DHCP server on i5/OS has not been updated to support IPv6.
Point-to-Point Protocol (PPP)	PPP supports dial-up interfaces over various modem and line types.	i5/OS does not support IPv6 over PPP.

	IPv4	IPv6
Layer 2 Tunnel Protocol (L2TP)	L2TP can be thought of as virtual PPP, and works over any supported line type.	i5/OS does not support IPv6 over L2TP.
Host table	A configurable table that associates an Internet address with a host name; for example, 127.0.0.1, loopback. This table is used by the sockets name resolver, either before a DNS lookup or after a DNS lookup fails (determined by host name search priority).	In the V5R4 release, this table does not support IPv6 addresses. Customers need to configure a AAAA record in a DNS for IPv6 domain resolution. You may run the DNS locally on the same system as the resolver or you may run it on a different system.
Services table	A configurable table that associates a service name with a port and protocol; for example, service name FTP-control, port 21, TCP and UDP. A large number of well-known services are listed in the services table. Many applications use this table to determine which port to use.	No updates to table are needed for IPv6. The table supports IPv6 without change. Applications use the same port number for both IPv4 and IPv6.
Protocol table	A configurable table that associates a protocol name with its assigned protocol number; for example, UDP, 17. The system is shipped with a small number of entries: IP, TCP, UDP, ICMP.	No updates to the table are needed for IPv6. The table supports IPv6 without change.
Network table	A configurable table that associates a network name with an IP address without mask. For example, host Network14 and IP address 1.2.3.4.	The table is rarely used on most systems. It has not been updated to support IPv6 addresses.

Using IPv6 Communications Trace

Use Communications Trace to troubleshoot IPv6. Communications Trace is a service function that enables data to be traced on a communications line. After the IPv6 data has been collected, the raw IPv6 data must be dumped into a stream file before it can be displayed or printed.

Communications Trace may be used for troubleshooting both IPv4 and IPv6 communications; however, the procedure is slightly different for each. This section describes the IPv6 Communications Trace method (dump of the raw data to stream file is required).

Use Communications Trace in these situations:

- Your problem analysis procedures do not give enough information about the problem.
- You suspect that a protocol violation is the problem.
- You suspect that line noise is the problem.

- You want to know whether your application is transmitting information correctly across the network.
- You want to know whether you have performance problems with network congestion or data throughput.

To use the CL commands to perform a communications trace, you must have *SERVICE special authority, or be authorized to the Service Trace function of i5/OS through iSeries Navigator.

For more information about using communications trace, refer to the TCP/IP Troubleshooting topic in the Information Center (http://publib.boulder.ibm.com/infocenter/iseries/v5r4) in the path **Networking** → **TCP/IP troubleshooting**.

Preliminary steps

Before starting to work with a communications trace, follow these steps:

1. If you have not created the library IBMLIB or output queue IBMOUTQ, specify the following commands:

```
CRTLIB LIB(IBMLIB)
CRTOUTQ OUTQ(IBMLIB/IBMOUTQ)
```

2. Specify the following commands to add the library IBMLIB to your library list and to change the output queue for your job to output queue IBMOUTQ:

```
ADDLIBLE IBMLIB
CHGJOB * OUTQ(IBMLIB/IBMOUTQ)
```

3. If the QTCPPRT printer file does not exist on your system, then specify the following commands to create it:

```
CRTPRTF FILE(QTCP/QTCPPRT) DEV(*JOB) RPLUNPRT(*YES) SCHEDULE(*FILEEND) FILESEP(0)
   LVLCHK(*NO) TEXT('TCP/IP printer file')

CHGOBJOWN OBJ(QTCP/QTCPPRT) OBJTYPE(*FILE) NEWOWN(QSYS)
```

4. Specify the following commands to send the spooled file QTCPPRT containing the communications trace to the output queue IBMOUTQ in library IBMLIB:

```
OVRPRTF FILE(QTCPPRT) OUTQ(IBMLIB/IBMOUTQ)
OVRPRTF FILE(QPCSMPRT) TOFILE(QTCP/QTCPPRT)
```

5. The printer file overrides are not in effect after your job ends.
6. Obtain the name of the line description associated with the TCP/IP interface with which you are having the problem or which is used by the application or network with which you are having a problem. Use NETSTAT *IFC to determine the name of the line description associated with the interface.
7. Ensure that the line is varied on and that the TCP/IP interface associated with the line has been started so that TCP/IP data can be sent and received over the interface and the line. Use NETSTAT *IFC to verify that the interface is active.

Performing the trace

Follow these steps to start gathering information for your communication trace.

Start the trace

To start a communications trace, follow these steps:

1. At the command line, specify Start Communications Trace (STRCMNTRC).

2. At Configuration object, specify the name of the line, such as ETHLINE.
3. At Type, specify the type of resource *LIN.
4. At Buffer size, specify a sufficient amount of storage for the anticipated volume of data. For most protocols, 16 MB is sufficient storage.

End the trace

To end a communications trace, follow these steps:

1. At the command line, specify End Communications Trace (ENDCMNTRC).
2. At Configuration object, specify the same line you specified when you started the trace, such as ETHLINE.
3. At Type, specify the type of resource *LIN.

Dump the trace

To dump a communications trace, follow these steps:

1. Create a directory, such as mydir using the command CRTDIR DIR('mydir/mytraces')
2. At the command line, specify Dump Communications Trace (DMPCMNTRC).
3. At Configuration object, specify the same line you specified when you started the trace, such as ETHLINE.
4. At Type, specify the type of resource *LIN.
5. At To stream file, specify the path name, such as /mydir/mytraces/trace1.

Print the data from a stream file

To print the data from the stream file, follow these steps:

1. At the command line, type Print Communications Trace (PRTCMNTRC) and press F4 (Prompt).
2. At From stream file, specify path name, such as /mydir/mytraces/trace1, and press Enter.
3. At Character code, specify *EBCDIC or *ASCII. You should print the data twice, once specifying *EBCDIC and then specifying *ASCII.
4. At Format TCP/IP data, specify *YES, and press Enter twice.
5. Perform Steps 1 through 4 again, but specify a different character code.

View the trace

To view the contents of a communications trace, follow these steps:

1. At the command line, specify WRKOUTQ OUTQ(IBMLIB/IBMOUTQ).
2. On the Work with Output Queue dialog, press F11 (View 2) to view the date and time of the spooled file with which you want to work.
3. Specify 5 in the Opt column next to the spooled file you want to display. The last files contain the most recent communications traces.
4. Verify that this is a communications trace for the line traced and that the times that the trace started and ended are correct.

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this Redbooks publication.

IBM Redbooks

For information about ordering these publications, see “How to get IBM Redbooks” on page 728.

- *AS/400 Internet Security: Implementing AS/400 Virtual Private Networks*, SG24-5404
- *AS/400 Internet Security: Protecting Your AS/400 from HARM in the Internet*, SG24-4929
- *AS/400 Internet Security: Developing a Digital Certificate Infrastructure*, SG24-5659
- *AS/400 Internet Security Scenarios: A Practical Approach*, SG24-5954
- *AS/400 Remote Access Configuration Examples*, SG24-6058
- *AS/400 TCP/IP Autoconfiguration: DNS and DHCP Support*, SG24-5147
- *Exploring NFS on AS/400*, SG24-2158
- *IBM eServer iSeries Wired Network Security: OS/400 V5R1 DCM and Cryptographic Enhancements*, SG24-6168
- *IBM eServer iSeries Universal Connection for Electronic Support and Service*, SG24-6224
- *IBM eServer iSeries e-business Handbook: A V5R1 Technology and Product Reference*, SG24-6711
- *Implementation and Practical Use of LDAP on the IBM eServer iSeries Server*, SG24-6193
- *LPAR Configuration and Management Working with IBM eServer iSeries Logical Partitions*, SG24-6251
- *V4 TCP/IP for AS/400: More Cool Things Than Ever*, SG24-5190
- *OS/400 V5R2 Virtual Private Networks: Remote Access to the IBM eServer iSeries Server with Windows 2000 VPN Clients*, REDP-0153
- *Remote Access to AS/400 with Windows 2000 VPN Clients*, REDP0036
- *TCP/IP Tutorial and Technical Overview*, GG24-3376

Other resources

This publication is also relevant as a further information source:

- *iSeries TCP/IP Configuration and Reference Version 5*, SC41-5420

Referenced Web sites

These Web sites are also relevant as further information sources:

- iSeries Information Center

  `http://publib.boulder.ibm.com/html/as400/infocenter.html`

- IBM WebSphere Edge Server

  http://www.ibm.com/software/webservers/edgeserver/
- RFC Index Search Engine

  http://www.rfc-editor.org/rfcsearch.html
- IPv6 Internet Drafts relating to mobility support

  http://www.ietf.org/ids.by.wg/ipv6.html
- CISCO offers IPv6-capable routers. Information regarding their support of IPv6 is available on their Web site at:

  http://www.cisco.com/warp/public/732/Tech/ipv6/

How to get IBM Redbooks

You can order hardcopy Redbooks, as well as view, download, or search for Redbooks at the following Web site:

ibm.com/redbooks

You can also download additional materials (code samples or diskette/CD-ROM images) from that site.

IBM Redbooks collections

Redbooks are also available on CD-ROMs. Click the CD-ROMs button on the Redbooks Web site for information about all the CD-ROMs offered, as well as updates and formats.

Index

Symbols

A

B

C

D

E

F

G

H

I

P

Q

R

S

T

Redbooks

IBM i5/OS IP Networks: Dynamic

IBM i5/OS IP Networks: Dynamic

Hints for secure, self-configuring, fault-tolerant IP networks

IPv6 and how you can use it on i5/OS

How-to: Virtual IP, DHCP, DDNS, QoS, IDS, and more

Over the course of many years, the developers in both the Endicott and Rochester labs have been working very hard adding functions to each release of OS/400 and i5/OS to make the configuration and use of the IBM System i in a TCP/IP network easier and more powerful. If you need to design an IP network that is self-configuring, fault-tolerant, secure, and efficient in its operation, then this IBM Redbook is for you.

We start low with the details of IP interface and route implementation on i5/OS. Through the study of these building blocks, we show how to create IP networks that are easier to configure, tolerant of faults, and can perform both inbound and outbound load balancing.

i5/OS has always had many built-in Network Security features. These features have been enhanced to include an Intrusion Detection System (IDS). This allows you to be notified of attempts to hack into, disrupt, or deny service to the system.

Moving up to the application layer, we demonstrate the dynamic power of IP by having the DHCP server assigning IP addresses and automatically updating the i5/OS Dynamic DNS. Now clients and servers can be added dynamically to the IP network and assigned a name automatically.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

SG24-6718-02 ISBN 0738486493