Spring COMMON, 2000: Session: 45MJ

AS/400 Routing and Load Balancing Techniques

Gary A. Diehl

diehl@us.ibm.com

AS/400 TCP/IP Development

This publication may refer to products that are not currently available in your country.
IBM makes no commitment to make available any products referred to herein.

Acknowledgements:

Much of this presentation is based upon an ITSO presentation by Fant Steele, entitled, "TCP/IP Routing and Load Balancing Tips and Techniques". Fant's contribution is gratefully acknowledged.

Objectives

Describe AS/400 TCP/IP routing capabilities

Describe how these capabilities can be used for:

- ➡ *Load Balancing*
- ➡ *Fault tolerance*
- ➡ *Consolidated Router & Server Functions*
- ➡ *Configuration Stability*
- ➡ *Simplified IP Address Management*

Agenda

AS/400 Routing Techniques (the "Building Blocks"):

- Proxy ARP
- Point-point: numbered & unnumbered networks
- RouteD (RIP)
- NAT/Masquerading
- VirtualIP
- Schowler Routes
- Route Binding
- CIDR/Supernetting

Applications :

- Load Balancing
- Fault Tolerance
- Advanced Point-Point Applications:
  - TCP/IP over Opticonnect & LPAR
  - Frame Relay

AS/400 TCP/IP Routing history

V3R1: Static route based packet forwarding

V3R7/V3R2:

SLIP: Proxy ARP Routing and Unnumbered network support

V4R1:

Dynamic Routing Information Protocol Ver 1 (RIPv1)

V4R2:

Dynamic Routing Information Protocol Ver II (RIPv2)

Twinax: Transparent subnetting

Duplicate route based load balancing

V4R3:

Virtual IP addresses

IP Address Masquerading and Network Address Translation (NAT)

CIDR / Supernetting

V4R4:

TCP/IP over Opticonnect

TCP/IP with LPAR

V4R3/V4R4 PTFs:

Local load balancing

Virtual IP extensions for local clients

Notes:

A brief history of routing developments on the AS/400. Before you plan to use a function, check here to make sure that your system is at the correct level to support the function. In some cases you can use a different approach to achieve the same results.

Note that the term "*routing*", as it is used in this presentation, implies more than simply the support of formal routing protocols like RIPv1 or RIPv2. Techniques like Proxy ARP routing, unnumbered interfaces and Virtual IP addresses provide powerful routing capabilities, independent of routing protocols and transparent to the rest of the network.

Routing Basics

Interfaces:

- TCP/IP connection to the external world
- The system does not have an IP address, only the interfaces have IP addresses (Exception: *Virtual IP)
- "Weak Multihoming" model: In general, no matter what interface a packet comes in on, the system will accept it if the destination IP address is defined on the system.

Routes:

- Path denoting an adapter which should be used for sending packets to a given external destination
- Route Destination & Subnet Mask define the host, or range of hosts, that are reachable via this route
- Each route is bound to an interface --> line description ->hardware resource (adapter)
  - Exception: Pseudo line descriptions like *LOOPBACK, *OPC, etc.
- Two general route classes:
  - *DIRECT Routes

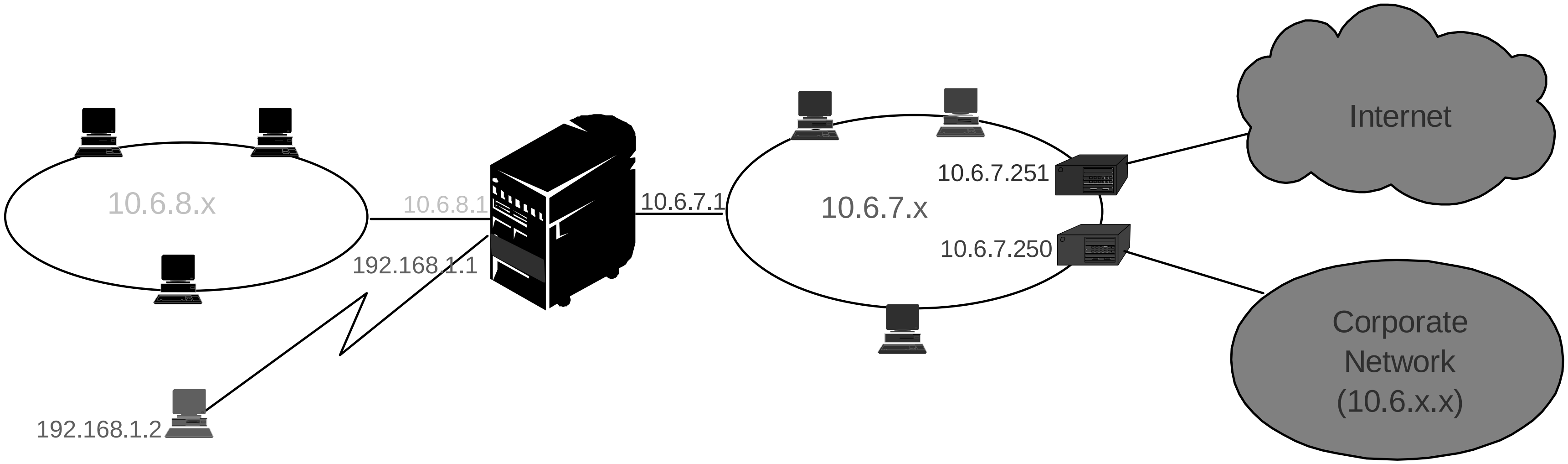
    Automatically added by the system when an interface is added

    Defines the range of external hosts that are locally connected, i.e.., "locally reachable"
  - Indirect Routes : (Sub)network routes and Default routes

    Manually configured, or automatically added via RIP, ICMP, etc.

    Defines range of hosts "remotely" connected, i.e., hosts reachable via an intermediate next hop gateway
- The route selected for sending packets is based upon the destination IP address in the packet and the Route Destination of the configured routes. Route selection is ordered based on:
  - Route group search order: Direct routes, then (sub)network routes, then default routes
  - Within group, the route with the most specific subnet mask is chosen
  - Among equally specific routes, the route bound to the preferred source IP address is chosen (more later)
  - Equally specific routes subject to list order or load balancing options

Routing Basics - Example



Sample Interface Table

IP Address	Network Address	Line
10.6.7.1	10.6.7.0	TRNLINE1
10.6.8.1	10.6.8.0	TRNLINE2
127.0.0.1	127.0.0.0	*LOOPBACK
192.168.1.1	192.168.1.1	PPPLINE

Sample Route Table

Route Destination	Subnet Mask	Next Hop
10.6.7.0	255.255.255.0	*DIRECT
10.6.8.0	255.255.255.0	*DIRECT
10.6.0.0	255.255.0.0	10.6.7.250
127.0.0.0	255.0.0.0	*DIRECT
192.168.1.1	*HOST	*DIRECT
192.168.1.2	*HOST	192.168.1.1
*DFTROUTE	*NONE	10.6.7.251

The Building Blocks

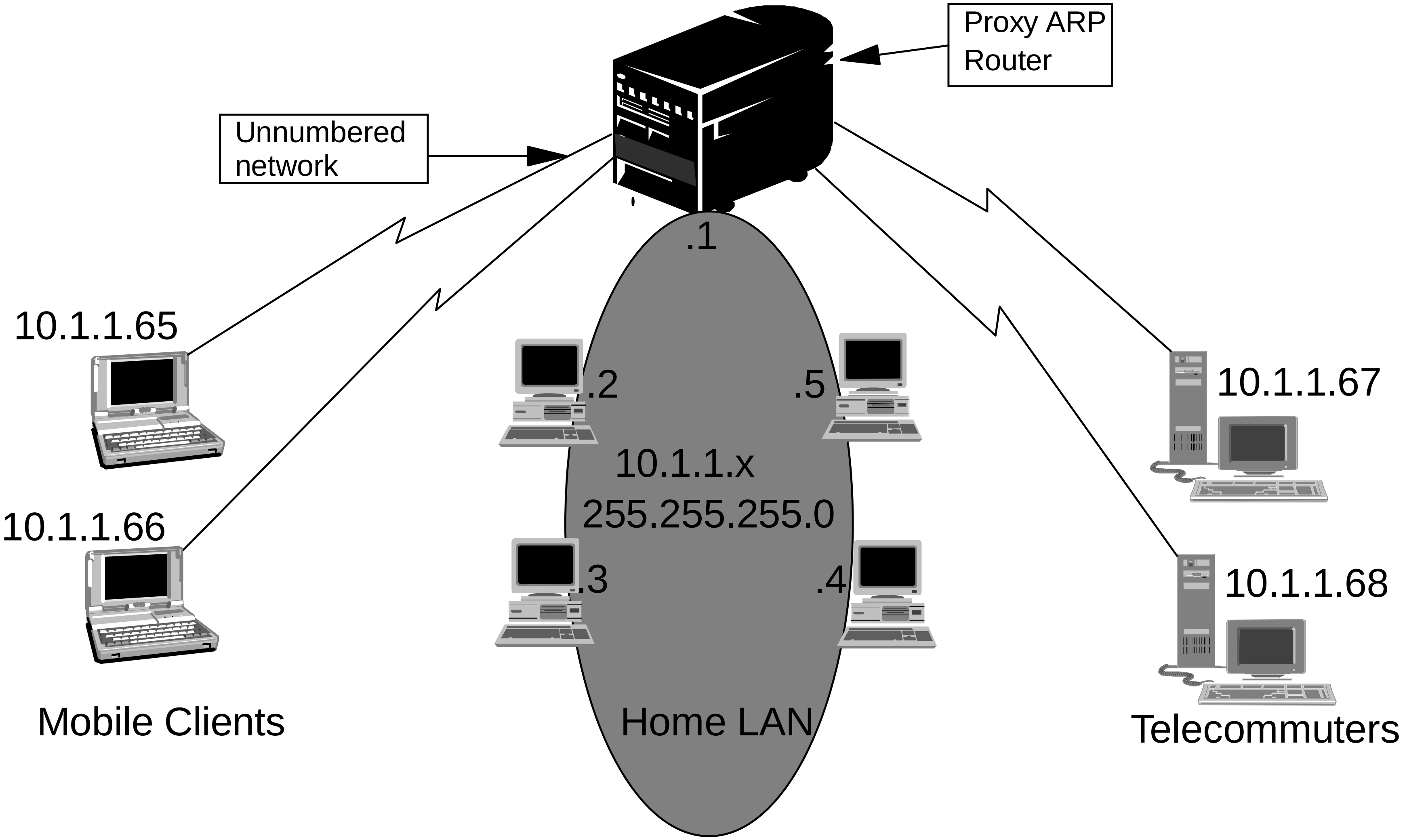
Notes:

This section will cover the basic AS/400 functions and concepts that can be used to implement advanced solutions to complex routing issues -- issues like network performance, availability and configuration stability

An understanding of these basic principles is required in order to appreciate the applications of these principles, given in subsequent pages. Moreover, such an understanding will enable you to apply these same principles in your own environments, that may not be covered in this presentation.

Proxy ARP Routing

Remote clients appear to be connected to the Home LAN



Notes:

Proxy ARP allows physically distinct networks to appear as if they are a single, logical network. The advantage of this technique is that it provides connectivity between these physically separate networks, without creating any new logical networks, and more importantly, without updating any route tables. Automatic, or "transparent" connectivity is provided between the two networks.

In the preceding chart, proxy ARP allows systems that are not directly connected to a LAN appear to other systems on the LAN as though they are connected. This is very useful in a dial up scenario to provide connections to the entire network from a dial in interface.

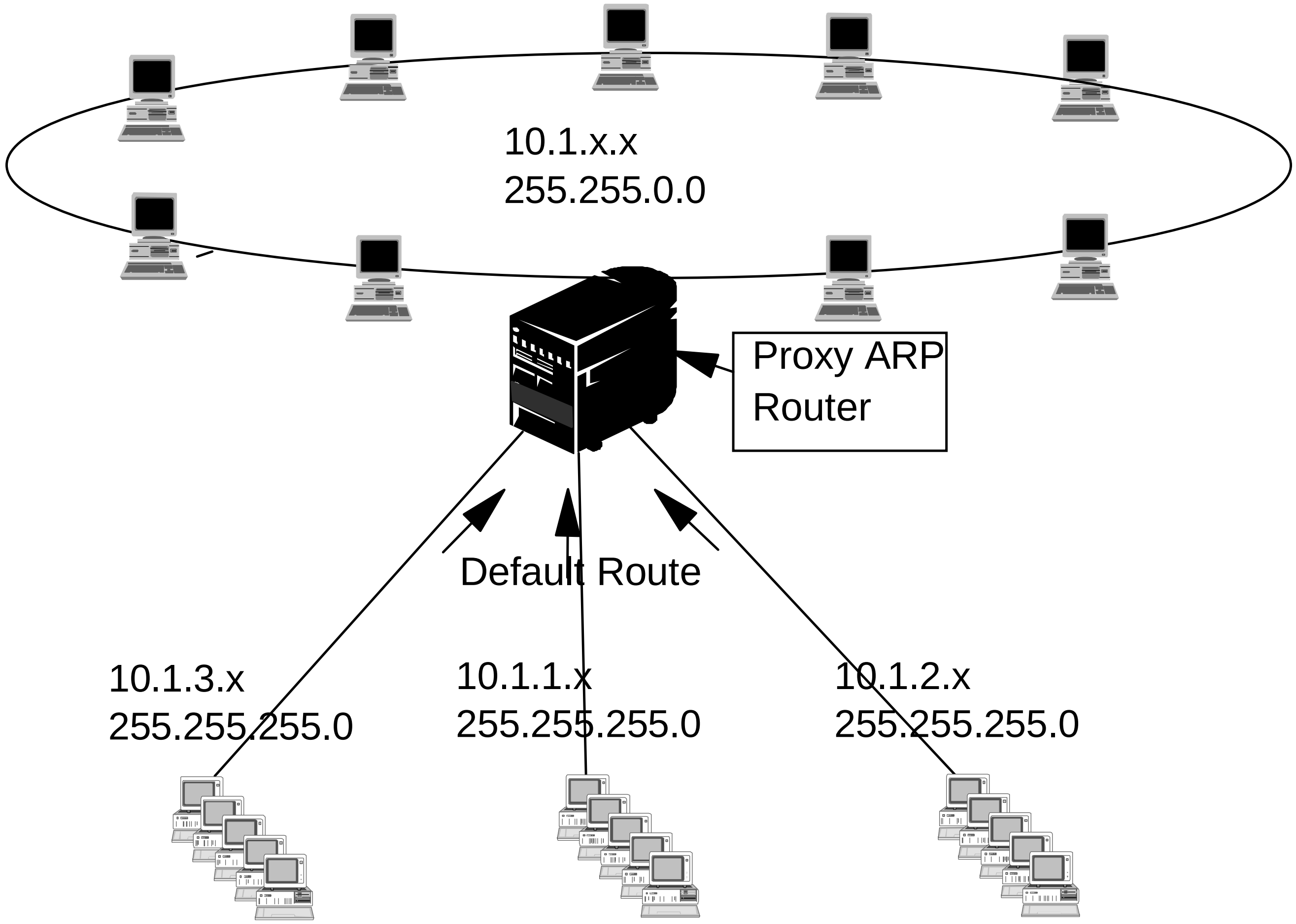
When a system on the home LAN 10.1.1.x wants to send data to one of the remote systems, it will first do an ARP (address resolution protocol) request. This is a broadcast that goes out to all the systems attached to the LAN segment to request the MAC address of the target system. In the case of a remote connected system, it will not see the broadcast. This is where Proxy ARP comes in. The AS/400 knows which systems are connected remotely. If The AS/400 sees a ARP request for one of the remote connected machines, the AS/400 will reply to the ARP request with it’s MAC address. The system requesting the ARP will then send to the AS/400 MAC address. The AS/400 in turn receives the data and forwards it to the remote system.

If the remote system is not connected, the AS/400 will not reply to the ARP request and the requesting system will not send data.

Note: For this forwarding to take place, the TCP/IP Attribute, Datagram Forwarding, must be set to *YES

Classic Transparent Subnetting

AS/400
V4R2
Twinax
offering



- Stub networks are assigned addresses out of the primary network address space -- subset of primary network
- **Note:** Requires LAN interface be configured as "Associated Local Interface on the TDLC interfaces

Notes:

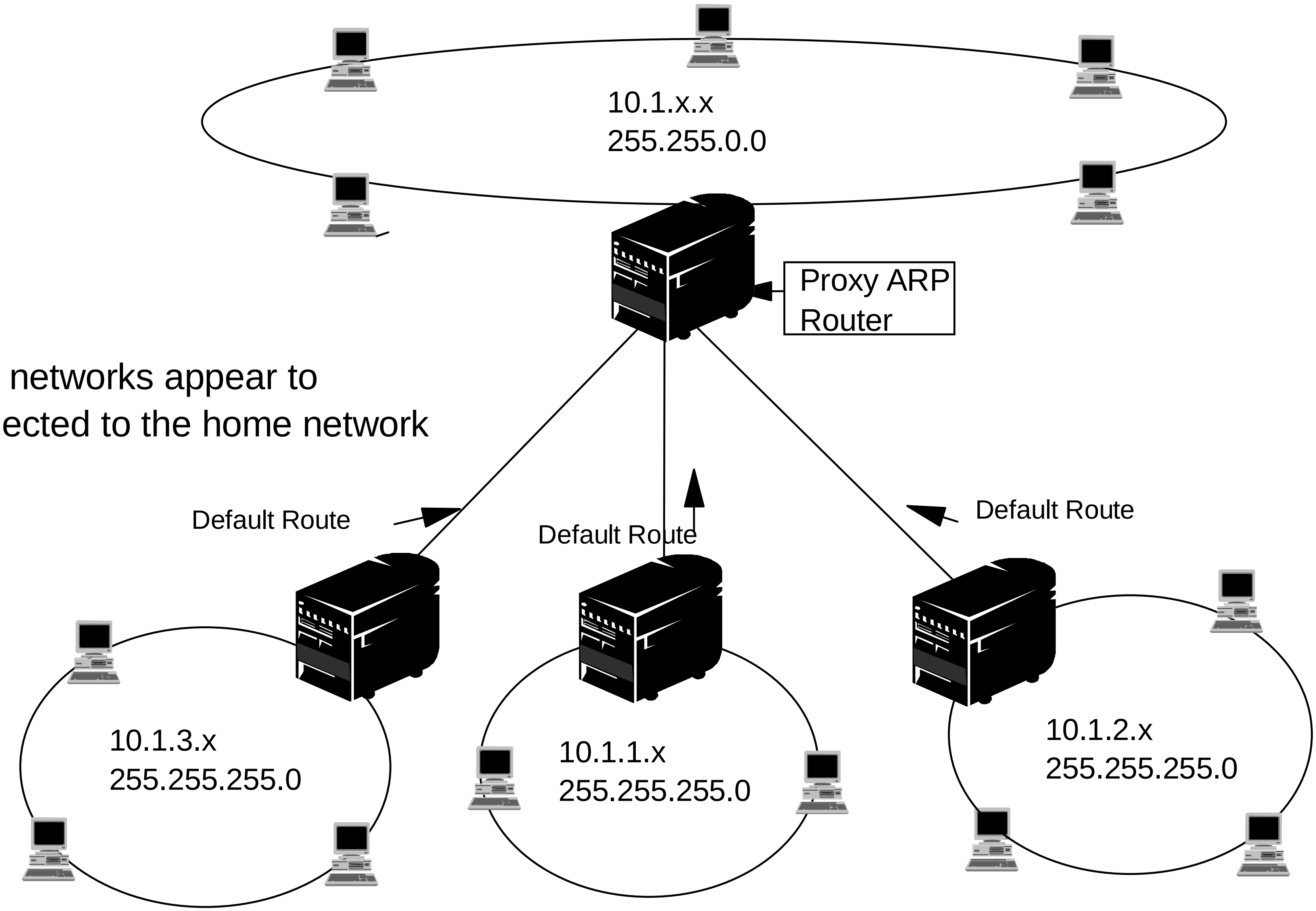
Transparent Subnetting was introduced on the AS/400 in V4R2. It is based on RFC1027, "Using ARP to Implement Transparent Subnet Gateways". It was added to provide twinax attached networkstations and PCs with twinax card access beyond the workstation controller.

The twinax "LANs" are defined in address ranges that are within the real LAN address range. Prior to V4R2 the edits on the add TCP/IP route 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 AS/400 sees this happen, it will automatically Proxy ARP for any systems that are attached behind the twinax controller.

This allows all the systems on the 10.1.x.x network to communicate with all the subnetted systems with no changes to the systems on the 10.1.x.x network.

Transparent subnetting is nothing more than extending the proxy ARP concept from proxying for a single host, to proxying for an entire subnet, or range of hosts.

Transparent Subnetting over WAN



Remote networks appear to
be connected to the home network

Notes:

The transparent subnet feature can be further expanded to handle "real" LANs that are remotely located.

In this example we have three networks that are attached to the home 10.1.x.x network via the AS/400. These networks are all defined using subnet mask that make them a transparent subnet to home network. Once again Proxy ARP will respond to any ARP request on the home network for systems in the 10.1.1.x, 10.1.2.x, and 10.1.3.x subnets. This will cause the traffic for the home network to be routed automatically to the AS/400 in the home network. This AS/400 will in turn route the data to the correct remote AS/400. The remote AS/400 will either process the data, or forward it to the correct system within the remote LAN.

The workstations in the remote LAN must have a default route that points to the remote AS/400 in their network as the first hop gateway.

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

Point to Point Connections

Point to Point connections are typically used to connect two systems together over a wide area network (WAN) connection

Point to point interfaces typically (but not always) have subnet masks of 255.255.255.255

There are two ways to configure the IP addresses for a point to point connection

- Numbered Network

- Unnumbered Network

Do not Confuse Point to Point connections with PPP. PPP is merely one type of point to point connection

Notes:

Point to point links are used for connecting two systems ,often in a Wide Area Network (WAN). Point to point connections are used across dial up lines, leased lines, as well as other types of non-broadcast networks such as X.25 and frame relay. As we will see later, these connections can also be used with Opticonnect, a non-WAN environment, to simplify routing.

There are two ways to configure the IP addresses for point to point connections. Numbered and Unnumbered

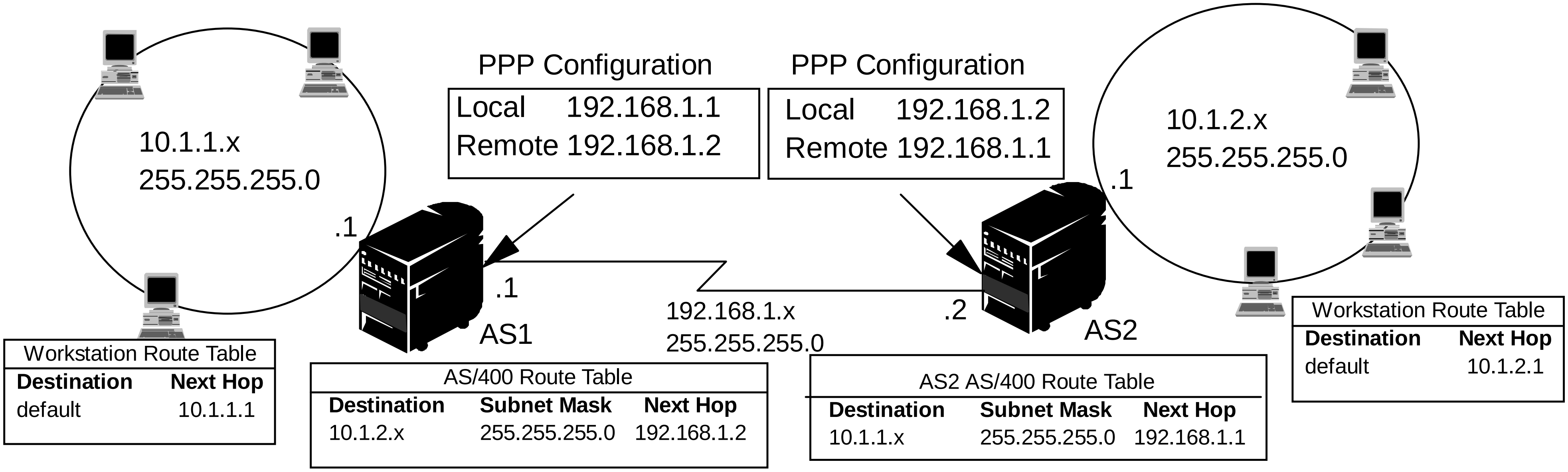
As the names imply, a numbered connection has a unique IP address defined for each interface. An unnumbered connection does not use IP additional addresses for the connecting.   In the following pages we will examine both of these techniques in more detail.

Remember that in this context, we use the term "point to point connection" to merely describe a network to which only two systems are connected, with each system denoted by an IP address. It is a "logical" model that can be applied to many different "physical" point to point link types, e.g., SLIP, PPP, frame relay, X.25 and sometimes Opticonnect. All of the concepts for point to point connections that are discussed in the subsequent pages apply equally well to any of the physical point to point link types.

Point to Point - Numbered Network

Numbered Interfaces:

- Each end of the connection has a unique IP address
- Route statements must be added to send packets to the remote system
- Addresses on the point to point link must be managed by the network administrator.
- Two addresses are used up just to connect two systems.



10.1.1.x
255.255.255.0

.1

PPP Configuration

Local 192.168.1.1
Remote 192.168.1.2

.1

AS1

192.168.1.x
255.255.255.0

PPP Configuration

Local 192.168.1.2
Remote 192.168.1.1

.2

AS2

10.1.2.x
255.255.255.0

.1

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

AS2 AS/400 Route Table

Destination	Subnet Mask	Next Hop
10.1.1.x	255.255.255.0	192.168.1.1

Workstation Route Table

Destination	Next Hop
default	10.1.2.1

Notes:

The simplest way (or so it first seems) to configure a point to point connection is by using a numbered connection.

The route selection process in the AS/400 depends on having a IP address for an interface.

A numbered connection is a point to point definition that has a unique IP address defined for each end of the connection. This has the potential of using up an entire subnet of addresses just to define an interface at each end. In a small network, these addresses are easy to keep up with and do not use many additional addresses. But in larger networks, these additional IP addresses may be unavailable or at least, difficult to manage.

A route entry is not needed if all we want to do is communicate from AS1 to AS2. But if we want to communicate with systems in the remote network (10.1.2.x), the routing entry shown in the graphic must be added to each system. This is because 10.1.2.x is not part of 192.168.1.x.

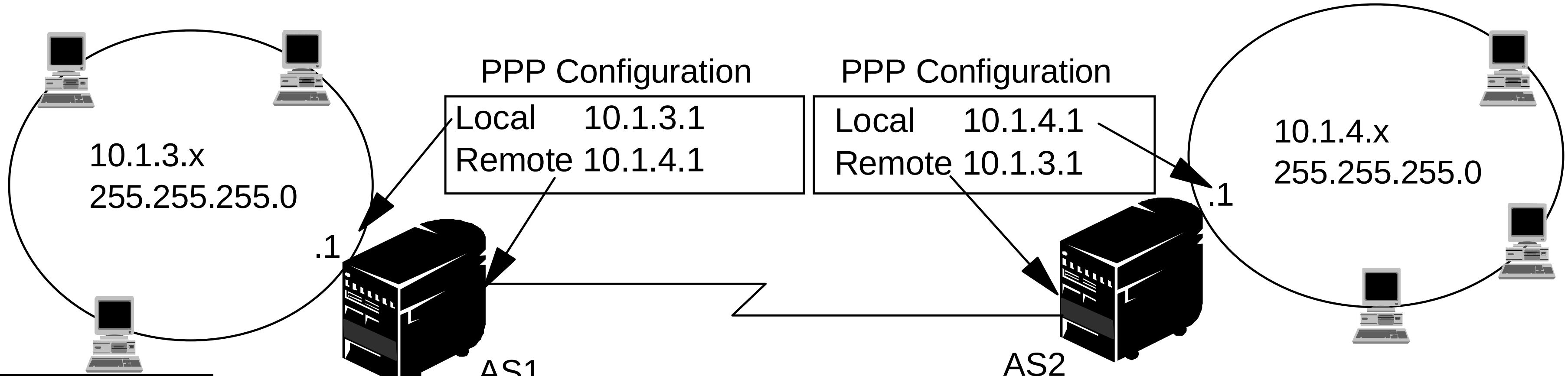
When each point to point connection is defined to the AS/400, a route entry must be made on each end to describe how to get to any network at the other end of the connection.

These addresses and routes must be managed by the network administrator.

Point to Point - Unnumbered Network

Unnumbered Interfaces:

- Point to point interface has an IP address from the remote network
- Manually added route statements are not needed in the system
- Any additional routes are automatically added when PPP or similar point-point interface is added.
- Simplifies network administration by not using extra IP addresses for the point-point link



PPP Configuration

Local 10.1.3.1
Remote 10.1.4.1

PPP Configuration

Local 10.1.4.1
Remote 10.1.3.1

Workstation Route Table

Destination	Next Hop
default	10.1.3.1

AS/400 Route Table

Destination	Subnet Mask	Next Hop
10.1.4.0	255.255.255.0	10.1.4.1

AS/400 Route Table

Destination	Subnet Mask	Next Hop
10.1.3.0	255.255.255.0	10.1.3.1

Notes:

A little more complex method of defining a point to point connection is to define a unnumbered connection. If you can get over the initial confusion you may find the unnumbered connection simpler.

The route selection process in the AS/400 depends on having a IP address for an interface. In a unnumbered connection, the point to point interface does not have a unique address. The IP address of the AS/400 interface for an unnumbered connection is actually the IP address of the remote system.

In the example shown, AS1 is connected to the 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 the 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.

Now we 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, we would simply add a LAN adapter to each system and plug the new interface into the correct LAN. If we did this AS1 and AS2 would not need any routing entries added. In our case however, the systems are in different cities so we must use a point to point connection. Even though we are using a point-point connection, we would still like to avoid creating a new network just for the point to point connection and manually adding route entries to route across that connection.

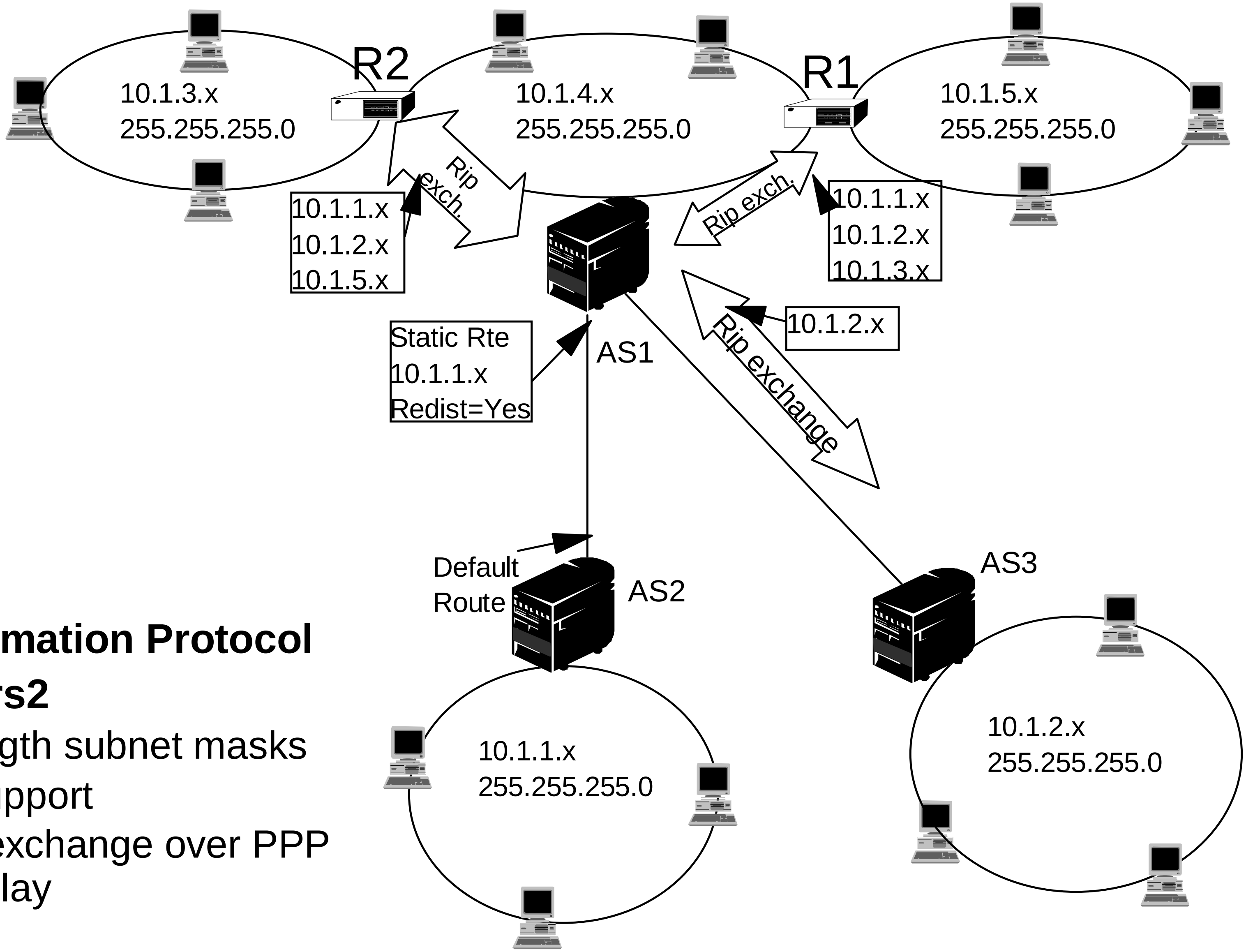
By defining the point-point connection as a unnumbered connection, we achieve the same results that we would have gotten if we could have used LAN adapters and do not have to add any route entries to the AS/400. To do this each system borrows the IP address of the remote system for use with route resolution.

Each system (AS1 and AS2) adds the remote IP address to it's route table as a local interface. The address is treated special so packets destined for that address will not be processed locally. The packets for the remote address will be placed on the interface and transported to the other end of the connection. When the packet arrives at the other end of the connection, normal packet processing is used.

AS1 looks like it has an interface in the 10.1.4.x network. AS2 looks like it has an interface in the 10.1.3.x network. Additional route table entries for AS1 & AS2 are added automatically when the unnumbered interfaces are added. For example, if these were PPP lines, the additional route is added when the PPP profile is started.

Again, no new networks, along with manually added route table entries, are created.

Dynamic Routing with/over WANs



- **Routing Information Protocol**
- **V4R2 RIP Vers2**
  - Variable length subnet masks
  - Multicast Support
  - RIP V1/V2 exchange over PPP & Frame Relay

Notes:

In V4R2 RIPv2 was added to the AS/400. This allows the AS/400 to send and receive RIP packets to update routes through the network.

In this example, a static route is added to the central system (AS1) that describes the connection to the network 10.1.1.x 255.255.255.0 via AS2. This is a static route (added by the network administrator) with Route redistribution set to *YES. This will cause this route to be shared with other routers and systems so that when they have traffic for 10.1.1.x they will route the traffic to the central AS/400 (AS1).

Or, AS1 can distribute also route information that it receives from a remote RouteD server.   In this example, AS3 has the Routed server started so it sends and receives RIP information. It sends the message that AS3 has a direct connection to 10.1.2.x.

AS1 receives this RIP packet and processes it. If it does not have a route to 10.1.2.x it will store this route. If it does have a path to 10.1.2.x that is the same number of hops or fewer, it will discard this new route information. In this example it keeps the route data.

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. AS 1 send 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.

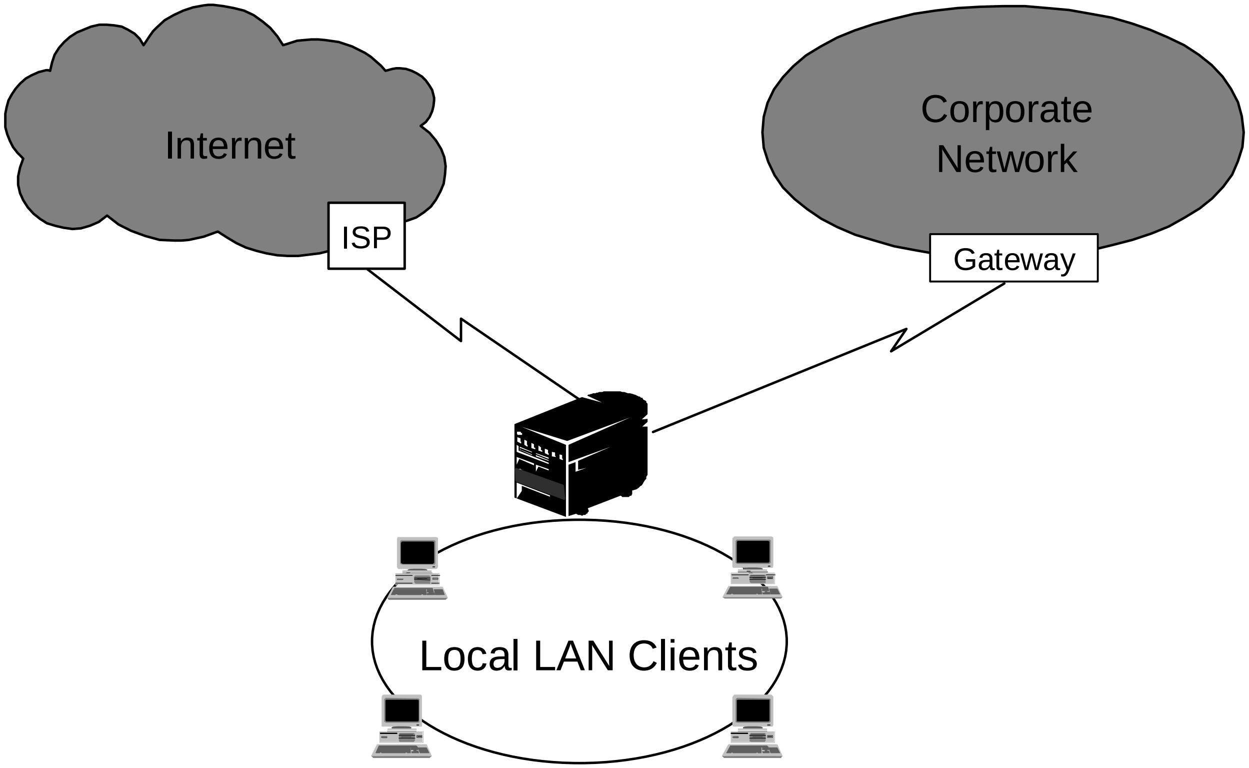
Similar information is sent to R2 and AS3.

Address Masquerading

Allows local clients to access remote networks using IP address assigned by ISP or remote gateway

AS/400 can serve as Internet Gateway for local clients

- Clients use their fixed private addresses
- AS/400 does NAT function
  - Modifies packets on the fly
- Hides Local LAN client IP addresses from remote network
- Smart/Effective use of public IP addresses
- Transparent to clients
  -so long as client IP addresses are not buried in data



Notes:

IP Masquerading is used to allow the private network to hide behind and be represented by the address bound to the public interface of the NAT machine. In most situations, this will be the address that has been assigned by an ISP which may be dynamic in the case of a PPP connection. This type of translation can only be used for connections originating within the private network destined for the outside public network. Each connection out, is maintained by using a different source (client) IP port number.

The following applies to the NAT masquerading technique.

- Swaps actual local ports with dummy ports and local addresses with the public interface address

- Provides a kind of native firewall since the external net knows nothing of your existence

- Supports TCP and UDP (recomputes TCP, UDP and IP checksums after packet is modified)

- Has a pool of port numbers available for the same physical interface

- Provides 1->M mapping of port numbers

- Maintains tables of outgoing and incoming traffic

- Connection initiated by locally attached workstations only

- Hosts at the internal network cannot advertise their addresses

- The range of port numbers for masquerading are
    - 55335 - 65335 (for TCP) and 60001 - 65335 (for UDP)

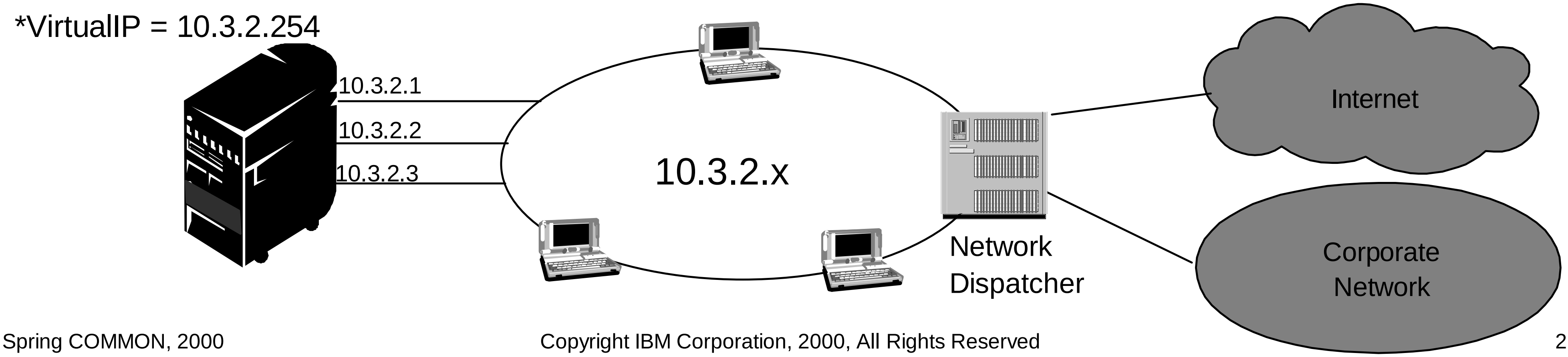
Virtual IP Addresses

Powerful tool for load balancing, fault tolerance, unnumbered interface anchor, etc.

- **Can be viewed as "primary" or "external" IP address -- "IP address of the system"**
- **Externally accessible local IP address unbound to a single physical interface**

- **VirtualIP interfaces : Not directly routable:**
  - Reachable only via indirect route through "physical IP address" (IP address of physical interface)
  - AS/400 will never answer ARP request to *VIRTUALIP address
  - ***Allows same *VirtualIP address to exist on multiple hosts***

- **Primarily for remote host access as introduced in V4R3. Recent PTFs extend to local host access**
- **VirtualIP is also supported by other IBM server platforms (AIX, MVS)**
- **VirtualIP interfaces advertsied by RIPv2,**



Notes:

Virtual IP is a very powerful new feature with many different applications. Since VirtualIP addresses are not bound to a single physical interface (i.e., line) they provide a simple way to define system wide IP addresses. They allow the AS/400 to be known by a single IP address, even when it is attached to many different networks. Some of the more common uses of Virtual IP interfaces are Load Balancing, Fault Tolerance, anchors for unnumbered interfaces and as an alternative to NAT. We will examine most of these applications in more detail in subsequent charts.

Virtual IP addresses are not directly routable, that is the AS/400 never responds to an ARP request to a VirtualIP address. For other systems to reach the Virtual IP address, they must have an indirect route defined which specifies the IP address of the physical adapter as the next hop to be used to reach the VirtualIP address. For remote clients, connecting through a gateway, only the local gateway needs to have the indirect route(s) defined for the AS/400’s VirtualIP address. On the other hand, locally connected clients would each need a host route defined for the VirtualIP address.

RIPV2 on the AS/400 will advertise networks defined by the VirtualIP. Thus, the indirect routes on the external hosts & gateways may be statically configured, or they may be added automatically through RIP exchanges

VirtualIP on the AS/400 was originally designed primarily for remotely connected clients. Configuring the VirtualIP address on the same network to which the AS/400 was directly attached was not recommended. In some cases, the wrong source IP address would be used in the outbound packets sent to the locally connected clients. However, recent PTFs extend the VirtualIP capabilities so that it can also be used with locally connected clients.

VirtualIP interfaces are also called "Circuitless" interfaces by Operations Navigator or "Loopback" interfaces by AIX

Schowler Routes

Extends Duplicate Route, Round Robin load balancing to local networks

- DRRR - Based on *"Duplicate Route Priority"* and *"Preferred Binding Interface"* parms
- Problem: Neither parameter is available on *DIRECT routes
- Solution: <u>"**Schowler**" Routes</u>
  - Special indirect route that replaces a *DIRECT route
  - Same Route Destination, Subnet Mask & TOS as equivalent *DIRECT route
  - Next Hop and Preferred Binding Interface are set to the IP address of the equivalent local interface
  - Same local network connectivity as *DIRECT route but allows user to set Duplicate Route Priority and Preferred Binding Interface options for local network load balancing
  - Requires recent (1Q00) PTFs for V4R3 or V4R4

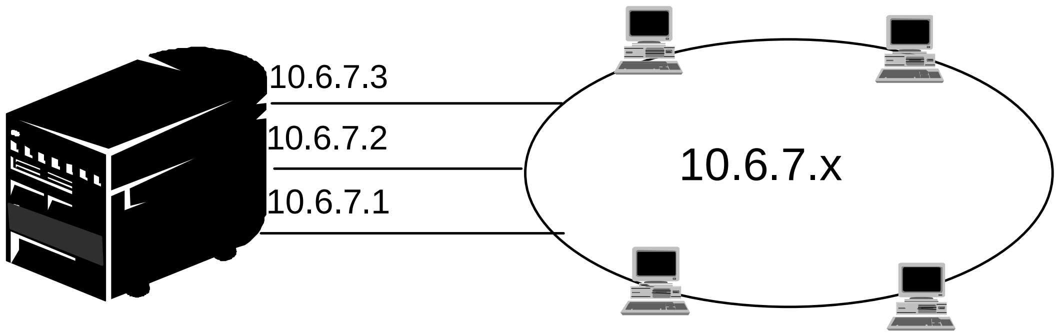
- Side Benefit: <u>Host routes may be prioritized over *DIRECT routes</u>

Standard *DIRECT Routes:

Rte Dest.	Subnet Mask	Next Hop
10.6.7.0	255.255.255.0	*DIRECT
10.6.7.0	255.255.255.0	*DIRECT
10.6.7.0	255.255.255.0	*DIRECT

Schowler Route Replacements for *DIRECTs:

Rte Dest.	Subnet Mask	Next Hop	Preferred IFC
10.6.7.0	255.255.255.0	10.6.7.1	10.6.7.1
10.6.7.0	255.255.255.0	10.6.7.2	10.6.7.2
10.6.7.0	255.255.255.0	10.6.7.3	10.6.7.3



<u>Private DNS Records:</u>
MyServer 10.6.7.3

Notes:

The Duplicate Route, Round Robin method of load balancing that was introduced in V4R2 was oriented towards remotely connected clients.   This method of load balancing is based on two indirect route parameters:

- Duplicate Route Priority
- Preferred Binding Interface.

Configuring multiple duplicate routes with the same priority caused the routes to be selected in a round robin fashion.

The problem was that these two parameters were not available for the *DIRECT routes that are automatically added when an interface is added. Thus, this form of load balancing did not work with locally connected hosts.

"Schowler" routes extend this load balancing capability to locally connected hosts.   A Schowler route is functionally equivalent to the *DIRECT route that it replaces, but since it is added just like any other indirect route, the above two load balancing parameters can now be configured by the user.   Schowler routes have two special characteristics:

- The same route destination, subnet mask and TOS setting as the equivalent *DIRECT route
- The Next Hop and Preferred Binding Interface IP addresses are both set to the IP address of the associated local interface.

When the Duplicate Route Priority is set greater than the default of 5, the equivalent Schowler routes are selected in a round robin fashion, identical to what can be done with other indirect routes.

In the previous chart, we have 3 interfaces configured, connecting the AS/400 to the 10.6.7.x network, 10.6.7.1, 10.6.7.2 and 10.6.7.3.   The first box in the lower left shows the standard *DIRECT routes that are automatically added with the interfaces. However, by adding 3 equivalent Schowler routes, shown in the lower box, the three *DIRECT routes disappear and are replaced by the Schowlers.

One final use of Schowler routes is to reverse the default AS/400 TCP/IP routing logic that always prioritizes *DIRECT routes over any indirect routes, even *HOST routes.   By replacing the *DIRECT routes with Schowler routes, no "highest priority" *DIRECT routes will be found during route lookup.   All candidate routes are now indirect, and prioritized by subnet mask. Thus, a *HOST route, with a subnet mask of 255.255.255.255 will be considered the highest priority route.

Explicit Route Binding

Provides user control of "*Route to Interface*" Binding

Pre-V4R2: No user control of indirect route - interface binding. Two common complaints:

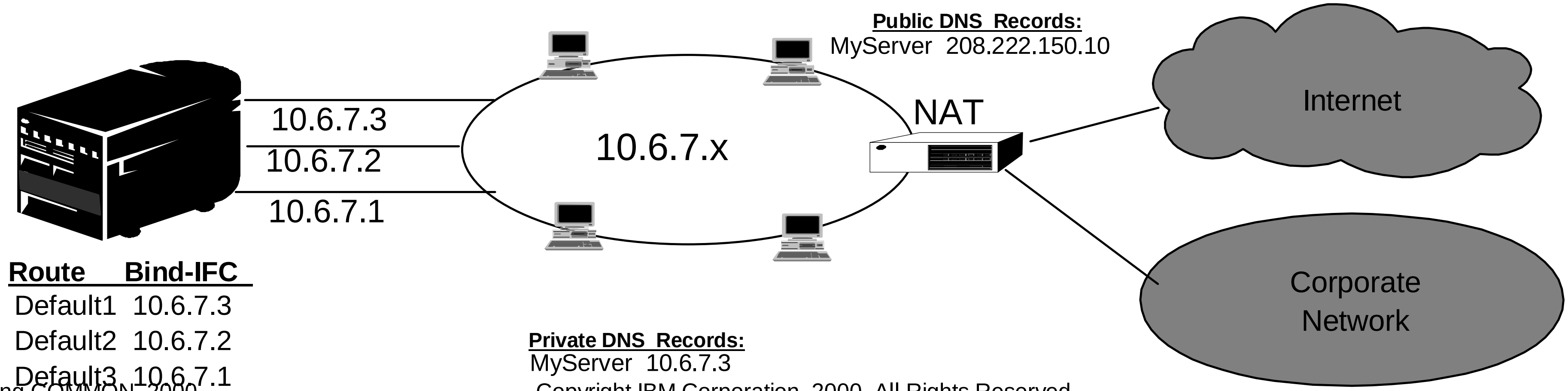
1) "Response packets are not sent back out the same interface that the request was received on"

- Caused by all indirect routes being bound to the 1st "acceptable" interface found.
- <u>All routes bound to 1st interfaces, others had no routes ==> no connectivity to remote host</u>

2) "The interface which is picked for sending the response is not consistent"

- Interface to which routes were bound dependent upon variable STR/END IFC order

V4R2 Solution: "*Preferred Binding Interface*" added to Add Route function



Route	Bind-IFC
Default1	10.6.7.3
Default2	10.6.7.2
Default3	10.6.7.1

Notes:

In this example we have three adapters on our system all connected to the same LAN segment. Each interface has connectivity to the same remote networks

How do we configure this:

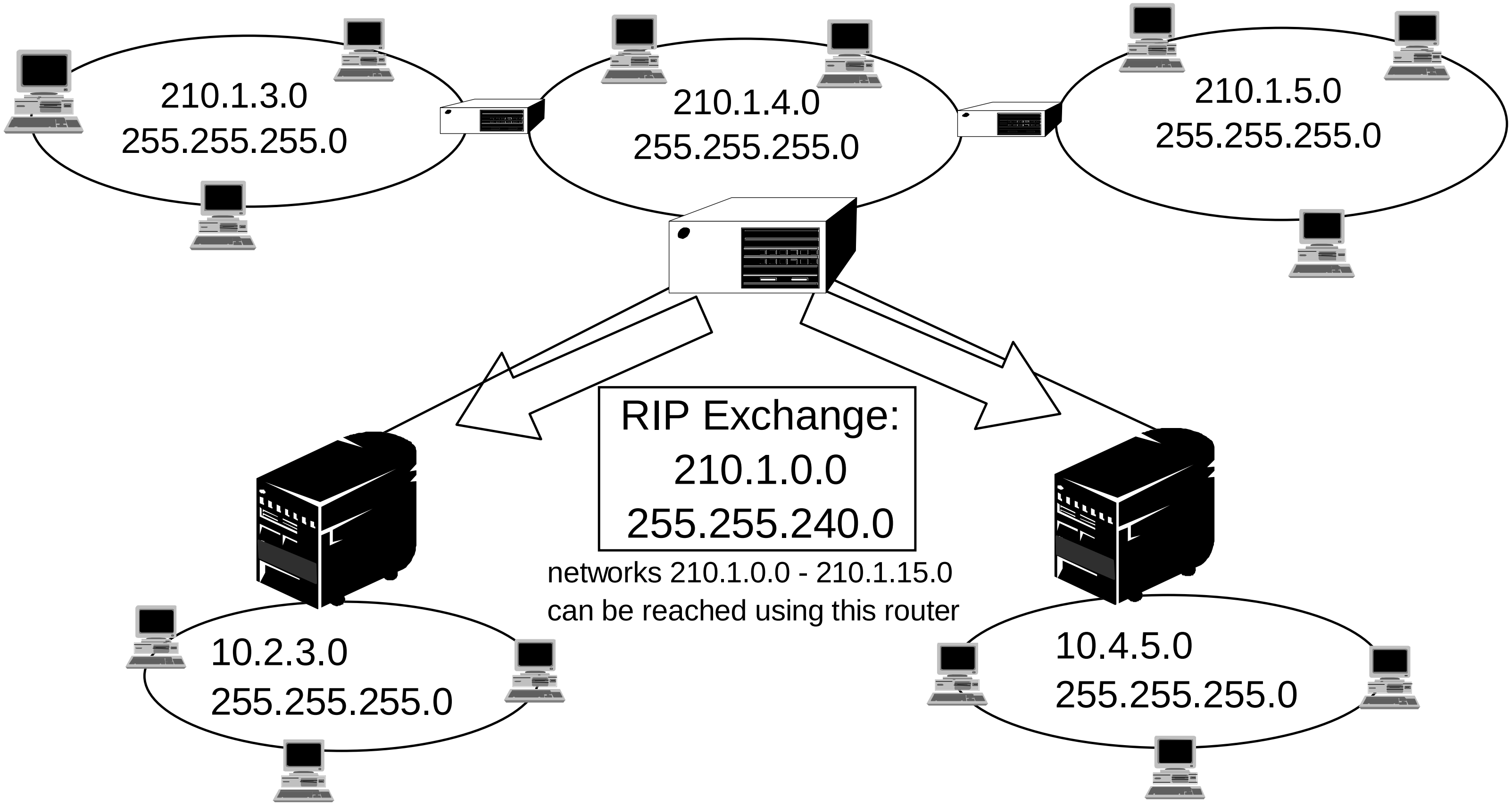
Two new parameters were added to the ADDTCPRTE command in V4R2 (also to the operations navigator GUI). One of these is the "Preferred Binding Interface" (BINDIFC).   The preferred binding interface allows the user to explicitly bind a route to a specific interface by IP address, rather than have it bound to the first one the system sees.

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

The other new parameter added to the ADDTCPRTE command in V4R2 is the "Duplicate Route Priority" (DUPRTEPTY).   This parameter is very important for load balancing and will be discussed later .

CIDR / Supernetting

- ► **CIDR = Classless Interdomain Routing**
- ► **Aggregates multiple, contiguous class C networks into a single class B network/route**



Notes:

CIDR or supernetting is a way to combine several class C network address ranges into a single network or route. This was implemented on the AS/400 in V4R3. In the past you were required to enter a subnet mask that was equal to or greater than the mask required for the network class. For Class C addresses this meant a subnet of 255.255.255.0 was the biggest (253 host) that could be specified. To conserve IP addresses, when companies needed more than 253 host in a network the Internic was issuing several class C addresses. This would make the configuration of routes and other things difficult. Supernetting 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 the following 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, I could ask my ISP to make them a supernet by using the subnet mask 255.255.252.0. This mask would combine the four network into one for routing.

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

Applications

Load Balancing

Load Balancing

- **Load Balancing => Splitting workload across :**
  - ➤ Multiple interface adapters
  - ➤ Multiple host servers
  - ➤ Many other options possible

- **<u>TCP/IP Based Load Balancing techniques:</u>**
  - ➤ ***DNS based round robining***
  - ➤ ***Duplicate Route based round robining***
  - ➤ ***Dispatcher Load balancing with Virtual IP***

- Cost effective approach to increase performance
  - May only require additional I/O adapters
- Same techniques can split traffic over multiple interfaces or servers

➤ Transparent to remote users

Notes:

Load balancing can mean many different things to different people. This graphic summarizes some of the different things people mean when they ask about load balancing.   The subsequent pages address load balancing solutions that are based upon TCP/IP routing and naming techniques.

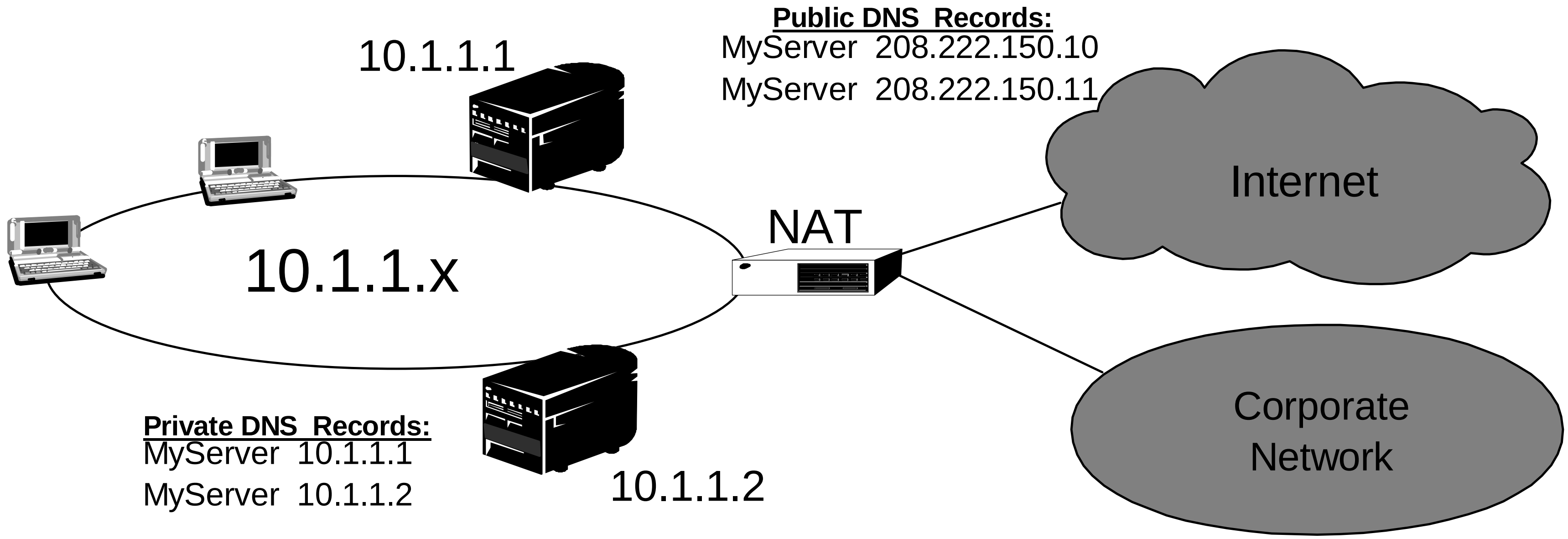
Depending on the version and release of the system, there are different tools to achieve load balancing.

In the following pages we will examine three different methods of network load balancing.   However the methods are not mutually exclusive

DNS Round Robining - Multiple Hosts

Inbound directed load balancing

- Multiple host IP addresses configured in DNS for a single host server name
- DNS alternates the host IP address returned to successive client host name resolve requests (A Record queries)



Public DNS Records:
MyServer 208.222.150.10
MyServer 208.222.150.11

Private DNS Records:
MyServer 10.1.1.1
MyServer 10.1.1.2

Pro: - Common DNS function
- V4R2 - Integrated DNS

Con: - IP address caching by client
- Connection, not load, based

Notes:

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 (A) record for the system name. In this example each address corresponds to a different system. This allows users to provide load balancing across two separate systems.

In the case of clients on the private network, they will receive a different address for each request.

This is a common DNS function. In V4R2 of OS/400 DNS was added to the operating system.

Notice that the public DNS also has two address entries. These addresses are translated using static NAT so that clients on the Internet can reach the two systems.

If the customer has programs that depend on getting to a specific system, or 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 page URLs for example could point to myserver2 after the first contact.

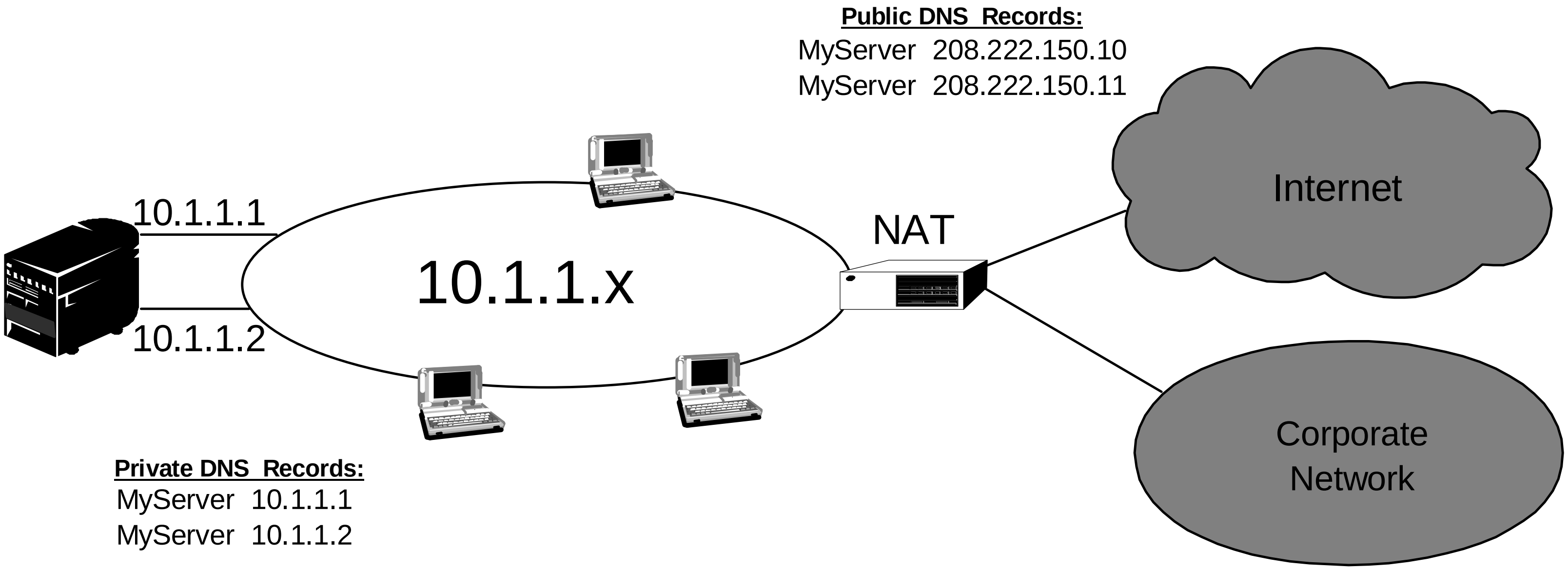
This type of load balancing provides balancing by the connect request. In most cases once a client has resolved the address the client caches the address and will not ask again. This type of load balancing does not consider the amount of traffic going to each system. On average it should do A 50 - 50 split between the systems.

This type of load balancing is oriented towards splitting the inbound traffic to the AS/400. However, if the AS/400 is being used primarily for server applications, outbound replies will be sent back out the same interface that received the inbound request.   Thus, a degree of load balancing of the outbound traffic also results.

DNS Round Robining - Multihomed, Single Host

Inbound directed load balancing

- ►Multiple host IP addresses configured in DNS for a single host server name
- ►DNS alternates the host IP address returned to successive client host name resolve requests (A Record queries)



Pro: - Common DNS function
- V4R2 - Integrated DNS

Con: - IP address caching by client
- Connection, not load, based

Notes:

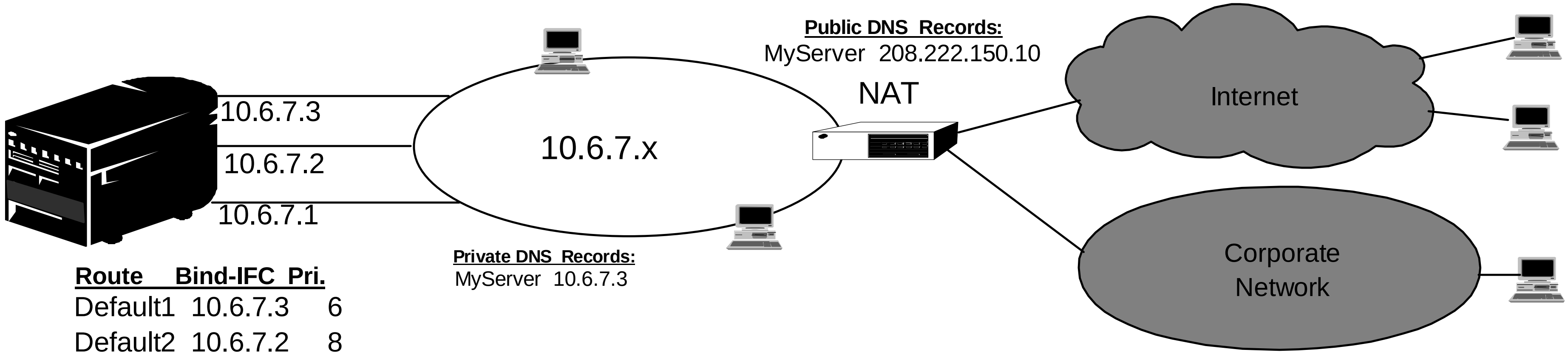
This is basically the same as the previous example. The only difference is that we have two adapters on one system rather than one adapter on two systems.

The same issues and concerns apply.

Duplicate Route Round Robining

Outbound load balancing across a route/interface pool

- ►Route Binding: Outgrowth of V4R2 enhancement to allow user control of interface to route binding
- ►**Allows user to create a "pool" of equivalent routes, each bound to a different interface**



Public DNS Records:
MyServer 208.222.150.10

Private DNS Records:
MyServer 10.6.7.3

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

Pro: - Total AS/400 solution
- More flexibility than DNS
- Good for HTTP, Telnet
- **Extended to local clients with Schowler route PTFs**

Con: - Connection, not load, based
- <u>Initially</u>, not active for local clients
- No effect on inbound requests

Notes:

This function provides **Outbound** load balancing across multiple interfaces.

In this example we have three adapters on our system all connected to the same LAN segment. We have set one of the adapters up as inbound only and set the other two adapters up as outbound.

How do we configure this:

Two new parameter was added to the ADDTCPRTE command in V4R2 (also to the operations navigator GUI) one called Duplicate route priority (DUPRTEPTY), the other called Preferred binding interface(BINDIFC). If the value for DUPRTEPTY is left at the default value of 5 nothing happens. If a value greater than 5 is set, then routed connections will round robin between all the 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 this example we want to use the 10.6.7.3 interface as our primary "inbound" adapter. This is accomplished by configuring only this IP address in the DNS for our host and by configuring a lower DUPRTEPTY of 6 in the default route that is bound to the 10.6.7.3 interface. We configured the other two adapters with a DUPRTEPTY of 8. Because the DUPRTEPTY on one adapter is 6 it will not be selected for outbound connections unless all the DUPRTEPTY = 8 interfaces are down. You should put all the outbound interfaces ate the same priority. If you put some a one value and some at another value, only the highest value interfaces will be used.

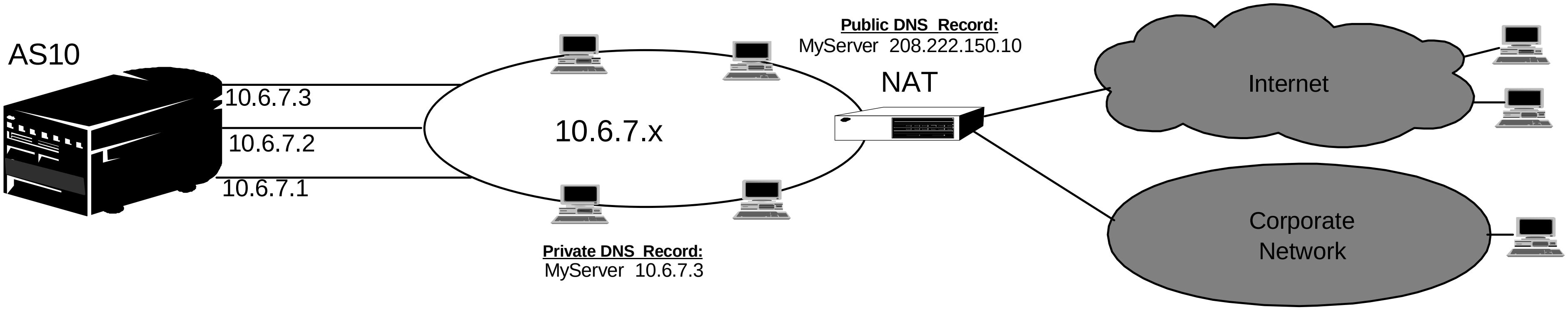
Notice that the DNS is pointing to the 10.6.7.3 interface making it the inbound interface.

Even if you decide not to use DUPRTEPTY you should always define a default route out of the system on each interface by using the BINDIFC parameter.

Finally, with the recent V4R3 and V4R4 PTFs that allow Schowler routes, this same method of load balancing works equally well with local or remote clients.

Duplicate Route Round Robining - Local Networks

- ► **Recent V4R3 & V4R4 PTFs extend DRRR to local networks**
- ► **Identical load balancing capabilities for local networks via Schowler routes that exists for remote networks.**



Sample Route Table for local and remote network DRRR load balancing

Rte Dest.	Subnet Mask	Next Hop	Preferred IFC	Rte Pri.	
10.6.7.0	255.255.255.0	10.6.7.1	10.6.7.1	9	Schowler Routes
10.6.7.0	255.255.255.0	10.6.7.2	10.6.7.2	9	
10.6.7.0	255.255.255.0	10.6.7.3	10.6.7.3	5	
Default	*NONE	10.6.7.250	10.6.7.1	8	
Default	*NONE	10.6.7.250	10.6.7.2	8	
Default	*NONE	10.6.7.250	10.6.7.3	6	

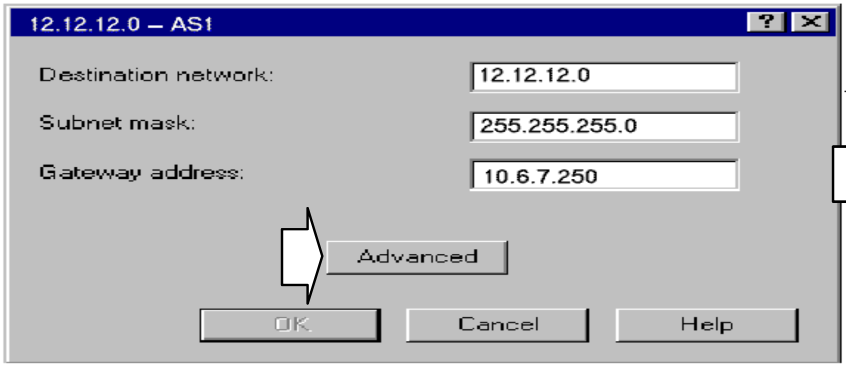
Notes:

This shows an example of using the same Duplicate Route Round Robin load balancing function for locally attached network traffic as well as for remote networks.

Three Schowler routes are configured that replace the *DIRECT routes.   Since only the 10.6.7.3 IP address is defined in the DNS for AS10, incoming packets should all be directed to this interface.   We would therefore like to direct all outbound traffic through the other two interfaces, 10.6.7.1 and 10.6.7.2.   To do this, we configure higher route priorities on the Schowler routes for 10.6.7.1 and 10.6.7.2.   When routes are selected for outbound traffic, only these two interfaces will be selected.

For the remotely connected clients, the default route priorities are configured as they were on the previous page.

Duplicate Route Configuration

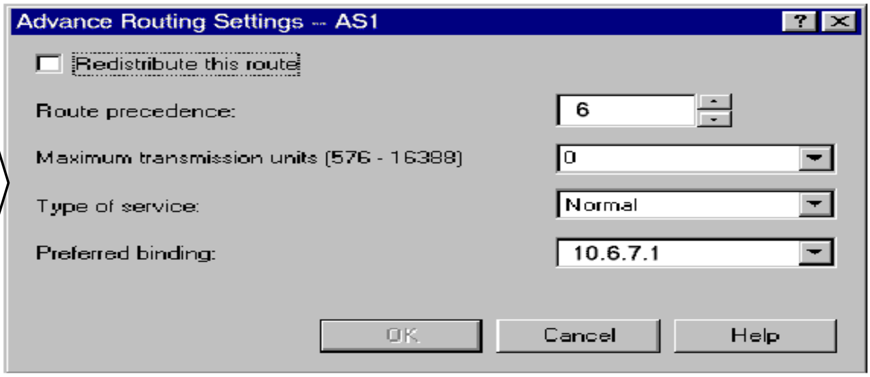


12.12.12.0 – AS1

Destination network: 12.12.12.0

Subnet mask: 255.255.255.0

Gateway address: 10.6.7.250



Advance Routing Settings -- AS1

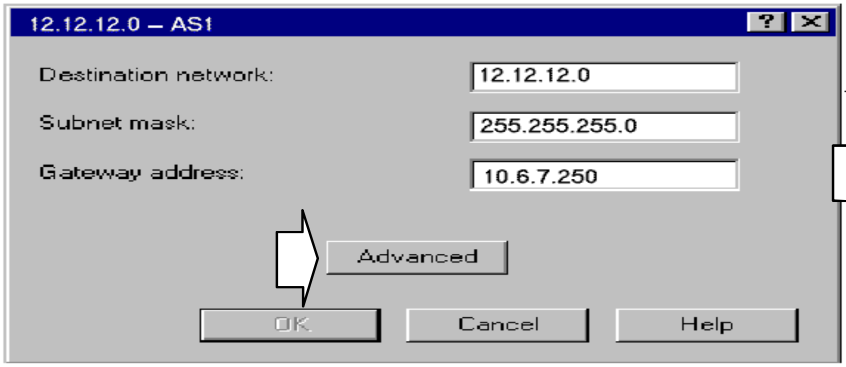
□ Redistribute this route

Route precedence: 6

Maximum transmission units (576 - 16388): 0

Type of service: Normal

Preferred binding: 10.6.7.1

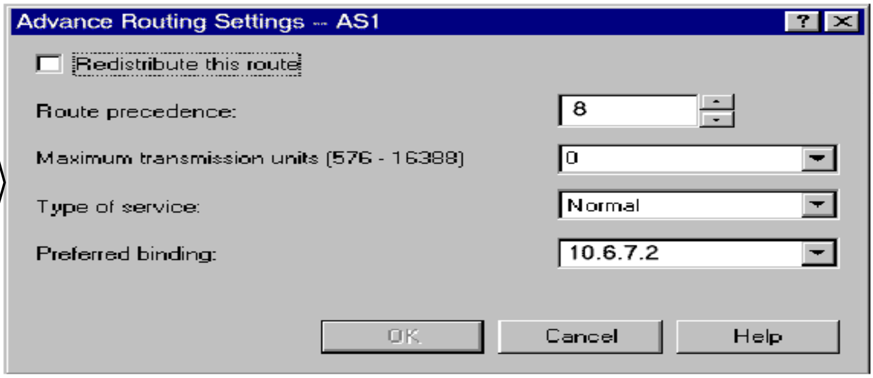


12.12.12.0 – AS1

Destination network: 12.12.12.0

Subnet mask: 255.255.255.0

Gateway address: 10.6.7.250



Advance Routing Settings -- AS1

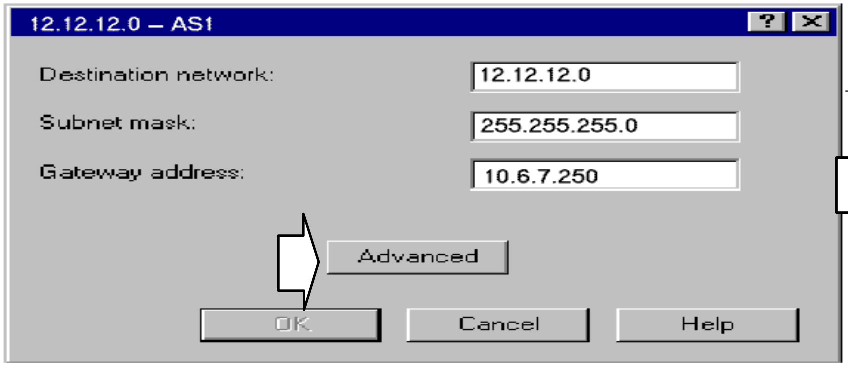
□ Redistribute this route

Route precedence: 8

Maximum transmission units (576 - 16388): 0

Type of service: Normal

Preferred binding: 10.6.7.2

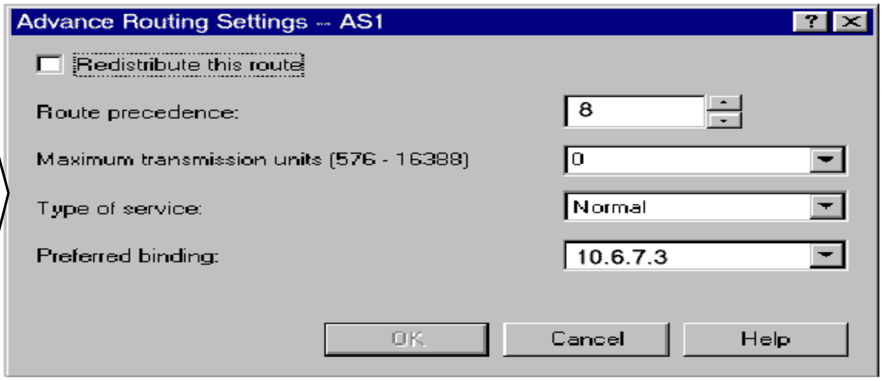


12.12.12.0 – AS1

Destination network: 12.12.12.0

Subnet mask: 255.255.255.0

Gateway address: 10.6.7.250



Advance Routing Settings -- AS1

□ Redistribute this route

Route precedence: 8

Maximum transmission units (576 - 16388): 0

Type of service: Normal

Preferred binding: 10.6.7.3

Notes:

This shows the Operations Navigator screens used to specify the route precedence (DUPRTEPTY) and Preferred binding (BINDIFC). You must click "advanced" to see the options.

Load Balancing with Virtual IP

Enables load balancing by front end dispatcher across servers/interfaces

- External front end dispatcher, like IBM Network Dispatcher, has multiple paths configured to the Virtual IP address.
- Paths are differentiated by unique next-hop IP address, i.e., IP address of physical interfaces that are connected to the network
- Paths may identify multiple host servers or multiple interfaces on single server
  - **Since *VirtualIP addresses are not directly routable (i.e., will never answer ARP request to a *VirtualIP address), the same *VIRTUAL IP address may be defined on multiple servers**
- Dispatcher monitors load across each path and dispatches new connections appropriately.
  - **IBM Network Dispatcher - Powerful load balancer supporting many customizable modes, options.**
  - **(For details, see: http://www.software.ibm.com/network/dispatcher/)**
- If *VirtualIP address is not on local network, it can be advertised out by AS/400 RouteD server
  - **Totally automatic route configuration for dispatcher and local hosts**

Notes:

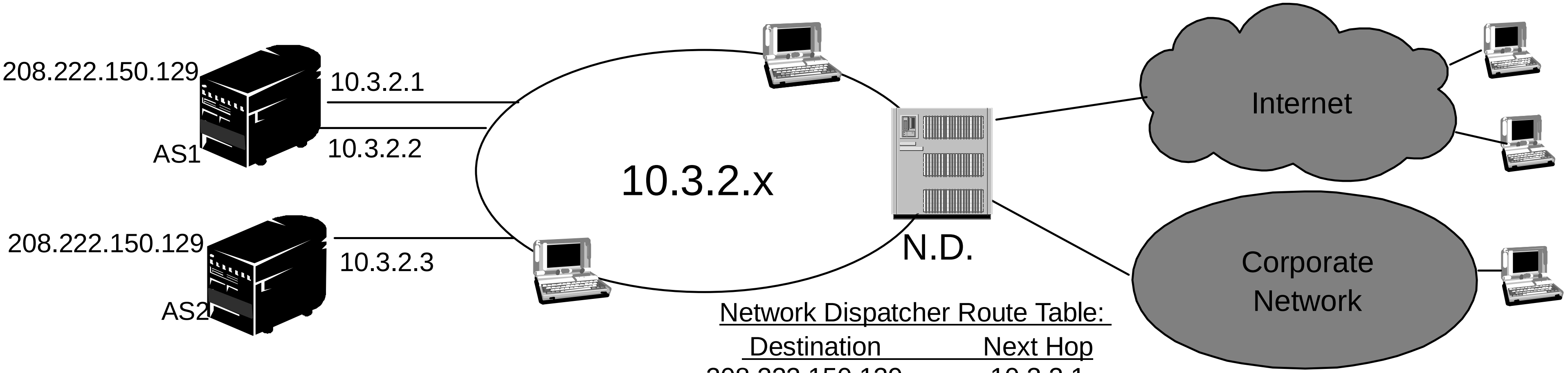
In V4R3, you have an additional tool to use for load balancing, Virtual IP.

Virtual IP addresses allow you to assign an address to the system rather than a specific interface.   Or you may define the same address to multiple servers. This allows many new options for load balancing.

*VirtualIP interfaces, alone, cannot perform any load balancing.   But *VirtualIP addresses allow the AS/400 to be used in multi-server environments where load balancing is done by a front end dispatcher like the IBM Network Dispatcher.   The IBM Network Dispatcher provides extensive and powerful user controls on how network connections should be distributed across multiple interfaces or servers. For additional details, see: http://www.software.ibm.com/network/dispatcher/

It was this type of load balancing that IBM used to maintain the web servers during the Atlanta Olympic games and during the 1998 Tour de Suisse

Load Balancing with Virtual IP



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
Inbound and outbound load balancing
More sophisticated than simple round robining

Con: - Requires external dispatcher

- **Local clients can still connect to servers via physical interface addresses (10.3.2.x) or if host route has been configured, may alternatively access servers via *VirtualIP address (208.222.150.129)**
- **Remote clients see only a single virtual server, identified by *VirtualIP address of 208.222.150.159, accessible only through dispatcher**
- **If Virtual IP address is reachable via multiple interfaces on the same host (e.g., AS1 above) , duplicate indirect routes should be defined, each bound to a different physical interface**
- **Guarantees same connectivity via all adapters**

Notes:

In this example we have used virtual IP addresses to assign the same address to two servers. AS1 has two interfaces to the LAN with real IP addresses. AS2 has one interface.

The local clients still connect using the real IP addresses or the *VirtualIP address.   However, if the latter is desired, the clients need to have host routes defined for the *VirtualIP address.

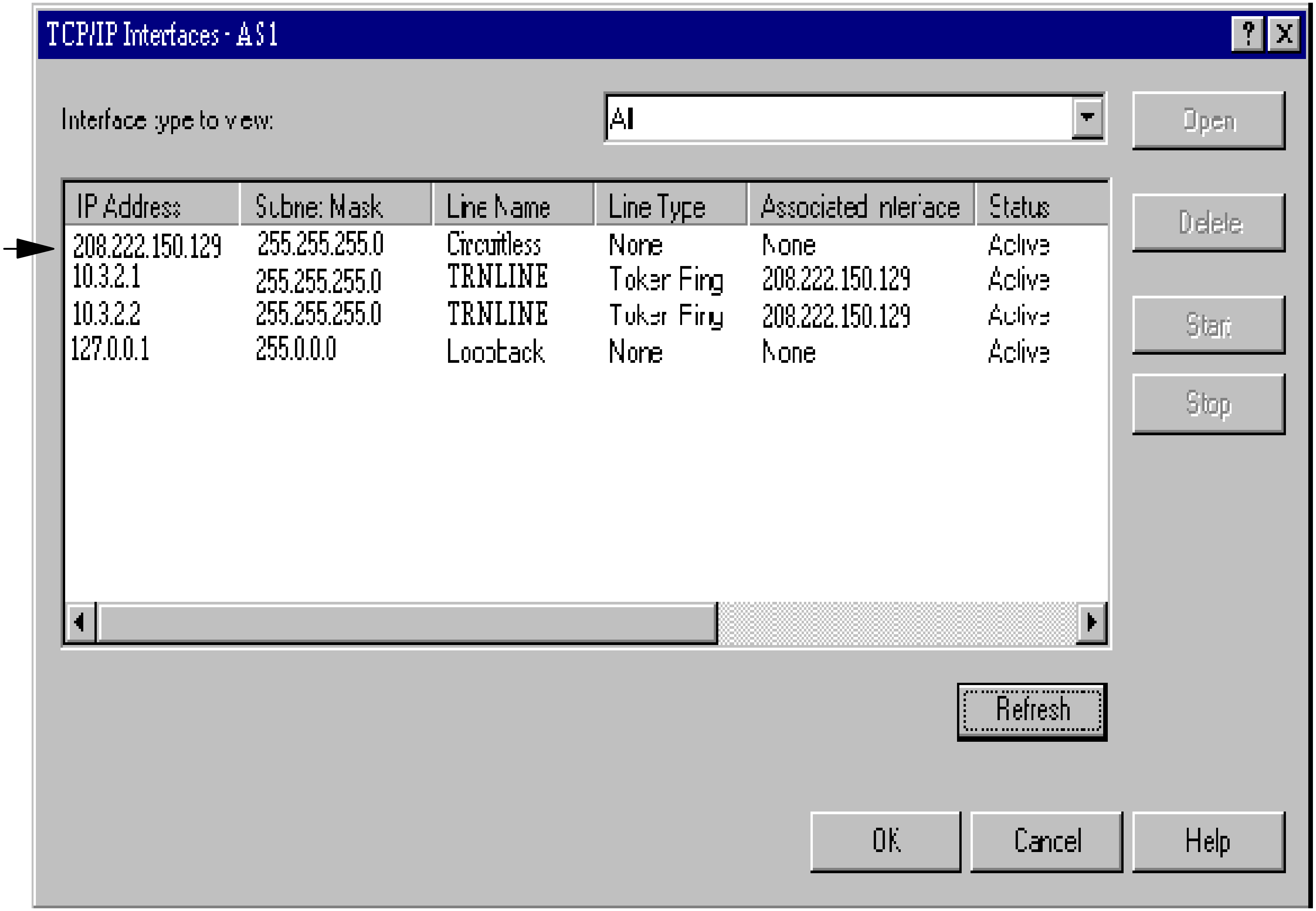
If load balancing is needed for local clients, any of the three load balancing options could be used.   Multiple DNS entries could be defined for the same server name. Or Duplicate Routes could be defined as discussed in earlier charts. Or finally, if the clients host route pointed back to the dispatcher, local traffic can be balanced just like remote network traffic

AS1 should have multiple default routes defined with a DUPRTEPTY value of 6 for all the routes and a BINDIFC of 10.3.2.1 and 10.3.2.2. This will allow the outbound traffic to go out the way it came in. If you want to set aside on of AS1's interfaces for inbound and one for outbound change the DUPRTEPTY of one of the routes and remove the IP address from the Network Dispatcher table.

This solution does require an external load balancing box such as the system shown here running Network Dispatcher (ND). The ND code does load balancing based on usage not just connections. SecureWay Network Dispatcher runs on AIX, Sun Solaris, and Windows NT. .

If you are planning to use the same *VirtualIP address for local as well as remote client access, you should review the rules for selecting source IP addresses, documented in the Appendix.

Virtual IP Configuration



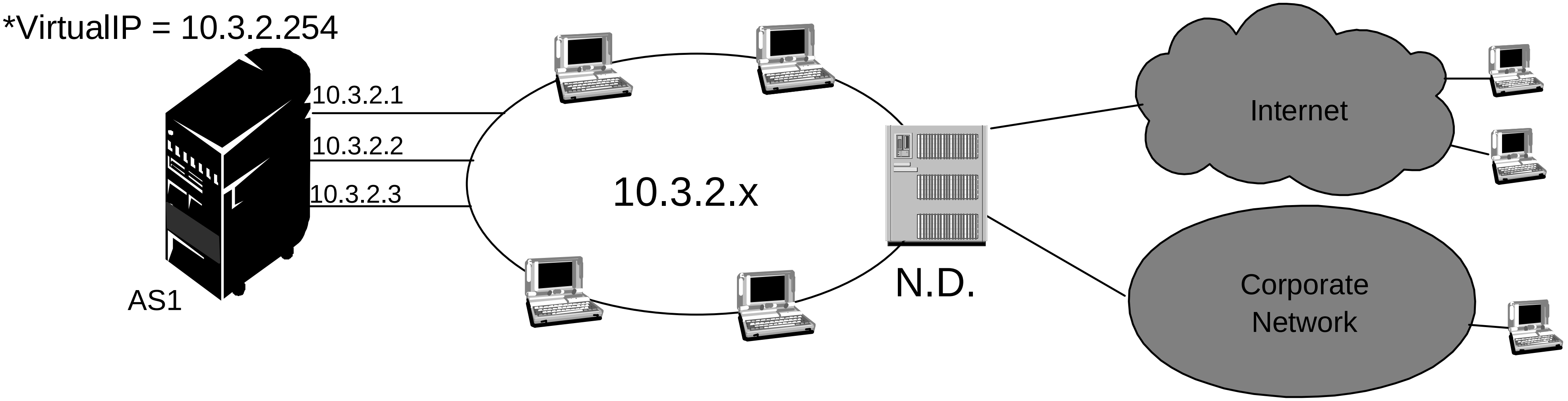
VirtualIP Interface

IP Address	Subnet Mask	Line Name	Line Type	Associated Interface	Status
208.222.150.129	255.255.255.0	Circuitless	None	None	Active
10.3.2.1	255.255.255.0	TRNLINE	Token Ring	208.222.150.129	Active
10.3.2.2	255.255.255.0	TRNLINE	Token Ring	208.222.150.129	Active
127.0.0.1	255.0.0.0	Loopback	None	None	Active

Notes:

This is a sample screen shot of AS1’s configuration. Notice that both real ports specify an associated local interface of 208.222.150.129 (the *VIRTUALIP address interface).

Load Balancing with Virtual IP - Local Networks



- ►Recent PTF extends access *VirtualIP interface to local clients
- ► Local clients can still connect to servers via physical interface addresses (10.3.2.1, 10.3.2.2 or 10.3.2.3) or if host route has been configured, may alternatively access servers via *VirtualIP address (10.3.2.254)
- ►Remote clients see only a single server, identified by *VirtualIP address of 10.3.2.254, accessible only through dispatcher
- ►*VirtualIP interface should have a subnet mask of 255.255.255.255 to avoid routing ambiguity
- ►**Removes the need to create a separate network just for the *VirtualIP address**

Notes:

The preceding page shows that the use of *VirtualIP is no longer restricted to remotely connected clients.   Assuming that the proper host routes are configured on the external hosts, *VirtualIP can be used without having to create a separate network just for the *VirtualIP interface.

Fault Tolerance

Notes:

The following pages will examine how AS/400 routing capabilities can be used for greater network fault tolerance

Fault Tolerance

What:

24x7 System Availability

How:

- Increase availability & reliability of individual components
- Deploy redundant componentry :
  - Routers, Networks, Interface adapters, Hosts, etc.
  - Requires automatic detection and rerouting function

AS/400 Routing Techniques :

- Dead Gateway
- Virtual IP
- IP Address Takeover

Notes:

Fault tolerance means keeping your systems available 24 hours / day, 7 days a week. This can be accomplished by many different techniques. One way is obviously to increase the reliability and availability of each individual component.

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

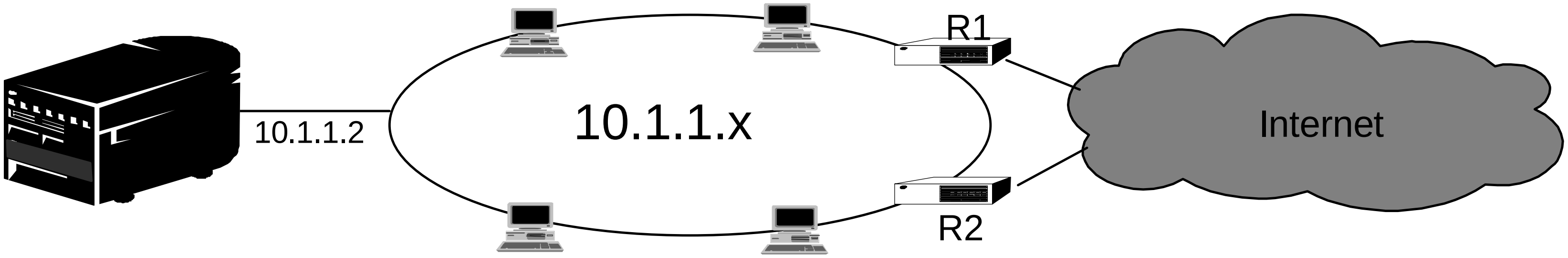
Various AS/400 routing capabilities can play a role in the design a such redundant networks. Among them are:

- Dead Gateway
- *VirtualIP addresses
- IP Address Takeover

Dead Gateway

Dead Gateway: AS/400 mechanism to detect local router failure, and reroute, if possible

- Triggered by ARP failure or excessive TCP retries
- Suspect gateway is Pinged -- if no Ping reply:
  - Gateway is marked down, all affected routes marked INACTIVE
  - Connections rerouted through alternate path, if one exists
- Slow Pings continue in order to detect gateway recovery



Router R1 fails:

Assuming redundant route through R2 exists, R1 connections are transparently rerouted through R2

Slow Pings will detect R1's recovery,
Routes defined through R1 will be reactivated
Active connections stay through R2, new connections will be established over R1 again

Note: If router is acting as a firewall and has Ping replies disabled, <u>Dead Gateway is fooled:</u>
Dead Gateway is fooled into thinking router is down, when in fact, it is still active (Fixed in recent PTFs)

Notes:

Per RFC 1122, the AS/400 employs a Dead Gateway detection mechanism to detect failures in locally connected gateways. Dead Gateway is triggered by either of two events:

Repeated ARP requests to a gateway fail to receive an ARP reply
Excessive TCP retries with a remote host

In the first case of an ARP failure, the gateway is immediately considered down, all routes defined through this gateway are marked INACTIVE and Dead Gateway starts slow pinging the gateway. In the second case, excessive retries may indicate a problem with the local gateway, or the problem may be further downstream. Therefore, before any routes are marked inactive, the gateway is quickly Pinged a few times. If no Ping reply is received, the gateway is considered down, all affected routes are marked INACTIVE, and slow pinging begins.

When routes are marked INACTIVE, TCP/IP will attempt to reroute connections over an alternate route. In the previous chart, when router R1 fails, if a route exists that goes through router R2, connections will get rerouted.

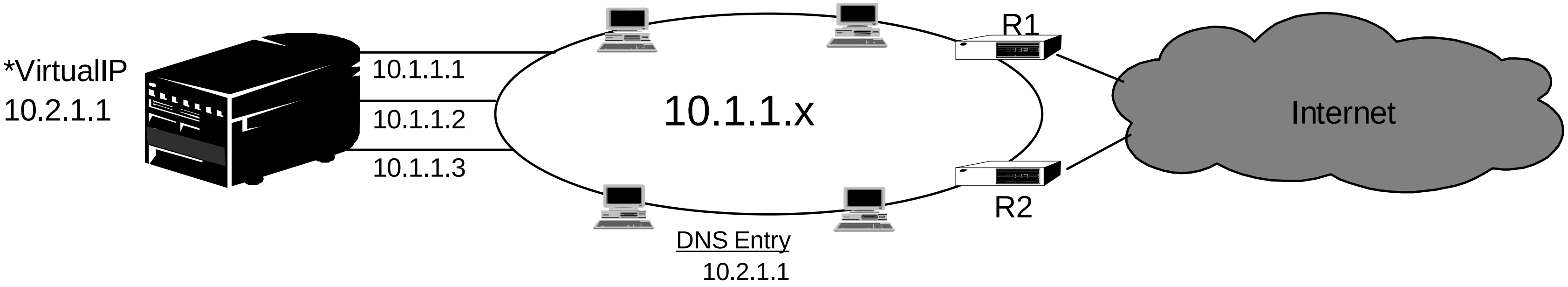
When R1 comes back, active connections will stay routed through R2. However, new connections will be routed over R1, just like prior to the failure.

Finally, as stated above, Dead Gateway uses Ping responses to determine whether a gateway is alive. But if the gateway has Ping replies disabled, Dead Gateway will incorrectly conclude that a gateway is down. This usually happens as a result of TCP initiated Dead Gateway. One way to fix this is to re-enable Ping replies in the gateway. If that cannot be done, two AS/400 circumventions are available via PTFs. In the V4R3 and V4R4 PTFs for APAR MA20996, TCP initiated dead gateway is disabled. Dead Gateway will only be invoked upon an ARP failure. Or in V4R4, PTF MF23501 changes the Dead Gateway to be ARP, rather than Ping based. Routes will not be marked INACTIVE unless an ARP failure has occurred

*Fault Tolerance using *VirtualIP*

Use *VirtualIP to provide continuous availability even through an interface failure

- **What if, instead of an external router, an interface adapter fails?**
- Unbound routes automatically switched to active interface. (Routes explicitly bound to interface not moved)
- **But IP address of failed interface is still unavailable -- "system still appears down"**
- **<u>Solution:</u> Use a "Virtual IP" address as the primary system address to which external users connect**
  - Primary IP address of system remains active as long as system is active
  - **<u>System stays accessible so long as at least one physical interface remains active</u>**



Interface 10.1.1.1 fails:

Any connections to 10.1.1.1 are lost, connections to 10.1.1.2, 10.1.1.3 remain active.

<u>But connections to 10.2.1.1, the *Virtual IP address, remain active , system stays available</u>

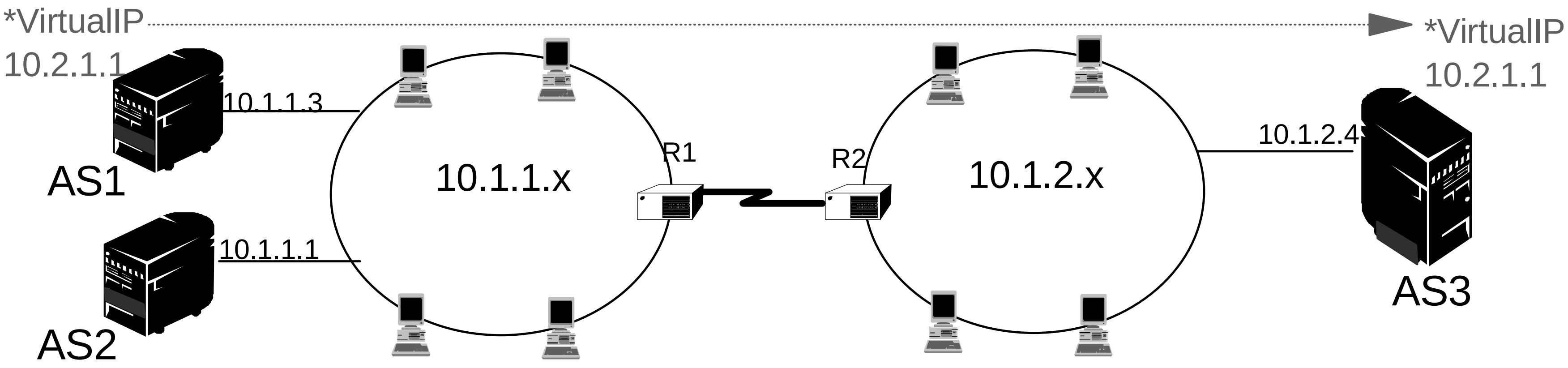
Notes:

This chart demonstrates yet another powerful use of *VirtualIP addresses.   Here, we define a *Virtual IP address as the primary address for the system.   In the DNS, only the *VirtualIP address is defined.   All external users access the system via the 10.2.1.1 *VirtualIP address.

If any of the local interfaces fail, the system remains accessible so long as at least one interface remains active. Connections can be transparently re-routed through any of the available interfaces as needed. The advantage of this is that because a *VirtualIP address is not tied to a hardware adapter, it remains active so long as TCP/IP is active.

IP Address Takeover using *VirtualIP

- **What if entire system is taken down?**
- **V4R4: IP Address Takeover -> Switch primary server address to physically different machine**
  - If backup machine is on the same network, route switchover is automatic (via ARP)
  - But backup machine can even be on a totally different network:
    - Define Primary server address as a *VirtualIP interface
    - With RIPV2, movement of *VirtualIP address is advertised throughout the network
  - Note: Also requires V4R4 Clustering product be installed



AS1 is taken down:

IP Address Takeover inactivates 10.2.1.1 *VirtualIP interface on AS1 and activates equivalent interface on AS3

RouteD on AS3 advertises that it can now reach 10.2.1.1

After route change is propagated, all traffic to 10.2.1.1 should be directed to AS3

Notes:

Finally, *VirtualIP addresses can improve system availability when used in conjunction the V4R4 Clustering product.   The Clustering application controls on which system is the *VirtualIP address active at any point in time.   When that system is taken down, the same *VirtualIP address is activated on a backup system.

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

But IP address takeover is not limited to both machines being on the same network. All we need is to define the takeover address as an address that is not directly accessible from either of the local networks -- in other words, a ***VirtualIP address.

For example, consider the backup system being AS3, rather than AS2.   In this case, we need to define the takeover address as a *VirtualIP address that is not part of either of the local networks to which the AS/400's are attached. That is why. on the previous page, the *VirtualIP address is defined as 10.2.1.1. This address is not part of either the 10.1.1.x or the 10.1.2.x networks.

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

TCP/IP and Opticonnect

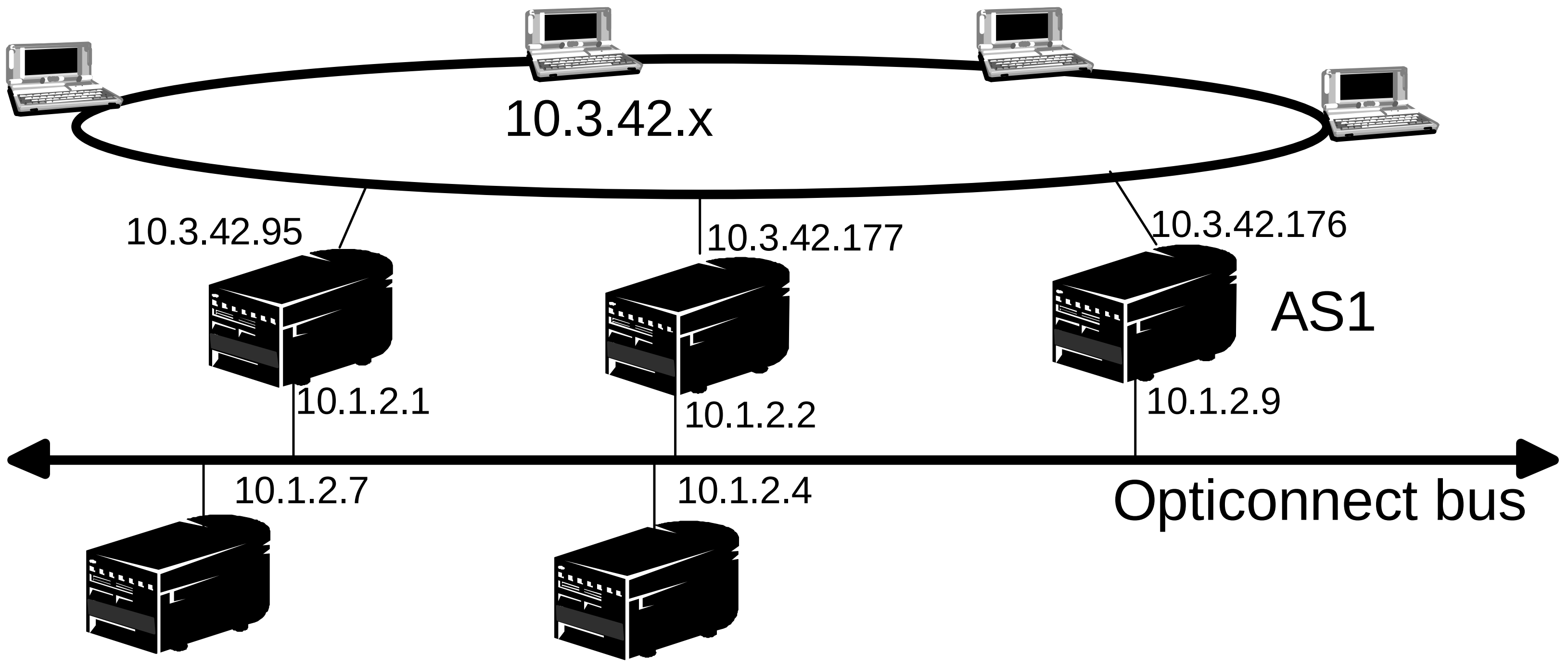
Notes:

Just added with V4R4 is the ability to define TCP/IP connections over the Opticonnect Bus. This section will take a brief look at this feature and how it can used.

TCP/IP over Opticonnect provides another application for the same routing building blocks discussed earlier -- proxy ARP, unnumbered point-point networks and *VirtualIP interfaces

The connection can be created as an emulated LAN or as an unnumbered point to point connection.

TCP/IP over Opticonnect - Emulated LAN configuration



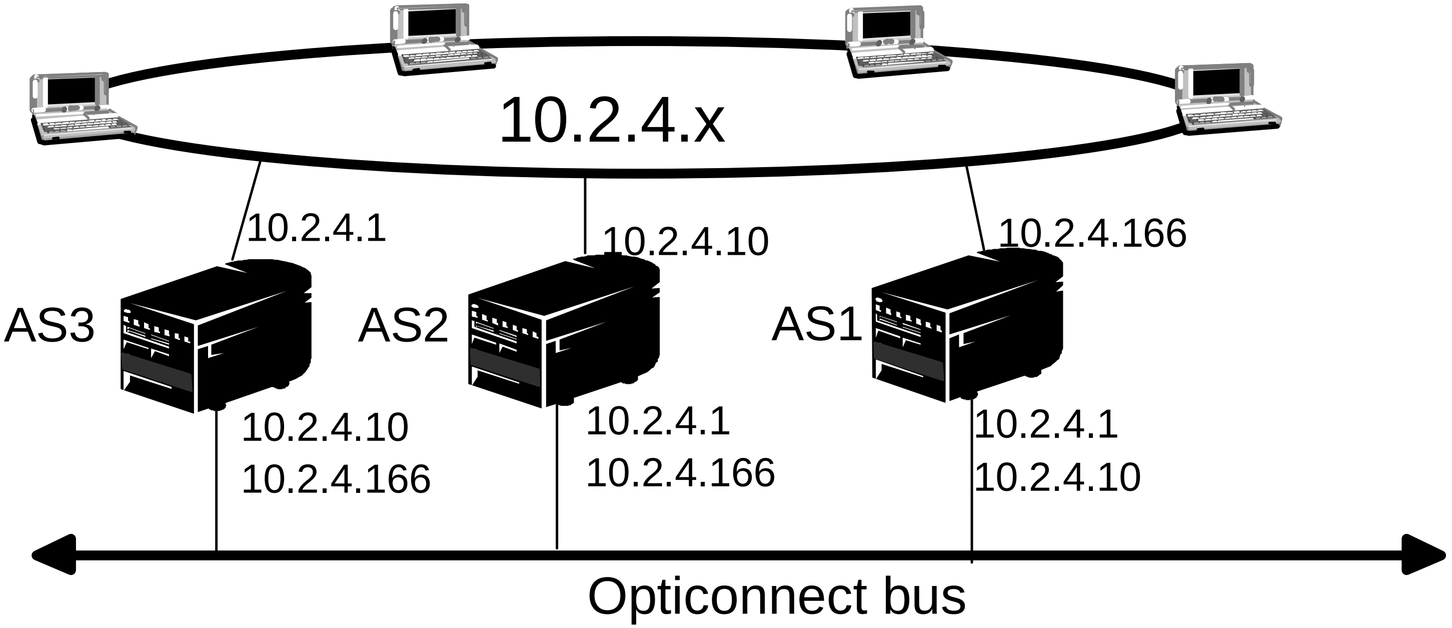
- Opticonnect bus appears as a LAN to TCP/IP
- Simple to configure, but LAN <-> Opticonnect connectivity is not automatic (requires RIP, static routes, and so on.)

Notes:

This connection appears like a LAN to TCP/IP.   Although Opticonnect is physically not a broadcast capable media, with this configuration, broadcasting is emulated, just like for a real LAN

Since a new network was created for the Opticonnect bus, connections between the Opticonnect bus and the LAN require route entries to be made. These connections can be advertised if Routed is started.

TCP/IP over Opticonnect - Point to Point Configuration



- Pt-Pt unnumbered interfaces configured for each pair of opticonnect hosts:
  - Subnet mask = 255.255.255.255
  - Associated Local Interface = Primary LAN interface
- No new networks created, LAN - Opticonnect connectivity automatic
- Traffic automatically switches to LAN if Opticonnect path is down

Notes:

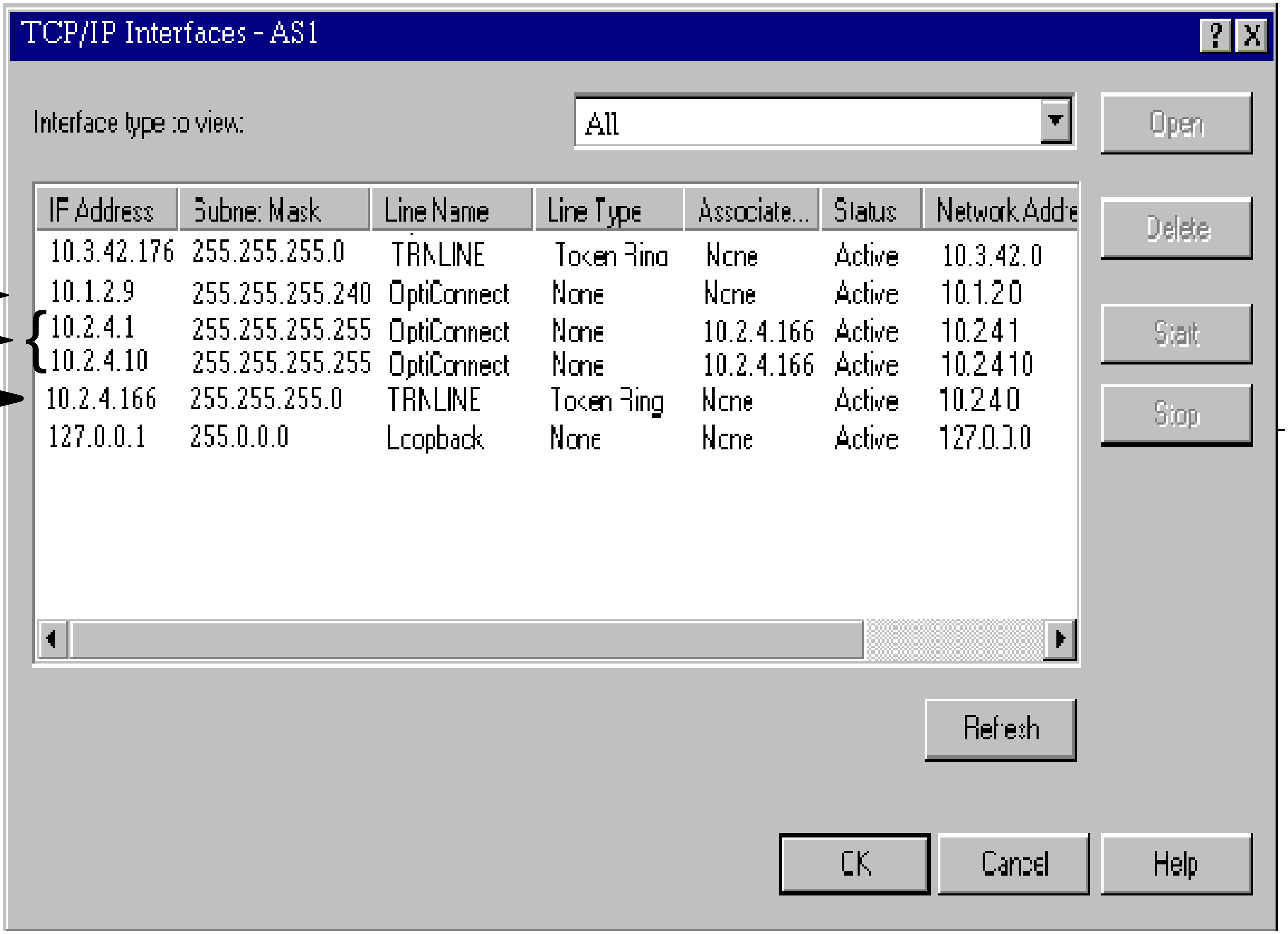
This is a variation on the unnumbered point to point connection that we saw earlier. In this case we are using the Opticonnect bus as a collection of Point to Point connections. We define an unnumbered connection for each pair of hosts. On AS1 we have defined the Real token ring interface at 10.2.4.166. We have a point to point connection defined between AS1 and AS2 (10.2.4.10) and a connection between AS1 and AS3 (10.2.4.1).   As before, the IP address assigned to the point-point interface is the IP address of the emote system.

These definitions are made using Operations Navigator.

One advantage of this configuration, using unnumbered point to point connections, no additional route definitions are required.   Connectivity between host on one network to hosts one the other network is automatic.

Another advantage is that if both networks are active, data sent between AS/400's will flow over the Opticonnect bus, because these routes have the most specific subnet mask. But if the Opticonnect bus goes down, traffic is automatically switched to the token ring LAN

Opticonnect Interface Configuration



Interface type to view: All

IP Address	Subnet Mask	Line Name	Line Type	Associate...	Status	Network Addre
10.3.42.176	255.255.255.0	TRNLINE	Token Ring	None	Active	10.3.42.0
10.1.2.9	255.255.255.240	OptiConnect	None	None	Active	10.1.2.0
10.2.4.1	255.255.255.255	OptiConnect	None	10.2.4.166	Active	10.2.4.1
10.2.4.10	255.255.255.255	OptiConnect	None	10.2.4.166	Active	10.2.4.10
10.2.4.166	255.255.255.0	TRNLINE	Token Ring	None	Active	10.2.4.0
127.0.0.1	255.0.0.0	Loopback	None	None	Active	127.0.0.0

Emul. LAN IFC

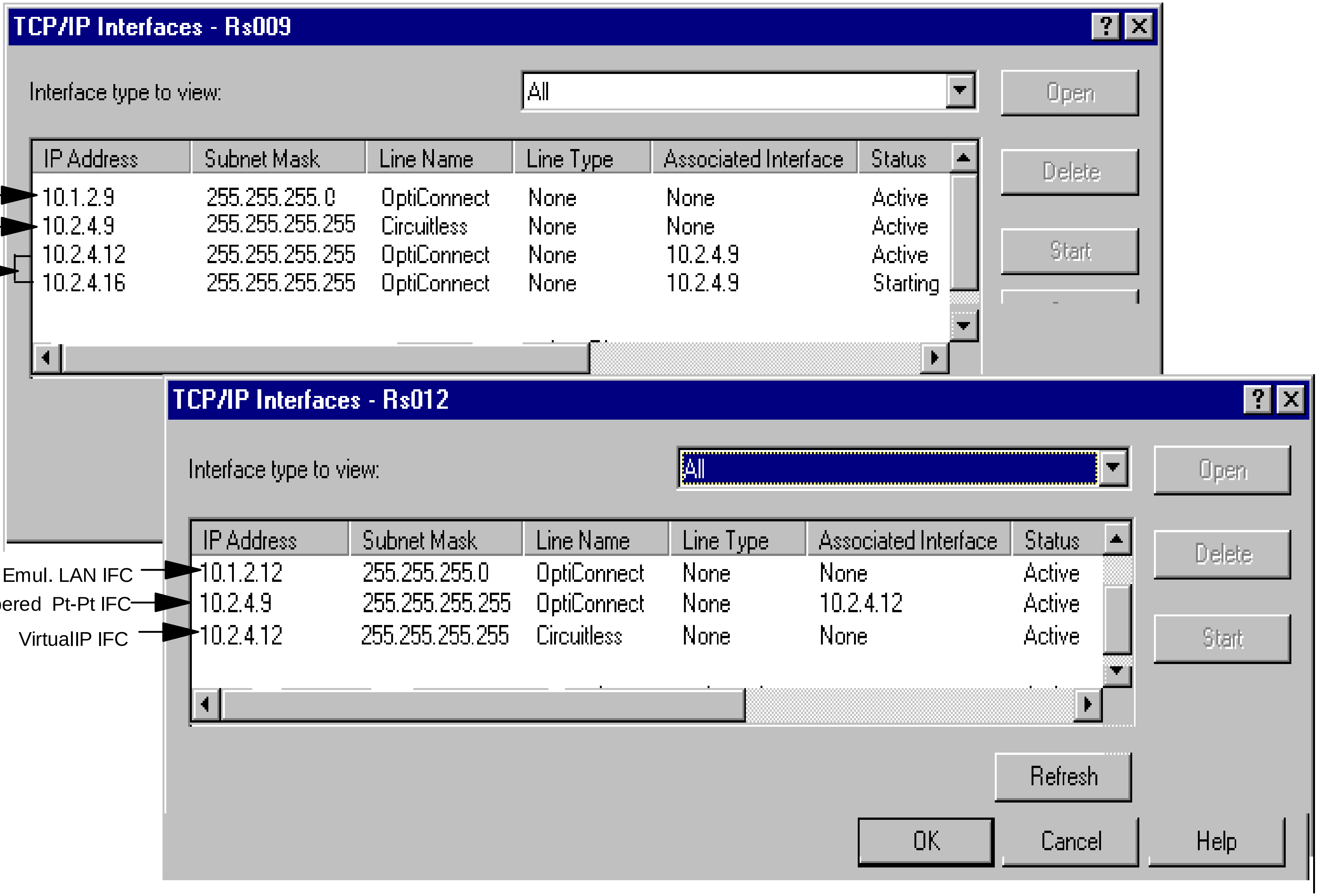
Pt. - Pt. IFC

Primary LAN IFC

Notes:

This is a screen shot from AS1 showing the two types of configurations. The 10.1.2.9 interface is an LAN emulation interface.   The next two entries are the point to point Interface definitions.

Opticonnect Configuration - Cont.



TCP/IP Interfaces - Rs009

Interface type to view: All

IP Address	Subnet Mask	Line Name	Line Type	Associated Interface	Status
10.1.2.9	255.255.255.0	OptiConnect	None	None	Active
10.2.4.9	255.255.255.255	Circuitless	None	None	Active
10.2.4.12	255.255.255.255	OptiConnect	None	10.2.4.9	Active
10.2.4.16	255.255.255.255	OptiConnect	None	10.2.4.9	Starting

Emul. LAN IFC

VirtualIP IFC

Unnumbered
Pt-Pt IFCs

TCP/IP Interfaces - Rs012

Interface type to view: All

IP Address	Subnet Mask	Line Name	Line Type	Associated Interface	Status
10.1.2.12	255.255.255.0	OptiConnect	None	None	Active
10.2.4.9	255.255.255.255	OptiConnect	None	10.2.4.12	Active
10.2.4.12	255.255.255.255	Circuitless	None	None	Active

Emul. LAN IFC

Unnumbered Pt-Pt IFC

VirtualIP IFC

Notes:

The preceding chart shows another example of a TCP/IP over Opticonnect configuration.

The preceding page shows two Interface displays. On RS009, interface 10.1.2.9 is an emulated LAN *OPC interfaces, indicated by the subnet mask being less than 255.255.255.255, in this case, 255.255.255.240 RS012 has a similar interface, 10.1.2.12, connecting it to the same emulated LAN.

On RS009, The 10.2.4.12 and 10.2.4.16 are unnumbered, point-point *OPC interfaces. RS012 has one unnumbered, point-point interface of 10.2.4.9 These are unnumbered, point-point opticonnect interfaces because:

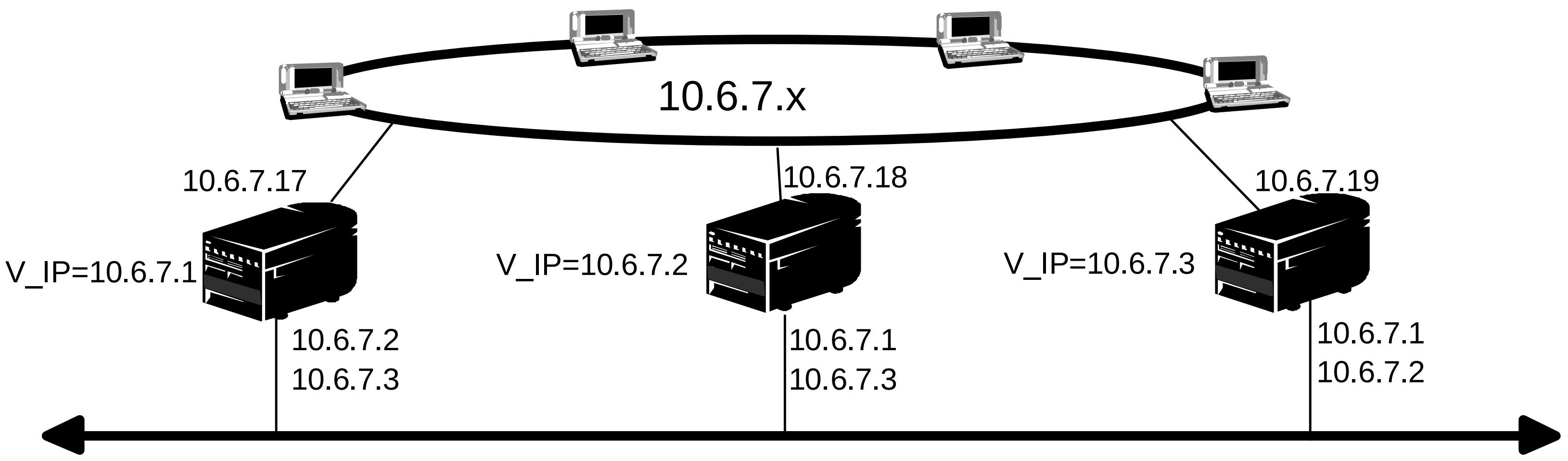
- the subnet mask is 255.255.255.255
- the associated local interface is not *NONE

As explained earlier, an unnumbered interface can be thought of as nothing more than a "routing handle". When the AS/400 has data to send to such an IP address, the data is just routed out that unnumbered interface. Inbound data that is destined for an unnumbered IP address is not accepted by the local AS/400. Instead it is forwarded out the unnumbered interface to the true owner of that IP address.

Another interesting item on the previous page is the one interface being in "Starting" state. The precise meaning of the "Starting" interface state is link specific. For Opticonnect interfaces it means:

- The QSOC subsystem is not yet active
- Or, for unnumbered, point-point *OPC interfaces, the remote end of the connection, is not yet active. The RS009 display shows unnumbered interface 10.2.4.16 in "Starting" state because the connection to the true owner of 10.2.4.16, i.e., RS016, is not yet active. (The remote end of the connection is through unnumbered interface 10.2.4.9 on RS016)

TCP/IP over Opticonnect using *VirtualIP



- ► Pt-Pt unnumbered interfaces configured for each pair of opticonnect hosts:
  - ■ Subnet mask = 255.255.255.255
  - ■ Associated Local Interface = *VIRTUALIP interfaces
- ► No new networks created, LAN - Opticonnect connectivity automatic
- ► Traffic automatically switches to alternate path if one interface is down
- ► *VirtualIP interface is used as the anchor for the unnumbered point-point interfaces

Notes:

This is a variation on the unnumbered point to point connection that we saw earlier. In this case we are using the Opticonnect bus as a collection of Point to Point connections. We define an unnumbered connection for each pair of hosts.

These definitions are made using Operations Navigator.

Like the previous configuration, no additional route definitions are required.   Connectivity between host on one network to hosts one the other network is automatic.

But an advantage of this configuration is that if either networks are active, a path will exist to reach any of the AS/400’s.

TCP/IP and LPAR

Notes:

The advent of LPAR provided yet another environment to apply the same routing concepts as previously discussed.

With LPAR, a single AS/400 is logically partitioned in multiple virtual machines. Each partition has its own address space its own instance of TCP/IP, and may have its own dedicated I/O adapters. To TCP/IP, each partition appears like a distinct AS/400

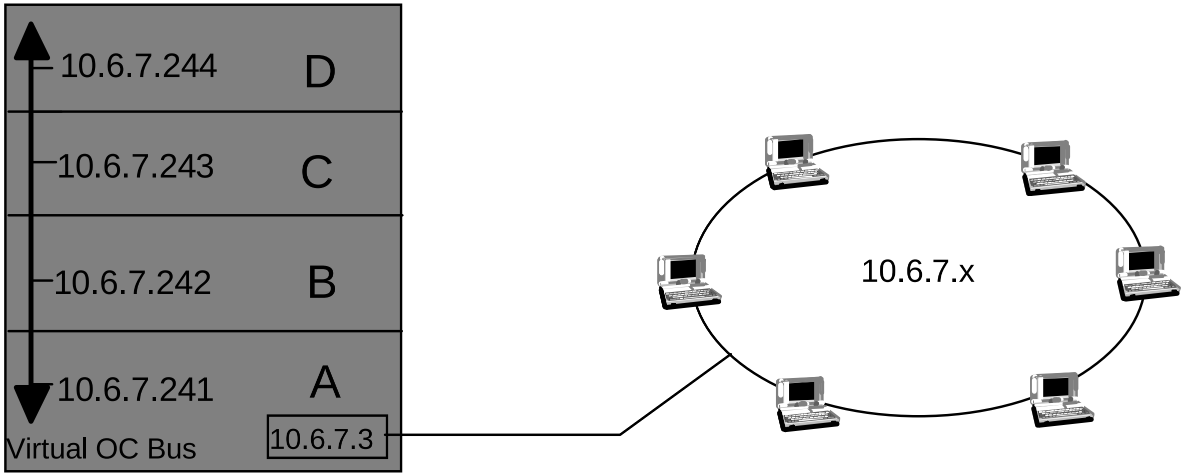
Moreover, TCP/IP communication between the different partitions is done via a virtual opticonnect bus. The TCP/IP routing code sees the path to another LPAR partition no differently than the path to another system connected via a physical opticonnect bus. All of the concepts and configurations that were previously described for "TCP/IP over Opticonnect" environments apply equally well to "TCP/IP with LPAR"

.

Virtual Opticonnect with LPAR

► **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



(Associated local interface = 10.6.7.3)

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

Transparent subnetting enabled when:

- *OPC subnet (e.g., 10.6.7.240) is subset of associated local interface subnet (10.6.7.0)
- *OPC MTU <= associated local interface MTU

Transparent subnetting provides connectivity from external LAN to "I/O less" partitions (B,C,D)

Thus, 10.6.7.3 TRN interface in partition A will proxy for all IP addresses in 10.6.7.240 subnet

If the LAN adapter 10.6.7.3 fails, connection to all partitions is lost

You should also consider the amount of traffic expected to the other partitions when choosing this configuration.

Notes:

In this example we only have one LAN adapter 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 we are going to define a transparent subnet on the Virtual Opticonnect Bus. The LAN has an network address of 10.6.7.x. We want to plan for additional partitions so we need 12 IP addresses. To get 12 addresses we must use a subnet mask of 255.255.255.240. This gives us 10.6.7.241 - 10.6.7.254, a total of 14 usable addresses.

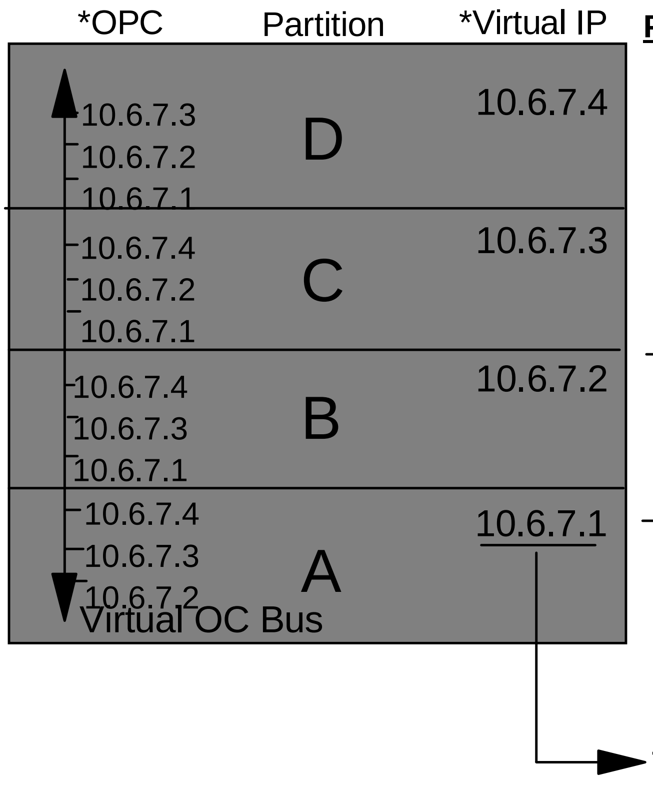
We must insure that these addresses are not already in use in the real LAN.

After we get the addresses, we assign one to each partition. We add an interface to each partition and define the address on the virtual Opticonnect bus (*OPC).

So long as the MTU size on the virtual Opticonnect Bus is less than or equal to the size of the MTU on the real LAN interface and the *OPC subnet is a subnet of the LAN network address, then transparent subnetting will automatically be enabled and the interface 10.6.7.3 will Proxy ARP for all the interfaces defined in the partitions. This will allow clients on the LAN to connect to the partitions.

Note: This configuration is an example of how to provide maximum connectivity between the LAN hosts and the partitions.   Alternatively, if maximum security or isolation is needed between the LAN and one or more partitions, different IP addresses or MTU can be used to accomplish this.

Virtual IP with Virtual Opticonnect & LPAR



To 10.6.7.x external LAN

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	10.6.7.3	*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

- Proxy ARP again provides connectivity from external LAN to "I/O less" partitions (B,C,D)
- 10.6.7.1 TRN interface in partition A will proxy for IP addresses in 10.6.7.2 - 10.6.7.4
- Indirect routes in partitions B, C and D provide connectivity to external LAN via 10.6.7.1

Notes:

This configuration is shown primarily to again illustrate that the same configurations that apply to TCP/IP over real opticonnect apply equally well to TCP/IP with LPAR.

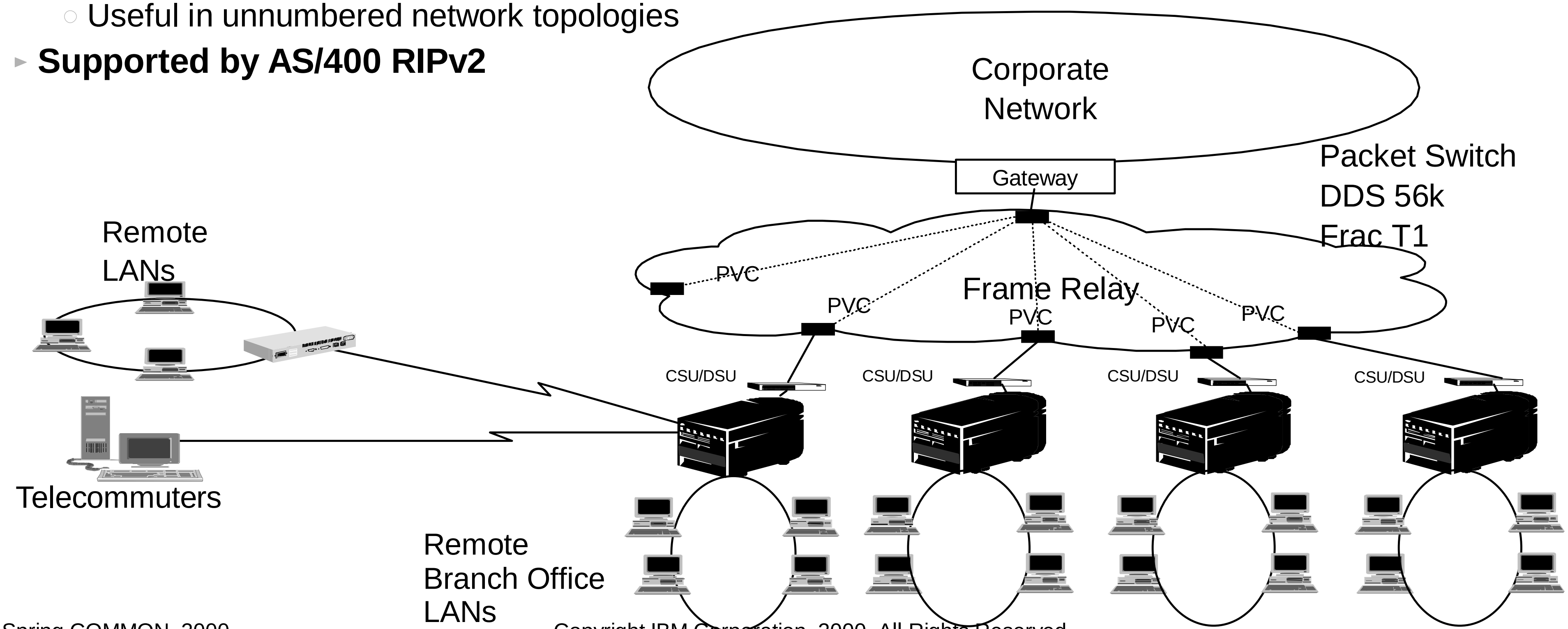
This example shows the earlier point-point opticonnect configuration with *VirtualIP mapped to an LPAR configuration

Frame Relay

AS/400 Frame Relay

High Speed WAN technology for interconnecting remote systems and networks

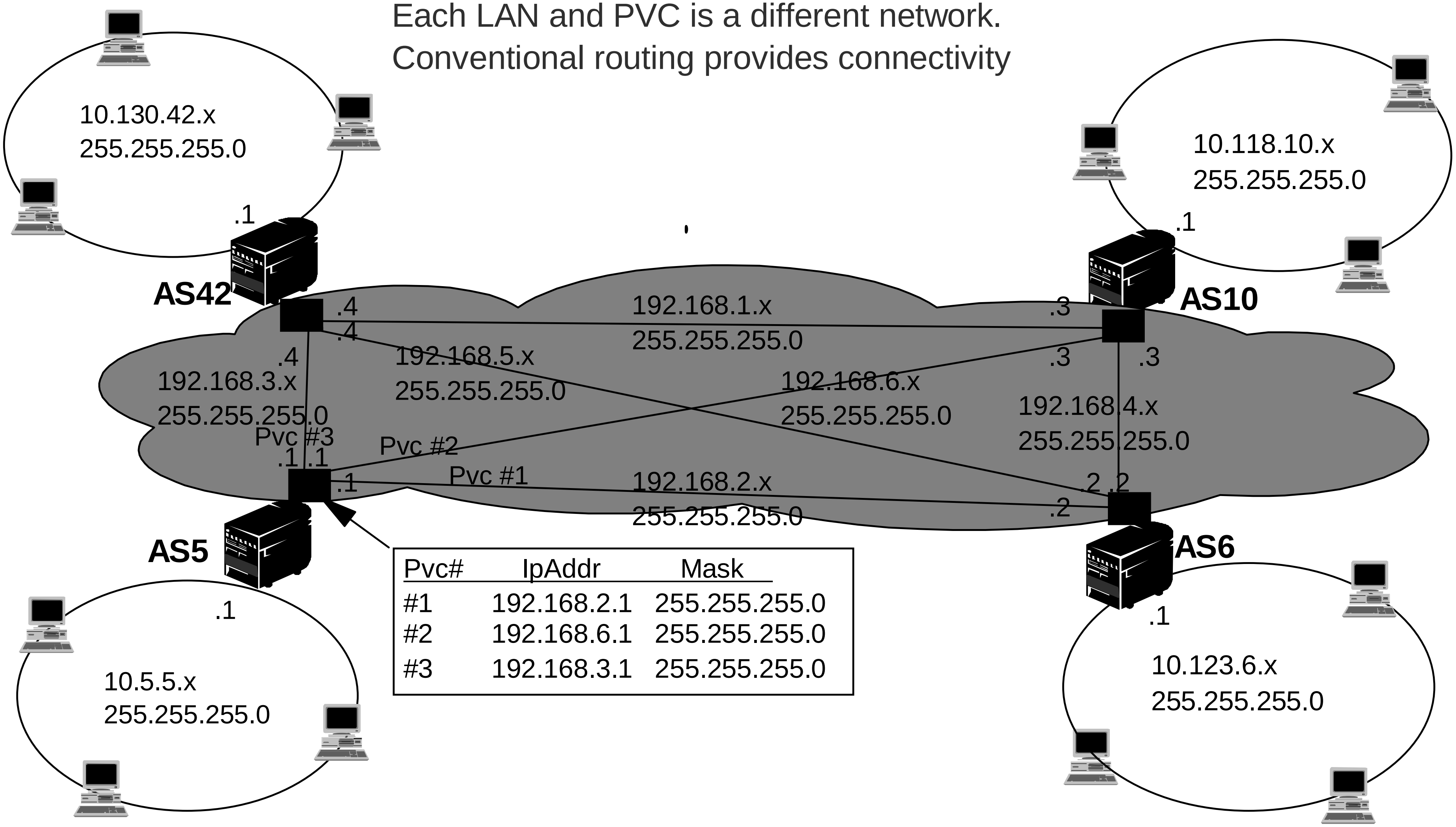
- **Supports multiprotocol (SNA, IPX, TCPIP) communications**
- **Many different network topologies possible, utilizing previously described routing techniques like proxy ARP, numbered & unnumbered networks and *VirtualIP**
- **"DCE mode" for direct host - host WAN connection without an actual frame relay network**
- **Supports RFC 2390, Inverse ARP protocol:**
  - Dynamic resolution of remote IP addresses,
  - Useful in unnumbered network topologies
- **Supported by AS/400 RIPv2**



Frame Relay Network Topologies:

Point-to-Point Numbered Network

Each LAN and PVC is a different network.
Conventional routing provides connectivity



Pvc#	IpAddr	Mask
#1	192.168.2.1	255.255.255.0
#2	192.168.6.1	255.255.255.0
#3	192.168.3.1	255.255.255.0

Notes:

The preceding page illustrates how one might use frame relay links to interconnect four different networks, using the AS/400's as the local network gateway to the other three networks.

The IP addresses are configured using the numbered network method. As previously stated, this method is easiest to understand, but not necessarily always the easiest overall solution.

Each AS/400 is connected to its local LAN.   Frame relay interfaces provide numbered network connections to each of the other three remote networks.   Each connection, or PVC, is a separate network.   For example, the PVC connecting systems AS5 and AS42 comprises the 192.168.3.0 network

Sample interface and route tables for system AS5 are shown below.

Interface Table

IP Address	Subnet Mask	Line
10.5.5.1	255.255.255.0	TRNLINE
192.168.2.1	255.255.255.0	FR_PVC1
192.168.6.1	255.255.255.0	FR_PVC2
192.168.3.1	255.255.255.0	FR_PVC3

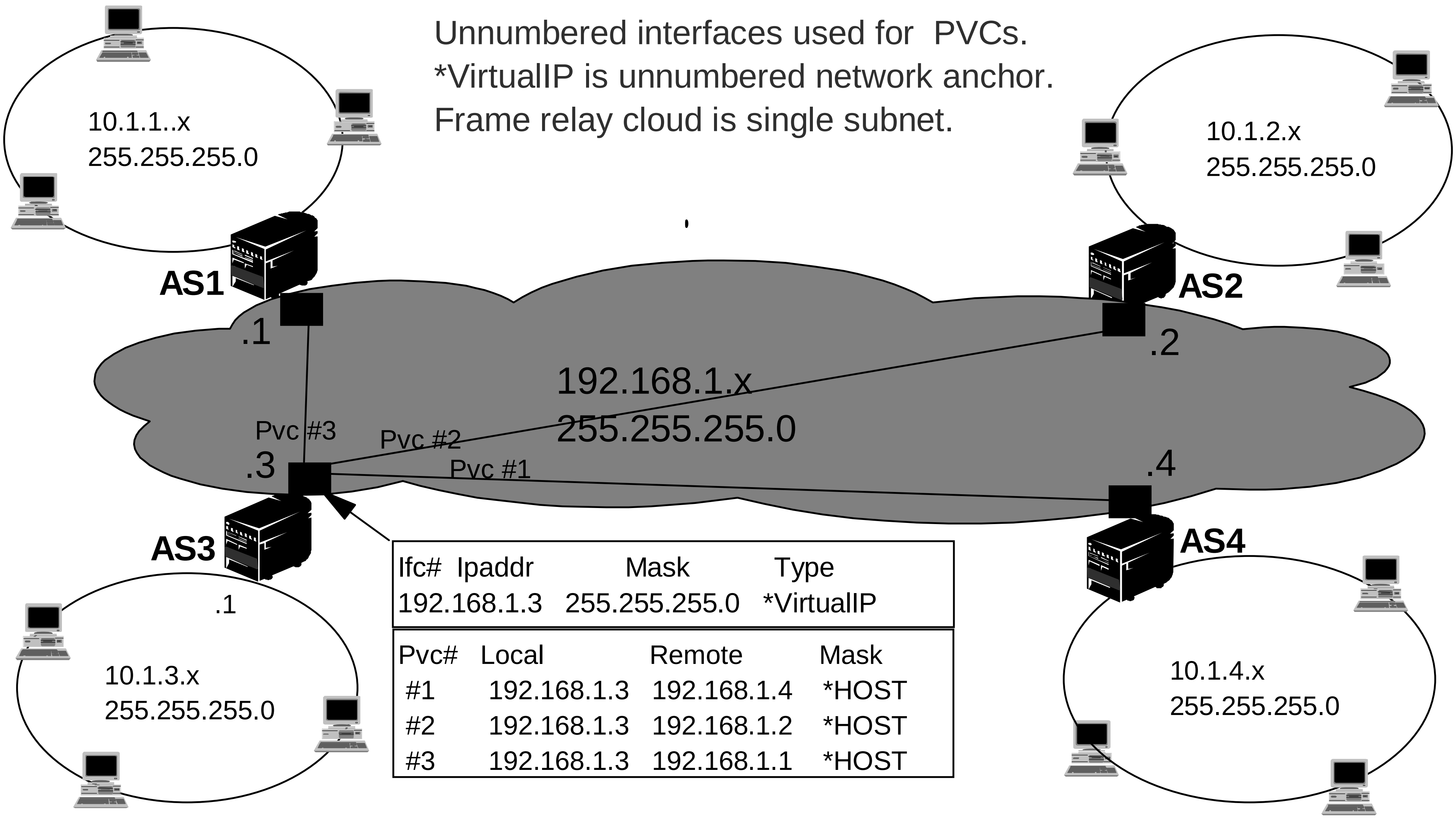
Route Table

Route Dest.	Subnet Mask	Next Hop
10.5.5.0	255.255.255.0	*DIRECT
192.168.2.0	255.255.255.0	*DIRECT
192.168.6.0	255.255.255.0	*DIRECT
192.168.3.0	255.255.255.0	*DIRECT
10.123.6.0	255.255.255.0	192.168.2.2
10.118.10.0	255.255.255.0	192.168.6.2
10.130.42.0	255.255.255.0	192.168.3.2

As you can see, connectivity between the different networks is accomplished by adding explicit routes to each remote network.

Frame Relay Network Topologies:

Non-broadcast Multi-Access, Partially Meshed



Unnumbered interfaces used for PVCs.
*VirtualIP is unnumbered network anchor.
Frame relay cloud is single subnet.

Ifc# Ipaddr	Mask	Type
192.168.1.3	255.255.255.0	*VirtualIP

Pvc#	Local	Remote	Mask
#1	192.168.1.3	192.168.1.4	*HOST
#2	192.168.1.3	192.168.1.2	*HOST
#3	192.168.1.3	192.168.1.1	*HOST

Notes:

The preceding page gives an example of configuring a frame relay network using unnumbered networks:

- This frame relay network is only partially meshed. Or in other words, PVCs exist only between the gateway system, AS3, and the remote systems, AS1, AS2 and AS4. Routes must be added for the remote systems to communicate.
- The IP addresses are assigned using unnumbered, rather than numbered, networks.

In this case, instead of configuring each PVC as a separate numbered network, we configure all of the PVCs as part of a single network -- the 192.168.1.0 network. Data is routed from one AS/400 to another using one of the *HOST routes configured on each system.

Because unnumbered point-point interfaces are being used, far fewer 192.168.x.x IP addresses are consumed. In fact, no additional IP addresses are required for the frame relay interfaces. Using the unnumbered, point-point scheme, the IP addresses assign to each frame relay interface is the *VirtualIP address of the remote system.

Sample interface and route tables for system AS3 and AS4 are shown below.

AS3 Interface Table

IP Address	Subnet Mask	Line	Assoc Lcl
10.1.3.1	255.255.255.0	TRNLINE	*NONE
192.168.1.3	255.255.255.0	*VIRTUALIP	*NONE
192.168.1.1	*HOST	FR_PVC31	192.169.1.3
192.168.1.2	*HOST	FR_PVC32	192.169.1.3
192.168.1.4	*HOST	FR_PVC23	192.169.1.3

AS3 Route Table

Route Dest.	Subnet Mask	Next Hop
10.1.3.0	255.255.255.0	*DIRECT
192.168.1.0	255.255.255.0	*DIRECT
192.168.1.1	*HOST	*DIRECT
192.168.1.2	*HOST	*DIRECT
192.168.1.4	*HOST	*DIRECT
10.1.1.0	255.255.255.0	192.168.1.1
10.1.2.0	255.255.255.0	192.168.1.2
10.1.4.0	255.255.255.0	192.168.1.4

AS4 Interface Table

IP Address	Subnet Mask	Line	Assoc Lcl
10.1.3.1	255.255.255.0	TRNLINE	*NONE
192.168.1.4	255.255.255.0	*VIRTUALIP	*NONE
192.168.1.3	*HOST	FR_PVC41	192.168.1.4

AS4 Route Table

Route Dest.	Subnet Mask	Next Hop
10.1.4.0	255.255.255.0	*DIRECT
192.168.1.0	255.255.255.0	*DIRECT
10.1.0.0	255.255.0.0	192.168.1.3
192.168.1.0	255.255.255.0	192.168.1.3

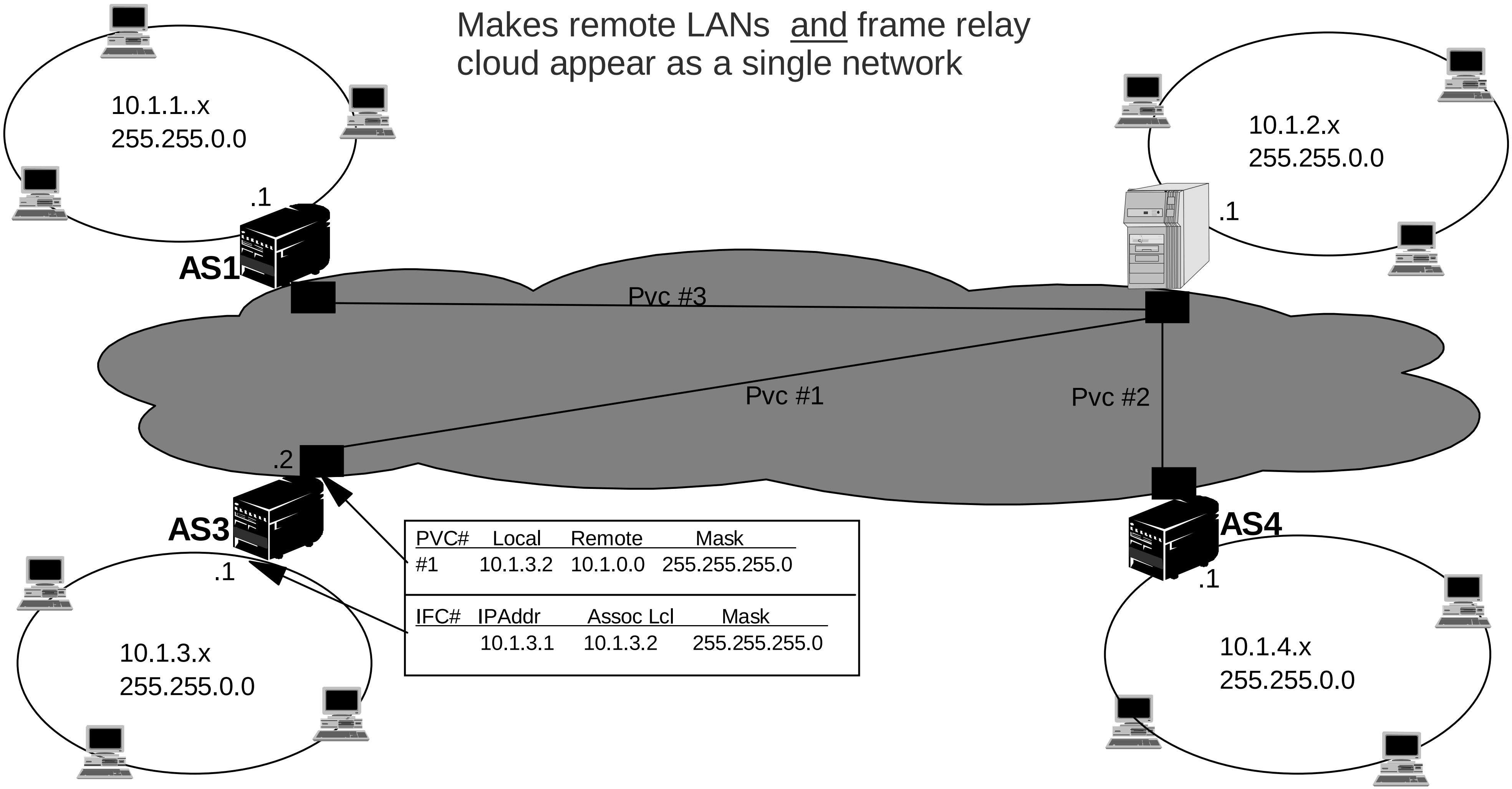
Note: Above AS4 configuration assumes no other 10.1.1.x networks than what are shown.

Frame Relay Network Topologies:

Partially Meshed with Transparent Subnetting

Transparent Subnetting:

Makes remote LANs <u>and</u> frame relay cloud appear as a single network



PVC#	Local	Remote	Mask
#1	10.1.3.2	10.1.0.0	255.255.255.0

IFC#	IPAddr	Assoc Lcl	Mask
	10.1.3.1	10.1.3.2	255.255.255.0

Notes:

Finally, this example illustrates the classic Proxy ARP concept of making disjoint physical networks appear as they are a single logical network. In this network, transparent subnetting is used to make not only the three remote LANs, but also the frame relay cloud appear to be a single network, i.e., a 10.1.0.0 network.

Consider the 10.1.3.x LAN. All hosts on the LAN, except AS3, are configured with a subnet mask of 255.255.0.0. Each of these hosts, except for the A/400, assume that they are directly connected to any 10.1.x.x host. AS3 is the only machine that is aware that it is only directly connected to the 10.1.3 subnet, so its' LAN interface is configured with a mask of 255.255.255.0. According to normal LAN protocol, when these hosts have data to send to a 10.1.x.x host, an ARP request is first sent.

Using transparent subnetting, AS3 will answer ARP requests for IP addresses on any of the remote LANs. When the subsequent IP packet is received by the AS/400, it is forwarded out the PVC1 frame relay interfaces to the gateway system, on the 10.1.2 network.

A sample interface and route table for system AS3 is shown below:

Interface Table

IP Address	Subnet Mask	Line	Assoc Local
10.1.3.1	255.255.255.0	TRNLINE	10.1.3.2
10.1.3.2	255.255.0.0	FR_PVC1	*NONE

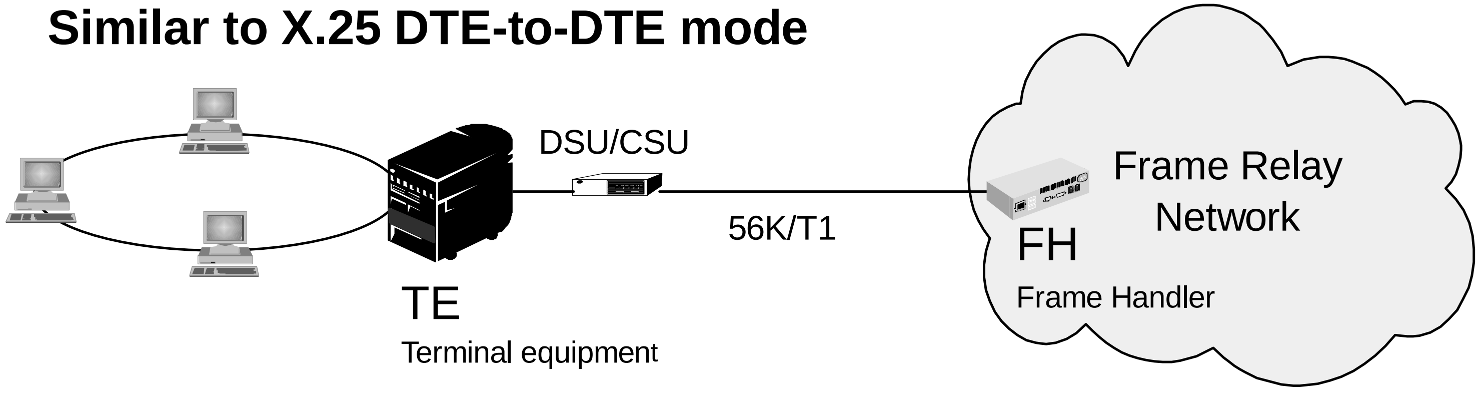
Route Table

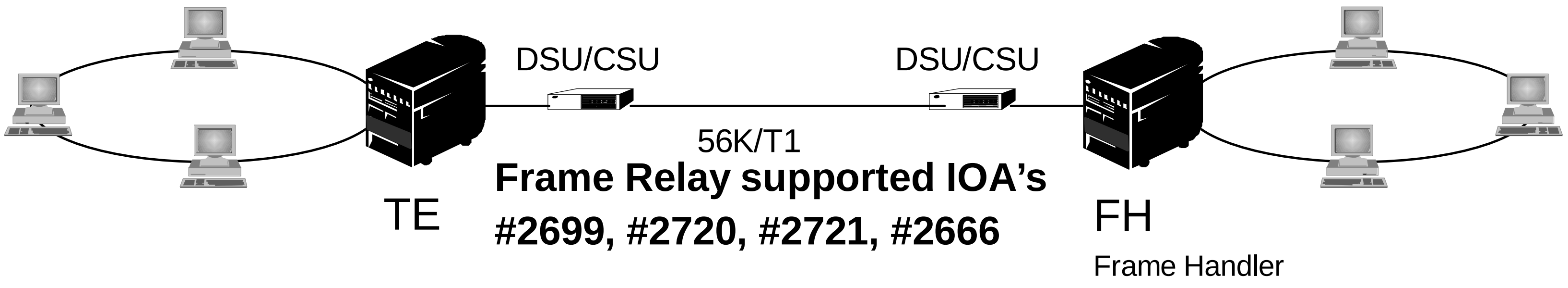
Route Dest.	Subnet Mask	Next Hop
10.1.3.0	255.255.255.0	*DIRECT
10.1.0.0	255.255.0.0	*DIRECT

As you can see, no additional indirect routes are needed to reach the remote networks -- the classic advantage of proxy ARP and transparent subnetting techniques

Frame Relay without Frame Relay Cloud

Similar to X.25 DTE-to-DTE mode





Frame Relay supported IOA’s #2699, #2720, #2721, #2666

- **Why**
  - Enables AS/400 multi-protocol(SNA,IP,IPX) point-to-point communications
- **How**
  - Use existing hardware(ie can easily convert existing SDLC links to Frame Relay)
  - AS/400 can be configured as Frame Handler

Notes:

The last topic we will look at is Frame Relay without a frame relay cloud. This support can be very useful to customers who have existing Point to Point lines that are being used for SNA with SDLC.

In a "real" frame relay connection the system connects to a frame relay cloud. The Cloud is provided by a network provider. In this configuration the system is configured as terminal equipment (TE). The AS/400 can also be configured as a Frame Handler (FH). This allows another AS/400 configured as TE to connect directly to the system configured as a FH with out going through the cloud. If a customer has one of the supported IOAs listed in the graphic, connected point to point, they can very easily switch it over from SDLC to Frame Relay. This will allow them to run TCP/IP, IPX, and SNA traffic across the link at the same time. This can be a great first step for a customer wanting to convert their WAN from SNA to TCP/IP. Frame Relay also gives them true full duplex support across the link, so they may see a speed improvement.

Appendix

Appendix A: Additional References

- **SG24-5190: " V4 TCP/IP for AS/400: More Cool Things Than Ever"**

  http://www.redbooks.ibm.com/ -> Redpieces

- **http://www.as400.ibm.com/infocenter/**

  Networking--> TCP/IP -->TCP/IP routing and workload balancing.

- **IBM Network Dispatcher:**
  - http://www.software.ibm.com/network/dispatcher/

- **GC24-3376: "TCP/IP Tutorial Technical Overview",**

  Chapter 11-- Availability, Scalability and Load Balancing

- **SG24-5147: " AS/400 Autoconfiguration: DNS and DHCP Support**

  Section 15.2: Transparent Subnet Masking

Appendix B: Recommended Routing PTFs

Routing Function :	APAR	V4R3 PTFs	V4R4 PTFs
Inhibit TCP initiated Dead Gateway	MA20996	MF23263	MF23322
ARP based Dead Gateway	MA21169	N/A	MF23501
Schowler Routes	MA21168 SA86580	MF23735 SF60765	MF23501 SF60267
*VirtualIP on local subnet	MA21170	MF23735	MF23501
RouteD Support of *VirtualIP interface	SA86310	SF60225	SF60207

Appendix C: Source IP Address Setting with *VirtualIP

Every packet that is sent by the AS/400 contains a source IP address field in the IP header. The logic for setting this field is usually obvious:

The source IP address in an outbound packet is set to the IP address of the interface to which the selected route is bound

However there are exceptions to the above rule. These are described below.

- The first exception is when Duplicate Route Round Robin load balancing is being used. In this case, the source IP address is set based upon the route selected <u>prior</u> to load balancing. Any subsequent route re-selection due to load balancing considerations does not affect the source IP address. Thus, this form of load balancing is transparent to the external host.

- The other class of exceptions to the general, "source IP address selection" rule given above is when the interface to which the selected route is bound, has an "Associated Local Interface" configured. When the selected route is bound to such an interface, the source IP address in the outbound packet will be the associated local IP address if:

  - The selected, physical interface is an unnumbered, point-point interface. This technique is commonly used with dialup PPP lines where an unnumbered point-point interface is created on the AS/400 with the same IP address as is assigned to the remote PPP client.

  - Or, the associated local interface is a *VIRTUALIP interface, and one of the following conditions is true:

    1) An external client explicitly connects to the *VIRTUALIP address.

       This is commonly the situation when an external client connects to a TCP server application on the AS/400. The client may have the AS/400’s *VIRTUALIP address permanently configured or the *VIRTUALIP address may have been returned by the DNS.

    2) A local client application sends a packet without binding to a specific local IP address and the destination is on a remote network.

       An example of this would be to Ping a host on a remote network without specifying a localIP address. The *VIRTUALIP address is used as the source IP address in this case because it is assumed that the *VIRTUALIP address is the IP address by which the AS/400 is known outside of the local network.

    3) A local client application sends a packet without binding to a specific local IP address, and both the *VIRTUALIP interface and the destination are on the same local subnet

       An FTP client connects to a server on the local network. The *VIRTUALIP address is again used as the source IP address because it is in the same subnet. Similar to 2) above, it is assumed that if the *VIRTUALIP interface is configured as part of the local subnet, then it is this IP address by which the AS/400 is to be know locally. It is assumed that if the *VIRTUALIP address is configured like this, then the external clients have also been configured to be able to route to the *VIRTUALIP address.

    4) A local client application binds to the *VIRTUALIP, associated local interface. The *VIRTUALIP address is again used as the source IP address because the application has explicitly bound to this IP address.

Appendix D:

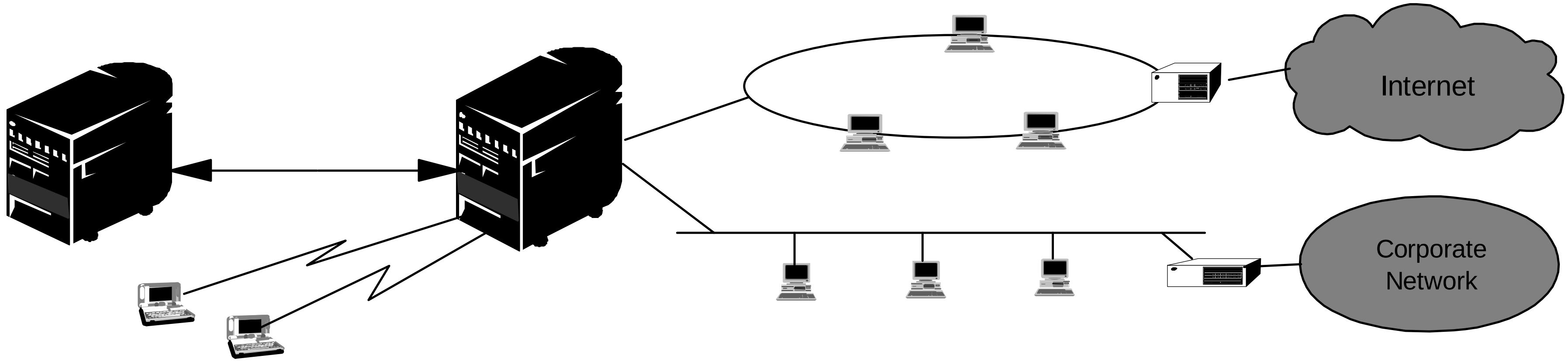
*VirtualIP, Unnumbered Point-Point, and other Mysteries

An alternate perspective on *VIRTUALIP and Unnumbered, point-point interfaces

IP Addresses: What are they used for ? Why do we need them ?

Four common uses:

- "Is it for me ?"
- "I am"
- "Where are you ?"
- "Where am I going ?"



D.1: "Is it for me ?"

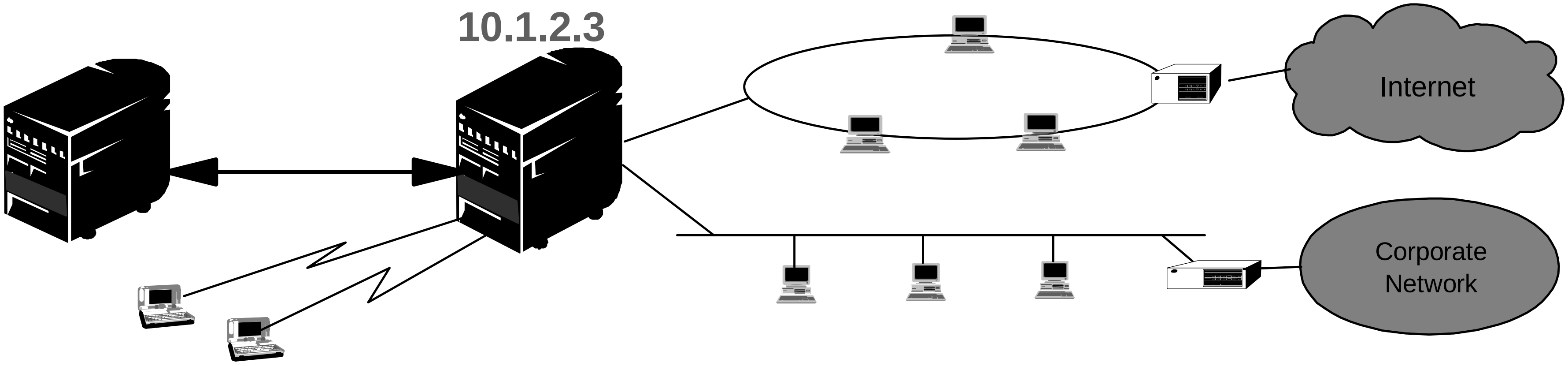
Inbound packet received:

- Must decide whether to accept and pass to local application, or discard/forward
- Examine IP header in received packet:
  - Does destination IP address match the AS/400's local IP address(es) ?
- "AS/400's local IP address" ==> Must define at least one "IP address for the system"

RFC 1122: Two models for multi-homed systems:

"Weak Multi-homing model: The adapter on which a packet is received is irrelevant

- **Thus, a single, system wide, IP Address would suffice:**



10.1.2.3: Single, system wide IP Address = ***VIRTUALIP Address***

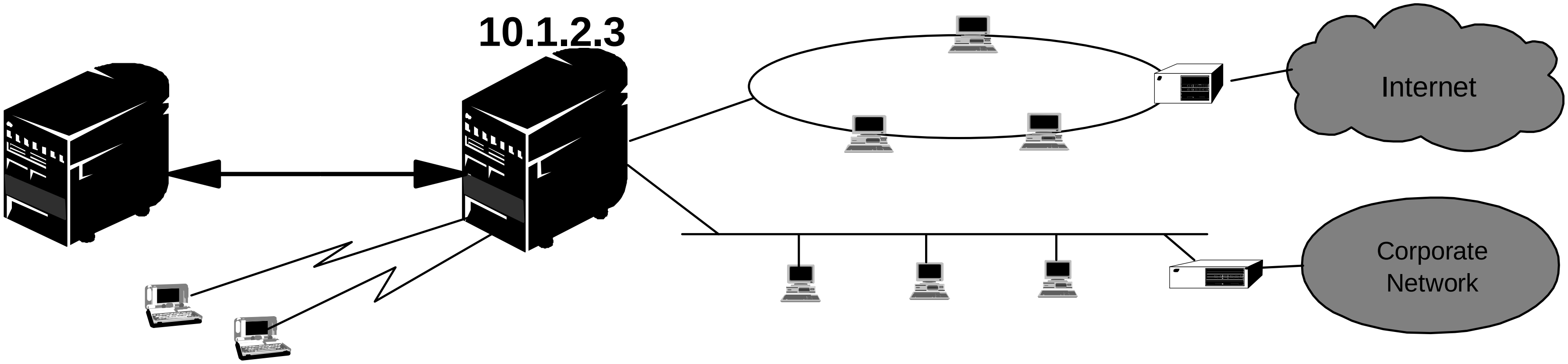
D.2: "I am"

Sending Outbound packets:

- **Must identify yourself when sending packets**
- **Receiver must be able to determine the IP address of the sender**
- **==> Set source IP address in the IP header to the "IP address of the system"**

- **Weak Multi-homing: May use the same "system IP address" regardless of which interface packet is being sent on.** (ignores external routing considerations)

- **Thus, again, a single system wide, IP Address would suffice**



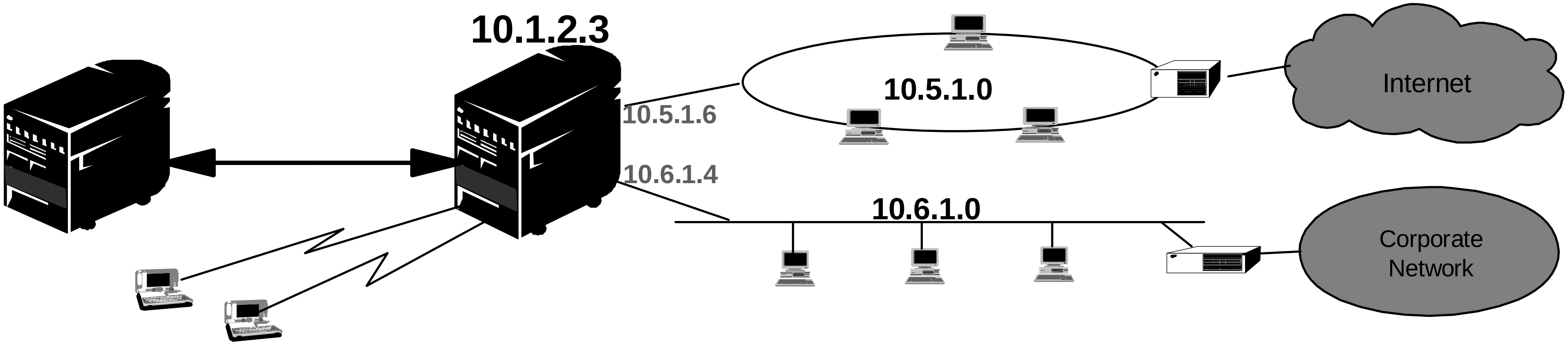
D.3: "Where are you ?"

Directing Outbound packets to a Target External Host:

- **How to direct outbound packets to the correct external host ?**
  - **Send as broadcast , or, send to the unique adapter address of external host**

- **How to determine adapter address of an external (LAN) host ?**
  - **ARP -- Address Resolution Protocol**
  - **ARP dynamically retrieves physical adapter address from external host**
  - **ARP is specific to physical network, i.e., ARP IP address must be part of the subnet to which adapter is physically connected**

- **Requires that subnet specific IP addresses be assigned to the LAN interfaces**
- **A single, system wide IP address is no longer sufficient**



Note, however, that no IP addresses have yet been assigned to the non-LAN, point-point interfaces

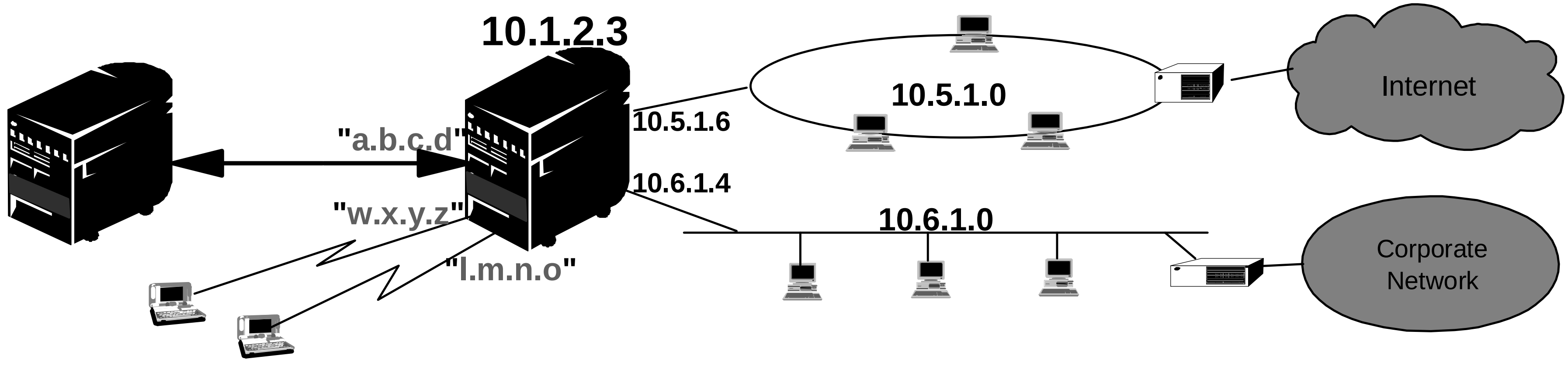
D.4: "Where am I going ?"

Routing:

To send data, a route must first be selected, which indirectly selects an adapter to use to send the packet

- **On the AS/400:**
  - ○ Routes are bound to interfaces, which are then bound to adapters.
  - ○ No direct "route to adapter" binding, route must first select an interface before adapter is determined
  - ○ Interfaces are denoted by IP address, not name

- **To route data out an adapter, there must be an interface, with an IP address, defined on that adapter**



- ARP forces IP addresses from the local LAN subnet to be assigned to the LAN interfaces
- Binding routes to interfaces, and denoting interfaces by IP addresses, forces an IP address to be assigned to all interfaces, even point-point interfaces

- **But what IP addresses should be assigned to the point-point interfaces ?**

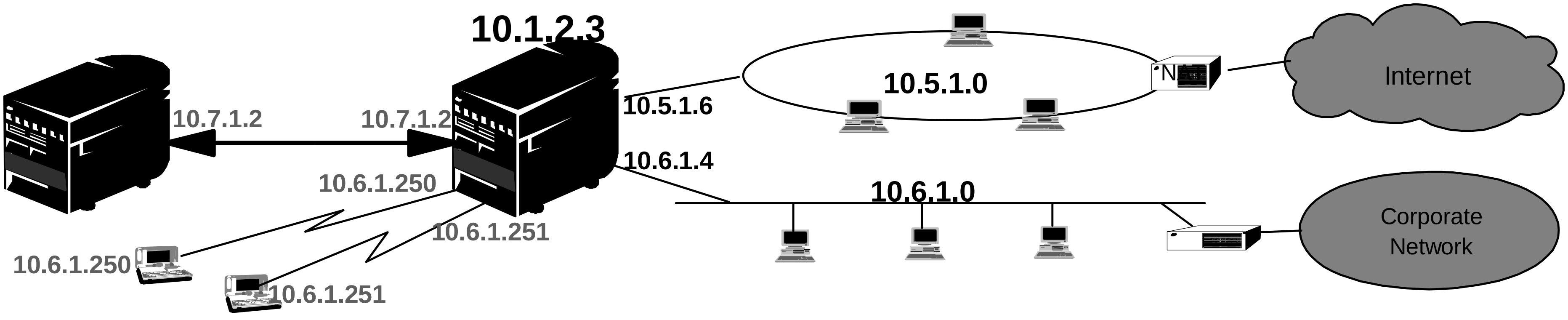
D.4: "Where am I going ?" *(Continued)*

What IP addresses should be assigned to the point-point interfaces ?

- **Would rather not consume 2 IP addresses for each and every point-point link**
- **Recall that the local IP address of a point-point interface is required only for routing purposes:**
  - **IP address is nothing more than a "route handle"**
  - **Local, "route handle" IP address is not externally visible, true owner of the IP address is a remote system**
  - **Packets to route handle IP address are never accepted, always forwarded**
  - **Route handle is never used as a source IP address (always use associated local IP address)**

- **If the true owner of a given IP address is the host at the remote end of a point-point link, then we could assign the same IP address to the local, point-point interface, and use this as the "route handle"**

- **This special local interface is an Unnumbered, Point-Point interface**



Unnumbered Point-Point interfaces: ***Conserve IP addresses***

When used with proxy, no new networks are created

Appendix E: Trademarks and Service Marks

AS/400, IBM, AIX and OS/400 are trademarks of the IBM Corporation in the United States or other countries or both.

Lotus and Domino are trademarks of the Lotus Corporation in the United States or other countries or both.

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