CHAPTER 7
IPX

Internetwork Packet Exchange (IPX) is a Layer 3 protocol that forms the basis for Novell's network operating system (NOS) called NetWare. IPX was developed from the earlier Xerox Network System (XNS). Today, it is used almost exclusively to support networks of Novell NetWare servers. It is primarily used for file and printer sharing, but the capabilities are broader. IPX is able to carry a large variety of applications.

Unfortunately, some of the terminology adopted by Novell is different from that used in IP networks. For example, Novell calls every device that provides IPX services a *router*. This term can cause some confusion. Thus, in this book, I continue with the already adopted language. I call a device that provides application services a *server*. In this book, a router is a device that primarily performs Layer 3 network functions. As always, I strongly caution the reader against using general-purpose application servers to perform network functions such as bridging and routing.

Just running applications creates a lot of work for application servers. At peak utilization times, they frequently are unable to perform their network functions as well. Therefore, when the network is needed the most, it is unavailable for all applications, not just the ones served by this device. I mention this fact specifically in the context of IPX because Novell servers are frequently configured to either bridge or route—a situation I do not recommend.

Every device in an IPX network has a globally unique address. This address is divided into two parts. A 32-bit address called the *network number* defines on which segment the device lives and the 48-bit *node number* defines the specification on the segment. Usually, the node number is identical to the device's MAC address.

The network number is similar to IP, which also uses 32-bit addresses, but IPX does not use the same notation for its addresses. While IP shows the four octets of the address as separate decimal numbers separated by dots, IPX network and node numbers are usually written in hexadecimal format. Furthermore, in most cases, leading "0" digits from the IPX network number are dropped when the address is written.

For example, a network might include the network number A1A. This number means that its address is really 00000A1A. The first and second octets of the address are both 00. The third octet has a value of 0A (10 in decimal) and the last has the value 1A (26 in decimal).

Another difference from IP is the absence of a subnet mask. The whole network number refers to a particular network. This reference causes confusion in networks that have more than one server on a LAN segment. To get around this problem, every Novell server also has an *internal network number*. This internal number need not have a relationship to the network number of the segment.

As I discuss later in this section, if the network uses Novell Link State Protocol (NLSP) areas, it is more efficient to allocate all addresses in blocks. This allocation includes both the LAN network numbers and these internal network numbers.

The services associated with any particular server are generally associated with the internal network number. The server acts like a router that connects the external LAN segment address to this internal network number that, in turn, represents the unique server.

One of the most important issues to remember when building an IPX network is that these IPX network numbers are not administered rigorously. Novell does offer a service called the Novell Network Registry that allocates and tracks IPX network numbers. This allocation and tracking permits different organizations to interconnect their IPX networks without worrying about address conflicts. However, participation in the Novell Network Registry is optional. Thus, merging two IPX networks together can be extremely challenging.

When merging occurs, it is quite common to find that both IPX network numbers and server names appear in both networks. The only solution to this problem is to adopt a single common standard for naming and addressing servers and IPX networks. However, this standard can take a long time to implement, and corporate priorities may make it necessary to merge the networks quickly.

There is a convenient way around this problem. Usually, in such situations just a few servers actually need to be simultaneously accessible to both networks. These few servers can then be readdressed and renamed according to the new common standard. You can then build a simple IPX routing firewall between the two networks by using a pair of routers with a LAN segment between them. One router connects this common segment to the first network and the other connects to the second network.

You can then implement Access lists on the exchange of IPX routing and service information (I discuss these protocols shortly). The two routers prevent those network numbers and server names that are in conflict from entering this common segment. Then, as IPX names and addresses are migrated to the new standard, the new addresses pass through the routing firewall setup. In this way, one can gradually migrate the entire networks to a common addressing scheme. When it is complete, the firewall connection can be replaced by a common LAN infrastructure.

There are three reasons for deploying two routers with a common segment (rather than a single router) between them. First, much greater control is possible over the routing Access lists because the first can see all of the first network's routes and services and only pick and choose those that will be permitted to pass to the second network. Second, this sort of filtering can be rather CPU- and memory-intensive in a large network. Splitting the task between two routers helps ensure stability. Third, it might become necessary to implement one or more servers on this intermediate segment. For example, if both networks use IPX for their email services, then the network designer can implement an email relay server on this intermediate segment. Since this LAN segment is visible to both sides, it is also a natural place to put central file servers or servers that download common information to remote user servers.

IPX itself is a connectionless protocol and similar in concept to IP. Programmers can build connectionless applications to run over IP using UDP. For applications that require connections, with the network protocol ensuring that all of the packets arrive intact and in order, an IP programmer would instead use TCP.

IPX also has a connection-based transport layer called Sequenced Packet Exchange (SPX). Like UDP and TCP, SPX includes the concept of well-known port numbers to ensure that a server knows to which application to connect the session.

Dynamic Routing

IPX has dynamic routing protocols that are in many ways similar to those that I already discussed for IP. They fall into the same general categories of Distance Vector and Link State protocols, and they apply many of the same loop-avoidance mechanisms. The basic goals are the same—to find the best multihop path to a particular destination automatically, to converge quickly after topology changes, and to eliminate loops.

Novell RIP and SAP

The services or applications that any particular server has to offer are described by Service Advertisement Protocol (SAP) packets that are sent around the network. End-user workstations receive these SAP broadcast packets and use them to build a list of all available services on the network.

Running parallel to these Service Advertisements is a routing protocol called Routing Information Protocol (RIP). IPX RIP shares several similarities to IP RIP. Both are Distance Vector algorithms. However, while IPX RIP keeps track of the number of hops to a destination, it doesn't use the information in exactly the same way as IP RIP. Both protocols use the hop count metric to avoid loops. In the case of IPX RIP, the actual routing decisions are made according to which route has a shorter time delay to reach the destination.

Time delay is measured in *ticks*. The length of a tick is selected so that there are 65,535 ticks in an hour (65,535 is the largest number that can be expressed in 16 bits). Thus, there are roughly 18.2 ticks in a second, each one being about 55 milliseconds long. RIP makes its routing decisions based on this time delay and uses the hop count only as a "tie breaker" when two paths have the same net time delay.

RIP routing updates are made on a schedule. Each RIP packet can contain up to 50 routing table entries, and each device attempts to pass its entire routing table along to all of its neighbors. The exception to this situation is the fact that IPX RIP employs a Split Horizon algorithm that does not pass routing information back to the device from which it was originally learned. At each successive hop, the devices increment the hop counts indicated in the received routing table. They also add to the time delay a new delay that is measured by the network-interface card.

The SAP protocol carries information about what devices support what application services. This information does not change as frequently as routing information. Generally, if a server is available, then all of the services it offers are also available. Thus, SAP generally works as a query and response.

When a new workstation appears on a network, it sends out a general query looking for information about what servers are available on the network and what services they support. When a new server appears on the network, its neighbors ask it what services are available on the network. When a server stops responding to queries, its services are eventually flushed from the SAP tables of other devices on the network.

Since NetWare is intended to operate across large network environments, a user on a remote LAN segment must be able to get information about the services supported by central servers in the computer room. To make this possible, SAP information is relayed around the entire network from router to router. In this way, every device is able to see a list of available services anywhere in the network.

This SAP information includes some routing information. It is not sufficient to say only that a server named ACCOUNTING supports a database application. The network has to know where that ACCOUNTING server is. However, although these SAP packets include routing information, this information is not used to route the packets. The information used to route packets comes from RIP. Therefore, one of the most confusing problems in an IPX network comes when the RIP and SAP information is inconsistent.

This is particularly true when filtering either RIP or SAP. This filtering is often done to control the size of the routing and service tables on routers and servers. Also, because RIP periodically updates all of its neighbors with its entire routing table, network engineers often want to filter RIP to control bandwidth. Later in this chapter, I explain why too much SAP on a large network is a potentially greater problem. Thus, SAP filtering is usually more restrictive than RIP filtering.

Unless properly controlled, RIP and SAP traffic can cause serious congestion problems, particularly on low-speed WAN links. RIP and SAP, however, are distinct protocols, so they must be filtered separately.

It is not uncommon to wind up with inconsistent filters. Then the network can get into a situation in which an end-user workstation sees that a server called ACCOUNTING offers a database service, but cannot reach that server. Conversely, if the RIP but not the SAP is present, then the user will not even see this service, but might connect to other services on the same LAN segment, or even the same server. This is one of the most common network problems on a large IPX network.

An up-to-date list of registered Novell SAP numbers can be found online at *http://www.isi.edu/in-notes/iana/assignments/novell-sap-numbers/*.

EIGRP

Cisco's EIGRP protocol is capable of supporting IPX, as well as IP (it also can distribute AppleTalk routing information). EIGRP distributes both route and service information. That is, it replaces both RIP and SAP. If a network uses EIGRP, it is important to disable IPX RIP and SAP on all router-to-router links.

However, on the router-to-server links, RIP and SAP must be enabled. Because RIP and EIGRP calculate metrics differently, routing tables can become terribly confused if both protocols are present between two adjacent routers. Always take care to disable or filter out the one that is not in use.

EIGRP can provide several important efficiencies over standard RIP and SAP. First, it supports a much larger network radius. A RIP network can have at most 15 hops between any two networks. This is for exactly the same reason that the IP RIP maximum size is 15 hops. The maximum size of an EIGRP network depends on the architecture. Usually one encounters problems due to too many devices before exhausting the theoretical maximum number of hops.

IPX EIGRP works essentially the same way as IP EIGRP. The main conceptual difference is that IPX EIGRP must carry SAP information, as well as routing information. Again, these updates are performed separately and can be filtered separately. Thus, the network actually still has the same potential problems with route and SAP information being inconsistent. This situation is almost impossible to avoid.

The chief advantage of using EIGRP over RIP and SAP is its bandwidth economy. EIGRP only distributes changes to its tables, rather than sending the entire table periodically. If there are no updates, then neighboring routers only exchange HELLO packets. Conversely, RIP and SAP must periodically distribute their entire tables to ensure consistency.

Another potential advantage of using EIGRP is the availability of equal-cost multipath routing. This routing represents a significant advantage in IP networks. However, I

usually try to vary routing costs so that one path is absolutely preferred in IPX. This is because some IPX applications do a poor job of recovering when packets are delivered out of order.

In general, when one has equal-cost multipath routing, the routers distribute the packets among all possible paths. This means that two successive packets will take different paths through the network. It is possible that they will arrive in inverted order. For a well-behaved application this rarely presents a problem. But some IPX applications do not cope well with packet-sequence errors.

It should be noted that some IP applications also suffer from this malady, but the IP world has had equal-cost multipath routing for a long time. Consequently, natural selection has eliminated most of these unfit applications. However, in the IPX universe, equal-cost multipath routing has been introduced relatively recently. Therefore, many legacy IPX applications behave poorly in this environment.

NLSP

Novell also has created a more efficient routing protocol to overcome some deficiencies of RIP and SAP. This protocol, called Novell Link State Protocol (NLSP), is derived from the OSI Intermediate System to Intermediate System protocol (IS-IS). IS-IS is not discussed in this book, but NLSP shares many similarities with OSPF, so I discuss it by analogy with OSPF.

As a replacement for RIP, NLSP carries all of the routing information for an IPX network. As a replacement for SAP, it also carries service advertisements. However, NLSP does not completely replace RIP and SAP. End stations still require these protocols to find their servers.

The usual mode of operation for NLSP is to run RIP and SAP on the local segments. Then the servers on these local segments speak NLSP back to the router (or routers) that provide network connectivity to this segment. Router-to-router communication then uses NLSP for the main infrastructure of the network.

NLSP works best when all servers and routers in the IPX network use NLSP and only the end station–to–server communication uses RIP and SAP.

Like OSPF, NLSP is organized hierarchically into an Autonomous System (AS) that holds several areas. Each AS has an associated NLSP System ID number that is common throughout the network. Areas in NLSP serve the same functions as they do in OSPF. They allow network address summarization, which in turn results in efficient routing. They allow the Link State database to be broken up.

All routers and servers in any particular NLSP area share a common Link State database that is updated by flooding incremental changes, exactly as in OSPF. However, like OSPF, routers and servers in one area do not see the Link State information for routers and servers in a different area.

NLSP areas are defined according to the IPX summary addresses for the enclosed networks. To use NLSP effectively, it is important to use areas for exactly the same reasons as in OSPF. As in OSPF, effective summarization is important for areas to work properly. However, unlike OSPF, areas do not function at all if the enclosed networks cannot be summarized.

An NLSP area is specified by an IPX network and mask that together summarize all IPX network addresses in the area. For example, one could specify an area with the address `00258A00` and mask `FFFFFF00`. Then this area would include the networks `00258A00`, `00258A01`, and so forth up to `00258AFF`.

As with IP address masks, you can use masks that break the range at any bit. So another valid area could be `030AC000` with a mask of `FFFFE000`. In this case, the range included in this area is `030AC000` to `030AC1FF`. Writing these pairs out in binary, as in Table 7-1, helps to show how they work.

Table 7-1. IPX address mask pair examples

	Address / Mask			
Hx	00258A00 / FFFFFF00			
Binary network	00000000	00100101	10001010	00000000
	(00)	(25)	(8A)	(00)
Binary mask	11111111	11111111	11111111	00000000
	(FF)	(FF)	(FF)	(00)
Allowed range	00000000	0010101	10001010	00000000 to 11111111 (00) to (FF)
	(00) only	(25) only	(8A) only	
Hex	030AC000 / FFFFE000			
Binary network	00000011	00001010	11000000	00000000
	(03)	(0A)	(C0)	(00)
Binary mask	11111111	11111111	11111110	00000000
	(FF)	(FF)	(FE)	(00)
Allowed range	00000011	00001010	11000000 and 11000001	00000000 to 11111111 (00) to (FF)
	(03) only	(0A) only	(C0) and (C1)	

This summarization property of areas has important design implications. It means that designers must be extremely careful about how they allocate their IPX network numbers. Most IPX networks that were initially implemented with RIP never had any requirement for this sort of summarization. Consequently, for many organizations, the conversion from RIP and SAP to NLSP requires that all servers be readdressed.

The language of NLSP diverges somewhat from OSPF. NLSP defines three different levels of routing. Level 1 routing occurs within an area, Level 2 routing occurs between areas, and Level 3 routing occurs between ASes.

OSPF requires that an Area 0 must sit at the center of the AS. Then all other areas are connected to this area directly by means of Area Border Routers. NLSP does not have this restriction. It is possible to construct NLSP areas in somewhat arbitrary configurations, with Level 2 routing taking place between them. However, the OSPF architectural model is good and should be followed in NLSP as well.

It might seem tempting to designate the central area with a network number and mask pair of 00000000 and 00000000 by analogy with OSPF's Area 0. In this way, the central area would effectively include all possible IPX network numbers. But including these numbers is not a good idea because it implies that the central area actually encloses all other areas, which is not possible. The central area is just another area, similar to all of the others. It contains a group of routers and servers that communicate using Level 1 routing. It also communicates to the other areas using Level 2 routing. Thus, the central area must have a summary address of its own that is distinct from every other area.

Figure 7-1 shows how one might use NLSP to build a hierarchical network. Note that in this picture only one connection exists between each "leaf" area and the central area. This arrangement is only to make the picture easier to read. As with OSPF, these key links should always be made redundant. In fact, NLSP supports an equal-cost multipath mode just as OSPF does. The same basic design principles for redundancy apply to both.

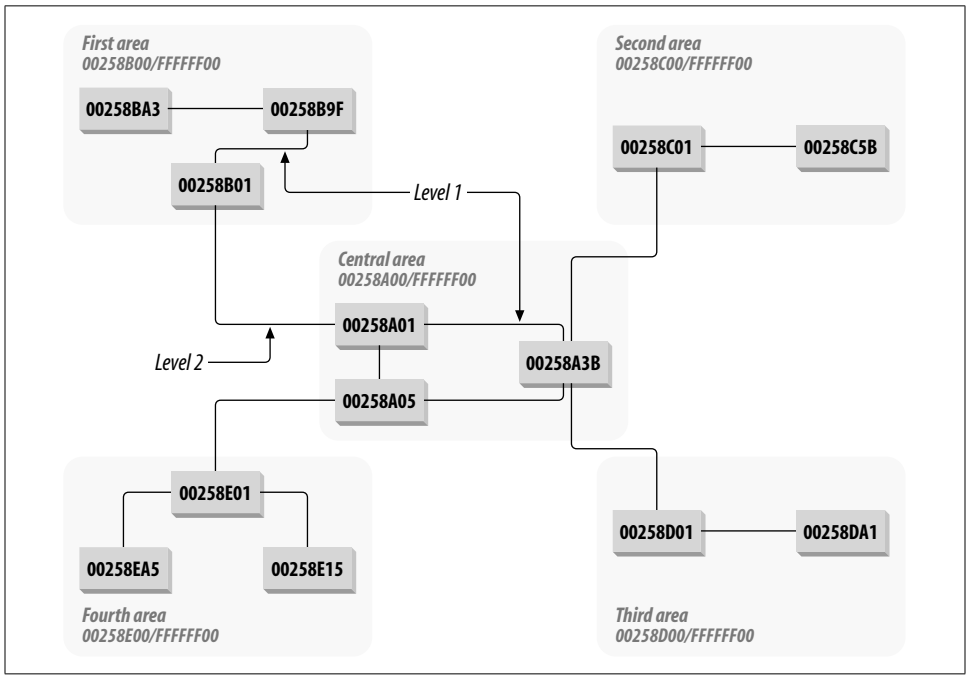

Figure 7-1. A hierarchical NLSP network design

In an IPX network of any size, it is important to limit the number of entries in the Service Advertisement table. This limitation is not merely for bandwidth reasons. Using NLSP or EIGRP makes it possible to drastically reduce the bandwidth taken to distribute this information.

The problem with a large IPX network is simply the size of the table. It is not uncommon for a large IPX network with hundreds of servers to have thousands or tens of thousands of advertised services. This is because every network-attached printer must send a SAP. Similarly, every Windows NT workstation running IPX sends out at least one SAP by default unless SAP is disabled, and every server generally runs several services besides simple file sharing, remote console, and directory services.

The table size for large networks adds up to a huge amount of information that must be distributed to every IPX router and server in the network. Each one of these devices is responsible for redistributing this information to every device downstream from it. In many cases, it represents more information than the routers can reliably handle. They start to run out of memory, and, worse, they start to run out of the CPU power required to process the data.

The vast majority of these advertised services originate with end devices such as workstations and printers. They are not required anywhere but in their originating segment. Thus, it is critically important for network stability that the routers must filter out all nonessential SAP information and prevent it from crossing the network.

The most appropriate place to do this filtering is usually on the router that connects the LAN Access segment to the network Distribution Level. Since a good network avoids using servers of any type as routers—preferring real routers—filtering on the servers isn't necessary. Rather, it all must be done on the routers.

General IPX Design Strategies

There are a few basic design strategies that can make an IPX network more efficient and reliable. Some of these strategies, like special addressing schemes, are analogous to good design principles that I have already discussed for IP networks. Others, such as those having to do with minimizing SAP traffic, are specific to IPX.

IPX Addressing Schemes

As discussed earlier in the section on NLSP, IPX route summarization can present a problem for many networks. However, there is a relatively tidy solution to this problem for networks that run IP and IPX in parallel. If the IP network runs OSPF and the IP addressing scheme has been properly constructed to allow area route summarization, then it is possible to derive IPX addresses from IP addresses.

You can derive these addresses easily. IP and IPX addresses contain the same number of bytes, and they both summarize from the left. Thus, you can do a decimal-to-hexadecimal conversion of the IP address to get the IPX network number.

For example, if the IP address of a file server is `10.1.21.15`, then you can convert the four bytes to hexadecimal notation as `0A01150F`. This is the address of the file server itself, so you can use this address for the IPX Internal Network Number. The External Network Number is effectively the address of the LAN segment.

In this case, the subnet's address is `10.1.21.0`, so the IPX External Network Number would be `0A011500`. If there was a second file server on this segment with an IP address of `10.1.21.16`, then its IPX Internal Network Number would simply be `0A011510`, and it would have the same External Network Number as the first server.

Now, if you have built your OSPF areas properly, you should be able to summarize all addresses inside of an area. If the area containing this example server's LAN is summarized as `10.1.0.0/16`, then the NLSP area will become `0A010000` with a mask of `FFFF0000`. Everything maps perfectly between the two protocols if you choose to number your servers in this way.

If the IPX network does not use NLSP, then there are essentially no restrictions on how the IPX network numbers are allocated. In many networks, IPX network numbers are also randomly distributed throughout the network.

This random distribution is a poor strategy, however. It makes sense to use IPX network numbers that correspond to IP addresses for several reasons. First, most modern networks using IPX also use IP, so there is a natural convergence between the two. Second, if one day the network is converted to NLSP, there is no need to readdress every device. Third, if a simple rule gives the correlation between the IP and IPX addresses, then troubleshooting is much simpler.

This last point deserves extra comment. It is relatively common to build IPX networks that also run IP, so it is natural to bind the IP protocol to an interface on the server. When troubleshooting network problems, it is often useful to send test packets to see whether point A can reach server B.

However, IPX does not have a universally standard equivalent to the IP PING utility. IPX PING does exist, but there are several different standards, and they do not work together. Furthermore, for a server to respond to an IPX PING request, it must have the appropriate NetWare Loadable Module (NLM) loaded. Therefore, there are many reasons why an IPX PING test might not work, even though nothing is wrong with the network.

However, all IP devices should support ICMP ping (firewalls are a notable exception to this rule). If a network administrator is concerned about network connectivity between two segments, using an IP ping test can be useful, even though this is a

different protocol. If an IP ping works, but there is no IPX connectivity, then the administrator can focus on IPX issues immediately. If neither work, then it is more likely that the problem is with physical connectivity.

Networks should always be designed as easy to manage, and troubleshooting is an important part of management. Anything you can do to make troubleshooting easier will give you a more reliable network in the long run.

RIP and SAP Accumulation Zones

In a large IPX network the routers are likely to have to employ significant amounts of route and SAP filtering. This filtering works in two directions. From the edges of the network into the Core, the network should prevent unnecessary SAP information from causing bandwidth or memory problems. From the Core out to the edges, it should distribute only the required routes and SAPs.

In most cases, it is not necessary for users in one department to see the local server for another department. They both may require access to a central server that handles global authentication, however.

To handle this situation, a designer can create a RIP and SAP Accumulation Zone somewhere in the Core of the network. An example of this configuration is shown in Figure 7-2. This figure shows four different user area groups. These groups may be NLSP areas, or there may be some other functional organization. Routers in each user area have redundant connections to a pair of Core routers. These connections provide extra fault tolerance.

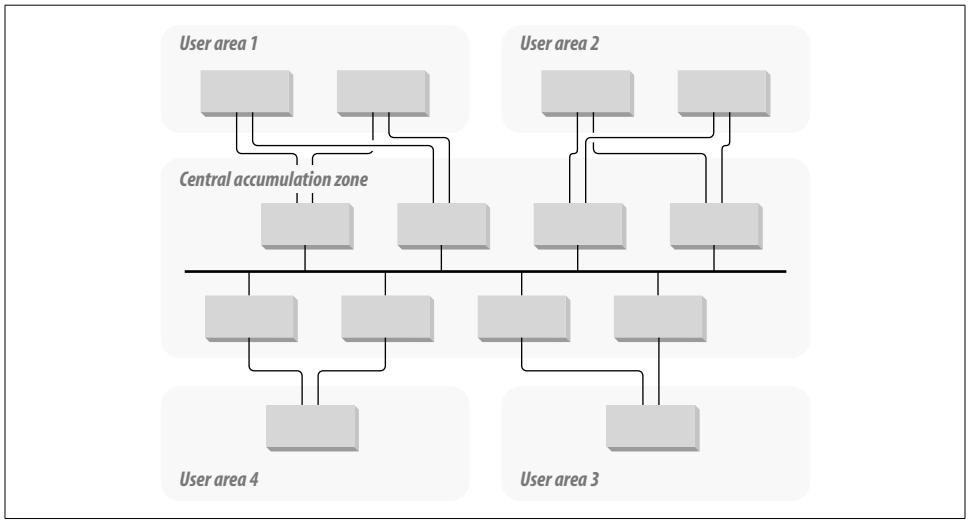

Figure 7-2. IPX network design with RIP and SAP accumulation zone

The user-area routers distribute RIP and SAP information inward to this redundant pair of routers. In the center of the picture is an Ethernet segment that I call the Accumulation Zone. Each router that is connected to this Accumulation Zone shares all of the RIP and SAP information that it receives with this segment. Note that this information need not be collected using either the RIP or SAP protocols. This model works equally well for any dynamic routing protocol.

Then RIP and SAP information is distributed back out to the user areas by the Accumulation Zone routers. Note that none of these routers actually needs to have the entire route or SAP tables. Rather, they can all just dump their tables onto this central segment for others to use as required. They can also take in only those routes and SAPs that they themselves require.

The advantage to this approach is that a central point in the network has all the RIP and SAP information that anybody could ever need. At the same time, though, keeping track of this information doesn't overwhelm any device.

Suppose there is a sudden requirement for a user in one user area to have access to a particular server in another area. Then you can simply allow its Accumulation Zone routers to receive and distribute this information to that user's own server. This provides maximum flexibility to managing the IPX network. At the same time, it avoids the most serious problems with having to support a large route or SAP table.

Efficiency in IPX Networks

I conclude this section with a brief discussion of other basic things you can do to improve efficiency in an IPX network.

I already mentioned the need to keep the number of SAPs to a bare minimum. This information can be gathered in an Accumulation Zone, as mentioned earlier. However, in a large IPX network, it can easily overwhelm the memory and bandwidth resources of the network devices to maintain all of it. Thus, the first thing to do is to create a thorough list of filters to restrict what routes and services are advertised.

Another aspect of the same issue is limiting the amount of server-to-server traffic. Usually this sort of traffic is not necessary. You often need to allow communication between user-area servers and central servers. However, you should try to keep this traffic flow in a star format as much as possible.

Finally, in any large IPX network, it is critical to avoid using RIP and SAP protocols. These protocols work well in small- to medium-sized networks. But remember that routers and servers must all take part in the routing protocols. Suppose, for example, that two servers communicate via RIP, but are separated by three routers. Then the internal network numbers of these servers are, in fact, five hops apart, so an IPX network is usually somewhat larger than it appears on the surface.

In even a moderate-sized network, it is not difficult to find distances that are greater than 15 hops, which is too large for RIP to handle. As I already mentioned, RIP and SAP scale rather poorly for bandwidth usage. It is a good idea to try to move away from these protocols in any large-scale IPX network.

It is possible to make an IPX network appear smaller than it is, however, by tunneling the IPX traffic inside of IP. For example, suppose that getting from a user-area router to the central Accumulation Zone requires several router-to-router hops. You can make it require a single hop by configuring an IPX tunnel between the Accumulation Zone router and the last user-area router before the server.

Even in this case, however, you should avoid using RIP and SAP. Although this design avoids the hop-count problem, you still have to be concerned about the bandwidth-usage problem with these protocols.