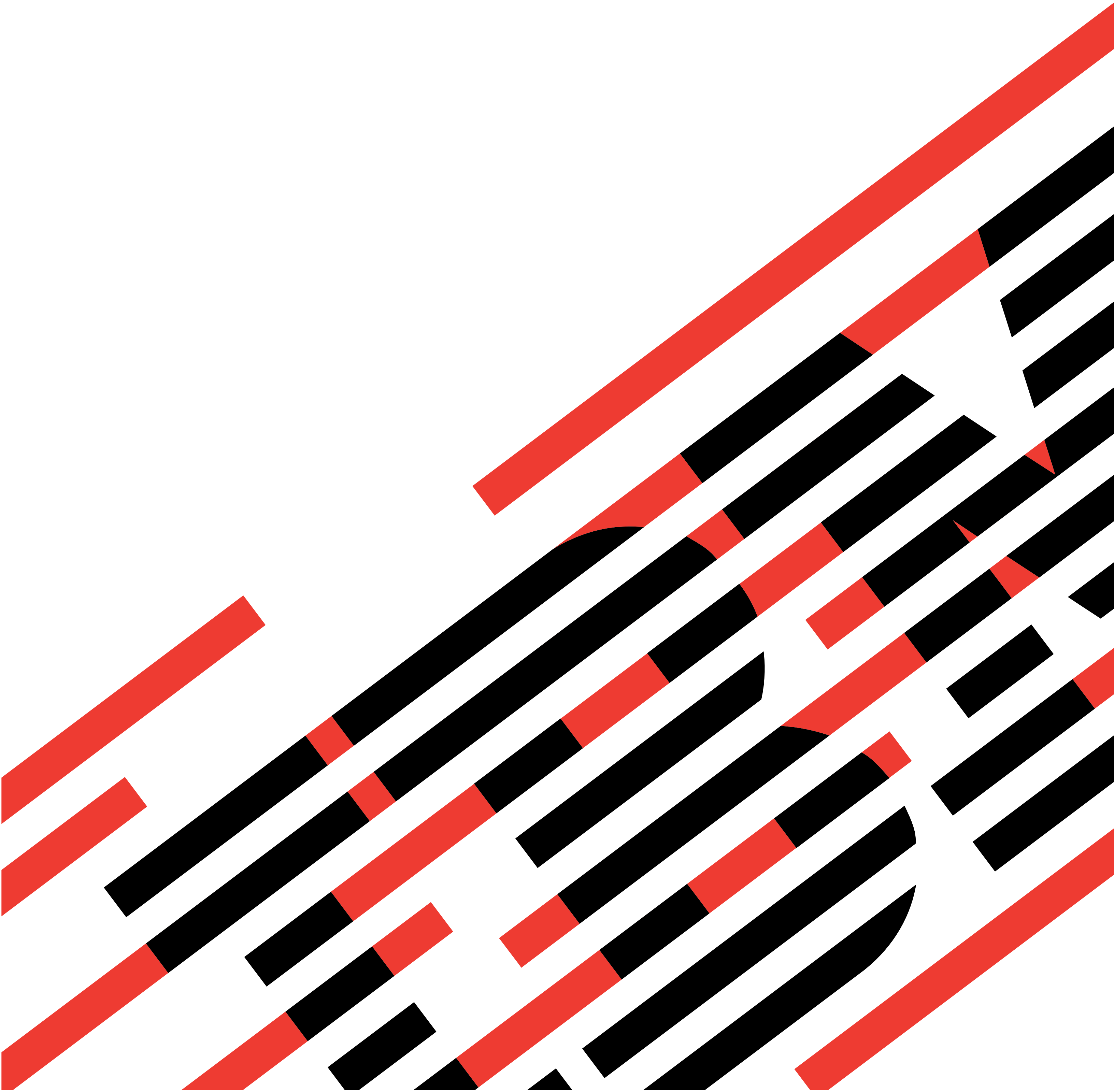
IBM

eserver

iSeries

TCP/IP routing and workload balancing



iSeries

TCP/IP routing and workload balancing

Contents

TCP/IP routing and workload balancing

Are you looking for better ways to route and balance the TCP/IP traffic of your iSeries(TM) server? Your iSeries server can be used for many things, but you should also know that its integrated routing capabilities can eliminate the need for an external router by connecting TCP/IP networks.

The routing and workload balancing methods as well as the background information will help you understand the options available for you to use on your iSeries server. 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 these routing techniques. The focus in these pages is on the routing principles and concepts you should know so your iSeries server works better for you.

Why are these methods important to you?

The techniques in these methods may cut down the overall cost of your connections because you could use fewer external routers and servers. Using these routing methods, you may free up IP addresses because you will learn how to manage them in a more effective way. By reading the workload balancing methods, you may get better overall iSeries server performance by balancing the communications workload on your system.

What if I want to print these pages?

You can easily print this topic and read it as one document. Just follow the instructions in Print this topic.

Before you start

If you are new to routing and workload balancing on iSeries server, you may want to look at these pages before you begin looking at the methods:

TCP/IP routing functions by release contains information on the routing functions available on each version and release of the iSeries server so that you know which functions are available for you to use.

Packet processing shows you how a packet of information is processed by your iSeries server.

General routing rules gives you some basic rules for iSeries server routing. You should consider these rules while you are reading the routing methods.

How do I know which method to use?

Many different methods are available to you. You can make your own decisions and apply these methods to your own network situation:

TCP/IP routing connectivity methods give you a better understanding of how your iSeries server is capable of routing data.

TCP/IP workload balancing methods help you understand different TCP/IP techniques that can be used to balance the communications workload of your iSeries server.

Want more information on iSeries server TCP/IP routing?

Other information about TCP/IP routing and workload balancing contains additional reference information related to TCP/IP routing and workload balancing.

Print this topic

You can view or download a PDF version of this document for viewing or printing. You must have Adobe(R) Acrobat(R) Reader installed to view PDF files. You can download a copy from the Adobe(R) Acrobat(R) Web site.

To view or download the PDF version, select Routing and workload balancing

(about 1.8 MB or 36 pages).

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

1. Open the PDF in your browser (click the link above).
2. In the menu of your browser, click **File**.
3. Click **Save As**.
4. Navigate to the directory in which you would like to save the PDF.
5. Click **Save**.

TCP/IP routing functions by release

The following list shows the functions supported by the release on iSeries server. Before you plan to use a function, check here to make sure that your system is at the correct release to support the function that you want to perform. In some cases, however, you can use a different approach to achieve the same results.

V3R1: Static route-based packet forwarding is introduced.

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).

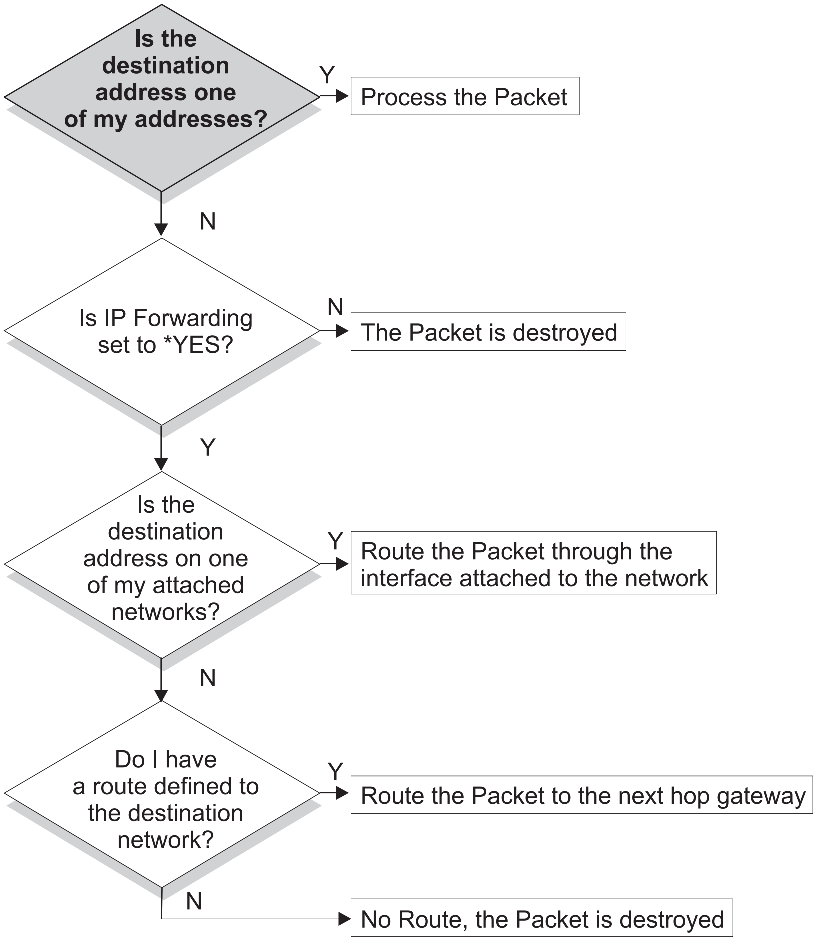
V4R2: Dynamic Routing Information Protocol Version 2 (RIPv2), transparent subnetting, and duplicated route-based load-balancing.

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

V4R4: IP over OptiConnect.

Packet processing

Having a better understanding of a packet process helps you decide how to implement routing functions. The simplified flow chart below shows the logical process that takes place when an IP packet (datagram) reaches your iSeries server. The actual flow may 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 may be slightly different.


Is the destination address one of my addresses?
Y
Process the Packet
N
Is IP Forwarding set to *YES?
N
The Packet is destroyed
Y
Is the destination address on one of my attached networks?
Y
Route the Packet through the interface attached to the network
N
Do I have a route defined to the destination network?
Y
Route the Packet to the next hop gateway
N
No Route, the Packet is destroyed

RZAJW523-0

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 was 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 the iSeries server is not directly connected to the remote host, the routing table is searched. Once again this is done from the most specific host (subnet mask 255.255.255.255) to the least specific different 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 are some of the basic rules that apply to TCP/IP in general and to TCP/IP on your iSeries server. You should consider these rules as you implement routing functions on your iSeries server. These rules will help you determine what is happening to the packets on your system and where they may be going. As with most rules, there are exceptions.

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

   The exception to this rule is Virtual IP (connectionless) addresses, which are assigned to the system. Virtual IP is available starting in V4R3.

2. 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 may be forwarded or discarded.

3. The IP address and mask define the address of the attached network.

4. 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 may 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. This page provides links to conceptual information on routing methods for you to consider using on your iSeries server.

- Routing with point-to-point connections
  You can get your data from your local system to a remote system or from a local network to a remote network with point-to-point connections. This explains two concepts used in configuring IP addresses for a point-to-point connection.
- Proxy ARP 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 also contains a description of transparent subnets, which is an extension to the proxy ARP routing technique.
- Dynamic routing
  Dynamic routing is a low maintenance method that automatically reconfigures routing tables as your network changes.
- Route binding
  Route binding gives you control over which interface is used to send out response packets of information.
- Classless Inter-Domain Routing (CIDR)
  Classless Inter-Domain Routing can reduce the size of your routing tables and make more IP addresses available within your business.
- Routing with virtual IP
  Virtual IP provides a way to assign one or more addresses to the system without the need of

binding the address to a physical interface. This can be used when you want to run multiple occurrences of a Domino Web server bound to different addresses or other services that need to bind to default ports.

- Fault tolerance
  Fault tolerance shows several different ways a route may be recovered after an outage.
- Routing with network address translation (NAT)
  Routing with NAT lets you access remote networks, such as the Internet, while protecting your private network by masking IP addresses that are used on the private network. This page discusses the kinds of NAT that iSeries server supports and why you may want to use them.
- Routing with OptiConnect and logical partitions
  OptiConnect can connect multiple iSeries servers by using a high-speed, fiber-optic bus. This information covers using OptiConnect with logical partitions and the advantages of using them.

Routing with point-to-point

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 on how to set up and manage your PPP connections.

You can use point-to-point connections across dial-up lines, leased 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:

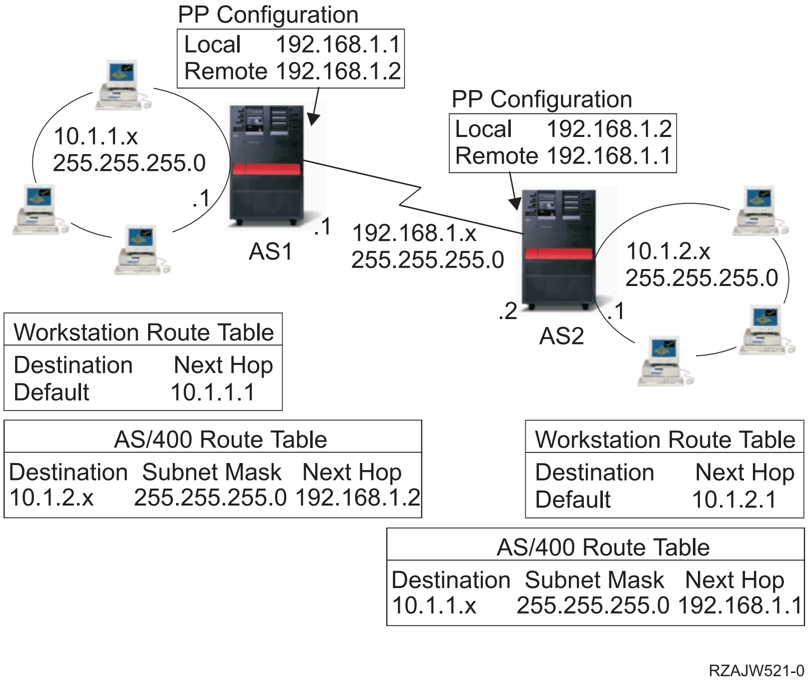
On the surface, it seems that the simplest way to configure a point-to-point connection is by using a numbered connection. A numbered connection is a point-to-point definition that has a unique IP address defined for each end of a connection.

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 iSeries server, 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 iSeries server 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, this may use an entire subnet of addresses just to define an interface at each end.

The figure below shows a numbered network connection between two iSeries servers. A routing entry is not needed if all you want to do is 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 a part of the 192.168.1.x connection.



Workstation Route Table	
Destination	Next Hop
Default	10.1.1.1

AS/400 Route Table		
Destination	Subnet Mask	Next Hop
10.1.2.x	255.255.255.0	192.168.1.2

Workstation Route Table	
Destination	Next Hop
Default	10.1.2.1

AS/400 Route Table		
Destination	Subnet Mask	Next Hop
10.1.1.x	255.255.255.0	192.168.1.1

Unnumbered network connections:

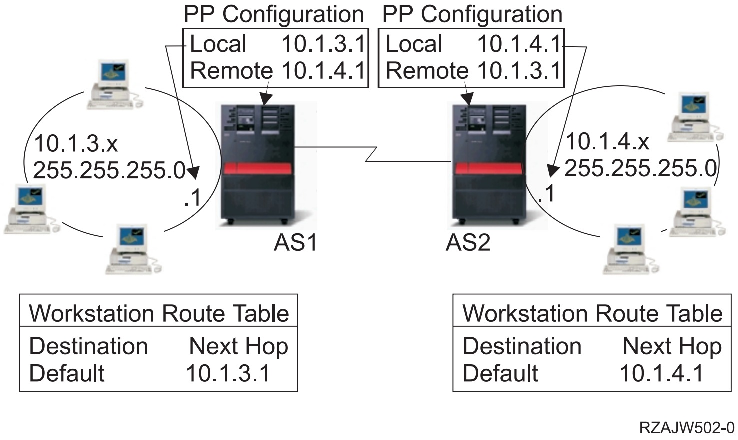
An unnumbered connection is a more complex method of defining a point-to-point connection than a numbered connection. However, you may find the unnumbered connection a simpler and better way to manage your network.

The routing selection process in the iSeries server 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 iSeries server 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.

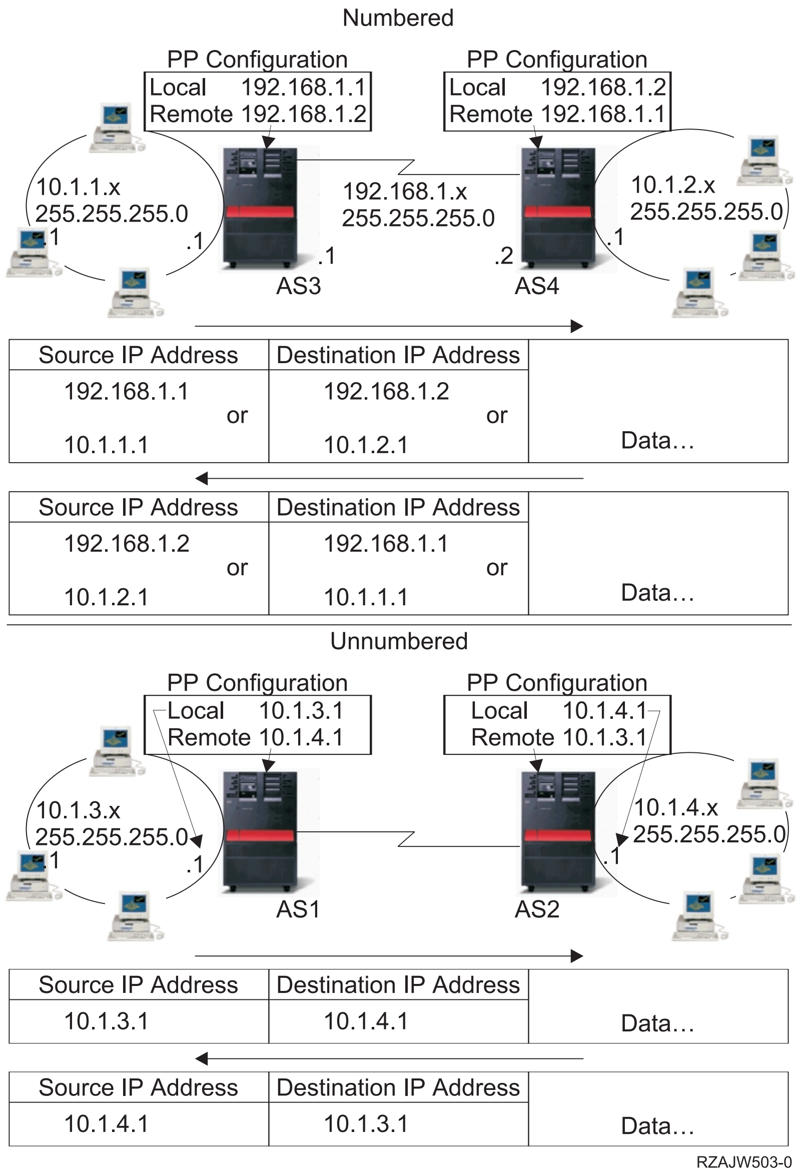


Also shown in the example is AS2. AS2 is connected to LAN network 10.1.4.x with an address of 10.1.4.1. This allows AS2 to communicate with any system on the 10.1.4.x network directly. Each system (AS1 and AS2) adds the remote address to its routing table as a local interface. The address is treated 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 would simply 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 would still like 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 would have gotten if you could have used LAN adapters without adding any routing entries to your iSeries server. 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 would 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.



Source IP Address	Destination IP Address	
192.168.1.1 or 10.1.1.1	192.168.1.2 or 10.1.2.1	Data…

Source IP Address	Destination IP Address	
192.168.1.2 or 10.1.2.1	192.168.1.1 or 10.1.1.1	Data…

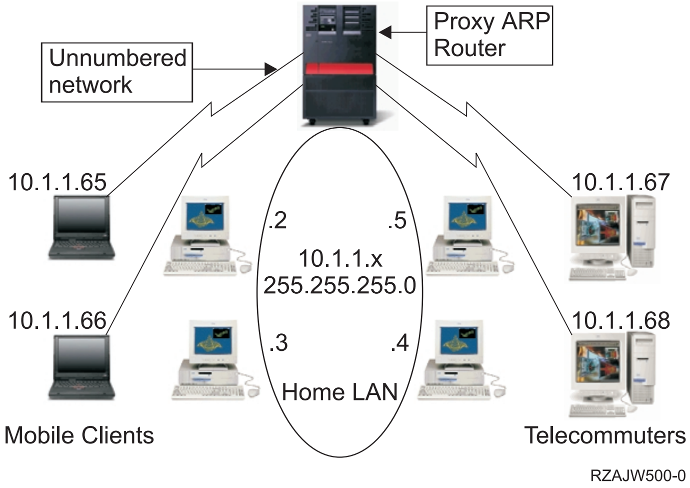
Source IP Address	Destination IP Address	
10.1.3.1	10.1.4.1	Data…

Source IP Address	Destination IP Address	
10.1.4.1	10.1.3.1	Data…

RZAJW503-0

Proxy Address Resolution Protocol routing

Proxy Address Resolution Protocol (ARP) routing allows physically distinct, separate networks to appear as if they were a single logical network. It provides connectivity between physically separate networks without creating any new logical networks and without updating any routing tables. Proxy ARP allows systems that are not directly connected to a 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.

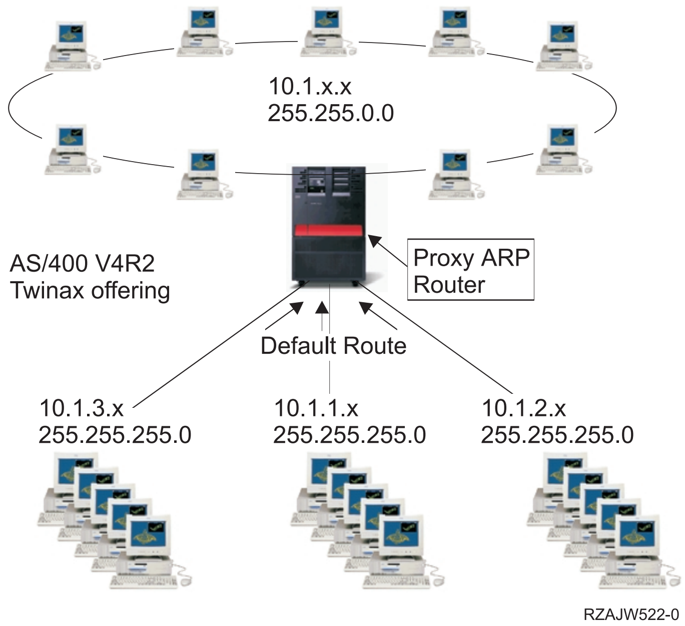


When a system on your home LAN (10.1.1.x) wants to send data to one of your remote systems, it will first do an ARP request. This is a broadcast that goes out to all your systems attached to the LAN segment to request the address of the target system. However, a remotely connected system will not see the broadcast. This is where the proxy ARP comes in. The iSeries server knows which systems are connected remotely. If your iSeries server sees an ARP request for one of your remotely connected machines, your iSeries server will reply to the ARP request with its address. Your iSeries server 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 iSeries server will not reply to the ARP request and the requesting system will not send data.

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

You can use transparent subnets as a way to extend the proxy ARP concept. 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 figure below 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 twinaxial 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 and add TCP/IP interface would not allow this to happen. In V4R2 the edits were relaxed. This allows two interfaces in different segments to have addresses that look like they are in the same segment. When the iSeries server sees this happen, it automatically performs a Proxy ARP for any systems that are attached behind the twinaxial controller. This allows all the systems on the 10.1.x.x network to communicate with all the subnet systems with no changes to the systems on the 10.1.x.x network.

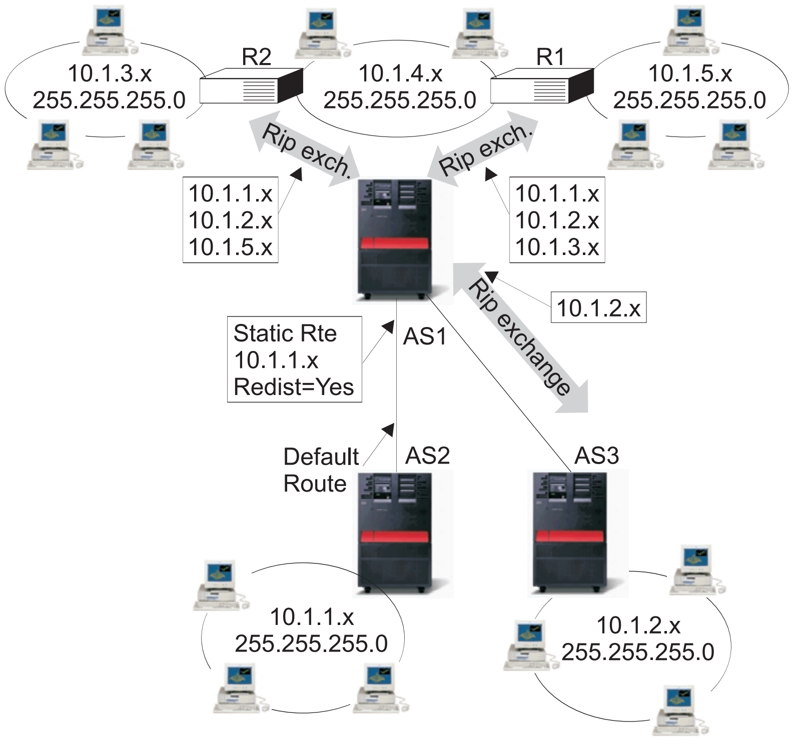
Transparent subnetting over WANs:

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 figure above, three networks are attached to the home 10.1.x.x network through the iSeries server. These networks are all defined using a subnet mask that makes them transparent to the home network. Proxy ARP responds 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 causes the traffic for the home network to be routed automatically to the iSeries server in the home network. This iSeries server in turn routes the data to the correct remote iSeries server. The remote iSeries server 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 iSeries server 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 provided by Interior Gateway Protocols (IGPs), such as Routing Internet Protocol (RIP). RIP allows you 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 crashes. RIPv2 was added to the iSeries server so you can send and receive RIP packets to update routes throughout your network.

In the figure below, 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 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 iSeries server (AS1). AS2 has the routed server 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.



RZAJW520-0

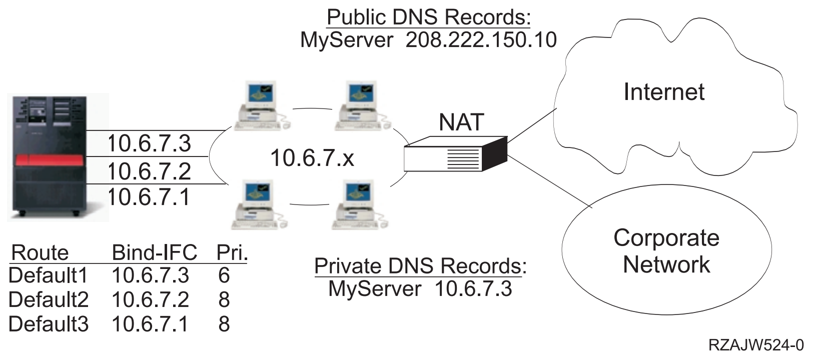
What’s happening in this example?

- 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 may 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.

Route binding

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 there are three interfaces 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. To do this, you must add the ″duplicate″ routes to each interface. In this example, we add three default routes, 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.

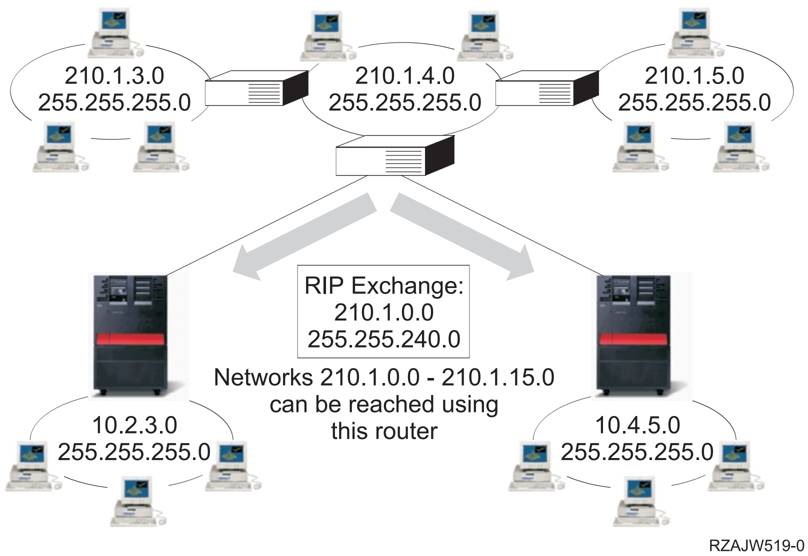


Classless Inter-Domain Routing

Classless Inter-Domain Routing (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 would make 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 could 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.



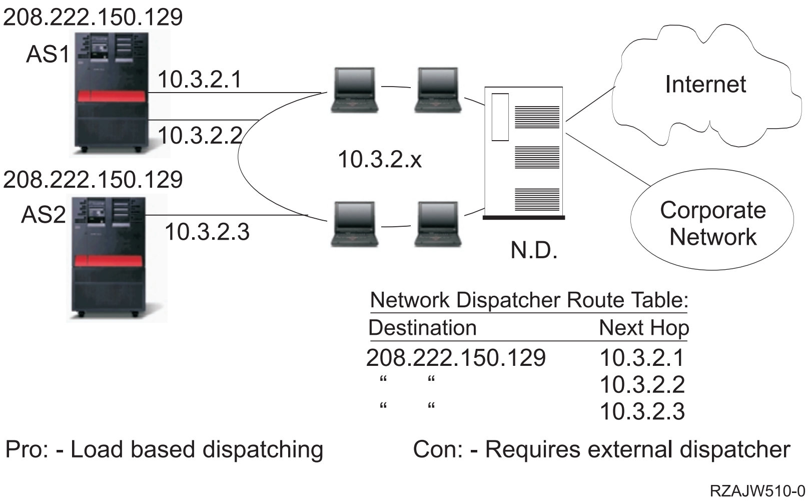
In this example, the router is set up to send one RIP message with the network address 210.1.0.0 and subnet mask 255.255.240.0. This tells your systems to receive the RIP messages for networks 210.1.0.0

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 can be used for many different things. It provides a way to assign one or more addresses to the system without the need of binding the address to a physical interface. This can be used when you want to run multiple occurrences of a Domino Web server 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 iSeries server, 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 iSeries server. The existence of these multiple interfaces should only be visible on the local network. You do not want the remote clients to have to be aware of the multiple IP addresses for the iSeries server. Ideally, you would like them to view your iSeries server as a single IP address. How the inbound packet gets routed through the gateway, over the local network, and to the iSeries server should be invisible to a remote client. The way to accomplish this is by using virtual IP. Local clients should communicate with the iSeries server by any of the physical IP addresses while remote clients would see only the virtual IP interface.



Network Dispatcher Route Table:

Destination	Next Hop
208.222.150.129	10.3.2.1
“ “	10.3.2.2
“ “	10.3.2.3

Pro: - Load based dispatching

Con: - Requires external dispatcher

The virtual IP environment is for the iSeries server 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 iSeries server 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 usually be set to 255.255.255.255.

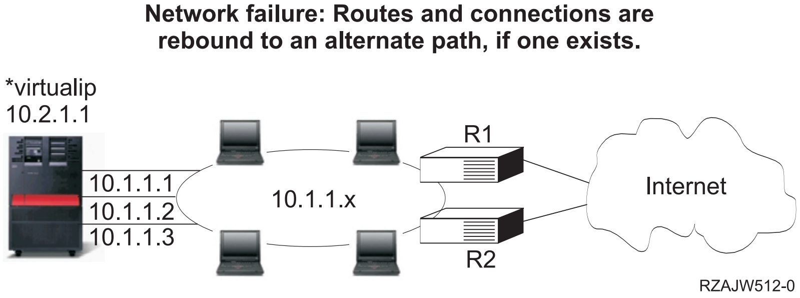
Since the virtual IP address is not bound to a single physical interface, iSeries server never responds to an Address Resolution Protocol (ARP) request to the virtual IP address. In other words, you cannot route directly to a virtual IP address. For other systems to reach the virtual IP address, they must have a route defined to reach the address. This is why Virtual IP is designed mostly for remotely attached clients. In the following example, the workstations all point to one of the 10.3.2 interfaces, on the iSeries server, as their next hop gateway. When a packet arrives at iSeries server, 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 DNS servers use the addresses of the requested server. In this case all the addresses represent the same system. The virtual IP function can be used when consolidating multiple systems into one larger system.

Fault tolerance

Another use for virtual IP addresses is to protect against route fault tolerance.

This example shows several different ways a route may 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.


Network failure: Routes and connections are rebound to an alternate path, if one exists.
*virtualip
10.2.1.1
10.1.1.1
10.1.1.2
10.1.1.3
10.1.1.x
R1
R2
Internet
RZAJW512-0

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.

Routing with network address translation

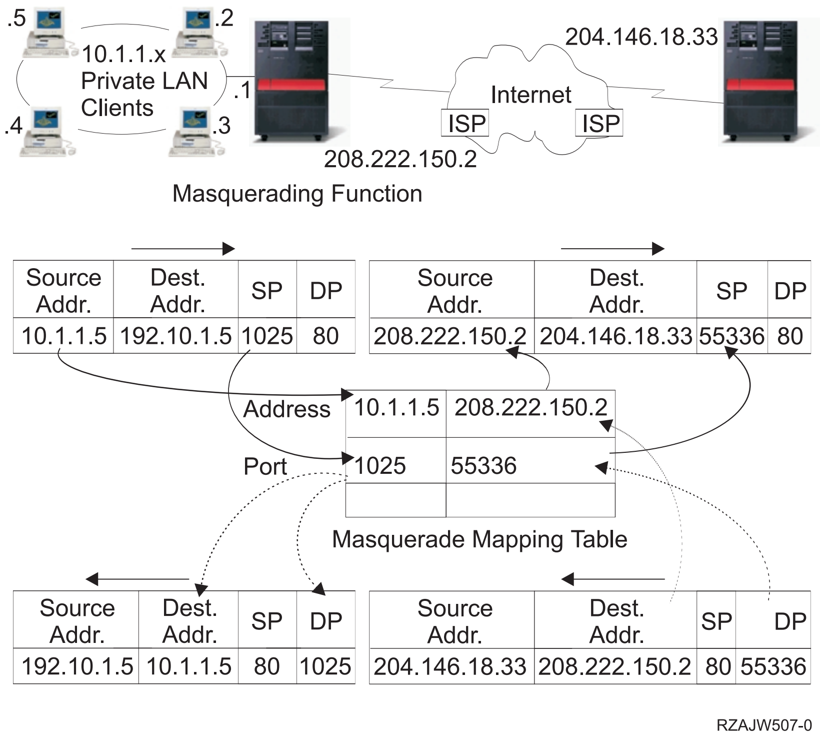
Network address translation (NAT) provides access to a remote network, usually the Internet, while protecting the private network by masking the IP addresses that are used inside your firewall. You can use these types of NAT for routing your iSeries server:

- Masquerade NAT
  Masquerade NAT allows your private network to hide behind and be represented by the address you bind to your public interface.
- Dynamic NAT
  Dynamic NAT establishes a connection from within a private network out to a public network. The difference is that a pool of public addresses is maintained and used when an outbound connection is made.
- Static NAT
  Static NAT supports inbound connections from a public network into a private network.

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, this is the address that has been assigned by an Internet Service Provider (ISP), and the address may 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 iSeries server. iSeries server has an IP address assigned by the local ISP as its Internet gateway. The term locally attached machine is used to refer to all machines on an internal network regardless of the method of attachment (LAN or WAN) and regardless of the distance of the connection. The term external machines is used to refer to machines located on the Internet. The following figure illustrates how Masquerade NAT works.



Source Addr.	Dest. Addr.	SP	DP
10.1.1.5	192.10.1.5	1025	80

Source Addr.	Dest. Addr.	SP	DP
208.222.150.2	204.146.18.33	55336	80

10.1.1.5	208.222.150.2
1025	55336

Source Addr.	Dest. Addr.	SP	DP
192.10.1.5	10.1.1.5	80	1025

Source Addr.	Dest. Addr.	SP	DP
204.146.18.33	208.222.150.2	80	55336

To the Internet, all of your workstations appear to be contained within your iSeries server; that is, only one IP address is associated with both your iSeries server 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 iSeries server 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 iSeries server 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 machines. 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.

Outbound masquerade NAT processing:

The outbound message in the figure above 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.

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 logcial 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.

Inbound masquerade NAT processing (response and other):

The inbound message in the figure above 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 machine 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.

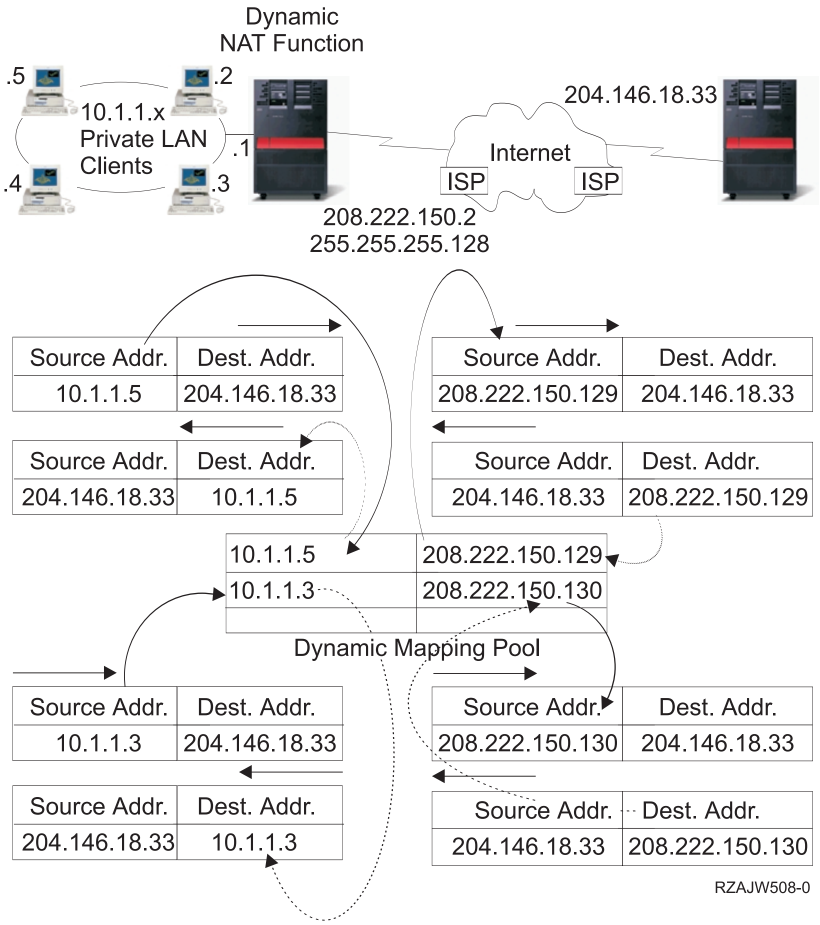
2. If a matching logical port number is found, a further check is made to determine that the source IP address matches the destination IP address of the existing logical port number table entry. If it matches, the original local machine’s port number replaces the source port in the IP header. If the 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 machine. 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.

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 figure below illustrates Dynamic NAT.



Source Addr.	Dest. Addr.
10.1.1.5	204.146.18.33

Source Addr.	Dest. Addr.
204.146.18.33	10.1.1.5

Source Addr.	Dest. Addr.
208.222.150.129	204.146.18.33

Source Addr.	Dest. Addr.
204.146.18.33	208.222.150.129

10.1.1.5	208.222.150.129
10.1.1.3	208.222.150.130

Dynamic Mapping Pool

Source Addr.	Dest. Addr.
10.1.1.3	204.146.18.33

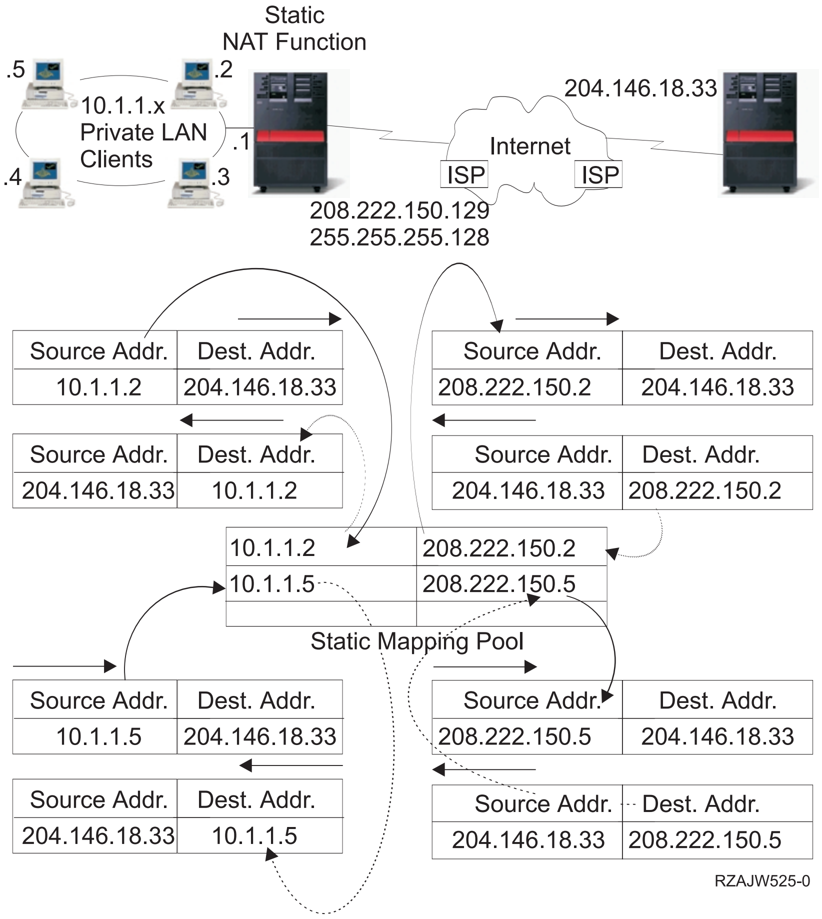
Source Addr.	Dest. Addr.
204.146.18.33	10.1.1.3

Source Addr.	Dest. Addr.
208.222.150.130	204.146.18.33

Source Addr.	Dest. Addr.
204.146.18.33	208.222.150.130

Static NAT

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.



Source Addr.	Dest. Addr.
10.1.1.2	204.146.18.33

Source Addr.	Dest. Addr.
204.146.18.33	10.1.1.2

Source Addr.	Dest. Addr.
208.222.150.2	204.146.18.33

Source Addr.	Dest. Addr.
204.146.18.33	208.222.150.2

10.1.1.2	208.222.150.2
10.1.1.5	208.222.150.5

Source Addr.	Dest. Addr.
10.1.1.5	204.146.18.33

Source Addr.	Dest. Addr.
204.146.18.33	10.1.1.5

Source Addr.	Dest. Addr.
208.222.150.5	204.146.18.33

Source Addr.	Dest. Addr.
204.146.18.33	208.222.150.5

Routing with OptiConnect and logical partitions

OptiConnect and logical partitions provide other environments for you to use the routing basics of proxy ARP, point-to-point, and virtual IP interfaces. Here are a few different methods of these basics.

- TCP/IP and OptiConnect
  OptiConnect can give you the ability to define TCP/IP connections over an OptiConnect bus. This page describes this feature and how it may be used.
- Virtual OptiConnect with logical partitions
  Virtual OptiConnect TCP/IP interfaces are used as interpartition communication paths. A single iSeries server is logically partitioned in multiple virtual machines. Each partition has its own address space. To TCP/IP, each partition appears as a distinct iSeries server. This page shows you how you can use this feature to your advantage.

TCP/IP and OptiConnect

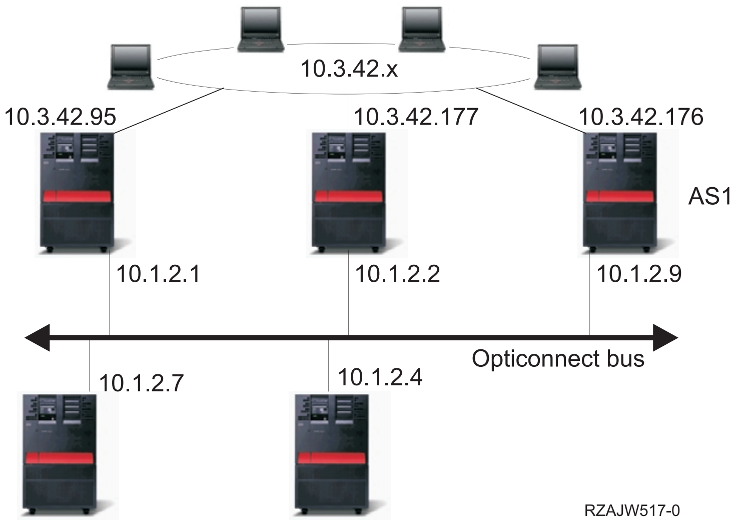
OptiConnect gives you the ability to define TCP/IP connections over an OptiConnect bus. TCP/IP over OptiConnect provides another method for the routing building blocks such as proxy ARP, unnumbered point-to-point networks, and virtual IP interfaces. You can configure this with an OptiConnect-emulated LAN configuration and an OptiConnect point-to-point configuration.

With an **OptiConnect-emulated LAN configuration**, the OptiConnect bus appears as a LAN to TCP/IP. 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 to 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 iSeries servers 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 on the unnumbered point-to-point configuration. Remember that whenever you use unnumbered, point-to-point interfaces, each interface has to 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 will know the local iSeries server. This associated local interface may be the iSeries server's primary LAN interface, as shown below. Or you can use a virtual IP interface as the associated local interface. In this configuration 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 previous 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 will exist to reach any iSeries server.

Opticonnect Emulated LAN configuration

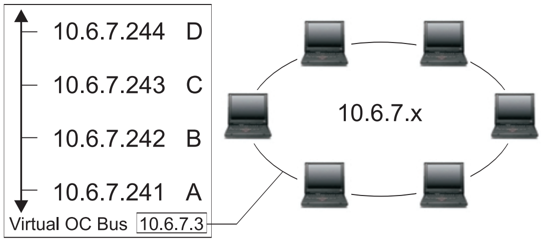


Routing with virtual OptiConnect and logical partitions

With logical partitions, a single iSeries server is logically partitioned into multiple virtual machines. 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 may have its own dedicated I/O adapters. To TCP/IP, each partition appears as a distinct iSeries server. 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.

LPAR: Virtual opticonnect TCP/IP interfaces are used as inter-partition communication paths.

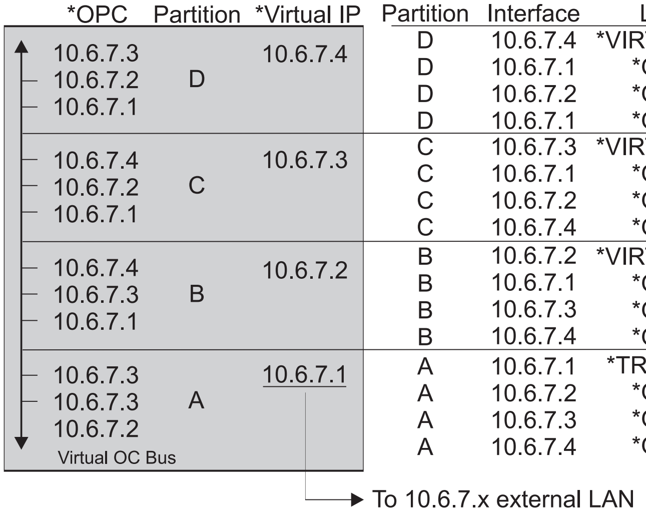
Virtual Opticonnect network= 10.6.7.241 - 10.6.7.254
This provides addresses for up to 14 partitions



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	(Associated local interface = 10.6.7.3)
A	10.6.7.3	TRNLINE	255.255.255.0	4096	

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.



Partition	Interface	Line	Subnet Mask	MTU	Assoc. LCL IFC
D	10.6.7.4	*VIRTUALIP	255.255.255.255	4096	*NONE
D	10.6.7.1	*OPC	255.255.255.255	4096	10.6.7.4
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
C	10.6.7.3	*VIRTUALIP	255.255.255.255	4096	*NONE
C	10.6.7.1	*OPC	255.255.255.255	4096	10.6.7.3
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
B	10.6.7.2	*VIRTUALIP	255.255.255.255	4096	*NONE
B	10.6.7.1	*OPC	255.255.255.255	4096	10.6.7.2
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
A	10.6.7.1	*TRNLINE	255.255.255.0	4096	*NONE
A	10.6.7.2	*OPC	255.255.255.255	4096	10.6.7.1
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

RZAJW516

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 machines across multiple processors, multiple interface adapters, or multiple host servers. If you want to get the best performance possible from your iSeries server, you should put the communications load on multiple parts of your server.

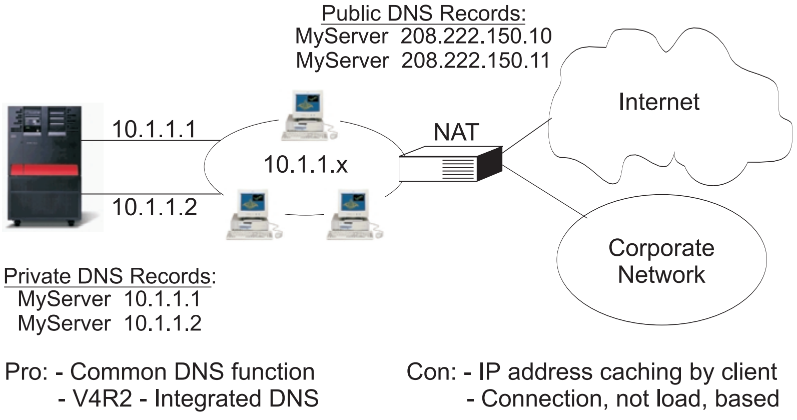
Several different TCP/IP routing methods can be used to balance the workload of your iSeries server:

- DNS-based load balancing
  You can use DNS-based load balancing for your inbound workload. If load balancing is needed for local clients, you should use DNS load balancing.
- Duplicate route-based load balancing
  Here you can learn about outbound workload balancing across multiple interfaces. This is a connection-based solution that has more flexibility than DNS-based load balancing but is not active for local clients.
- Load balancing with virtual IP
  This solution requires you to have an external load balancing machine, such as IBM eNetwork Dispatcher. Virtual IP addresses allow you to assign an address to the system rather than to a specific interface. You may define the same address to multiple servers, which allows many new options for load balancing.

DNS-based load balancing

DNS-based load balancing is used for inbound load balancing. Multiple host IP addresses are configured in DNS for a single host server 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 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 example below, 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.



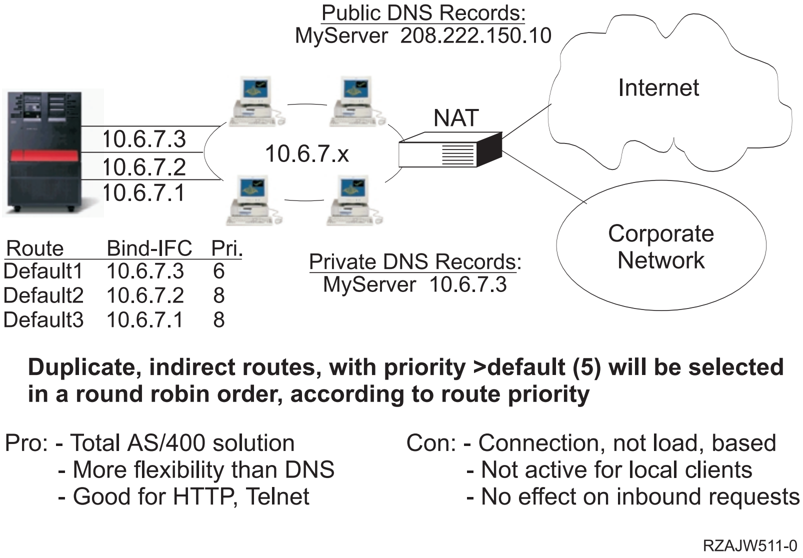
RZAJW518-0

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 site should be coded to send a different system name after the first contact is made. Additional DNS entries could be added for MyServer1 208.222.150.10 and MyServer2 208.222.150.11. By doing this, the Web sites, for example, could point to MyServer2 after the first contact. This type of load balancing provides balancing by the connection request. In most cases, once 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.

Duplicate route-based load balancing

You can use duplicate route-based load balancing for outbound workload balancing 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 iSeries server 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 (not a load-based solution) it is not active for local clients; and it has no effect on inbound requests.

In the example below, 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 could be a very large network if switches were used rather than routers.



Route	Bind-IFC	Pri.
Default1	10.6.7.3	6
Default2	10.6.7.2	8
Default3	10.6.7.1	8

Duplicate, indirect routes, with priority >default (5) will be selected in a round robin order, according to route priority

Pro: - Total AS/400 solution
- More flexibility than DNS
- Good for HTTP, Telnet

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

RZAJW511-0

Where do I go to configure this?

You can configure this in the Add TCP/IP Route command line and also in the iSeries Navigator interface. One is called duplicate route priority, the other is called the preferred binding interface. If the value for 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 rather than the first one the system sees.

In the example above, 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.

Adapter failover using virtual IP and Proxy ARP

Situation

Your production iSeries 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 iSeries and the network. Because of the growth, it has become imperative that this iSeries be available on the network without an unscheduled down time. If, for any reason, a network adapter becomes unavailable, other network adapters on the iSeries should take over and the network clients should be unaware of any failures.

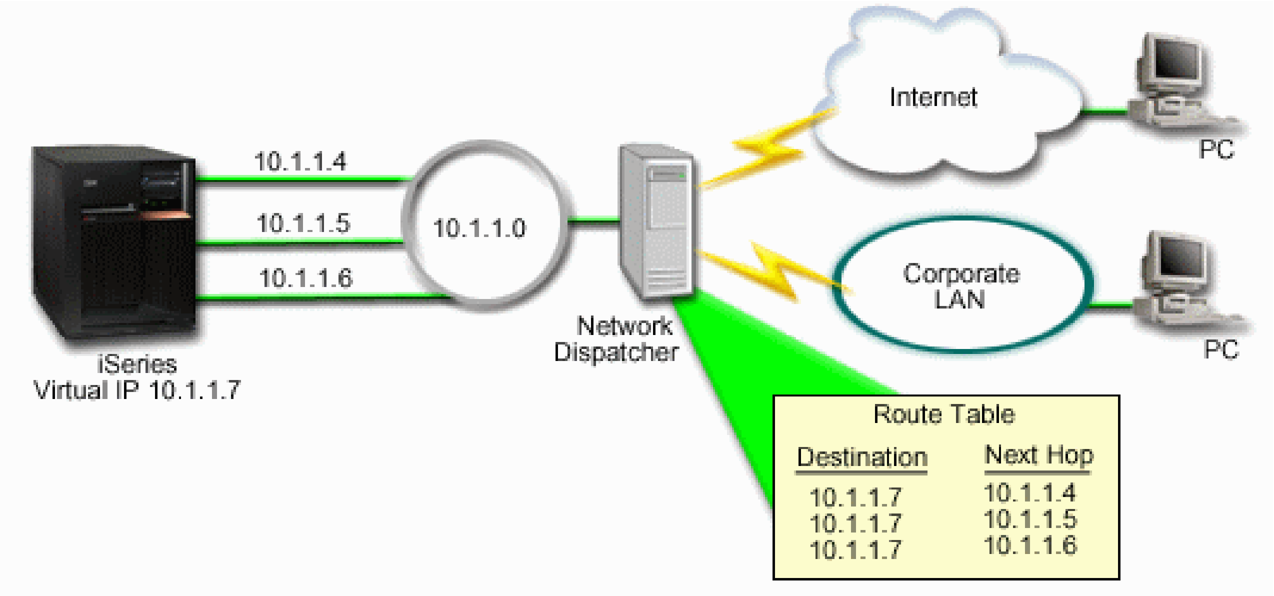
Objectives

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 iSeries for its clients in the event of an adapter failure.

Details

One way to handle the above scenario is have multiple physical connections to the LAN from your iSeries. Consider the following figure.

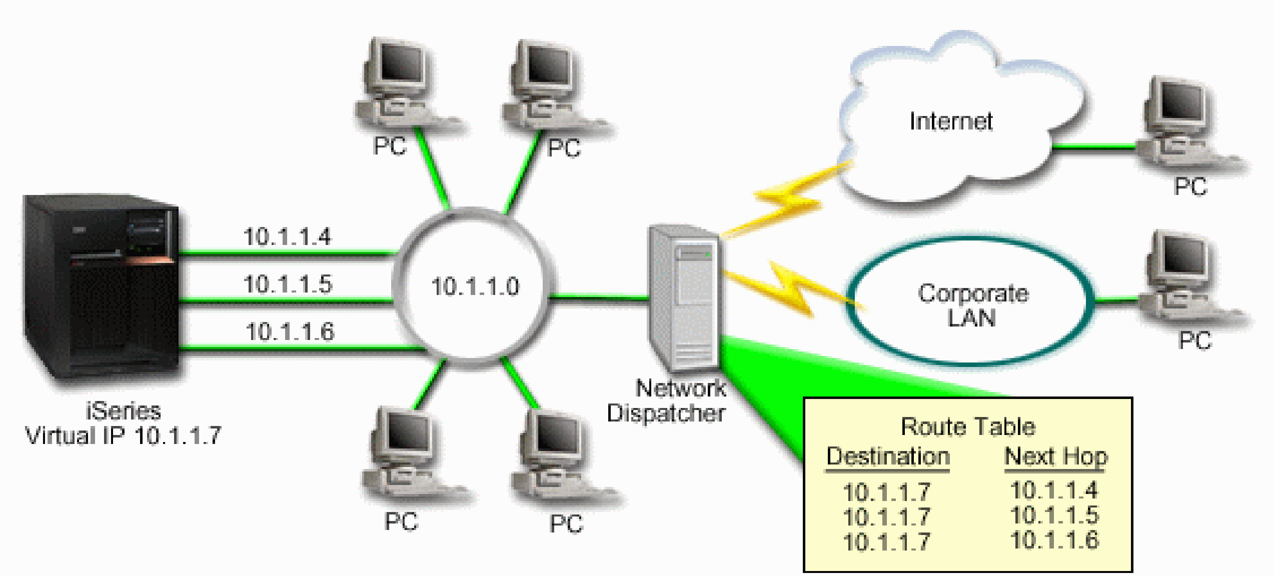
Figure 1. Adapter failover without local clients



Each of these physical connections would have a different IP address. Then you can assign a virtual IP address to the system. This virtual IP address is the IP address that all of its clients will recognize it by. All remote clients (clients that are not physically attached to the same LAN as the iSeries) will communicate with the iSeries through a external load balancing server such as a network dispatcher. When the IP requests from the remote clients go through the network dispatcher, the network dispatcher will route the virtual IP addresses to one of network adapters on the iSeries.

If the LAN that the iSeries is connected to has clients, these clients would not use the network dispatcher to direct their locally bound traffic because that would unnecessarily overload the network dispatcher. You could create route entries on each client that were similar to the route tables in the network dispatcher, but given a large number of clients, this would be a very impractical practice. This situation is described in the following figure.

Figure 2. Adapter failover with local clients

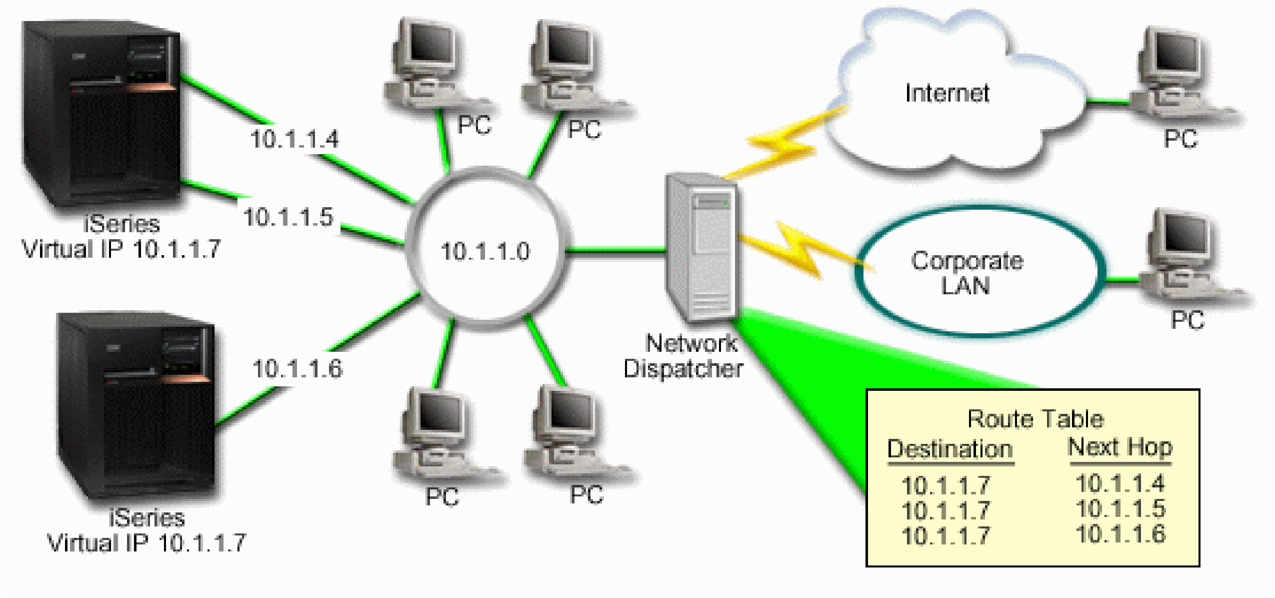


Route Table	
Destination	Next Hop
10.1.1.7	10.1.1.4
10.1.1.7	10.1.1.5
10.1.1.7	10.1.1.6

As of OS/400 V5R2, local clients (clients that are attached to the same LAN as the iSeries) can now connect to the iSeries virtual IP address through ARP. This allows local clients to have an adapter failover solution as well.

The solution can also involve using two or more iSeries servers to support each other. If one of the iSeries systems become unavailable, then the second system can serve as a failover. The following figure shows the same setup using two iSeries serves:

Figure 3. Adapter failover with multiple iSeries and local clients



Route Table

Destination	Next Hop
10.1.1.7	10.1.1.4
10.1.1.7	10.1.1.5
10.1.1.7	10.1.1.6

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

Configuration steps

The configuration for load balancing using virtual IP and proxy arp is very similar to standard TCP/IP configurations with the addition of a virtual TCP/IP interface. In the above case of adapter failover with local clients (See 24), the general configuration steps would be:

1. **Configure a virtual TCP/IP interface.**

   Using iSeries 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 mouse click **Interfaces** and choose **New Interface**->**Virtual IP**.

   For our example, we would enter an IP address of 10.1.1.7 with a subnet mask of 255.255.255.255. Once you have created the Virtual interface, right mouse click on the interface and select **Properties**. Click on the **Advanced** tab and check the **Enable Proxy ARP** checkbox.

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 iSeries Navigator and can be found at:
   Network—>**TCP/IP Configuration**->**IPv4**->**Interfaces**. Then right mouse click **Interfaces** and choose **New Interface**->**Local Area Network**. Complete the wizard for each of your LAN connections.

   For our example, you would 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 have completed each interface, right mouse click on the interface and choose **Properties**. On the **Advanced** tab, associate the interface with the Virtual IP interface you created in step 1. You can associate the interfaces with the **Associated local interface** select box.

Other information about TCP/IP routing and workload balancing

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 provides you with more background information and detail.

NAT and IP filter administration helps you manage your filter rules. Some of the functions include adding comments, editing, and viewing.

OptiConnect

, provides you with information on OptiConnect routing. This is an iSeries server online book called *OptiConnect for OS/400 V4R4*.

Point-to-point protocol 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).

Printed in U.S.A.