



System i
Networking
TCP/IP routing and workload balancing

Version 6 Release 1





System i

Networking

TCP/IP routing and workload balancing

Version 6 Release 1

Note

Before using this information and the product it supports, read the information in "Notices," on page 35.

This edition applies to version 6, release 1, modification 0 of IBM i5/OS (product number 5761-SS1) and to all subsequent releases and modifications until otherwise indicated in new editions. This version does not run on all reduced instruction set computer (RISC) models nor does it run on CISC models.

© **Copyright International Business Machines Corporation 1998, 2008.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

TCP/IP routing and workload balancing 1

What's new for V6R1	1
PDF file for TCP/IP routing and workload balancing	2
TCP/IP routing functions by release	2
Packet processing.	3
General routing rules	4
Routing connectivity methods	4
Routing with point-to-point connections	4
Proxy Address Resolution Protocol routing	9
Transparent subnets	9
Dynamic routing	10
Routing Information Protocol	10
Open Shortest Path First	11
Route binding	15
Classless Inter-Domain Routing.	16
Routing with virtual IP	17
Fault tolerance	19
Routing with network address translation	19
Masquerade NAT	19
Inbound masquerade NAT processing (response and other)	21
Outbound masquerade NAT processing	21

Dynamic NAT	22
Static NAT.	23
Routing with OptiConnect and logical partitions	24
TCP/IP and OptiConnect.	24
Routing with virtual OptiConnect and logical partitions	25
TCP/IP workload balancing methods.	26
DNS-based load balancing	26
Duplicate route-based load balancing.	27
Load balancing using virtual IP and proxy ARP	28
Scenario: Adapter failover using virtual IP and proxy ARP	30
Failover using automatic interface selection.	33
Failover using a preferred interface list	33
Related information for TCP/IP routing and workload balancing.	34

Appendix. Notices 35

Programming interface information	36
Trademarks	37
Terms and conditions	37

TCP/IP routing and workload balancing

You can route and balance the TCP/IP traffic of your system by using its integrated routing capabilities to eliminate the need for an external router.

The routing and workload balancing methods, as well as the background information, can help you understand the options available for you to use on your system. Each method is described using a figure so that you can see how the connections are made. These methods do not include instructions on configuring the routing techniques. The focus of this topic collection is on the routing principles and concepts you should know so that your system works better for you.

Why these methods are important to you

The techniques in these methods might cut down the overall cost of your connections because you can use fewer external routers and servers. Using these routing methods, you can free up IP addresses because you will be managing them in a more effective way. By reading the workload balancing methods, you can get better overall system performance by balancing the communications workload on your system.

What's new for V6R1

Read about new or significantly changed information for the TCP/IP routing and workload balancing topic collection.

New routing protocol supported

The i5/OS[®] operating system has been extended to support the Open Shortest Path First (OSPF) routing protocol. *Open Shortest Path First* (OSPF) is a link-state routing protocol in which routers or systems within the same area maintain an identical link-state database that describes the topology of the area.

Virtual IP enhancements

Virtual IP enhancements that affect the TCP/IP routing and workload balancing topic collection are as follows:

- Virtual IP address support has been extended to include IPv6 addresses.
- A Point-to-Point Protocol (PPP) interface or a Layer Two Tunneling Protocol (L2TP) interface can use a virtual IP address as the local IP address to provide fault tolerance for remote connections.
- You can configure virtual IP Proxy ARP while the virtual IP interface is active.

You can find these IPv6 enhancements in the “Routing with virtual IP” on page 17 and “Fault tolerance” on page 19 topics.

New load balancing method documented

Although using virtual IP and proxy ARP as a load balancing method is not new for V6R1, this load balancing method was not documented in this document before. A “Load balancing using virtual IP and proxy ARP” on page 28 topic has been added to introduce this load balancing method.

How to see what's new or changed

To help you see where technical changes have been made, the information center uses:

- The  image to mark where new or changed information begins.

- The  image to mark where new or changed information ends.

In PDF files, you might see revision bars (|) in the left margin of new and changed information.

To find other information about what's new or changed this release, see the Memo to users.

PDF file for TCP/IP routing and workload balancing

You can view and print a PDF file of this information.

To view or download the PDF version of this document, select TCP/IP Routing and workload balancing (about 1.40 MB).

Saving PDF files

To save a PDF on your workstation for viewing or printing:

1. Right-click the PDF link in your browser.
2. Click the option that saves the PDF locally.
3. Navigate to the directory in which you want to save the PDF.
4. Click **Save**.

Downloading Adobe Reader

You need Adobe® Reader installed on your system to view or print these PDFs. You can download a free copy from the Adobe Web site (www.adobe.com/products/acrobat/readstep.html) .

Related reference

“Related information for TCP/IP routing and workload balancing” on page 34

Other information center topic collections contain information that relates to the TCP/IP routing and workload balancing topic collection.

TCP/IP routing functions by release

Before you plan to use a routing function, make sure that your system is at the correct release to support the function that you want to perform.

V3R1: Static route-based packet forwarding

V3R7/V3R2: Serial Line Internet Protocol (SLIP), Proxy Address Resolution Protocol (ARP) routing, and unnumbered connection network support

V4R1: Dynamic Routing Information Protocol Version 1 (RIPv1).

V4R2: Dynamic Routing Information Protocol Version 2 (RIPv2), transparent subnetting, and duplicated route-based load-balancing

V4R3: Virtual IP addresses, IP address masquerading, network address translation (NAT), and Classless Inter-Domain Routing (CIDR)

V4R4: IP over OptiConnect

V5R4: Preferred interface list

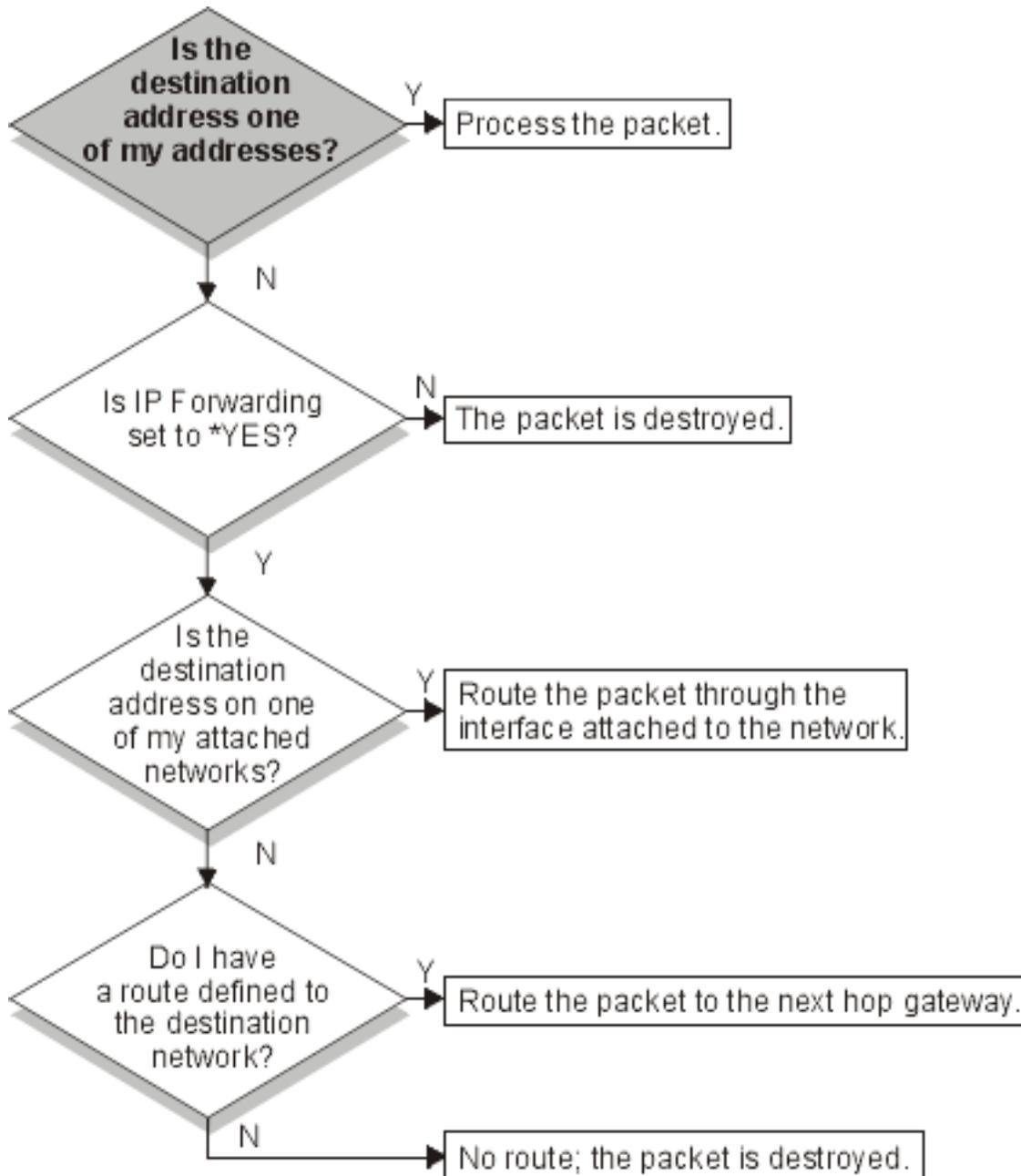
| **V6R1:** Open Shortest Path First (OSPF) routing protocol and virtual IP address support for IPv6
| addresses

2 System i: Networking TCP/IP routing and workload balancing

Packet processing

Having a better understanding of packet processing helps you decide how to implement routing functions.

The following simplified flow chart shows the logical process that takes place when the i5/OS operating system receives an IP packet (datagram). The actual flow might be different, but the outcome should be the same. The following logic only describes the default packet processing scenarios. If advanced routing techniques are used, packet processing might be slightly different.



First, the destination address in the IP header is compared to all the defined addresses on the system. If it is determined that the packet is destined for your system, the packet is passed up the IP stack to a higher level software, such as TCP, and then to the application that is listening on the destination port.

If the packet is not accepted locally, the next check that is performed is the IP forwarding attribute. If IP forwarding is set to *YES, then this system is configured to forward packets like a router. If the attribute is set to *NO in the TCP/IP attributes or in the PPP profile, the packet is destroyed.

The destination address of the packet is compared to all the *DIRECT routes known to your system. This is done by including the destination address of the packet with the subnet mask specified in the *DIRECT routing entries of the defined interfaces to determine if the packet is destined for a network that is directly attached to this system. Checking is done from the most specific routes to the least specific.

Then, if i5/OS is not directly connected to the remote host, the routing table is searched. Checking is done from the most specific host (subnet mask 255.255.255.255) to the least specific route (subnet mask 0.0.0.0). If a route is found, the packet is forwarded to the next hop gateway.

The last point in the flow chart shows that if no matching routing entry is found, the packet is destroyed.

General routing rules

These rules apply to TCP/IP in general and to TCP/IP on the i5/OS operating system.

To manage packets on your system, you should consider these rules as you implement routing functions on your system. These rules can help you determine what is happening to the packets on your system and where they might be going. As with most rules, there are exceptions.

- Your system does not have an IP address; only interfaces have IP addresses.

Note: Virtual IP (connectionless) addresses are assigned to the system.

- In general, if the destination IP address is defined on your system, your system will process it regardless of what interface a packet comes in on.

The exception in this case is that if the address is associated with an unnumbered interface, or if IP NAT or filtering is active, the packet might be forwarded or discarded.

- The IP address and mask define the address of the attached network.
- The route out of a system is selected based on the network address attached to an interface. The route selected is based on the following items:
 - Route group search order: direct routes, subnetwork routes, and then default routes.
 - Within a group, the route with the most specific subnet mask is chosen.
 - Equally specific routes are subject to list order or load-balancing techniques.
 - Routes can be added manually or dynamically by the system.

Routing connectivity methods

Routing deals with what path the network traffic follows from its source to its destination and how that path is connected.

Routing with point-to-point connections

Using point-to-point connections, you can send your data from your local system to a remote system or from a local network to a remote network.

Point-to-point connections are typically used to connect two systems together over a wide area network (WAN). You can use a point-to-point connection to get data from your local system to a remote system or to get data from a local network to a remote network. Do not confuse point-to-point connections with Point-to-Point Protocol. Point-to-Point Protocol (PPP) is one type of a point-to-point connection that is commonly used to connect a computer to the Internet. See PPP connections for more information about how to set up and manage your PPP connections.

You can use point-to-point connections across dial-up lines, nonswitched lines, and other types of networks such as frame relay. There are two ways that you can configure the IP addresses for a point-to-point connection: a numbered connection or an unnumbered connection. As the names imply, a numbered connection has a unique IP address defined for each interface. An unnumbered connection does not use additional IP addresses for a connection.

Numbered network connections

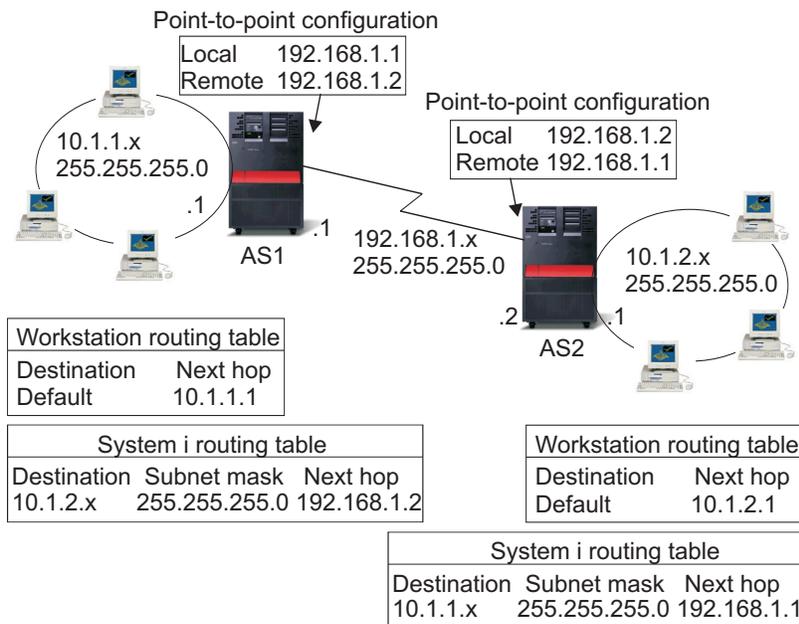
On the surface, it seems that the simplest way to configure a point-to-point connection is by using a numbered connection. A numbered connection is a point-to-point definition that has a unique IP address defined for each end of a connection.

Here are some points to keep in mind when you consider a numbered point-to-point connection:

- Each end of the connection has a unique IP address.
- Routing statements must be added to your system to flow the traffic to the remote system.
- Addresses on the point-to-point link must be managed by your network administrator.
- Addresses are used up just to connect two systems.

When each point-to-point connection is defined to your system, a routing entry must be made on each end to describe how to get to any network at the other end of the connection. The routing selection process on your system depends on having an IP address for each interface. These addresses and routes must be managed by your network administrator. In a small network, these addresses are easy to keep track of and do not use many additional addresses. In a large network, however, it might take an entire subnet of addresses just to define an interface at each end.

The following figure shows a numbered network connection between two System i® platforms. A routing entry is not needed if you only want to communicate from AS1 to AS2. If you want to communicate with systems in the remote network (10.1.2.x), the routing entry included in the figure must be added to each system. This is because the remote network, 10.1.2.x, is part of the 192.168.1.x connection.



RZAJW521-1

Unnumbered network connections

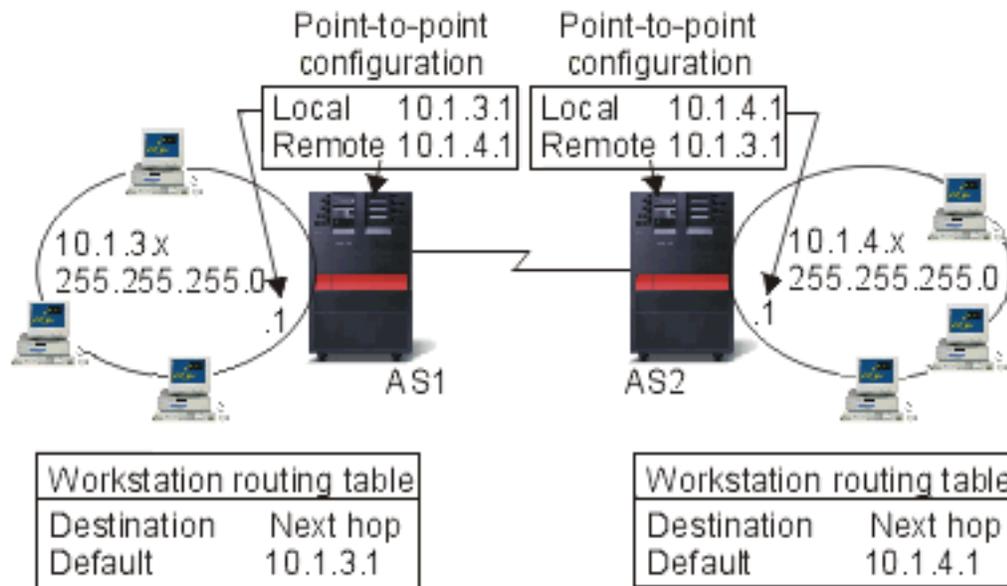
An unnumbered connection is a more complex method of defining a point-to-point connection than a numbered connection. However, you might find the unnumbered connection a simpler and better way to manage your network.

The routing selection process on i5/OS depends on having an IP address for an interface. In an unnumbered connection, the point-to-point interface does not have a unique address. The IP address of your system interface for an unnumbered connection is actually the IP address of the remote system.

Points to keep in mind while considering an unnumbered connection:

- The point-to-point interface has an address that appears to be in the remote network.
- Routing statements are not needed in the system.
- Your network administration is simplified by not using up IP addresses for the link.

In the following example, AS1 appears to have an interface in the 10.1.4.x network and AS2 appears to have an interface in the 10.1.3.x network. The AS1 is connected to LAN network 10.1.3.x with an address of 10.1.3.1. This allows AS1 to communicate with any system on the 10.1.3.x network directly.



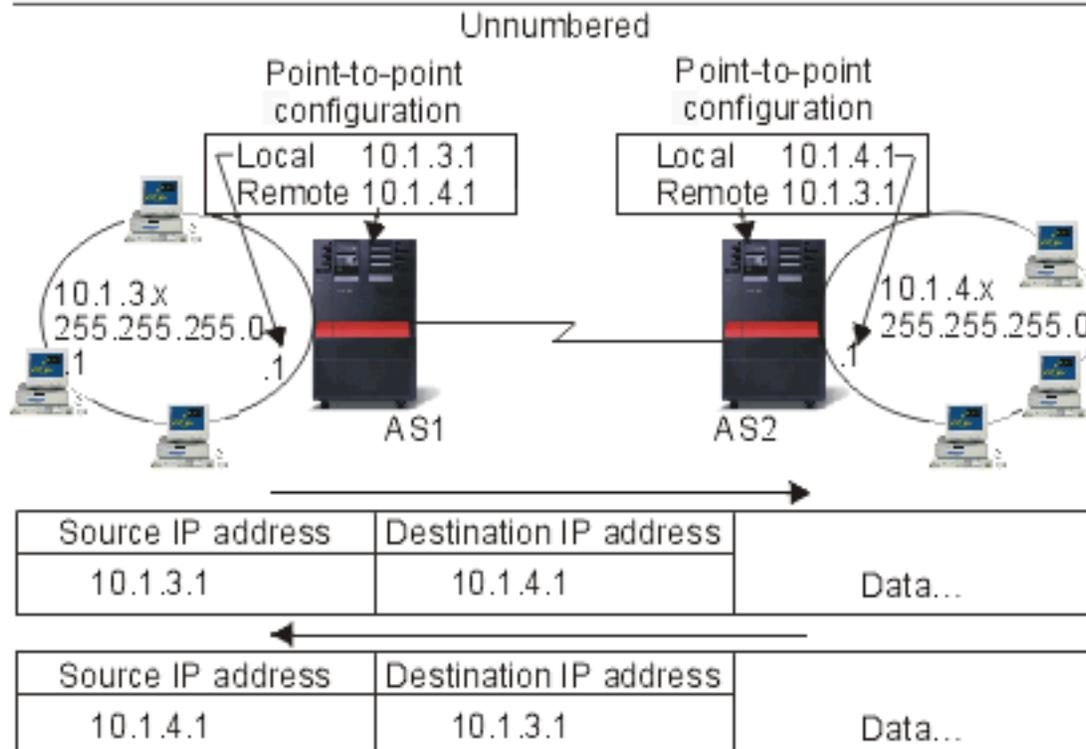
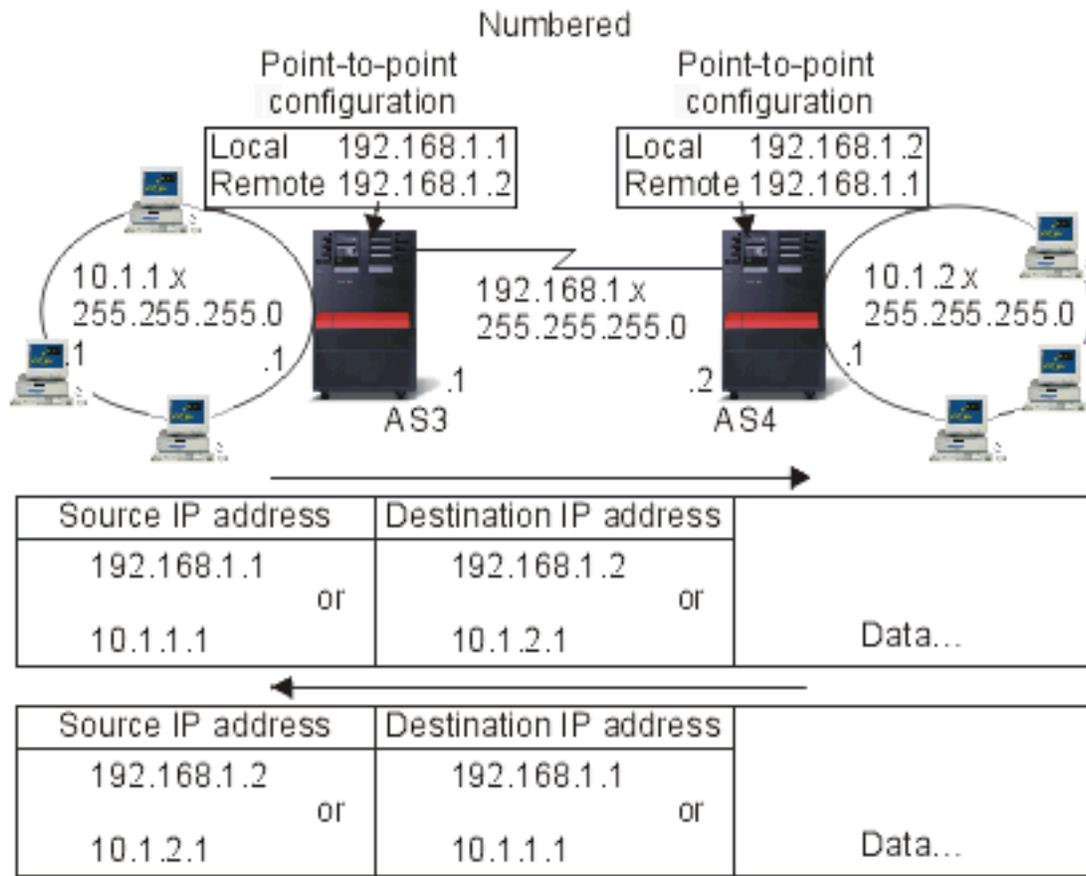
Also shown in the example is AS2. AS2 is connected to LAN network 10.1.4.x with an address of 10.1.4.1. This allows AS2 to communicate with any system on the 10.1.4.x network directly. Each system (AS1 and AS2) adds the remote address to its routing table as a local interface. The address is treated specially so that 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.

Now you have a need to connect AS1 to the 10.1.4.x network and to connect AS2 to the 10.1.3.x network. If these two systems were in the same room, you can add a LAN adapter to each system and plug the new interface into the correct LAN. If you did this, AS1 and AS2 would not need any routing entries added. In this example, however, the systems are in different cities so you must use a point-to-point connection. Even though you are using a point-to-point connection, you might still want to avoid adding routing entries. By defining the Point-to-Point Protocol (PPP) connection as an unnumbered connection, you achieve the same results that you can get if you use LAN adapters without adding any routing

entries to your system. To do this, each system borrows the IP address of the remote system for use with route resolution.

Unnumbered versus numbered connection data flow

The following figure shows the addresses that will be used in a numbered and unnumbered point-to-point connection. The top half of the picture shows, that with a numbered connection, the remote system address of 192.168.1.2 or 10.1.2.1 could be used to reach the remote system. This is because there is a routing entry in AS3 that directs packets for 10.1.2.1 to 192.168.1.2 as the next hop. The addresses used in the return packet are based on the received packet. The bottom of the figure shows the addresses used with an unnumbered connection. The outbound packet has a source of 10.1.3.1 and a destination of 10.1.4.1. No routing entries are needed on either system because the systems have a direct interface to the remote network by using the remote system address of the point-to-point connection.



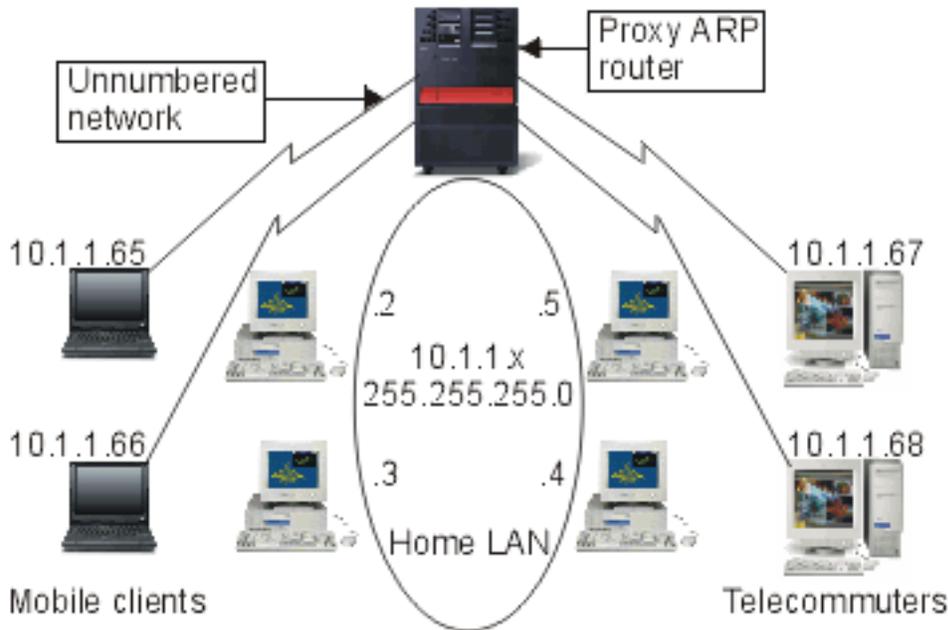
Related concepts

PPP connections

Proxy Address Resolution Protocol routing

Proxy Address Resolution Protocol (ARP) provides connectivity between physically separate networks without creating any new logical networks and without updating any routing tables. This topic also contains a description of transparent subnets, which is an extension to the proxy ARP routing technique.

ARP routing allows physically distinct, separate networks to appear as if they were a single logical network. It allows systems that are not directly connected to a local area network (LAN) to appear to other systems on the LAN as though they are connected. This is useful in dial-up scenarios to provide connections to the entire network from a dial-in interface. The following figure shows a possible scenario. The 10.1.1.x is your home LAN and the 10.1.1.65 through 10.1.1.68 are your remote systems.

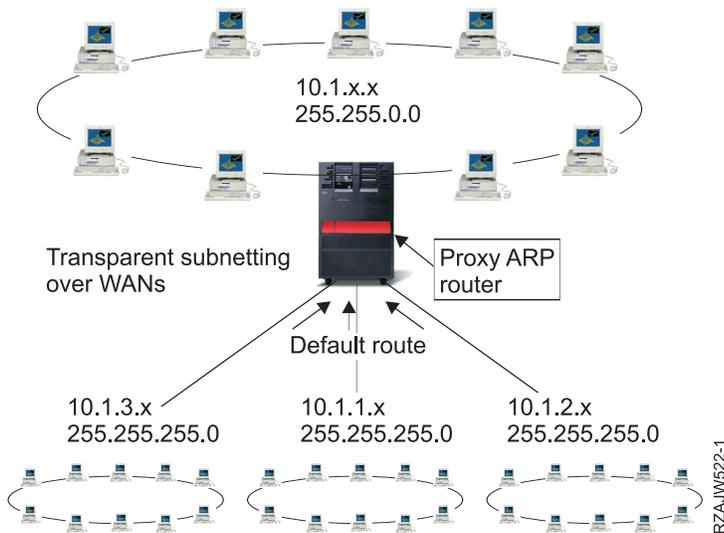


When a system on your home LAN (10.1.1.x) wants to send data to one of your remote systems, it will first do an ARP request. This is a broadcast that goes out to all your systems attached to the LAN segment to request the address of the target system. A remotely connected system cannot see the broadcast. But with proxy ARP, your system knows which systems are connected remotely. If your system sees an ARP request for one of your remotely connected systems, your system replies to the ARP request with its address. Your system in turn receives the data and forwards it to the remote system. For this forwarding to take place, IP forwarding must be set to *YES. If your remote system is not connected, your system cannot reply to the ARP request and the requesting system cannot send data.

Transparent subnets

Transparent subnets can be used as a way to extend the proxy ARP concept. You can use transparent subnets as a proxy for an entire subnet, or range of hosts. Transparent subnetting allows stub networks to be assigned addresses out of the primary network address space.

Transparent subnets work for a single host so that you can connect to an entire subnet or range of hosts. You can see in the following figure that the stub networks (10.1.1.x through 10.1.3.x) are assigned addresses out of the primary network address space (10.1.x.x).



The transparent subnet function can be further expanded to handle real LANs that are remotely located. Transparent subnetting over WANs makes remote networks appear to be connected to the home network. In the preceding figure, three networks are attached to the home 10.1.x.x network through the System i platform. These networks are all defined using a subnet mask that makes them transparent to the home network. Proxy ARP responds to any ARP request in the home network for systems in the 10.1.1.x, 10.1.2.x, and 10.1.3.x subnets. This action causes the traffic of the home network to be routed automatically to the system in the home network. This system, in turn, routes the data to the correct remote system. The remote system either processes the data, or forwards 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 in their network as the first hop gateway. The workstations in the home LAN do not need any additional routing entries because no new logical networks are created.

Dynamic routing

Dynamic routing is a low-maintenance method that automatically reconfigures routing tables when your network changes.

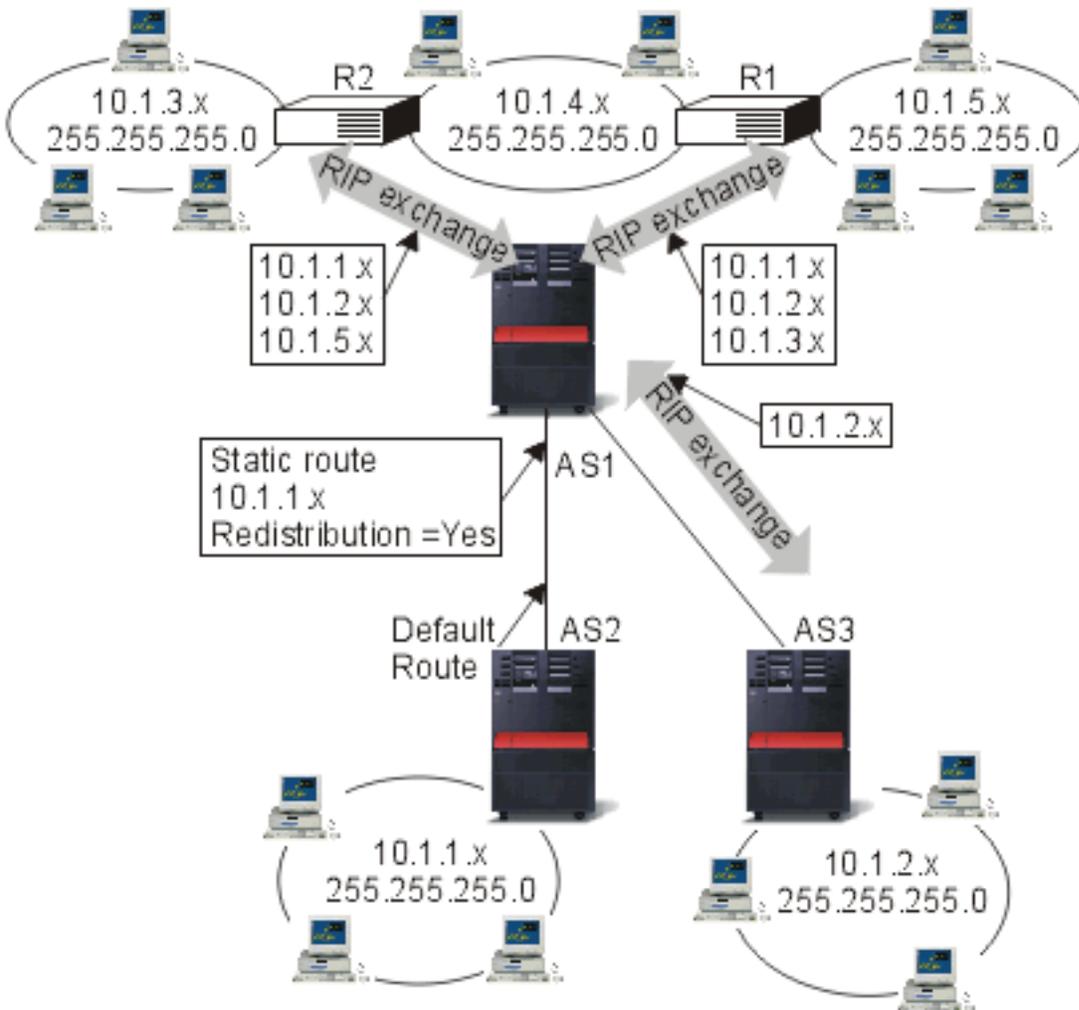
- | Dynamic routing is provided by Interior Gateway Protocols (IGPs). Routing Information Protocol (RIP)
- | and the Open Shortest Path First (OSPF) protocol are the two IGPs that the i5/OS operating system
- | supports.

Routing Information Protocol

Routing Information Protocol (RIP) is a distance-vector routing protocol. Routers running the distance-vector protocol send all or a portion of their routing tables in routing-update messages to their neighbors.

You can use RIP to configure the hosts as part of a RIP network. This type of routing requires little maintenance and also automatically reconfigures routing tables when your network changes or network communication stops. RIPv2 was added to the System i product so you can send and receive RIP packets to update routes throughout your network.

In the following figure, a static route is added to the central system (AS1) that describes the connection to the network 10.1.1.x by way of AS2. This is a static route (added by your network administrator) with route redistribution set to yes. This setting causes this route to be shared with other routers and systems so that when they have traffic for 10.1.1.x, they route the traffic to your central System i platform (AS1). AS2 has the routed system started so that it sends and receives RIP information. In this example, AS1 is sending the message that AS2 has a direct connection to 10.1.2.x.



The following process describes the routing of traffic in the preceding figure.

- AS1 receives this RIP packet from AS2 and processes it. If AS1 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, AS1 keeps the route data.
- AS1 receives information from R1 with route information to 10.1.5.x. AS1 keeps this route information.
- AS1 receives information from R2 with route information to 10.1.3.x. AS1 keeps this route information.
- The next time AS1 sends RIP messages, it will send information to R1 that describes all the connections AS1 knows about that R1 might not know about. AS1 sends route information about 10.1.1.x, 10.1.2.x, and 10.1.3.x. AS1 does not send information about 10.1.4.x to R1 because AS1 knows that R1 is connected to 10.1.4.x and does not need a route. Similar information is sent to R2 and AS3.

| Open Shortest Path First

| *Open Shortest Path First* (OSPF) is a link-state routing protocol that was developed for IP networks and is based on the Shortest Path First (SPF) algorithm. OSPF is an Interior Gateway Protocol (IGP).

| In an OSPF network, routers or systems within the same area maintain an identical link-state database that describes the topology of the area. Each router or system in the area generates its link-state database from the link-state advertisements (LSAs) that it receives from all the other routers or systems in the same area and the LSAs that itself generates. An LSA is a packet that contains information about neighbors and path costs. Based on the link-state database, each router or system calculates a shortest-path spanning tree, with itself as the root, using the SPF algorithm.

| OSPF has the following key advantages:

- | • Compared with distance-vector routing protocols such as the Routing Information Protocol (RIP), OSPF is more suitable for serving large, heterogeneous internetworks. OSPF can recalculate the routes in a short amount of time when the network topology changes.
- | • With OSPF, you can divide an Autonomous System (AS) into areas and keep area topologies separate to decrease the OSPF routing traffic and the size of the link-state database of each area.
- | • OSPF provides equal-cost multipath routing. You can add duplicate routes to the TCP stack using different next hops.

| **OSPF Hello protocol and link-state database exchange**

| After routers or systems in an OSPF network ensure that their interfaces are functional, they first send out Hello packets, using the Hello protocol over their OSPF interfaces, to discover neighbors. Neighbors are routers or systems that have interfaces to the common network. After that, neighboring routers or systems exchange their link-state databases to establish adjacencies.

| The following figure illustrates the process of discovering neighbors and establishing adjacencies for two systems in the 9.7.85.0 subnet. Each system has an OSPF interface to the common subnet 9.7.85.0 (interface 9.7.85.1 for system A and interface 9.7.85.2 for system B). Subnet 9.7.85.0 belongs to area 1.1.1.1.

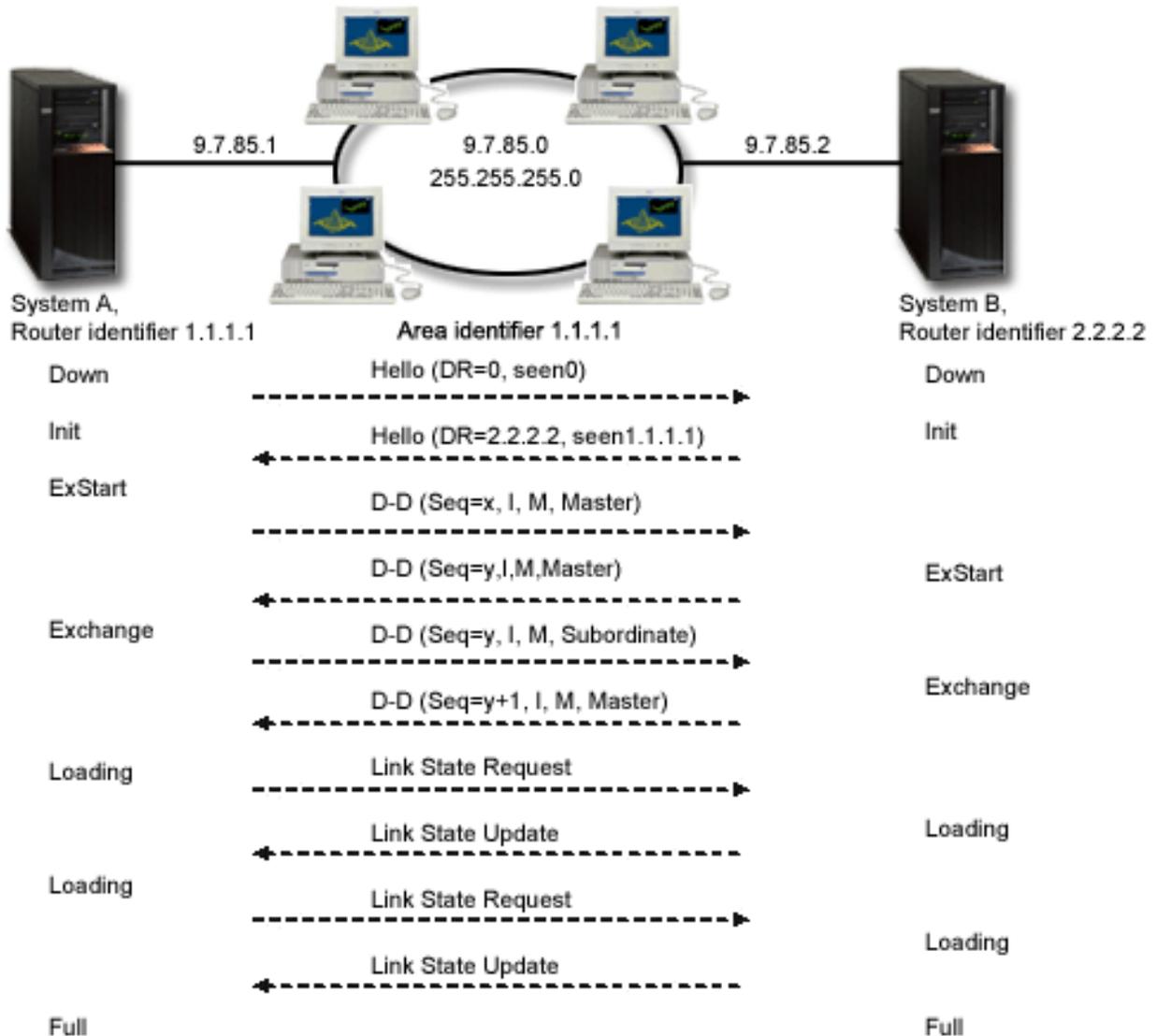


Figure 1. OSPF Hello protocol and database exchange

EXSTART phase

This is the first step of the link-state database exchange. The two systems negotiate who is the master and who is the subordinate.

EXCHANGE phase

The two systems exchange Database Description packets to find out the LSAs that the link-state database of each system does not include. Each system stores the LSAs that are not included in its link-state database in the retransmission list.

LOADING phase

Each system sends Link State Request packets to request the neighbor (the other system in this example) to send to it the entire LSAs that were stored in the retransmission list during the EXCHANGE phase. The neighbor responds to the request with the LSAs in Link State Update packets.

FULL phase

When the two systems finish exchanging LSAs and their link-state databases are synchronized, adjacency is established between the two systems.

| After adjacencies are established between all the routers or systems in an area, each router or system in the area periodically sends an LSA to share its adjacencies or to report its state change. By comparing the established adjacencies with the LSAs, routers or systems in the area can discover the area topology changes and update their link-state databases accordingly.

| **Designated router and backup designated router**

| In a multiaccess OSPF network that has at least two attached routers, the routers elect a designated router and a backup designated router using the Hello protocol. (A multiaccess network is a network in which multiple devices can connect and communicate simultaneously.)

| The designated router generates LSAs for the entire multiaccess network, floods the LSAs to the other routers in the network, and determines which routers should become adjacent. All the other routers in the network are adjacent to the designated router. The designated router decreases the network traffic and the size of the link-state database for this network.

| The backup designated router does not have any differences from the other routers except that it needs to establish adjacencies with all the routers in the network (including the designated router). The backup designated router is promoted to the designated router when the current designated router fails.

| In Figure 1, the 9.7.85.0 subnet is a broadcast network. Therefore, the routers in the 9.7.85.0 subnet elect a designated router and a backup designated router using the Hello protocol. In this example, system A is elected as the designated router and system B is elected as the backup designated router.

| **Splitting an OSPF AS into areas**

| Unlike RIP, OSPF can operate within a hierarchy. The largest entity within the hierarchy is the AS. An AS is a group of networks under a common administration that share a common routing strategy. An AS can be divided into areas, which are connected to each other by routers. An area consists of groups of contiguous networks and attached hosts. The topology of an area is invisible to entities outside the area. Routers within the same area have an identical link-state database. Separate area topologies allow for less routing traffic and smaller link-state database for each area.

| A router that is located on the border of OSPF areas and connects these areas to the backbone network is called Area Border Router. An Area Border Router has multiple interfaces to multiple areas and maintains separate link-state databases for each area.

| In the following figure, two areas (area 1.1.1.1 and area 2.2.2.2) are configured. System B is an Area Border Router, with interface 9.7.85.2 attached to area 1.1.1.1 and with interface 9.5.104.241 attached to area 2.2.2.2. System B has two link-state databases, one for each area. System B establishes adjacencies with system A and router C in area 1.1.1.1 through interface 9.7.85.2, and establishes adjacency with system D in area 2.2.2.2 through interface 9.5.104.241.

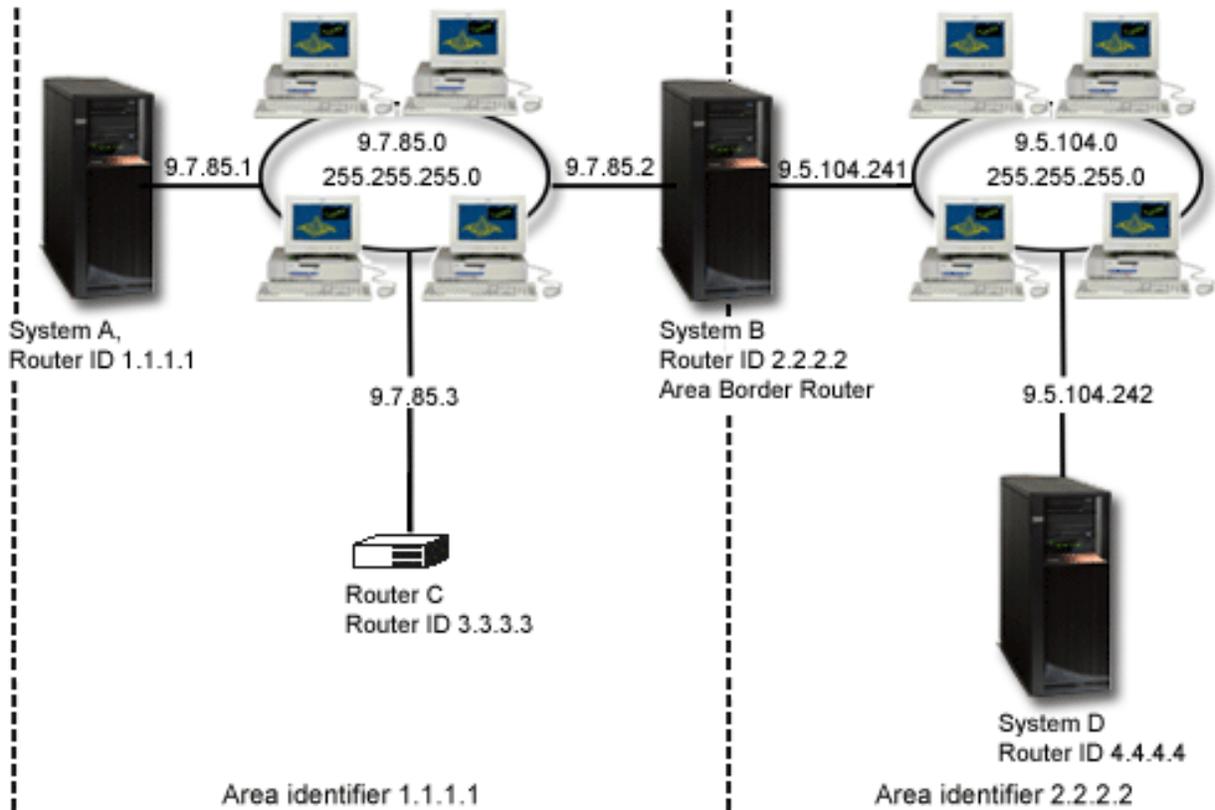


Figure 2. Splitting an OSPF AS into areas

Related concepts

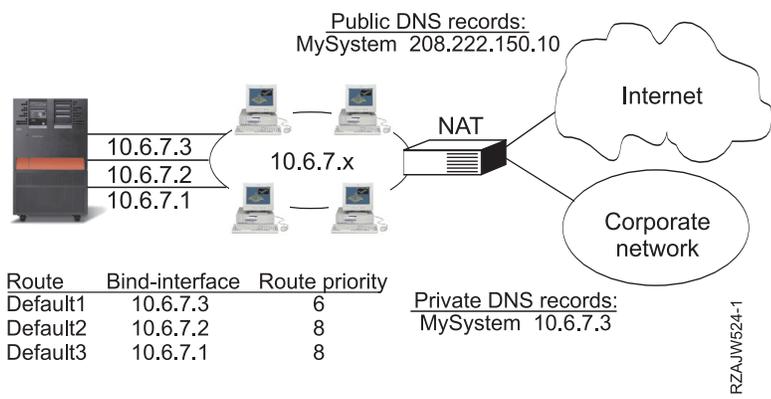
Open Shortest Path First (OSPF)

Route binding

Route binding gives you control over which interface is used to send out response packets of information.

Before preferred route binding came along, you did not have complete control over which interface was used to send out response packets of information. The Preferred Route Binding Interface, added to the add route function, gives you more control over which interface is used to send out your packets by allowing you to explicitly bind routes to interfaces.

In the following figure, three interfaces are connected to the same network. To guarantee that no matter which interface receives the inbound request, the reply can be sent back to the same interface, you must add the duplicate routes to each interface. In this example, three default routes are added; each one is explicitly bound to a different interface. This binding does not change regardless of the order in which interfaces are started or ended.



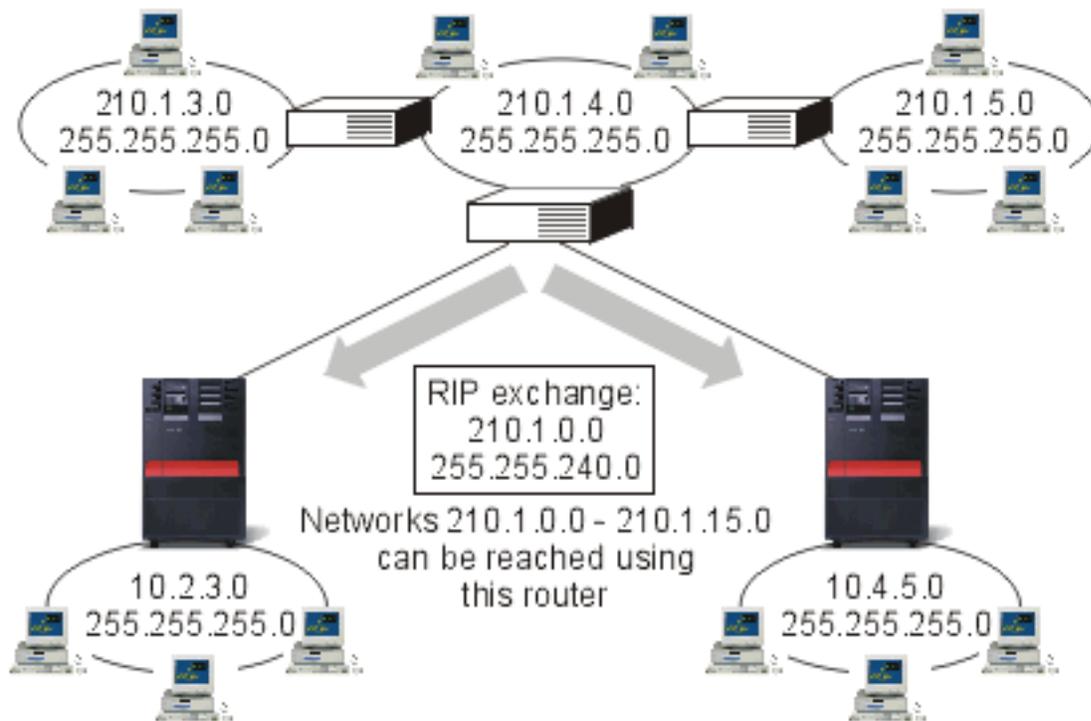
Classless Inter-Domain Routing

Classless Inter-Domain Routing can reduce the size of your routing tables and make more IP addresses available within your business.

Classless Inter-Domain Routing (CIDR or supernetting) is a way to combine several class-C address ranges into a single network or route. This method of routing adds class-C Internet Protocol (IP) addresses. These addresses are given out by Internet service providers (ISPs) for use by their customers. CIDR addresses can reduce the size of your routing tables and make more IP addresses available within your business.

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 a subnet of 255.255.255.0 was the largest (253 host) that could be specified. To conserve IP addresses, when companies needed more than 253 hosts in a network, the Internet was issuing several class-C addresses. This made the configuration of routes and other things difficult.

Now, CIDR allows these contiguous class-C addresses to be combined into a single network address range by using the subnet mask. For example, if you are giving out four 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 can ask your ISP to make them a supernet by using the subnet mask 255.255.252.0. This mask combines the four networks into one for routing purposes. CIDR is beneficial because it reduces the number of assigned but unnecessary IP addresses.

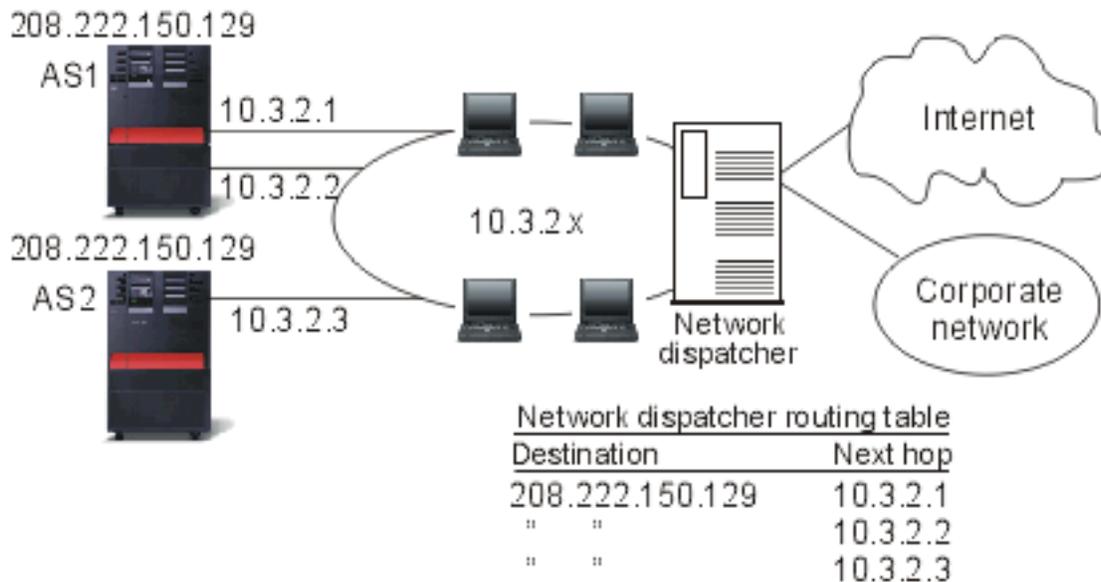


In this example, the router is set up to send one RIP message with the network address 210.1.0.0 and subnet mask 255.255.240.0. This tells your systems to receive the RIP messages for networks 210.1.0.0 through 210.1.15.0 through this router. This sends one message rather than the 16 that it would take to convey the same information if CIDR were not available.

Routing with virtual IP

Virtual IP, also called a circuitless or loopback interface, is a powerful function that provides a way to assign one or more addresses to the system without the need of binding the address to a physical interface.

You can use this function when you want to run multiple occurrences of a system bound to different addresses, or if you want to run other services that need to bind to default ports. Most environments where you might want to use virtual IP are cases where you want to provide multiple paths between the local gateway and the System i platform, for example, load balancing and fault tolerance. In this context, each path implies an additional interface, and consequently, an additional, nonvirtual IP address on the system, as is shown in the following figure.



Advantage:

- Load-based dispatching

Disadvantage:

- Requires external dispatcher

The existence of these multiple interfaces should only be visible on the local network. You do not want the remote clients to be aware of the multiple IP addresses for the system. Ideally, you want them to view your system as a single IP address. How the inbound packet gets routed through the gateway, over the local network, and to the system should be invisible to a remote client. The way to accomplish this is by using virtual IP. Local clients should communicate with the system by any of the physical IP addresses while remote clients see only the virtual IP interface.

The virtual IP environment is for the system that acts as the server for remotely connected clients. More importantly, the virtual IP address is on a different subnet than the physical interfaces. Moreover, the virtual IP address makes your system appear as a single host, not necessarily as one attached to a larger network or subnetwork. Therefore, the subnet mask for the virtual IP interface should typically be set to 255.255.255.255.

- | Because the virtual IP address is not bound to a single physical interface, the system never responds to an Address Resolution Protocol (ARP) request to the virtual IP address unless you enable proxy ARP for the virtual IP address. In other words, by enabling proxy ARP, a local interface can respond to the ARP requests on behalf of the virtual IP address. Otherwise, remote systems must have a route defined to reach the address. You can now configure virtual IP proxy ARP for a virtual IP interface while it is active.

In the preceding example, the workstations all point to one of the 10.3.2 interfaces on the system as their next hop gateway. When a packet arrives at the system, it goes through the packet processing. If the destination address matches any address defined on the system (including virtual IP addresses), the system processes the packet.

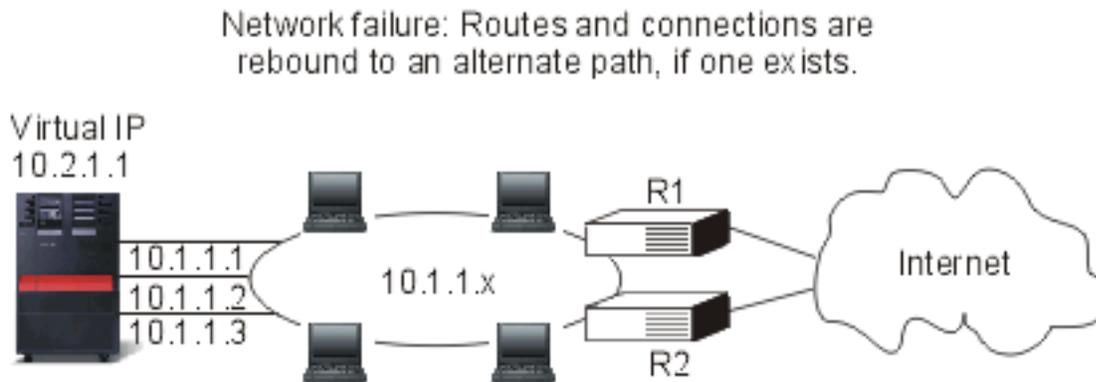
The Domain Name System (DNS) servers use the addresses of the requested system. In this case, all the addresses represent the same system. You can use the virtual IP function when consolidating multiple systems into one larger system.

- | Virtual IP address support now includes IPv6 addresses.

Fault tolerance

Another use of virtual IP addresses is to protect against router failures. Fault tolerance shows several different ways a route might be recovered after an outage.

This example shows several different ways a route can be recovered after an outage. The most reliable connection is when a virtual IP address is defined on the system. With virtual IP's support, even if an interface fails, the session can still communicate using different interfaces.



What happens if router R1 fails

- Connections through R1 are rerouted through R2.
- The failed gateway will detect R1 recovery, but active connections will continue to run through R2.

What happens if interface 10.1.1.1 fails

- Active connections to 10.1.1.1 are lost, but other connections to 10.1.1.2, 10.1.1.3, and 10.2.1.1 remain.
- Route rebinding:
 - Pre-V4R2: Indirect routes are rebound to 10.1.1.2 or 10.1.1.3.
 - V4R2: Routes are rebound only if Preferred Binding Interface is set to NONE.
 - V4R3 and higher: You need to define 10.2.1.1 as the virtual IP address and primary system address.
 - The primary IP address of the system remains active.
 - The system stays accessible as long as at least one physical interface remains active.

- | A Point-to-Point Protocol (PPP) interface or a Layer Two Tunneling Protocol (L2TP) interface can now use
- | a virtual IP address as the local IP address to provide fault tolerance for remote connections.

Routing with network address translation

Routing with network address translation (NAT) allows you to access remote networks, such as the Internet, while protecting your private network by masking IP addresses that are used on the private network.

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.

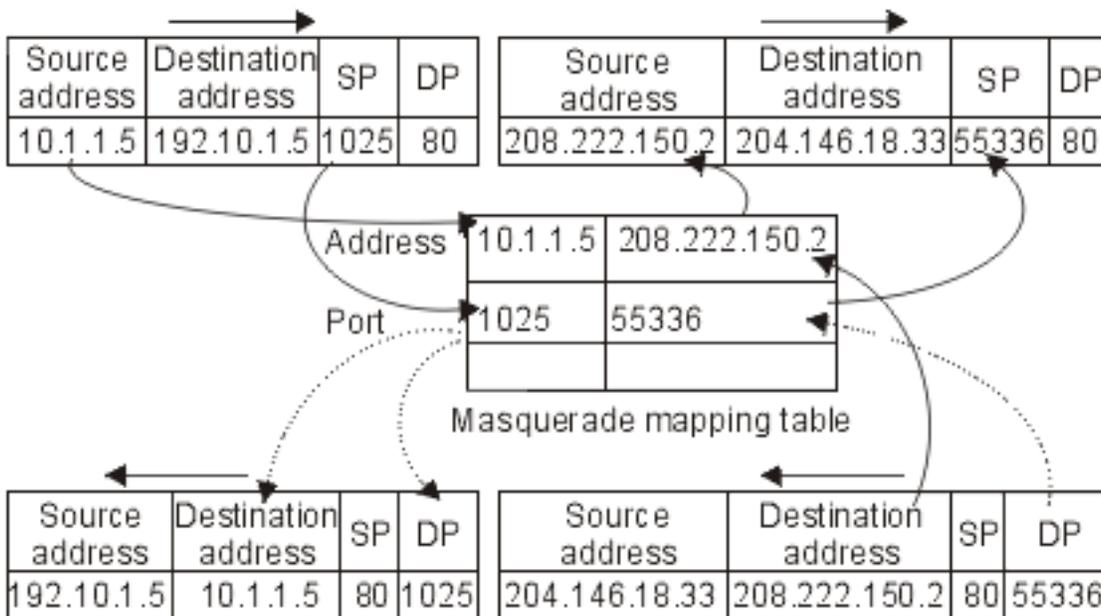
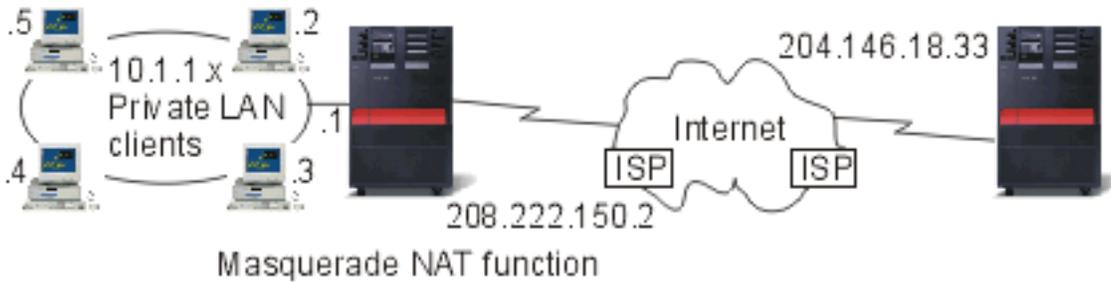
Masquerade NAT

Masquerade NAT is used to allow your private network to hide behind, as well as be represented by, the address bound to the public interface.

In many situations, the address bound to the public interface is the address that has been assigned by an Internet service provider (ISP), and the address can be dynamic in the case of a Point-to-Point Protocol

(PPP) connection. This type of translation can only be used for connections originating within the private network destined for the outside public network. Each outbound connection is maintained by using a different source IP port number.

Masquerade NAT allows workstations with private IP addresses to communicate with hosts on the Internet using the i5/OS operating system. i5/OS has an IP address assigned by the local ISP as its Internet gateway. The term *locally attached system* refers to all systems on an internal network regardless of the method of attachment (local area network or wide area network) and regardless of the distance of the connection. The term *external systems* refers to systems located on the Internet. The following figure illustrates how masquerade NAT works.



To the Internet, all of your workstations appear to be contained within your system; that is, only one IP address is associated with both your system and your workstations. When a router receives a packet intended for your workstation, it attempts to determine what address on the internal LAN should receive the packet and sends it there.

Each workstation must be set up so that i5/OS is its gateway and also its default destination. The correspondence between a particular communication connection (port) and a workstation is set up when one of your workstations sends a packet to i5/OS to be sent to the Internet. The masquerade NAT function saves the port number so that when it receives responses to your workstation's packet over that connection, it can send the response to the correct workstation.

A record of active port connections and the last access time by either end of the connection is created and maintained by masquerade NAT. These records are periodically purged of all connections that are idle for a predetermined amount of time based on the assumption that an idle link is no longer in use.

All communication between your workstation and the Internet must be initiated by locally attached systems. This is an effective security firewall; the Internet knows nothing of the existence of your workstations, and it cannot broadcast those addresses to the Internet.

A key to masquerade NAT implementation is the use of logical ports, issued by masquerade NAT to distinguish between the various communication streams. TCP contains a source and a destination port number. To these designations, NAT adds a logical port number.

Inbound masquerade NAT processing (response and other):

This process, which is the partner of outbound masquerade NAT processing, unfolds the corresponding outbound message to get right source workstation information.

About this task

The inbound message in the previous figure is a packet from the Internet to your private LAN. For inbound datagrams, the destination port number is the local port number. (For inbound messages, the source port number is the external port number. For outbound messages, the destination port number is the external port number.)

Response messages returning from the Internet bound for a locally attached system have a masquerade-assigned logical port number as the destination port number in the transport layer header. The masquerade NAT inbound processing steps are:

1. Masquerade NAT searches its database for this logical port number (source port). If it is not found, the packet is assumed to be an unsolicited packet, and the packet is returned to the caller unchanged. It is then handled as a normal unknown destination.
2. If a matching logical port number is found, a further check is made to determine that the source IP address matches the destination IP address of the existing logical port number table entry. If it matches, the original local system's port number replaces the source port in the IP header. If the check fails, the packet is returned unchanged.
3. The local matching IP addresses are placed in the packet IP destination.
4. The packet is then processed, as usual by IP or TCP, and ends up at the correct locally attached system. Because masquerade NAT requires a logical port number to determine the correct source and destination port addresses, masquerade NAT is incapable of handling unsolicited datagrams from the Internet.

Outbound masquerade NAT processing:

This process replaces the source port of an outbound message with a unique logical port number when the message is sent from the private LAN to the Internet.

About this task

The outbound message in the previous figure is a packet from the private LAN to the Internet. An outbound message (local to external) contains the source port used by the originating workstation. NAT saves this number and replaces it in the transport header with a unique logical port number. For outbound datagrams, the source port number is the local port number. The masquerade NAT outbound processing steps are:

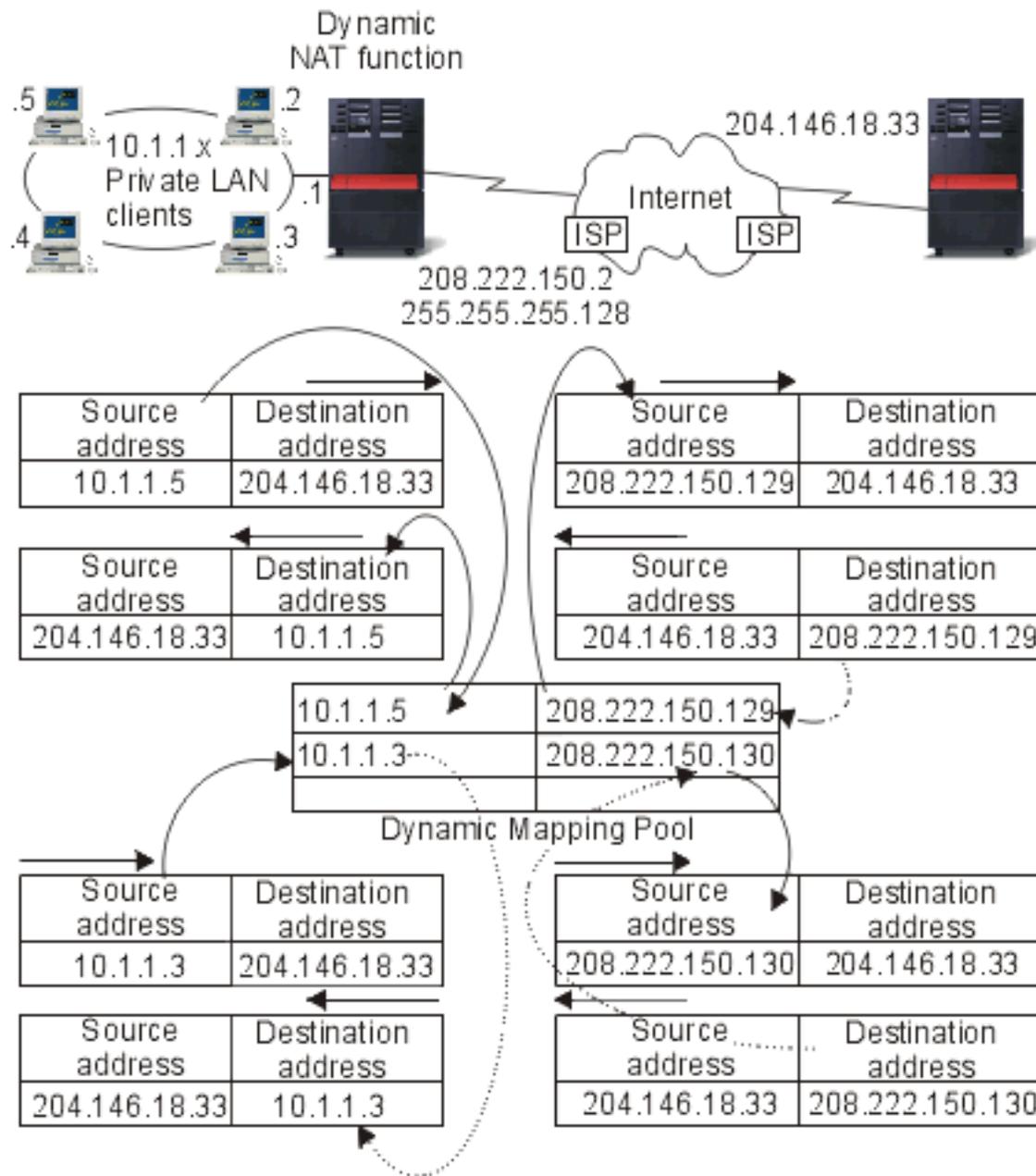
1. Outbound masquerade NAT processing assumes that all IP packets it receives are bound for external IP addresses, and therefore does not check to determine whether a packet should be routed locally.
2. The set of logical port numbers searches for a match on the transport layer as well as a source IP address and source port. If found, the corresponding logical port number is substituted for the source port. If no matching port number is found, a new one is created, and a new logical port number is selected and substituted for the source port.

3. The source IP address is translated.
4. The packet is then processed as usual by IP and is sent to the correct external system.

Dynamic NAT

Dynamic NAT can only be used to establish connections from within the private network out to the public network.

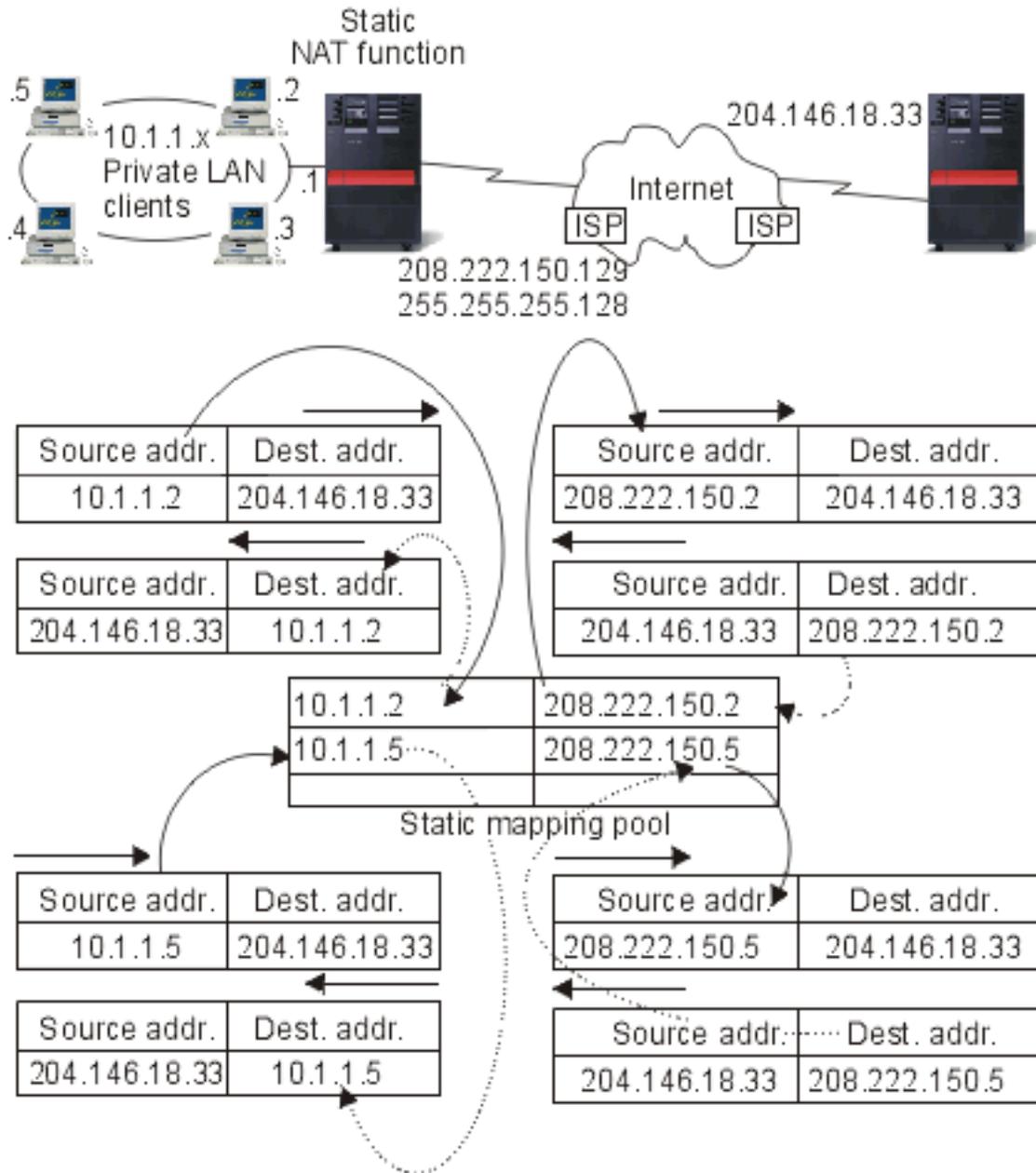
A pool of network addresses is maintained and used when an outbound connection is made. Each connection is assigned a unique public address. The maximum number of simultaneous connections is equal to the number of public addresses in the pool. This is similar to a one-to-one correspondence between addresses. Dynamic NAT allows you to communicate with the Internet through a dynamic NAT address. The following figure illustrates dynamic NAT.



Static NAT

Static NAT can use inbound connections from a public network into a private network.

Static NAT is a simple one-to-one mapping of private and public addresses. This is required to support inbound connections from your public network into your private network. For each local address defined, there has to be an associated globally unique address.



Related concepts

“DNS-based load balancing” on page 26

You can use DNS-based load balancing for your inbound workload. If load balancing is needed for local clients, use DNS load balancing.

Routing with OptiConnect and logical partitions

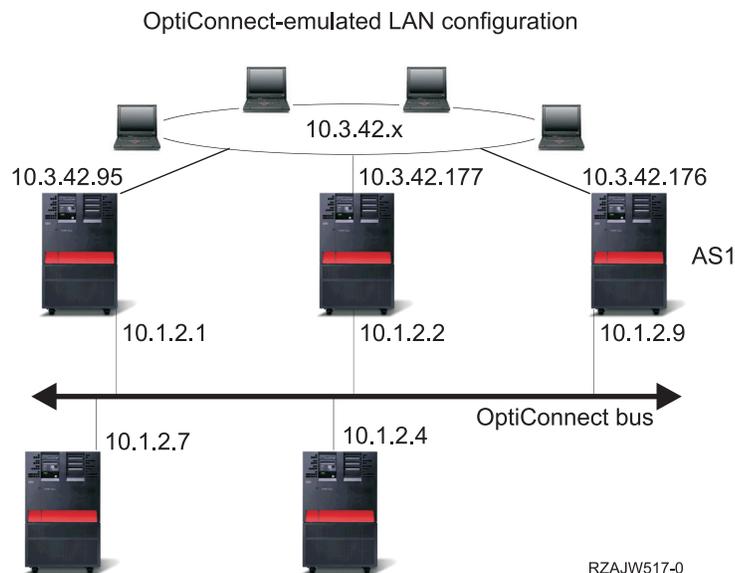
OptiConnect can connect multiple System i platforms by using a high-speed, fiber-optic bus. OptiConnect and logical partitions provide other environments for you to use the routing basics of proxy ARP, point-to-point, and virtual IP interfaces.

TCP/IP and OptiConnect

You can define TCP/IP connections over an OptiConnect bus. TCP/IP over OptiConnect provides another method for the routing building blocks, such as proxy ARP, unnumbered point-to-point networks, and virtual IP interfaces.

You can configure TCP/IP over OptiConnect using an OptiConnect-emulated LAN configuration or an OptiConnect point-to-point configuration.

With an **OptiConnect-emulated LAN configuration**, the OptiConnect bus appears as a LAN to TCP/IP, as is shown in the following figure. This is simple to configure, but the LAN OptiConnect connectivity is not automatic because it requires Routing Information Protocol (RIP) or static routes.



The **OptiConnect point-to-point configuration** uses point-to-point unnumbered interfaces that are configured for each pair of OptiConnect hosts. No new networks are created and so the LAN OptiConnect connectivity is automatic. One advantage of this configuration is that no additional route definitions are required. Connectivity between a host on one network to hosts on another network is automatic. Another advantage is that if both networks are active, data sent between the systems flows over the OptiConnect bus because these routes have the most specific subnet mask. If the OptiConnect bus goes down, traffic is automatically switched to the token-ring LAN.

OptiConnect point-to-point configuration using virtual IP is a variation of the unnumbered point-to-point configuration. Whenever you use unnumbered, point-to-point interfaces, each interface must have an associated local interface specified. This is the IP address by which the system on the remote end of the point-to-point link knows the local system. This associated local interface might be the system's primary LAN interface, as shown in the previous figure. Or you can use a virtual IP interface as the associated local interface.

In the OptiConnect point-to-point configuration using virtual IP, you use the OptiConnect bus as a collection of point-to-point connections. You define an unnumbered connection for each pair of hosts. Like the OptiConnect point-to-point configuration, no additional route definitions are required, and

connectivity between a host on one network to hosts on the other network are automatic. An advantage of this configuration is that if either network is active, a path exists to reach any system running on the i5/OS operating system.

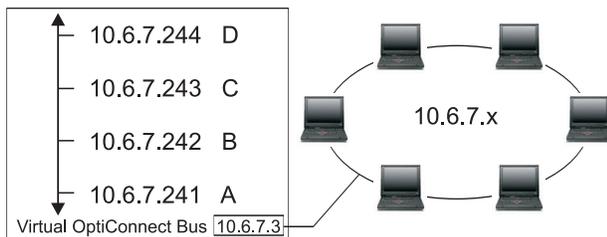
Routing with virtual OptiConnect and logical partitions

With logical partitions, a single system is logically partitioned into multiple virtual systems. Virtual OptiConnect TCP/IP interfaces are used as interpartition communication paths.

Each partition has its own address space, its own instance of TCP/IP, and might have its own dedicated I/O adapters. To TCP/IP, each partition appears as a distinct system. TCP/IP communication between the different partitions is done using a virtual OptiConnect bus. The TCP/IP routing code uses the path to another partition no differently than the path to another system connected by a physical OptiConnect bus.

Logical partitions: Virtual OptiConnect TCP/IP interfaces are used as inter partition communication paths.

Virtual OptiConnect network = 10.6.7.241 - 10.6.7.254
This provides addresses for up to 14 partitions



Partition	Interface	Line	Subnet mask	MTU
D	10.6.7.244	*OPC	255.255.255.240	4096
C	10.6.7.243	*OPC	255.255.255.240	4096
B	10.6.7.242	*OPC	255.255.255.240	4096
A	10.6.7.241	*OPC	255.255.255.240	4096
A	10.6.7.3	TRNLINE	255.255.255.0	4096

(Associated local interface = 10.6.7.3)

RZAJW515-0

In these examples, only one LAN adapter is installed in the system. It is allocated to partition A. The clients in the LAN need to communicate with the other partitions defined on the system. To do this, you define a transparent subnet on the virtual OptiConnect bus. The LAN has a network address of 10.6.7.x. You want to plan for additional partitions, so IP addresses are then needed. To get 12 addresses, you must use a subnet mask of 255.255.255.240. This gives you addresses 10.6.7.241 through 10.6.7.254, a total of 14 usable addresses. You must ensure that these addresses are not already in use on the LAN. After you get the addresses, you assign one to each partition. You add an interface to each partition and define the address on the virtual OptiConnect bus.

OPC	Partition	Virtual IP	Partition	Interface	Line	Subnet mask	MTU	Associated Local Interface
10.6.7.3	D	10.6.7.4	D	10.6.7.4	VIRTUALIP	255.255.255.255	4096	NONE
10.6.7.2			D	10.6.7.1	OPC	255.255.255.255	4096	10.6.7.4
10.6.7.1			D	10.6.7.2	OPC	255.255.255.255	4096	10.6.7.4
			D	10.6.7.1	OPC	255.255.255.255	4096	10.6.7.4
10.6.7.4	C	10.6.7.3	C	10.6.7.3	VIRTUALIP	255.255.255.255	4096	NONE
10.6.7.2			C	10.6.7.1	OPC	255.255.255.255	4096	10.6.7.3
10.6.7.1			C	10.6.7.2	OPC	255.255.255.255	4096	10.6.7.3
			C	10.6.7.4	OPC	255.255.255.255	4096	10.6.7.3
10.6.7.4	B	10.6.7.2	B	10.6.7.2	VIRTUALIP	255.255.255.255	4096	NONE
10.6.7.3			B	10.6.7.1	OPC	255.255.255.255	4096	10.6.7.2
10.6.7.1			B	10.6.7.3	OPC	255.255.255.255	4096	10.6.7.2
			B	10.6.7.4	OPC	255.255.255.255	4096	10.6.7.2
10.6.7.3	A	10.6.7.1	A	10.6.7.1	TRNLIN	255.255.255.0	4096	NONE
10.6.7.3			A	10.6.7.2	OPC	255.255.255.255	4096	10.6.7.1
10.6.7.2			A	10.6.7.3	OPC	255.255.255.255	4096	10.6.7.1
			A	10.6.7.4	OPC	255.255.255.255	4096	10.6.7.1

Transparent subnetting is automatically enabled when the following statements are true. First, the virtual OptiConnect bus is less than or equal to the size of the MTU on the real LAN interface. Second, the OptiConnect bus subnet is a subnet of the LAN network address. If both statements are true, then transparent subnetting is automatically enabled. The interface 10.6.7.3 performs a proxy for all the interfaces defined in the partitions. This allows clients on the LAN to connect to the partitions.

TCP/IP workload balancing methods

Workload balancing is redistributing network traffic and workload of heavily accessed systems across multiple processors, multiple interface adapters, or multiple host systems.

To get the best performance possible from the i5/OS operating system, put the communications workload on multiple parts of your system.

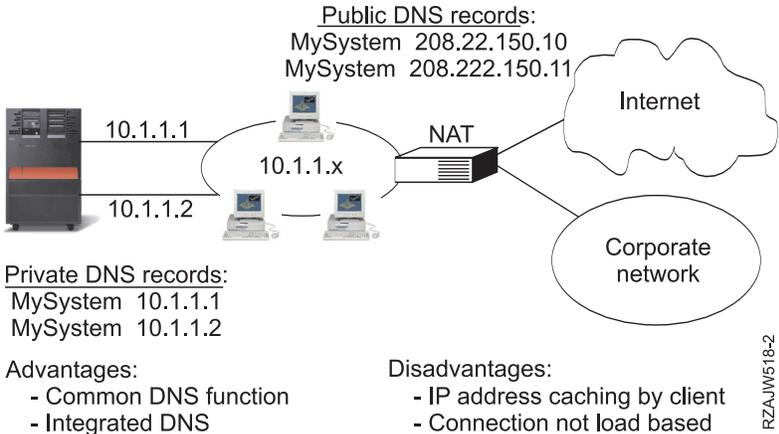
Several different TCP/IP routing methods can be used to balance the workload of your system.

DNS-based load balancing

You can use DNS-based load balancing for your inbound workload. If load balancing is needed for local clients, use DNS load balancing.

DNS-based load balancing is used for inbound load balancing. Multiple host IP addresses are configured in DNS for a single host system name. DNS alternates the host IP address returned to a successive client host name resolution request. An advantage to this type of load balancing is that it is a common DNS function. Disadvantages to this solution are that IP addresses can be cached by a client and that it is a connection-based solution, not a load-based solution.

The first way to achieve load balancing is to use a DNS function to pass out multiple addresses for the same system name. The DNS will serve a different IP address each time a request is made for the address record for your system name. In the following example, each address corresponds to a different system. This allows you to provide load balancing across two separate systems. In the case of clients on the private networks, they receive a different address for each request. This is a common DNS function. Notice that the public DNS also has two address entries. These addresses are translated using static NAT so that if you are on the Internet, you can reach the two systems.



If your programs depend on getting to a specific system or depend on returning to the same system after the initial connection, the Web pages and sites should be coded to send a different system name after the first contact is made. Additional DNS entries can be added for MyServer1 208.222.150.10 and MyServer2 208.222.150.11. By doing this, the Web sites, for example, can point to MyServer2 after the first contact. This type of load balancing provides balancing by the connection request. In most cases, after you have 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 system. Note that this type of load balancing only considers inbound traffic and also that you can have two adapters on one system rather than one adapter on two systems.

Related concepts

“Static NAT” on page 23

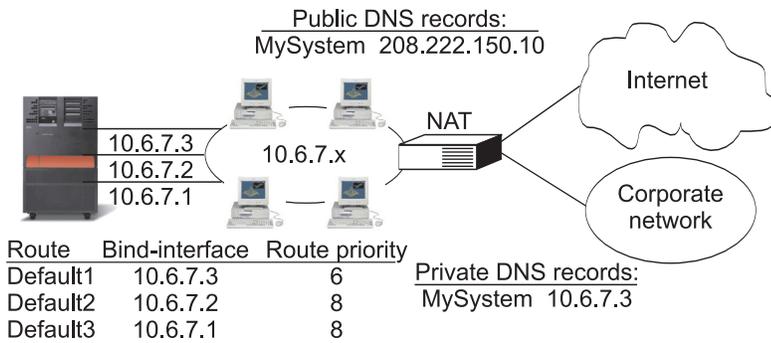
Static NAT can use inbound connections from a public network into a private network.

Duplicate route-based load balancing

You can use duplicate route-based load balancing to balance outbound workload across multiple interfaces.

This is a connection-based solution that has more flexibility than DNS-based load balancing, but it is not active for local clients. The advantages of using this type of load balancing are that it is a total i5/OS solution, it has more flexibility than DNS, and it is good for applications where most of the traffic is outbound, like HTTP and Telnet. The disadvantages to it are that it is a connection-based solution (not a load-based solution), it is not active for local clients, and it has no effect on inbound requests.

In the following example, three adapters on your system are all connected to the same LAN segment. You have set up one of the adapters as an inbound line only and set up the other two adapters as outbound. Local clients continue to work the same way as in the past. That is to say the outbound interface is the same as the inbound interface. Remember that a local client is any system that does not require a router to reach it. This can be a very large network if switches were used rather than routers.



Duplicate, indirect routes, with priority >(5) default will be selected according to route priority

Advantages:

- More flexibility than DNS
- Good for HTTP, Telnet

Disadvantages:

- Connection based, not load based
- Not active for local clients
- No effect on inbound requests

RZAJW511-2

You can configure duplicate route-based load balancing with the Add TCP/IP Route (ADDTCPRTE) command or with the System i Navigator interface. It is accomplished by setting either the duplicate route priority or the preferred binding interface. If the value for the duplicate route priority is left at the default value of 5, nothing happens. If a value greater than 5 is set, then connections are distributed between routes at the same priority. The preferred binding interface is used to bind a route to a specific interface by IP address.

In the preceding example, there is an "inbound" adapter (10.6.7.3) with a duplicate route priority of 6. The other two adapters are configured with a duplicate route priority of 8. Because the duplicate route priority on one adapter is 6, it will not be selected for an outbound connection unless all the single route priority interfaces of 8 are down.

You should put all the outbound interfaces at the same priority. If you put some at one value and some at another value, only the highest value interfaces will be used.

Notice that the DNS is pointing to the 10.6.7.3 interface, making it the inbound interface. Even if you decide not to use duplicate route priority, you should always define a default route out of the system on each interface by using the preferred binding interface parameter.

| Load balancing using virtual IP and proxy ARP

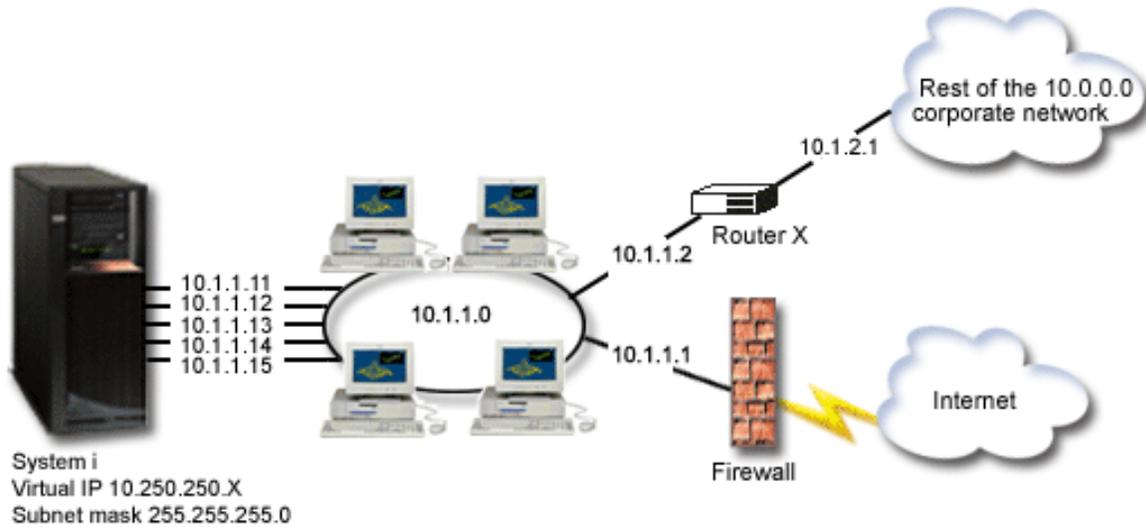
| You can use virtual IP and proxy ARP to achieve load balancing across multiple interfaces. This workload balancing method supports both inbound and outbound workload.

| These are the advantages of using virtual IP and proxy ARP as the workload balancing method:

- | • It supports both inbound and outbound workload.
- | • It supports local clients.
- | • It provides more flexibility than the DNS-based and duplicate route-based load balancing methods.

| The disadvantage of this workload balancing method is that it is a connection-based solution, not a load-based solution. The load on each interface is not considered. The assumption is that the traffic load is similar for all connections.

| The following example takes full advantage of using virtual IP addresses. In addition to binding a unique virtual IP address to each application, this example provides inbound and outbound connection balancing and some level of fault tolerance.



i5/OS TCP/IP route entries				
Destination	Subnet mask	Next hop	Preferred binding interface	Duplicate route priority
10.1.1.0	255.255.255.0	10.1.1.11	10.1.1.11	6
10.1.1.0	255.255.255.0	10.1.1.12	10.1.1.12	6
10.1.1.0	255.255.255.0	10.1.1.13	10.1.1.13	7
10.1.1.0	255.255.255.0	10.1.1.14	10.1.1.14	7
10.1.1.0	255.255.255.0	10.1.1.15	10.1.1.15	7
10.0.0.0	255.0.0.0	10.1.1.2	10.1.1.11	6
10.0.0.0	255.0.0.0	10.1.1.2	10.1.1.12	6
10.0.0.0	255.0.0.0	10.1.1.2	10.1.1.13	7
10.0.0.0	255.0.0.0	10.1.1.2	10.1.1.14	7
10.0.0.0	255.0.0.0	10.1.1.2	10.1.1.15	7
*dftroute	*none	10.1.1.1	10.1.1.11	6
*dftroute	*none	10.1.1.1	10.1.1.12	6
*dftroute	*none	10.1.1.1	10.1.1.13	7
*dftroute	*none	10.1.1.1	10.1.1.14	7
*dftroute	*none	10.1.1.1	10.1.1.15	7

X

Virtual IP	Application
10.250.250.1	SYSNAME
10.250.250.2	HTTPSVR1
10.250.250.2	HTTPSVR2
10.250.250.11	DOM1
10.250.250.12	DOM2
10.250.250.13	DOM3

Y

Router X routing table		
Destination	Subnet mask	Next hop
10.250.250.0	255.255.255.0	10.1.1.11
10.250.250.0	255.255.255.0	10.1.1.12

Z

Advantages:

- Effective for both inbound and outbound workload.
- Effective for local clients.
- More flexibility than the DNS-based and duplicate route-based load balancing methods.

Disadvantage:

- Connection-based, not load-based.

Figure 3. Load balancing using virtual IP and proxy ARP

In this example, inbound connection balancing is achieved by using virtual IP addresses defined on the system and by using the external router, firewall, and switch that can perform layer three (network layer) routing. Outbound connection balancing is achieved by using the preferred binding interface and

| duplicate route priority parameters on the i5/OS TCP/IP route entries. Outbound connections are
| distributed in a round-robin fashion between all the interfaces at the same duplicate route priority when
| the value for the duplicate route priority is set to be greater than the default value of 5. If all the
| interfaces at one value become unavailable, the system switches to the interfaces at the next lower value.

| According to the route directives that are configured on router X, interfaces 10.1.1.11 and 10.1.1.12 are set
| up as the primary inbound interfaces. Inbound connections are distributed in a round-robin fashion
| between interfaces 10.1.1.11 and 10.1.1.12, which is a function that most routers provide.

| According to the i5/OS TCP/IP route entries, interfaces 10.1.1.13, 10.1.1.14, and 10.1.1.15 with a duplicate
| route priority of 7 are set up as the primary outbound interfaces. Outbound connections are distributed
| in a round-robin fashion between interfaces 10.1.1.13, 10.1.1.14, and 10.1.1.15. If all of these three
| interfaces are down, interfaces 10.1.1.11 and 10.1.1.12 with a duplicate route priority of 6 are used for
| both the outbound and inbound connections.

| In this example, the i5/OS TCP/IP route entries consist of three groups. Group X provides outbound
| connection balancing to the local segment of the corporate network (10.1.1.0). Group Y provides outbound
| connection balancing to the rest of the corporate network (10.0.0.0) through the router. Group Z provides
| outbound connection balancing to the Internet through the firewall.

| **Related concepts**

| “Scenario: Adapter failover using virtual IP and proxy ARP”

| Virtual IP addresses allow you to assign an address to the system rather than to a specific interface.

| You can define the same address to multiple systems, which allows many new options for load
| balancing.

Scenario: Adapter failover using virtual IP and proxy ARP

Virtual IP addresses allow you to assign an address to the system rather than to a specific interface. You can define the same address to multiple systems, which allows many new options for load balancing.

Note: This failover scenario is referring to a single LAN adapter rather than a major type of system outage like clustering would cover. This solution requires you to have an external load balancing system.

Situation

Your production system handles data entry from both remote and LAN clients. It has the company’s critical application on it. As the company has grown, so has its demand on the System i hardware and the network. Because of the growth, it has become imperative that this system be available on the network without an unscheduled downtime. If, for any reason, a network adapter becomes unavailable, other network adapters on the system should take over and the network clients should be unaware of any failures.

Objectives

The concept of availability has many different aspects of redundancy and backup for failing components. In this scenario, the goal is to provide network availability to the system for its clients in the event of an adapter failure.

Details

One way to handle the preceding situation is to have multiple physical connections to the LAN from the System i platform. Consider the following figure.

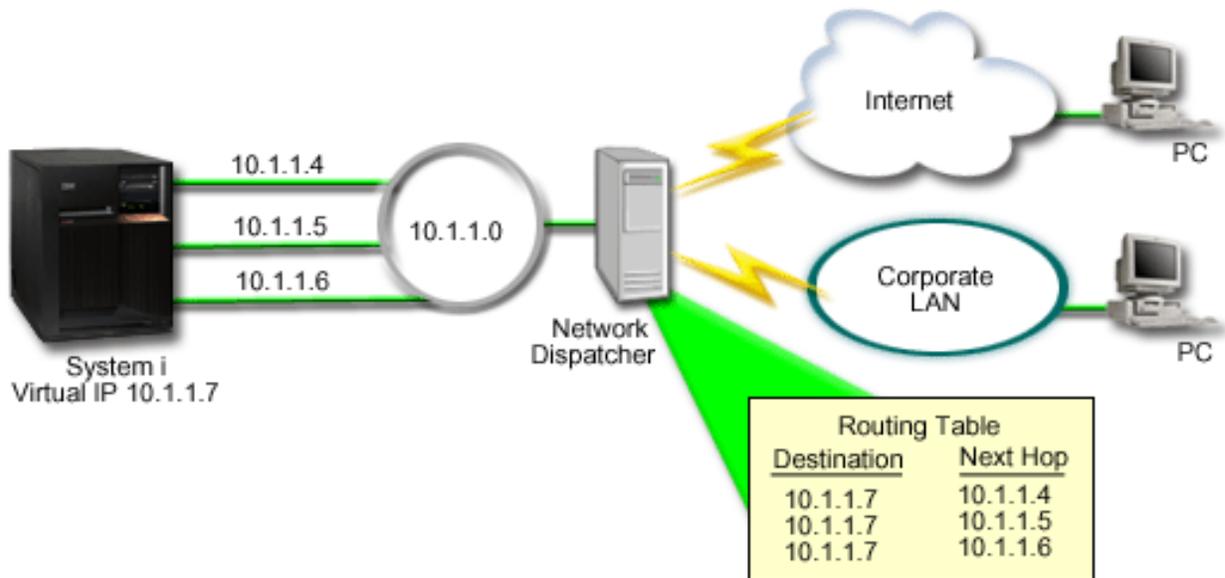


Figure 4. Adapter failover without local clients

Each physical connection has a different IP address. Then you can assign a virtual IP address to the system. This virtual IP address is the IP address by which all of its clients recognize it. All remote clients (clients that are not physically attached to the same LAN as the System i platform) communicate with the system through an external load balancing server such as a network dispatcher. When IP requests from remote clients go through the network dispatcher, the network dispatcher routes the virtual IP addresses to one of the network adapters on the system.

If the LAN that the system is connected to has clients, these clients will not use the network dispatcher to direct their locally bound traffic because that unnecessarily overloads the network dispatcher. You can create route entries on each client that are similar to the route tables in the network dispatcher. However, this is impractical if the LAN has a large number of local clients. This situation is shown in the following figure.

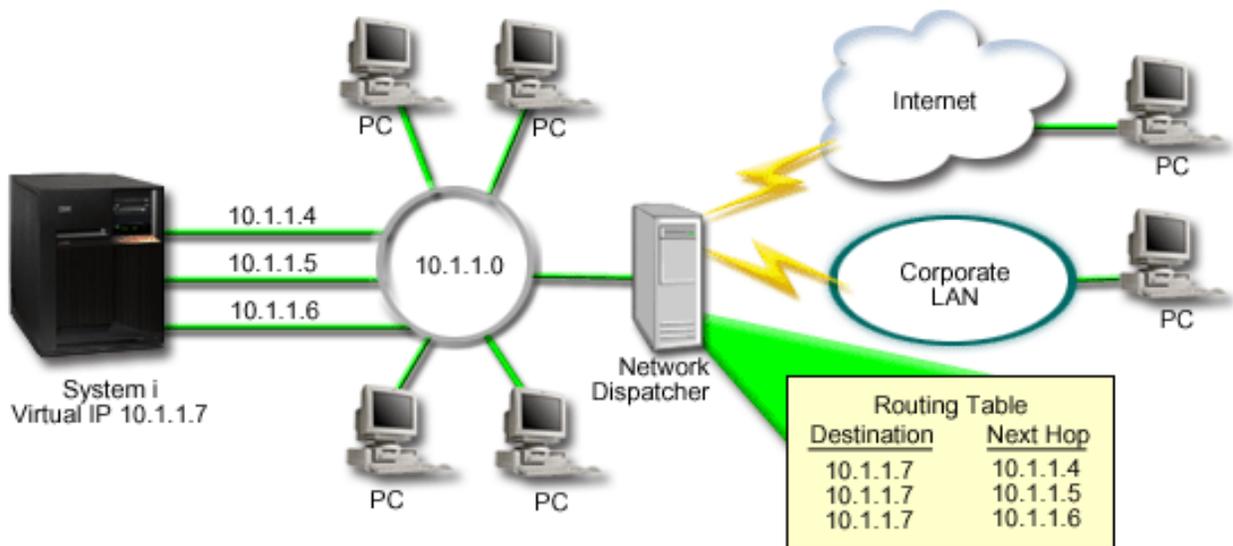


Figure 5. Adapter failover with local clients

Local clients (clients that are attached to the same LAN as the system) can connect to the virtual IP address of the system through ARP. This allows local clients to have an adapter failover solution as well.

In each case, neither local clients nor remote clients are aware of the failover when it occurs. The system chooses which adapters and IP addresses are the preferred interface for virtual IP address (VIPA) Proxy Address Resolution Protocol (ARP) agent selection.

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 if an adapter failure occurs. A preferred interface list is an ordered list of the interface addresses that take over for the failed adapters. You can use either System i Navigator or the Change TCP/IP IPv4 Interface (QTOCC4IF) application programming interface (API) to configure a preferred interface list. The preferred interface list is also configurable for both virtual Ethernet and virtual IP address interfaces.

Using Figure 2 as an example, remote clients are communicating with the local system using virtual IP address 10.1.1.7. Suppose 10.1.1.4 is the initial local adapter being used for this communication, and you want 10.1.1.5 to take over if 10.1.1.4 fails. You also want interface 10.1.1.6 to take over if both adapters for 10.1.1.4 and 10.1.1.5 fail. To control the order in which these interfaces are used in a failover situation, you can define a preferred interface list for virtual IP address 10.1.1.7. In this case, it is an ordered list of interface addresses that consists of 10.1.1.4, 10.1.1.5, and 10.1.1.6.

The solution can also involve using two or more System i platforms to support each other. If one of the systems becomes unavailable, then the second system can serve as a failover. The following figure shows the same setup using two systems.

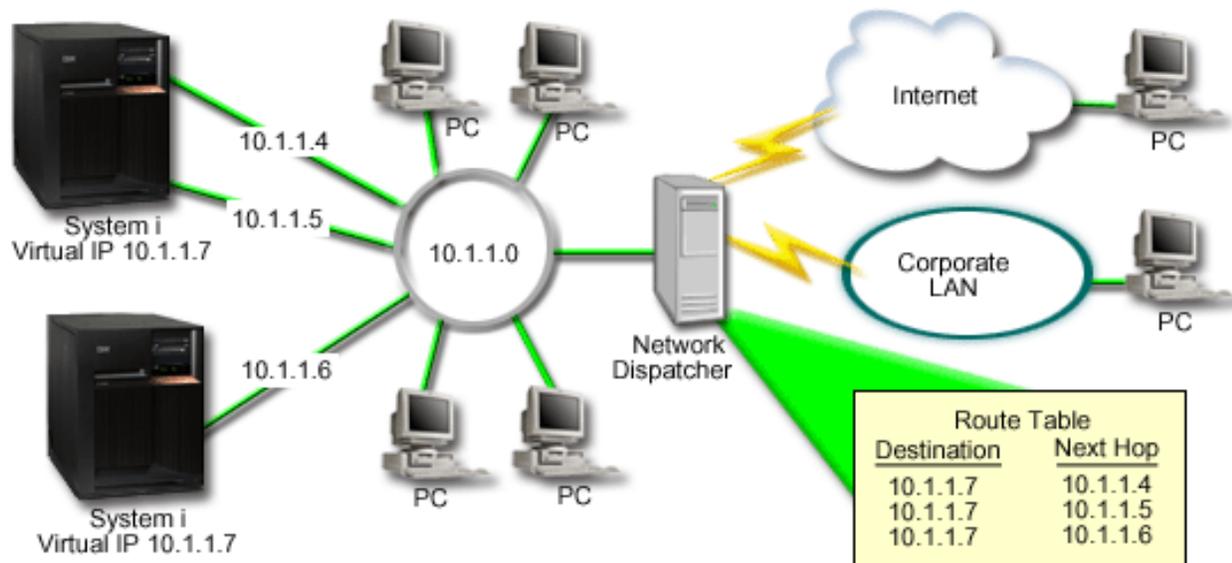


Figure 6. Adapter failover with multiple System i platforms and local clients

The packet routing is the same as routing for a single system and its remote clients; however, there is a distinct difference for the local clients. If you have multiple systems using the same virtual IP address, you can only use proxy for one of the systems. In this case, you want the system with the two LAN connections to serve as the proxy.

Configuration steps

The configuration for load balancing using virtual IP and proxy ARP is similar to standard TCP/IP configurations with the addition of a virtual TCP/IP interface.

Related concepts

“Load balancing using virtual IP and proxy ARP” on page 28

You can use virtual IP and proxy ARP to achieve load balancing across multiple interfaces. This workload balancing method supports both inbound and outbound workload.

Failover using automatic interface selection

Use these steps to configure virtual IP and proxy ARP for adapter failover situations in this scenario.

About this task

Using Figure 2 as an example, the general configuration steps would be:

1. Configure a virtual TCP/IP interface.

Using System i Navigator, create a virtual TCP/IP interface. The new Virtual IP interface wizard can be found at: **Network** → **TCP/IP Configuration** → **IPv4** → **Interfaces**. Then, right-click **Interfaces** and select **New Interface** → **Virtual IP**.

For our example, enter an IP address of 10.1.1.7 with a subnet mask of 255.255.255.255. After you create the virtual interface, right-click the interface and select **Properties**. Click the **Advanced** tab and select the **Enable proxy ARP** check box.

2. Create TCP/IP interfaces for all of your physical LAN connections.

Use the Create TCP/IP interface wizard to create your TCP/IP interfaces. The wizard is in System i Navigator and can be found at: **Network** → **TCP/IP Configuration** → **IPv4** → **Interfaces**. Then, right-click **Interfaces** and select **New Interface** → **Local Area Network**. Complete the wizard for each of your LAN connections.

For this example, you will run the wizard three times entering the IP addresses of 10.1.1.4, 10.1.1.5, and 10.1.1.6 with a subnet mask of 255.255.255.0. After you complete each interface, right-click the interface and select **Properties**. Click the **Advanced** tab and select the **Associated local interface** check box to associate the interface with the virtual IP interface that you created in step 1.

Failover using a preferred interface list

You can create a preferred interface list to control the order in which the local interfaces are used when an adapter failure occurs.

About this task

To create a preferred interface list, follow these steps:

1. In System i Navigator, expand **Network** → **TCP/IP Configuration** → **IPv4**.

2. Click **Interfaces**.

3. From the lists of interfaces that are displayed, select an interface for the virtual IP address or virtual Ethernet for which you want to create the preferred interface list.

Using Figure 2 as an example, select the virtual IP address 10.1.1.7.

4. Right-click the interface, and then select **Properties**.

5. Click the **Advanced** tab.

6. In the panel, select the interface addresses from the Available interface list, and click **Add**.

Using Figure 2 as an example, select interfaces 10.1.1.4, 10.1.1.5, and 10.1.1.6, and add them to the preferred interface list one by one.

You can also remove an interface from the preferred interface list in the right pane by using the **Remove** button, or move an interface up and down the list to change the order by using the **Move up** and **Move down** buttons.

7. Select the **Enable proxy ARP** check box above the Available interfaces list to enable the list.

8. Click **OK** to save the preferred interface list that you have just created.

| **Note:** You can only include 10 interfaces in the preferred interface list. If you configure more than 10,
| the list is truncated to the first 10.

Related information for TCP/IP routing and workload balancing

Other information center topic collections contain information that relates to the TCP/IP routing and workload balancing topic collection.

Other information

- Domain Name System
DNS is an advanced system for managing the host names that are associated with Internet Protocol (IP) addresses on TCP/IP networks. Here you will find basic concepts and procedures that you need to know to configure and administer DNS.
- Logical partitions
This topic collection provides you with more background information and detail.
- IP filtering and network address translation
Information in this topic collection helps you manage your filter rules. Some of the functions include adding comments, editing, and viewing.
- OptiConnect
This topic collection provides you with information about OptiConnect routing.
- Remote Access Services: PPP connections
Point-to-Point Protocol (PPP) is commonly used to connect a computer to the Internet. PPP is an Internet standard and is the most widely used connection protocol among Internet service providers (ISPs).

Related reference

“PDF file for TCP/IP routing and workload balancing” on page 2
You can view and print a PDF file of this information.

Appendix. 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 grant 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.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

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.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation

Software Interoperability Coordinator, Department YBWA
3605 Highway 52 N
Rochester, MN 55901
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, IBM License Agreement for Machine Code, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

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.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Programming interface information

This TCP/IP routing and workload balancing publication documents intended Programming Interfaces that allow the customer to write programs to obtain the services of IBM i5/OS.

Trademarks

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

i5/OS
IBM
IBM (logo)
System i

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

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

Terms and conditions

Permissions for the use of these publications is granted subject to the following terms and conditions.

Personal Use: You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative works of these publications, or any portion thereof, without the express consent of IBM.

Commercial Use: You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.



Printed in USA