CHAPTER

Routing and Peering 13

What You Will Learn

In this chapter, you will learn about how routing differs from switching, the other network layer technology. We'll compare connectionless and connection-oriented networking characteristics and see how *quality of service* (QoS) can be supported on both.

You will learn what a *routing protocol* is and what they do. We'll investigate the differences between interior and exterior routing protocols as the terms apply to an ISP. We'll also talk about *routing policies* and the role they play on the modern Internet.

In Chapter 9, we introduced the concept of forwarding packets hop by hop across a network of interconnected routers and LANs. This process is loosely called "routing," and that chapter comprised a first look at routing tables (and the associated forwarding tables). In this chapter, we'll discuss how ISPs manipulate their routing tables with routing policies to influence the flow of traffic on the Internet. This chapter will focus more closely on the routing tables on hosts. In Chapters 14 and 15, we discuss in more detail the routing tables and routing policies on the network routers.

This chapter will look at the routing tables on the hosts on the LANs, as shown in Figure 13.1. But we'll also discuss, for the first time, how the two ISPs on the network (called Ace ISP and Best ISP) relate to each other and how their routing tables ensure that traffic flows most efficiently between LAN1 and LAN2. For example, it's obviously more effective to send LAN1–LAN2 traffic over the link between P4 and P2 instead of shuttling onto the Internet from P4 and relying on routers beyond the control of either Best or Ace ISP to route the packets back to P2. (Of course, traffic could flow from P4 to P7, or even end up at P9 to be forwarded to P7, but this is just an example.) But how do the routers know how P2 and P4 are connected? More importantly, how do the routers PE5 and PE1 know how the other routers are connected? What keeps router PE5 from forwarding Internet-bound traffic to P9 instead of P4? And, because P9 is also connected to P4, why should it be a big deal anyway?

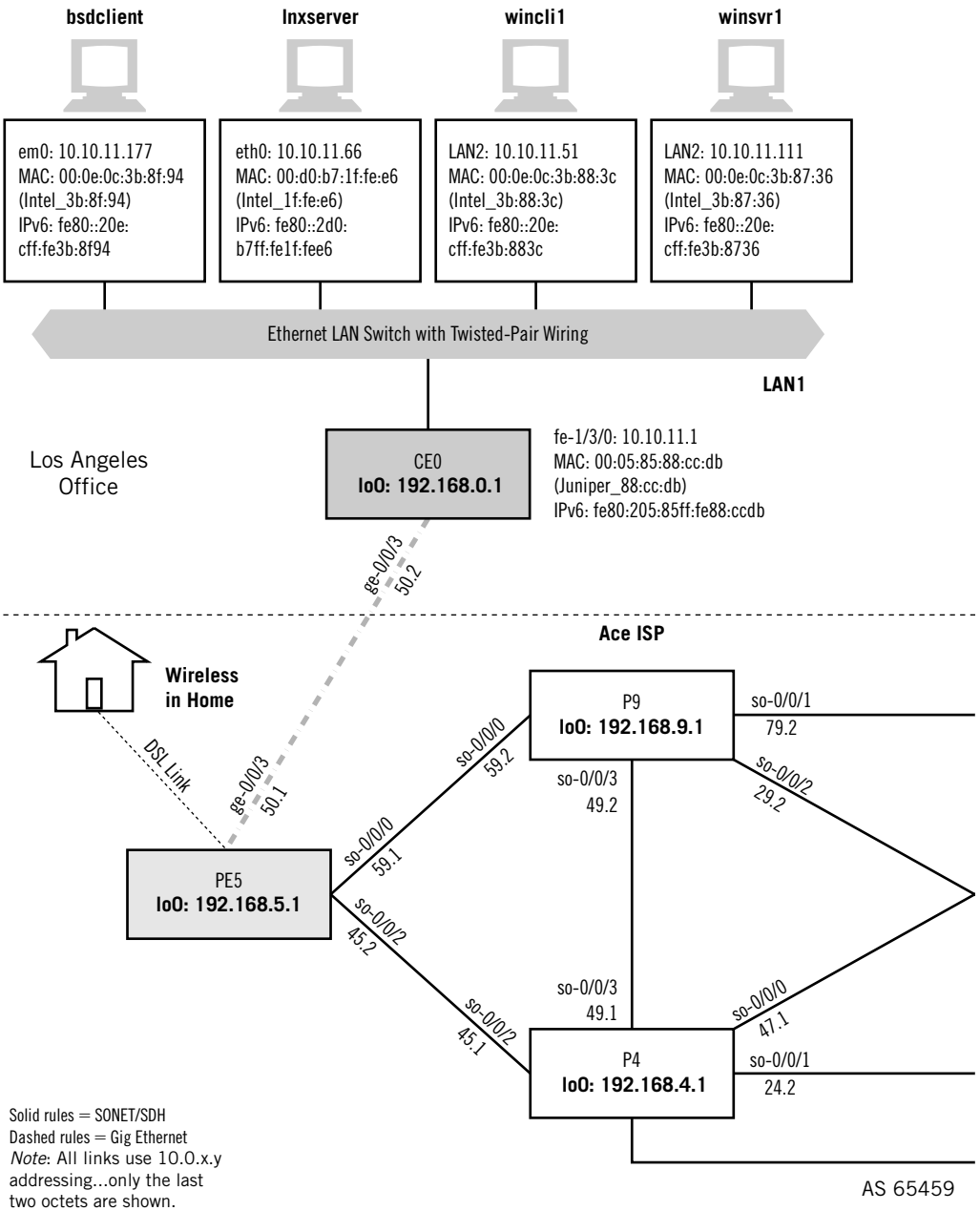
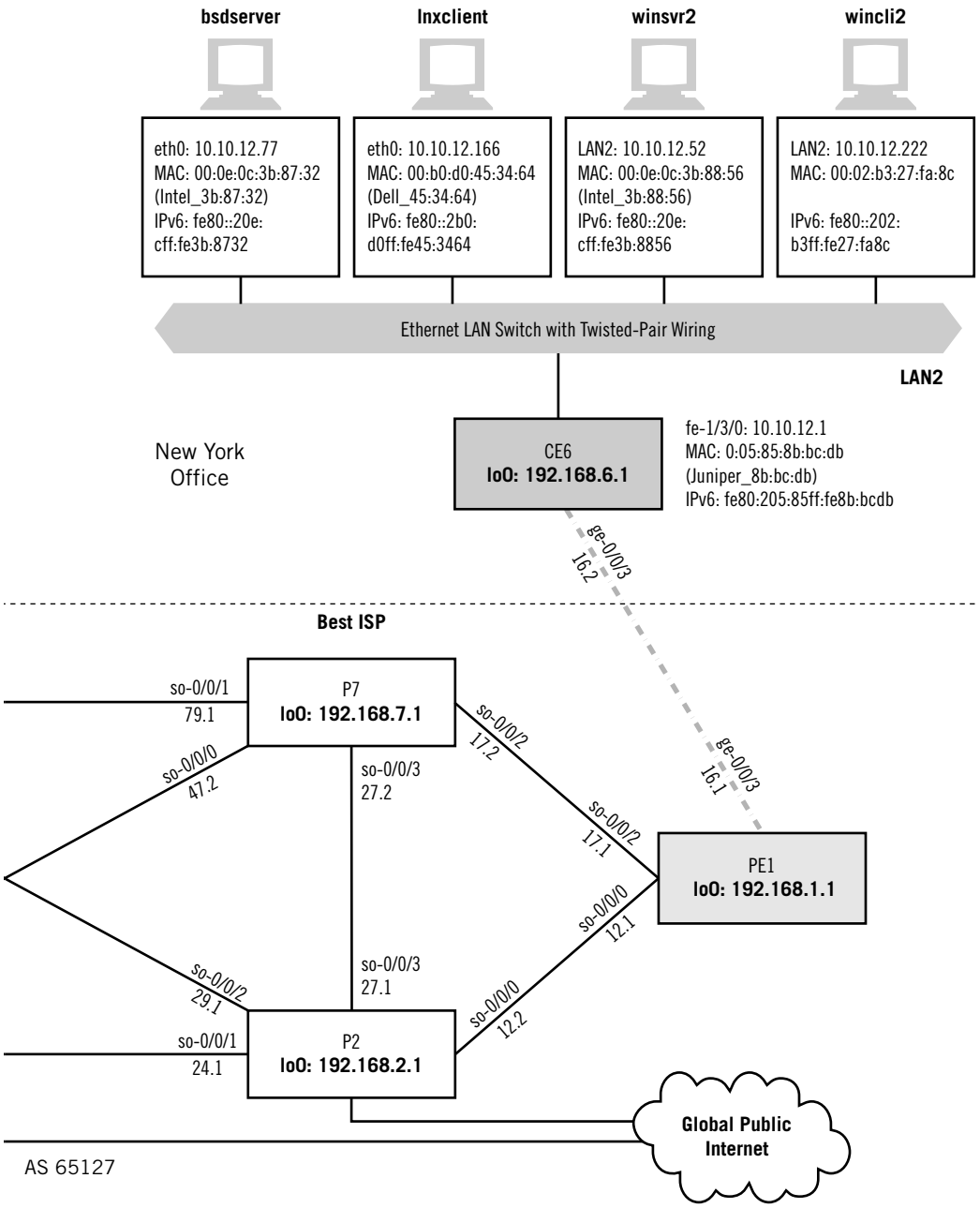


FIGURE 13.1

The hosts on the LANs have routing tables as well as the routers. The ISPs on the Illustrated Network have chosen to implement an ISP peering arrangement.


bsdserver
lnxclient
winsvr2
wincli2
eth0: 10.10.12.77
MAC: 00:0e:0c:3b:87:32
(Intel_3b:87:32)
IPv6: fe80::20e:
cff:fe3b:8732
eth0: 10.10.12.166
MAC: 00:b0:d0:45:34:64
(Dell_45:34:64)
IPv6: fe80::2b0:
d0ff:fe45:3464
LAN2: 10.10.12.52
MAC: 00:0e:0c:3b:88:56
(Intel_3b:88:56)
IPv6: fe80::20e:
cff:fe3b:8856
LAN2: 10.10.12.222
MAC: 00:02:b3:27:fa:8c
IPv6: fe80::202:
b3ff:fe27:fa8c
Ethernet LAN Switch with Twisted-Pair Wiring
LAN2
New York
Office
CE6
lo0: 192.168.6.1
fe-1/3/0: 10.10.12.1
MAC: 0:05:85:8b:bc:db
(Juniper_8b:bc:db)
IPv6: fe80:205:85ff:fe8b:bcdb
ge-0/0/3
16.2
Best ISP
so-0/0/1
79.1
P7
lo0: 192.168.7.1
so-0/0/2
17.2
so-0/0/0
47.2
so-0/0/3
27.2
ge-0/0/3
16.1
so-0/0/2
17.1
PE1
lo0: 192.168.1.1
so-0/0/0
12.1
so-0/0/2
29.1
so-0/0/3
27.1
so-0/0/0
12.2
so-0/0/1
24.1
P2
lo0: 192.168.2.1
Global Public
Internet
AS 65127

This chapter will begin to answer these questions, and the next two chapters will complete the investigation. However, it should be mentioned right away that connectionless routers that route (forward) each packet independently through the network are not the only way ISPs can connect LANs on the Internet. The network nodes can be connection-oriented switches that forward packets along fixed paths set up through the network nodes from source to destination.

We've already discussed connectionless and connection-oriented services at the transport layer (UDP and TCP). Let's see what the differences are between connectionless and connection-oriented services at the network layer.

NETWORK LAYER ROUTING AND SWITCHING

Are the differences between connection-oriented and connectionless networking at the network layer really that important? Actually, yes. The difference between the way connectionless router networks handle traffic (and link and node failures) is a major reason that IP has basically taken over the entire world of networking.

A *switch* in modern networking is a network node that forwards packets toward a destination depending on a locally significant connection identifier over a fixed path. This fixed path is called a *virtual circuit* and is set up by a signaling protocol (a *switched virtual circuit*, or SVC) or by manual configuration (a *permanent virtual circuit*, or PVC). A *connection* is a logical association of two endpoints. Connections only need be referenced, not identified by "to" and "from" information. A data unit sent on "connection 22" can only flow between the two endpoints where it is established—there is no need to specify more. (We've seen this already at Layer 2 when we looked at the connection-oriented PPP frame.) As long as there is no confusion in the switch, connection identifiers can be reused, and therefore have what is called *local significance only*.

Packets on SVCs or PVCs are often checked for errors hop by hop and are resent as necessary from node to node (the originator plays no role in the process). Packet switching networks offer guaranteed delivery (as least as error-free as possible). The network is also reliable in the sense that certain performance guarantees in terms of bandwidth, delay, and so on can be enforced on the connection because packets always follow the same path through the network. A good example of a switched network is the public switched telephone network (PSTN). SVCs are normal voice calls and PVCs are the leased lines used to link data devices, but frame relay and ATM are also switched network technologies. We'll talk about *public switched network* technologies such as frame relay and ATM in a later chapter.

On the other hand, a *router* is a network node that independently forwards packets toward a destination based on a globally unique address (in IP, the IP address) over a dynamic path that can change from packet to packet, but usually is fairly stable over time. Packets on router networks are seldom checked for errors hop by hop and are only resent (if necessary) from host to host (the originator plays a key role in the process). Packet routing networks offer only "best-effort" delivery (but as error-free as possible). The network is also considered "unreliable" in the sense that certain

performance guarantees in terms of bandwidth, delay, and so on cannot be enforced from end to end because packets often follow different paths through the network. A good example of a router-based network is the global, public Internet.

CONNECTION-ORIENTED AND CONNECTIONLESS NETWORKS

Many layers of a protocol stack, especially the lower layers, offer a choice of connection-oriented or connectionless protocols. These choices are often independent. We've seen that connectionless IP can use connection-oriented PPP at Layer 2. But what is it that makes a *network* connectionless? Not surprisingly, it's the implantation of the network layer. IP, the Internet protocol suite's network layer protocol, is connectionless, so TCP/IP networks are connectionless.

Connection-oriented networks are sometimes called *switched networks*, and connectionless networks are often called *router-based networks*. The signaling protocol messages used on switched networks to set up SVCs are themselves routed between switches in a connectionless manner using globally unique addresses (such as telephone numbers). These call setup messages must be routed, because obviously there are no connection paths to follow yet. Every switched network that offers SVCs must also be a connectionless, router-based network as well.

One of the major reasons to build a connectionless network like the Internet was that it was inherently simpler than connection-oriented networks that must route signaling setups messages and forward traffic on connections. The Internet essentially handles everything as if it were a signaling protocol message. The differences between connection-oriented switched networks and connectionless router networks are shown in Table 13.1.

Table 13.1 Switched and Connectionless Networks Compared by Major Characteristics

Characteristic	Switched Network	Connectionless Network
Design philosophy	Connection oriented	Connectionless
Addressing unit	Circuit identifiers	Network and host address
Scope of address	Local significance	Globally unique
Network nodes	Switches	Routers
Bandwidth use	As allowed by "circuit"	Varies with number and size of frames
Traffic processing	Signaling for path setup	Every packet routed independently
Examples	Frame relay, ATM, ISDN, PSTN, most other WANs	IP, Ethernet, most other LANs

Note that every characteristic listed for a connectionless network applies to the signaling network for a switched network. It would not be wrong to think of the Internet as a signaling network with packets that can carry data instead of connection (call) setup information. The whole architecture is vastly simplified by using the connectionless network for everything.

The simplified router network, in contrast to the switched network, would automatically route around failed links and nodes. In contrast, connection-oriented networks lost every connection that was mapped to a particular link or switch. These had to be re-established through signaling (SVCs) or manual configuration (PVCs), both of which involved considerable additional traffic loads (SVCs) or delays (PVCs) for all affected users. One of the original aims of the early "Internet" was explicitly to demonstrate that packet networks were more robust when faced with failures. Therefore, connectionless networks could be built more cheaply with relatively "unreliable" components and still be resistant to failure. Today, "best-effort" and "unreliable" packet delivery over the Internet is much better than any other connection-oriented public data network not so long ago.

Of course, an Internet router has to maintain a list of every possible reachable destination in the world (and so did signaling nodes in connection-oriented networks), but processors have kept up with the burden imposed by the growth in the scale of the routing tables. A switch only has to keep track of local associations of two endpoints (connections) currently established. We'll talk about multiprotocol label switching (MPLS) in Chapter 17 as an attempt to introduce the efficiencies of switching into router-based networking. (MPLS does not really relieve the main burdens of interdomain routing, but we will see that MPLS has *traffic engineering* capabilities that allow ISPs to shift the paths that carry this burden.)

In only one respect is there even any discussion about the merits of connection-oriented networks versus the connectionless Internet. This is in the area of the ability of connectionless router networks to deliver *quality of service* (QoS).

Quality of Service

It might seem odd to talk about QoS in a chapter on connectionless Internet routing and forwarding. But the point is that in spite of the movement to converge all types of information (voice and video as well as data) onto the Internet, no functional interdomain QoS mechanism exists. QoS is at heart a queue management mechanism, and only by applying these strategies across an entire routing domain will QoS result in any route optimization at all. Even then, no ISP can impose its own QoS methodology on any other.

One of the biggest challenges in quality of service (QoS) discussions is that there is no universal, accepted agreement of just what network QoS actually means. Some sources define QoS quite narrowly, and others define it more broadly. For the purposes of this discussion, a broader definition is more desirable. We'll use six parameters in this book.

CoS or QoS?

Should the term for network support of performance parameters be "class of service" (CoS) or "quality of service" (QoS)? Many people use the terms interchangeably, but in this book QoS is used to mean that parameters can take on almost any value between maximum and minimum. CoS, on the other hand, establishes groups of parameters based on real world values (e.g., bandwidth at 10, 100, or 1000 Mbps with associated delays), and is offered as a "class" to customers (e.g., bronze, silver, or gold service).

Our working definition of QoS in this book is the "ability of an application to specify required values of certain parameters to the network, values without which the application will not be able to function properly." The network either agrees to provide these parameters for the applications data flow, or not. These parameters include things like minimum bandwidth, maximum delay, and security. It makes no sense to put delay-sensitive voice traffic onto a network that cannot deliver delays less than 2 or 3 seconds one way (voice suffers at delays far less than full seconds), or to put digital, wide-screen video onto a network of low-bandwidth, dial-up analog connections.

Table 13.2 shows some typical example values that are used often. In some cases, an array of values is offered to customers as a CoS.

Bandwidth is usually the first and foremost QoS parameters, for the simple reason that bandwidth was for a long time the *only* QoS parameter that could be delivered by networks with any degree of consistency. It has also been argued that, given enough bandwidth (just how much is part of the argument), every other QoS parameter becomes irrelevant.

Jitter is just delay variation, or how much the end-to-end network latency varies from time to time due to effects such as network queuing and link failures, which cause alternate routes to be used. Information loss is just the effect of network errors. Some

Table 13.2 The Six QoS Parameters

QoS Parameter	Example Values (Typical)
Bandwidth (minimum)	1.5 Mbps, 155 Mbps, 1 Gbps
Delay (maximum)	50-millisecond (ms) round-trip delay, 150-ms delay
Jitter (delay variation)	10% of maximum delay, 5-ms variation
Information loss (error effects)	1 in 10,000 packets undelivered
Security	All data streams encrypted and authenticated

applications can recover from network errors by retransmission and related strategies. Other applications, most notably voice and video, cannot realistically resend information and must deal with errors in other ways, such as the use of forward error correction codes. Either way, the application must be able to rely on the network to lose only a limited amount of information, either to minimize resends (data) or to maximize the quality of the service (voice/video).

Availability and reliability are related. Some interpret reliability as a local network quality and availability as global quality. In other words, if my local link fails often, I cannot rely on the network, but global availability to the whole pool of users might be very good. There is another way that reliability is important in TCP/IP. IP is often called an *unreliable* network layer service. This does not imply that the network fails often, but that, at the IP layer, the network cannot be relied on to deliver any QoS parameter values at all, not even minimum bandwidth. But keep in mind that a system built of unreliable components can still be reliable, and QoS is often delivered in just this fashion.

Security is the last QoS parameter to be added, and some would say that it is the most important of all.

Many discussions of QoS focus on the first four items on the parameter list. But reliability and security also belong with the others, for a number of reasons. Security concerns play a large part in much of IPv6. And reliability can be maximized in IP routing tables. There are several other areas where security and reliability impact QoS parameters; the items discussed here are just a few examples.

Service providers seldom allow user application to pick and choose values from every QoS category. Instead, many service providers will gather the typical values of the characteristics for voice, video, and several types of data applications (bulk transfer, Web access, and so on), and bundle these as a *class of service* (CoS) appropriate for that traffic flow. (On the other hand, some sources treat QoS and CoS as synonyms.) Usually, the elements in a CoS suite that a service provider offers have distinctive names, either by type (voice, video) or characteristic ("gold" level availability), or even in combination ("silver-level video service").

The promise of widespread and consistent QoS has been constantly derailed by the continuing drop in the cost (and availability) of network links of higher and higher bandwidth. Bandwidth is a well-understood network resource (some would say the *only* well-understood network resource), and those who control network budgets would rather spend a dollar on bandwidth (known effects, low risk, etc.) than on other QoS schemes such as DiffServ (spotty support, difficult to implement, etc.).

HOST ROUTING TABLES

Now that we've shown that the Illustrated Network is firmly based on connectionless networking concepts, let's look at the routing tables (*not* switching tables) on some of the hosts. Host routing tables can be very short. When initially configured, many of them have only four types of entries.

Loopback—Usually called `lo0` on Unix-based systems (and routers), this is the prefix `127/8` in IPv4 and `::1` in IPv6. Not only used for testing, the loopback is a stable interface on a router (or host) that should not change even if the interface addresses do.

The host itself—There will be one entry for every interface on the host with an IP address. This is a `/32` address in IPv4 and a `/128` address in IPv6.

The network—Each host address has a network portion that gets its own routing table entry.

The default gateway—This tells the host which router to use when the network portion of the destination IP address does not match the network portion of the source address.

Gateway or Edge Router?

A lot of texts simply say that the term "router" is the new term for "gateway" on the Internet, but that this old term still shows up in a number of acronyms (such as IGP). Other sources use the term "gateway" as a kind of synonym for what we've been calling the customer-edge router, meaning a router with only two types of routing decisions, that is, local or Internet. A DSL "router" is really just a "gateway" in this terminology, translating between local LAN protocols and service provider protocols. On the other hand, a backbone router without customer LANs is definitely a router in any sense of the term.

In this book, we'll use the terms "gateway" and "router" interchangeably, keeping in mind that the gateway terminology is still used for the entry or egress point of a particular subnet.

Routing Tables and FreeBSD

FreeBSD systems keep this fundamental information in the `/etc/default/rc.conf` file. But this information can be manipulated with the `ifconfig` command, which we've used already. However, interface information does not automatically jump into the routing table unless the changes are made to the `rc.conf` file. (If the `network_interfaces` variable is kept to the default of `auto`, the system finds its network interfaces at boot time.)

Let's use the `netstat -nr` command to take a closer look at the routing table on `bsdserver`.

```
bsdserver# netstat -nr
Routing tables

Internet:
Destination        Gateway            Flags     Refs      Use  Netif Expire
default            10.10.12.1         UGSc         1       97   em0
10.10.12/24        link#1             UC           2        0   em0
```

```
10.10.12.1        00:05:85:8b:bc:db  UHLW       2        0    em0    335
10.10.12.52       00:0e:0c:3b:88:56  UHLW       0        4    em0   1016
127.0.0.1         127.0.0.1          UH         0     6306    lo0

Internet6:
Destination                       Gateway               Flags   Netif Expire
::1                               ::1                   UH      lo0
fe80::%em0/64                     link#1                UC      em0
fe80::20e:cff:fe3b:8732%em0       00:0e:0c:3b:87:32     UHL     lo0
fe80::%xl0/64                     link#2                UC      xl0
fe80::2b0:d0ff:fec5:9073%xl0      00:b0:d0:c5:90:73     UHL     lo0
fe80::%lo0/64                     fe80::1%lo0           Uc      lo0
fe80::1%lo0                       link#4                UHL     lo0
ff01::/32                         ::1                   U       lo0
ff02::%em0/32                     link#1                UC      em0
ff02::%xl0/32                     link#2                UC      xl0
ff02::%lo0/32                     ::1                   UC      lo0
```

FreeBSD merges the routing and ARP tables, which is why hardware addresses (and their timeouts) appear in the output. The `C` and `c` flags are host routes, and the `S` is a static entry.

To manually configure an Ethernet interface and add the route to the routing table, we use the `ifconfig` and `route` commands.

```
bsdserver# ifconfig em0 inet 10.10.12.77/24
bsdserver# route add -net 10.10.12.77 10.10.12.1
```

Routing and Forwarding Tables

Remember, the routing tables we're looking at here are tables of routing information and mainly for human inspection. Generally, everything the system learns about the network from a routing protocol is put into the routing table. But not all of the information is used for packet forwarding.

At the software level, the system creates a forwarding table in a much more compact and machine-useable format. The forwarding table is used to determine the output, the next-hop interface (if the system is not the destination). However, we'll use the friendly routing tables to illustrate the routing process, as is normally done.

Routing Tables and RedHat Linux

RedHat Linux systems keep most network configuration information in the `/etc/sysconfig` and `/etc/sysconfig/network-scripts` directories. The hostname, default gateway, and other information are kept in the `/etc/sysconfig/network` file. The Ethernet

interface-specific information, such as IP address and network mask for `eth0`, is in the `/etc/sysconfig/network-scripts/ifcfg-eth0` file (loopback is in `ifcfg-lo0`).

Let's look at the `lnxclient` routing table with the `netstat -nr` command.

```
[root@lnxclient admin]# netstat -nr
Kernel IP routing table
Destination  Gateway     Genmask         Flags  MSS  Window  irtt  Iface
10.10.12.0   0.0.0.0     255.255.255.0   U        0  0          0  eth0
127.0.0.0    0.0.0.0     255.0.0.0       U        0  0          0  lo
0.0.0.0      10.10.12.1  0.0.0.0         UG       0  0          0  eth0
```

Oddly, the host address isn't here. This system does not require a route for the interface address bound to the interface. The loopback entries are slightly different as well. Only network entries are in the Linux routing table. If we added a second Ethernet interface (`eth1`) with IPv4 address `172.16.44.98` and a different default router (`172.16.44.1`), we'd add that information with the `ipconfig` and `route` commands.

```
[root@lnxclient admin]# ifconfig eth1 172.16.44.98 netmask 255.255.255.0
[root@lnxclient admin]# route add default gw 172.16.44.0 eth1
```

We're not running IPv6 on the Linux systems, so no IPv6 information is displayed.

Routing and Windows XP

Windows XP, of course, handles things a little differently. We've already used `ipconfig` to assign addresses, and Windows XP uses the `route print` command to display routing table information, such as on `wincli2`.

```
C:\Documents and Settings\Owner>route print
===========================================================================
Interface List
0x1 ........................... MS TCP Loopback interface
0x2 ...00 02 b3 27 fa 8c ...... Intel(R) PRO/100 S Desktop Adapter - Packet
Scheduler Miniport
===========================================================================
===========================================================================
Active Routes:
Network Destination        Netmask          Gateway       Interface   Metric
          0.0.0.0          0.0.0.0      10.10.12.1    10.10.12.222       20
       10.10.12.0    255.255.255.0    10.10.12.222    10.10.12.222       20
     10.10.12.222  255.255.255.255      127.0.0.1       127.0.0.1       20
   10.255.255.255  255.255.255.255    10.10.12.222    10.10.12.222       20
        127.0.0.0        255.0.0.0      127.0.0.1       127.0.0.1       1
        224.0.0.0        240.0.0.0    10.10.12.222    10.10.12.222       20
  255.255.255.255  255.255.255.255    10.10.12.222    10.10.12.222       20
Default Gateway:        10.10.12.1
===========================================================================
Persistent Routes:
  None
```

The table is an odd mix of loopbacks, multicast, and host and router information. Persistent routes are static routes that are not purged from the table. We can delete information, add to it, or change it. If no gateway is provided for a new route, the system attempts to figure it out on its own.

The IPv6 routing table is not displayed with `route print`. To see that, we need to use the `IPv6 rt` command. The table on `wincli2` reveals only a single entry for the link-local–derived IPv6 address of the default router.

```
C:\Documents and Settings\Owner>ipv6 rt
::/0 -> 5/fe80:5:85ff:fe8b:bcdb pref 256 life 25m52s <autoconf>
```

This won't even let us ping the `wincli1` system on LAN1, even though we know to what router to send the IPv6 packets.

```
C:\Documents and Settings\Owner>ping6 fe80::20c:cff:fe3b:883c

Pinging fe80::20c:cff:fe3b:883c with 32 bytes of data:

No route to destination.
   Specify correct scope-id or use -s to specify source address.
No route to destination.
   Specify correct scope-id or use -s to specify source address.
No route to destination.
   Specify correct scope-id or use -s to specify source address.
No route to destination.
   Specify correct scope-id or use -s to specify source address.

Ping statistics for fe80::20c:cff:fe3b:883c:
   Packets: Sent = 4, Received = 0, Lost = 4 (100% loss)
```

What's wrong? Well, we're using link-local addresses, for one thing. Also, we have no way to get the routing information known about LAN2 and router CE6 to LAN1 and router CE0. That's the job of the Interior Gateway Protocols (IGPs), the types of routing protocols that run between ISP's routers. Why do we need them? Let's look at the Internet first, and then we'll use an IPG in the next chapter so that the IPv6 ping works.

THE INTERNET AND THE AUTONOMOUS SYSTEM

Before taking a more detailed look at the routing protocols that TCP/IP uses to ensure that every router knows how to forward packets closer to their ultimate destination, it's a good idea to have a firm grasp of just what routing protocols are trying to accomplish on the modern Internet. The Internet today is composed of interlocking network pieces, much like a jigsaw puzzle of global proportions. Each piece is called an *autonomous system* (AS), and it's convenient to think of each ISP as an AS, although this is not strictly true.

Routing Protocols and Routing Policies

A routing *protocol* is run on a router (and can be run on a host) to allow the router to dynamically learn about its network neighborhood and pass this knowledge on until every router has built a consistent view of the network "map" and the least cost ("best") place to forward traffic toward any reachable destination. Until the protocol *converges* there is always the possibility that some routers do not have the latest view of the network and might forward packets incorrectly. Actually, it's possible that some of the "maps" never converge and that some less-than-optimal path might be taken. But that need not be a disaster, although the reasons are far beyond this simple introduction.

A routing *policy* can be defined as "a rule implemented on the router to determine the handling of routing protocol information." An example of an ISP's routing policy rule is to "accept no routing protocol updates from hosts or routers not part of this ISP's network." This rule, intended to minimize the effects of malicious users, can be combined with others to create an overall routing policy for the whole ISP.

The term should not be confused with *policy routing*. Policy routing is usually defined as the forwarding of packets based not only on destination address, but also on some other fields in the TCP/IP header, especially the IPv4 ToS bits. Confusingly, policy routing can be made more effective with routing policies, but this book will not deal with policy routing or QoS issues.

Routing protocols do not and cannot blend all these ASs together into a seamless whole all on their own. Routing *protocols* allow routers or networks to share adjacency information with their neighbors. They establish the global connectivity between routers, within an AS and without, and ASs in turn establish the global connectivity that characterizes the Internet. Routing *policies* change the behavior of the routing protocols so AS connectivity is made into what the ISPs want (usually, ISPs add some term like "AS connectivity is made more effective and efficient" but many times routing policy doesn't do this, as we'll see).

Routers are the network nodes of the global public Internet, and they pass IP address information back and forth as needed. The result is that every router knows how to reach every IP network (really, the IP prefix) anywhere in the world, or at least those that advertise that they are willing to accept traffic for that prefix. They also know when a link or router has failed, and thus other networks might then be (temporarily) unreachable. Routers can dynamically route around failed links and routers, unless the destination network is connected to the Internet by only one link or happens to be right there on the local router.

There are no users on the router itself that originate or read email (as an example), although routers routinely take on a client or a server role (or both) for configuration and administrative purposes. Routers almost always just pass IP packet traffic through

from one interface to another, input port to output port, while trying to ensure that the packets are making progress through the network and moving one step closer to its destination. It is said that routers route packets "hop by hop" through the Internet. In a very real sense, routers don't care if the packet ever reaches the destination or not: All the router knows is that if the IP address prefix is X, that packet goes out port Y.

THE INTERNET TODAY

There is really no such thing as *the* Internet today. The *concept* of "the Internet" is a valid one, and people still use the term all the time. But the Internet is no longer a thing to be charted and understood and controlled and administered. What we have is an interlocking grid of ISPs, an *ISP* "*grid-net*," so to speak. Actually, the graph of the Internet is a bit less organized than this, although ISPs closer to the core have a higher level of interconnection than those at the edge. This is an interconnected mesh of ISPs and related Internet-connected entities such as government bureaus and learning institutions. Also, keep in mind that in addition to the "big-I internet," there are other internetworks that are not part of this global, public whole.

If we think of the Internet as a unity, and have no appreciation of actual ISP connectivity, then the role of routing protocols and routing policies on the Internet today cannot be understood. Today, Internet talk is peppered with terms like *peers*, *aggregates*, *summaries*, *Internet exchange points (IXPs)*, *backbones*, *border routers*, *edge routers*, and *points of presence (POPs)*. These terms don't make much sense in the context of the Internet as a unified network.

The Internet as the spaghetti bowl of connected ISPs is shown in Figure 13.2. There are large national ISPs, smaller regional ISPs, and even tiny local ISPs. There are also pieces of the Internet that act as exchange points for traffic, such as the Network Access Points NAPs and IXPs. IXPs can by housed in POPs, formal places dedicated for this purpose, and in various *collocation facilities*, where the organizations rent floor space for a rack of equipment ("broom closet") or larger floor space for more elaborate arrangements, such as redundant links and power supplies. The IXPs are often run by former telephone companies.

Each cloud, except the one at the top of the figure, basically represents an ISP's AS. Within these clouds, the routing protocol can be an IGP such as OSPF, because it is presumed that each and every network device (such as the backbone routers) in the cloud is controlled by the ISP. However, between the clouds, an EGP such as BGP must be used, because no ISP can or should be able to directly control a router in another ISP's network.

The ISPs are all chained together by a complex series of links with only a few hard and fast rules (although there are exceptions). As long as local rules are followed, as determined by contract, the smallest ISP can link to another ISP and thus give their users the ability to participate in the global public Internet. Increasingly, the nature of the linking between these ISPs is governed by a series of agreements known as *peering arrangements*. Peers are equals, and national ISPs may be peers to each other, but

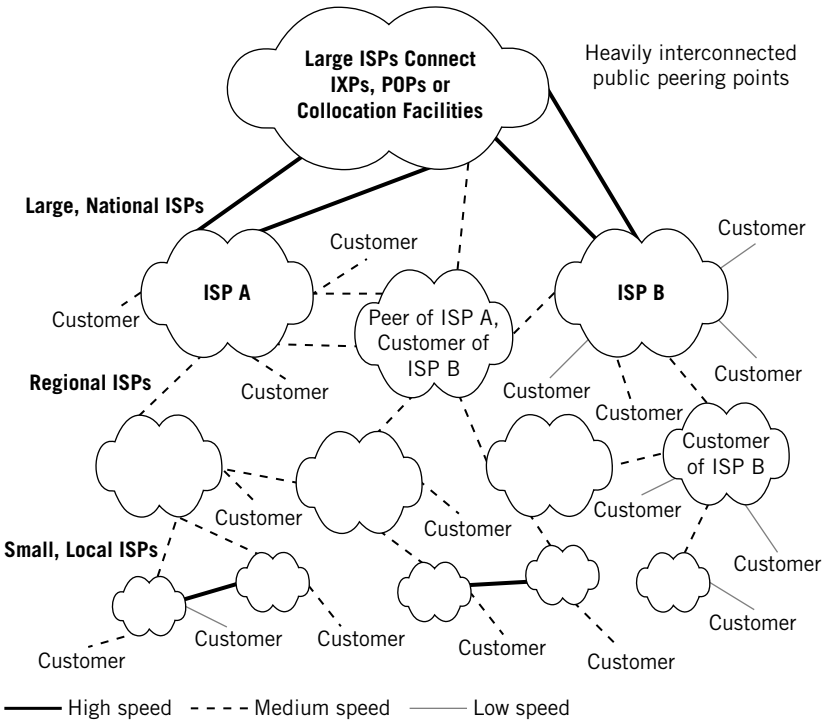


FIGURE 13.2

The haphazard way that ISPs are connected on today's Internet, showing IXPs at the top. Customers can be individuals, organizations, or other ISPs.

treat smaller ISPs as just another customer, although it's not all that unusual for small regional ISPs to peer with each other.

Peering arrangements detail the reciprocal way that traffic is handed off from one ISP (and that means AS) to another. Peers might agree to deliver each other's packets for no charge, but bill non-peer ISPs for this privilege, because it is assumed that the national ISP's backbone will be shuttling a large number of the smaller ISPs' packets. But the national ISP won't be using the small ISP much. A few examples of national ISPs, peer ISPs, and customer ISPs are shown in the figure. This is just an example, and very large ISPs often have plenty of very small customers and some of those will be attached to more than one other ISP and employ high capacity links. There will also be "stub AS" networks with no downstream customers.

Millions of PCs and Unix systems act as clients, servers, or both on the Internet. These hosts are attached to LANs (typically) and linked by routers to the Internet. The LANs and "site routers" are just "customers" to the ISPs. Now, a customer of even moderate size could have a topology similar to that of an ISP with a distinct border, core, and aggregation or services routers. Although all attached hosts conform to the

client–server architecture, many of them are strictly Web clients (browsers) or Web servers (Web sites), but the Web is only one part of the Internet (although probably the most important one). It is important to realize that the clients and servers are on LANs, and that routers are the network nodes of the Internet. The number of client hosts greatly exceeds the number of servers.

The link from the client user to the ISP is often a simple cable or DSL link. In contrast, the link from a server LAN's router to the ISP could be a leased, private line, but there are important exceptions to this (Metro Ethernet at speeds greater than 10 Mbps is very popular). There are also a variety of Web servers within the ISP's own network. For example, the Web server for the ISP's customers to create and maintain their own Web pages is located inside the ISP cloud.

The smaller ISPs link to the backbones of the larger, national ISPs. Some small ISPs link directly to national backbones, but others are forced for technical or financial reasons to link in a "daisy-chain" fashion to other ISPs, which link to other ISPs, and so on until an ISP with direct access to an IXP is reached. Peering bypasses the need to use the IXP structure to deliver traffic.

Many other countries obtain Internet connectivity by linking to an IXP in the United States, although many countries have established their own IXPs. Large ISPs routinely link to more than one IXP for redundancy, while truly small ones rarely link to more than one other ISP for cost reasons. Peer ISPs often have multiple, redundant links between their border routers. (Border routers are routers that have links to more than one AS.) For a good listing of the world's major IXPs, see *http://en.wikipedia.org* under Internet Exchange Point.

Speeds vary greatly in different parts of the Internet. Client access by way of low-speed dial-up telephone lines is typically 33.6 to 56 kbps. Servers are connected by Metro Ethernet or by medium-speed private leased lines, typically 1.5 Mbps. The high-speed backbone links between national ISPs run at yet higher speeds, and between the IXPs themselves, speeds of 155 Mbps (known as OC-3c), 622 Mbps (OC-12c), 2.4 Gbps (OC-48c), and 10 Gbps (OC-192c) can be used, although "$n \times 10$" Gbps Ethernet trunks are less expensive. Higher speeds are always needed, both to minimize large Web site content-transfer latency times (like video and audio files) and because the backbones concentrate and aggregate traffic from millions of clients and servers onto a single network.

THE ROLE OF ROUTING POLICIES

Today, it is impossible for all routers to know all details of the Internet. The Internet now consists of an increasing number of *routing domains*. Each routing domain has its own internal and external routing policies. The sizes of routing domains vary greatly, from only one IP address space to thousands, and each domain is an AS. Many ISPs have only one AS, but national or global ISPs might have several AS numbers. A global ISP might have one AS for North America, another for Europe, and one for the rest of the world. Each AS has a uniquely assigned AS number, although there can be various,

logical "sub-ASs" called *confederations* or *subconfederations* (both terms are used) inside a single AS.

We will not have a lot to say about routing policies, as this is a vast and complex topic. But some basics are necessary when the operation of routers on the network is considered in more detail.

An AS forms a group of IP networks sharing a unified *routing policy framework*. A routing policy framework is a series of guidelines (or hard rules) used by the ISP to formulate the actual routing policies that are configured on the routers. Among different ASs, which are often administered by different ISPs, things are more complex. Careful coordination of routing policies is needed to communicate complicated policies among ASs.

Why? Because some router somewhere must know all the details of all the IPv4 or IPv6 addresses used in the routing domain. These routes can be aggregated (or summarized) as shorter and shorter prefixes for advertisement to other routers, but some routers must retain all the details.

Routes, or prefixes, not only need to be advertised to another AS, but need to be accepted. The decision on which routes to advertise and which routes to accept is determined by routing policy. The situation is summarized in the extremely simple exchange of routing information between two peer ASs shown in Figure 13.3. (Note that the labels "AS #1" and "AS #2" are not saying "this is AS1" or "this is AS2"—AS numbers are reserved and assigned centrally.) The routing information is transferred by the routing protocol running between the routers, usually the Border Gateway Protocol (BGP).

The exchange of routing information is typically bidirectional, but not always. In some cases, the routing policy might completely suppress or ignore the flow of routing information in one direction because of the routing policy of the sender (suppress the advertising of a route or routes) or the receiver (ignore the routing information from the sender). If routing information is not sent or accepted between ASs, then clients or servers in one AS cannot reach other hosts on the networks represented by that routing information in the other AS.

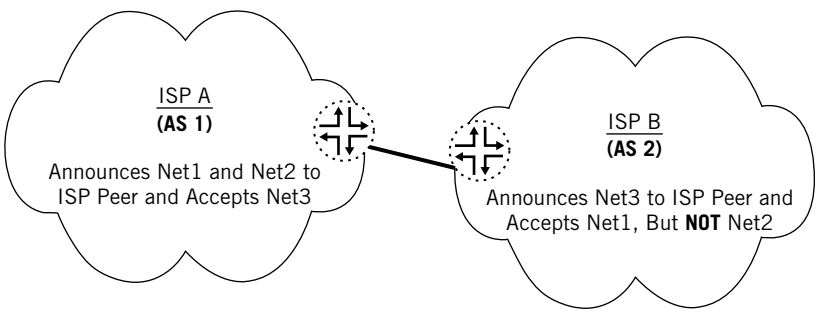


FIGURE 13.3

A simple example of a routing policy, showing how routes are announced (sent) and accepted (received). ISP A and ISP B are peers.

Economic considerations often play a role in routing policies as well. In the old days, there were always subsidies and grants available for continued support for the research and educational network. Now the ISP grid-net has ISPs with their own customers, and they can also be customers of other ISPs as well. Who pays whom, and how much?

PEERING

Telephony faced the same problem and solved it with a concept called *settlements*. This is where one telephone company bills the call originator and shares a portion of the billed amount with other telephone companies as an *access charge*. Access charges compensate the other telephone companies, long distance and local, that carry the call for the loss of the use of their own facilities (which could otherwise make money for the company directly) for the duration of the call. Now, in the IP world the source and destination share the cost of delivering packets, but the point is that telephony solved a similar issue and the terminology has been borrowed by the ISPs, which are often telephone companies as well.

The issue on the Internet becomes one of how one ISP should compensate another ISP for delivering packets that originate on the other ISP (if at all). The issue is complicated because the "call" is now a stream of packets, and an ISP might just be a transit ISP for packets that originate in one ISP's AS and are destined for a third ISP's AS.

ISP peers have tried three ways to translate this telephony "settlements" model to the Internet. First, there are very popular bilateral (between two sides) settlements based on the "call," usually defined as some aspect of IP packet flows. In this settlement arrangement, the first ISP, where the packet originates at a client, gets all of the revenue from the customer. However, the first ISP shares some of this money with the other ISP (where the server is located). Second, there is the idea of *sender keeps all* (SKA), where the flow of packets from client to server one way is supposedly balanced by the flow of packets from client to server the other way. So each ISP might as well just keep all of the revenue from their customers. Finally, there are *transit fees*, which are just settlements between one ISP and another, usually paid by a smaller ISP to a larger (because this traffic flow is seldom symmetrical).

Unfortunately, none of these methods have worked out well on the Internet. TCP/IP is not telephony and routers are not telephone switches. There are often many more than just two or three ISPs involved between client and server. There is no easy way to track and account for the packets that should constitute a "call," and even TCP sessions leave a lot to be desired because a simple Web page load might involve many rapid TCP connections between client and server. It is often hard to determine the "origin" because a packet and packets do not always follow stable network paths. Packets are often dropped, and it seems unfair to bill the originating ISP for resent packets replacing those that were not delivered by the billing ISP in the first place. Finally, dynamic routing might not be symmetric: So-called "hot potato" routing seeks to pass packets off to another ISP as soon as possible. So the path from client to server often passes through

different ISPs rather than keeping requests and replies all on one ISP's network. This common practice has real consequences for QoS enforcement.

These drawbacks of the telephony settlements model resulted in a movement to more simplistic arrangements among ISP *peers*, which usually means ISPs of roughly equal size. These are often called *peering arrangements* or just *peering*. There is no strict definition of what a peer is or is not, but it often describes two ISPs that are directly connected and have instituted some routing policies between them. In addition, there is nearly endless variation in settlement arrangements. These are just some of the broad categories. The key is that any traffic that a small network can offload onto a peer costs less than traffic that stays on internal transit links.

Economically, there is often also a sender-keeps-all arrangement in place, and no money changes hands. An ISP that is not a peer is just another *customer* of the ISP, and customers pay for services rendered. An interesting and common situation arises when three peers share a "transit peer" member. This situation is shown in Figure 13.4. There are typically no financial arrangements for peer ISPs providing transit services to the third peer, so peer ISPs will not provide transit to a third peer ISP (unless, of course, the third peer ISP is willing to pay and become a customer of one of the other ISPs).

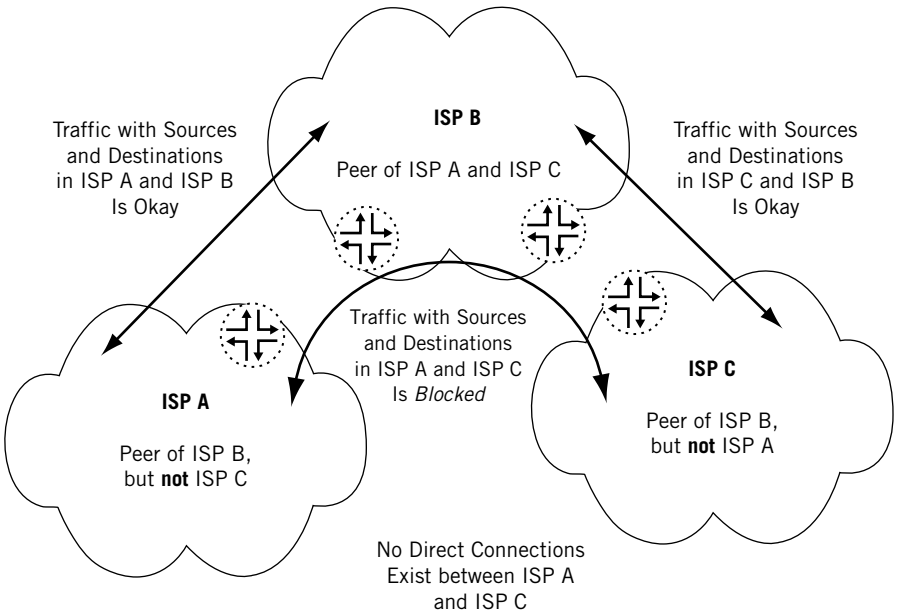


FIGURE 13.4

ISPs do not provide free transit services, and generally are either peers or customers of other ISPs. Unless "arrangements" are made, ISP B will routinely block transit traffic between ISP A and ISP C.

All three of these ISPs are "peers" in the sense that they are roughly equal in terms of network resources. They could all be small or regional or national ISPs. ISP A peers with ISP B and ISP B peers with ISP C, but ISP A has no peering arrangement (or direct link) with ISP C. So packet deliveries from hosts in ISP A to ISP B (and back) are allowed, as are packet deliveries from hosts in ISP C to and from ISP B. But ISP B has routing policies in place to prevent transit traffic from ISP A to and from ISP C through ISP B. How would that be of any benefit to ISP B? Unless ISP A and ISP C are willing to peer with each other, or ISP A or ISP C is willing to become a customer of ISP B, there will be no routing information sent to ISP A or ISP C to allow these ISPs to reach each other through ISP B. The routing policies enforced on the routers in ISP B will make sure of this, telling ISP A (for example) "you can't get to ISP C's hosts through me!"

The real world of the Internet, without a clearly defined hierarchy, complicates peering drastically. Peering is often a political issue. The politics of peering began in 1997, when a large ISP informed about 15 other ISPs that its current, easy-going peering arrangements would be terminated. New agreements for transit traffic were now required, the ISP said, and the former peers were effectively transformed into customers. As the trend spread among the larger ISPs, direct connections were favored over public peering points such as the IXPs.

This is one reason that Ace ISP and Best ISP in Figure 13.1 at the beginning of the chapter maintain multiple links between the four routers in the "quad" between their border routers. Suppose for a moment that routers P2 and P4 only have a single, direct link between them to connect the two ISPs. What would happen if that link were down? Well, at first glance, the situation doesn't seem very drastic. Both have links to "the Internet," which we know now is just a collection of other ISPs just like Ace and Best.

Can LAN1 reach LAN2 through "the Internet"? Maybe. It all depends on the arrangements between our two ISPs and the ISPs at the end of the "Internet" links. These ISPs might not deliver transit traffic between Ace and Best, and may even demand payment for these packets as "customers" of these other ISPs. The best thing for Ace and Best to do—if they don't have multiple backup links in their "quad"—is to make more peers of other ISPs.

PICKING A PEER

All larger ISPs often want to be peers, and peers of the biggest ISPs around. (For many, buying transit and becoming a customer of some other ISP is a much less expensive and effective way to get access to the global public Internet if being a transit provider is not your core business.) When it comes to peering, bigger is better, so a series of mergers and acquisitions (it is often claimed that there are no mergers, only acquisitions) among the ISPs took place as each ISP sought to become a "bigger peer" than another. This consolidation decreased the number of huge ISPs and also reduced the number of potential peers considerably.

Potential partners for peering arrangements are usually closely examined in several areas. ISPs being considered for potential peering must have high capacity backbones, be of roughly the same size, cover key areas, have a good network operations center (NOC), have about the same quality of service (QoS) in terms of delay and dropped packets, and (most importantly), exchange traffic roughly symmetrically. Nobody wants their routers, the workhorse of the ISP, to peer with an ISP that supplies 10,000 packets for every 1000 packets it accepts. Servers, especially Web sites, tend to generate much more traffic than they consume, so ISPs with "tight" networks with many server farms or Web hosting sites often have a hard time peering with anyone. On the other hand, ISPs with many casual, intermittent client users are courted by many peering suitors. Even if match is not quite the same in size, if the traffic flows are symmetrical, peering is always possible. The peering situation is often as shown in Figure 13.5. Keep in mind that other types of networks (such as cable TV operators and DSL providers) have different peering goals than presented here.

Without peering arrangements in place, ISPs rely on public exchange and peering points like the IXPs for connectivity. The trend is toward more private peering between pairs of peer ISPs.

Private peering can be accomplished by installing a WAN link between the AS border routers of the two ISPs. Alternatively, peering can be done at a collocation site where the two peers' routers basically sit side by side. Both types of private peering are common.

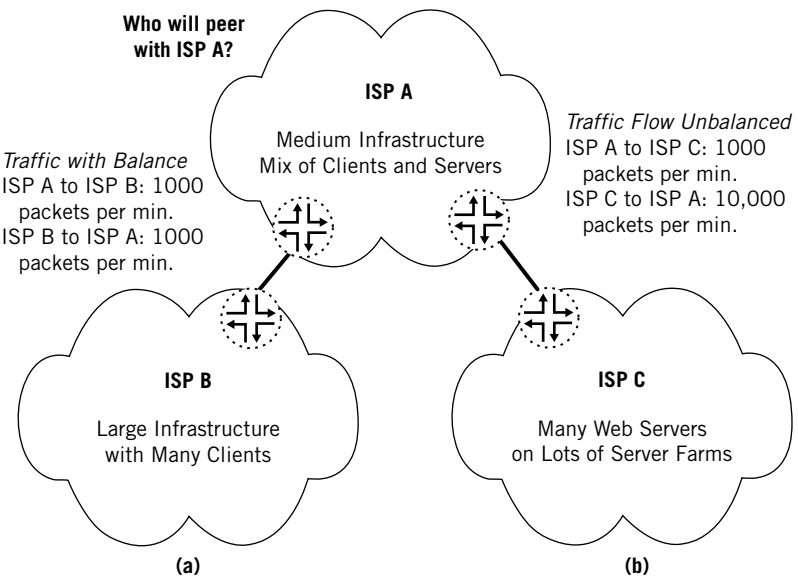


FIGURE 13.5

Good and bad peering candidates. Note that the goal is to balance the traffic flow as much as possible. Generally, the more servers the ISP maintains, the harder it is to peer. (a) ISP A will propose peering to ISP B; (b) ISP A will not want to peer with ISP C but will take them on as a customer.

The Internet today has more routes than there were *computers* attached to the Internet in early 1989. Routing policies are necessary whether the peering relationship is public or private (through an IXP or through a WAN link between border routers). Routing information simply cannot be easily distributed everywhere all at once. Even the routing protocols play a role. Some routing protocols send much more information than others, although protocols can be "tuned" by adjusting parameters and with routing policies.

Routing policies help interior gateway protocols (IGPs) such as OSPF and IS–IS distribute routing information within an AS more efficiently. The flow of routing information between routing domains must be controlled by routing policies to enforce the public or private peering arrangements in place between ISPs.

In the next chapter, we'll see how an IGP works within an AS or routing domain.

QUESTIONS FOR READERS

Figure 13.6 shows some of the concepts discussed in this chapter and can be used to help you answer the following questions.

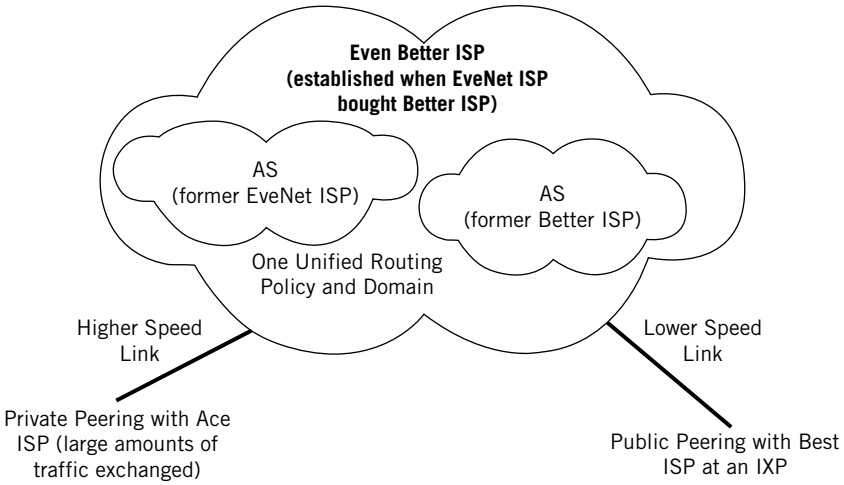


FIGURE 13.6

Even Better ISP, showing peering arrangements and routing domains.

1. What is an Internet autonomous system (AS)?
2. Why might a single ISP like Even Better ISP have more than one routing domain?
3. What is the purpose of a routing policy?
4. What does "ISP peering" mean?
5. What is the difference between public and private peering? Are both necessary?