CHAPTER

Network Link Technologies

3

What You Will Learn

In this chapter, you will learn more about the links used to connect the nodes of the Illustrated Network. We'll investigate the frame types used in various technologies and how they carry packets. We'll take a long look at Ethernet, and mention many other link types used primarily in private networks.

You will learn about SONET/SDH, DSL, and wireless technologies as well as Ethernet. All four link types are used on the Illustrated Network.

This chapter explores the physical and data link layer technologies used in the Illustrated Network. We investigate the methods used to link hosts and intermediate nodes together over shorter LAN distances and longer WAN distances to make a complete network.

For most of the rest of the book, we'll deal with packets and their contents. This is our only chance to take a detailed look at the frames employed on our network, and even peer inside them. Because the Illustrated Network is a real network, we'll emphasize the link types used on the network and take a more cursory look at link types that might be very important in the TCP/IP protocol suite, but are not used on our network. We'll look at Ethernet and the Synchronous Optical Network/Synchronous Digital Hierarchy (SONET/SDH) link technologies, and explore the variations on the access theme that digital subscriber line (DSL) and wireless technologies represent.

We'll look at public network services like frame relay and Asynchronous Transfer Mode (ATM) in a later chapter. In this book, the term *private network* is used to characterize network links that are owned or directly leased by the user organization, while a *public network* is characterized by shared user access to facilities controlled by a service provider. The question of *Who owns the intermediate nodes?* is often used as a rough distinguisher between private and public network elements.

Because of the way the TCP/IP protocol stack is specified, as seen in Chapter 1, we won't talk much about physical layer elements such as modems, network interface cards (NICs), and connectors. As important as these aspects of networking are, they

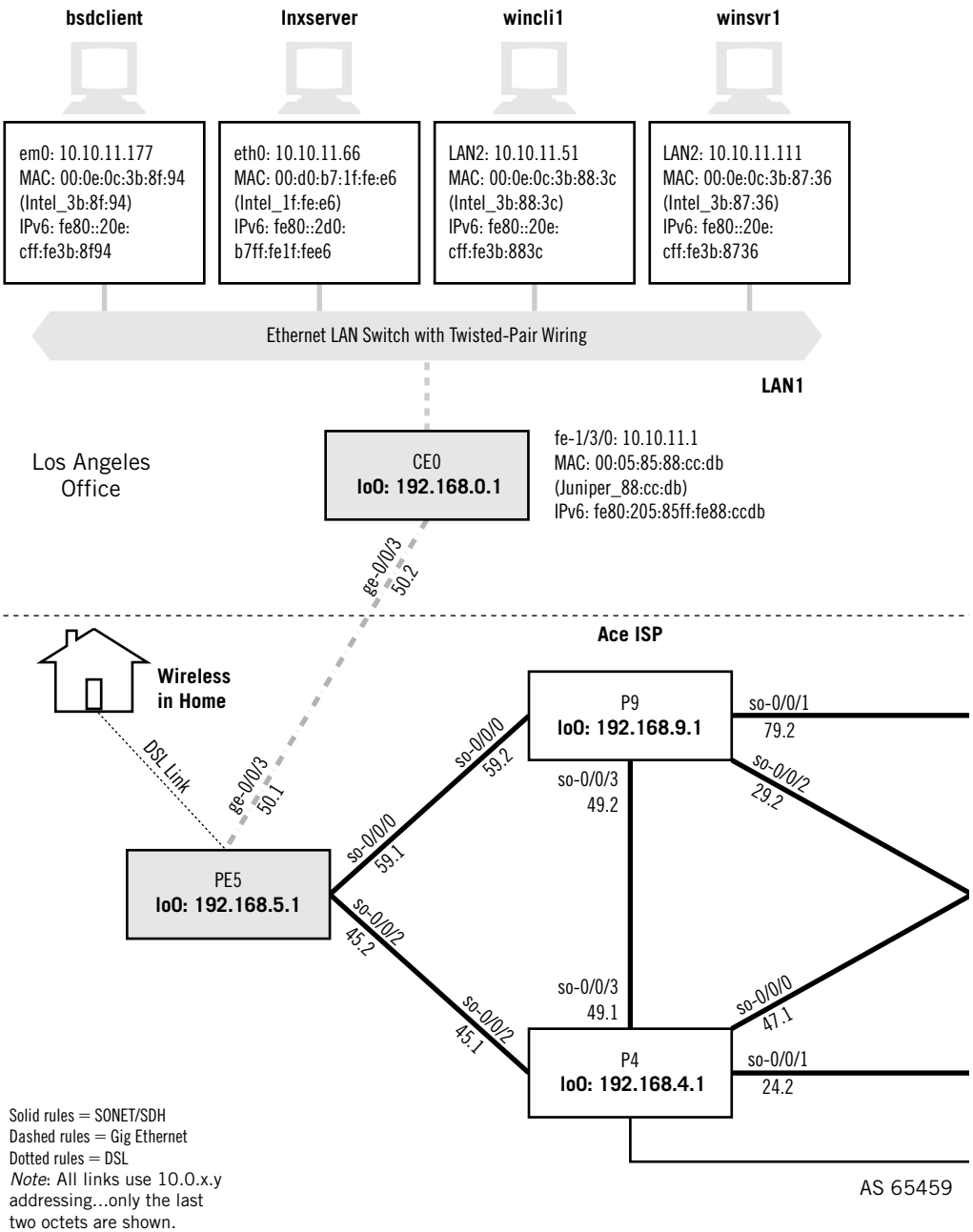
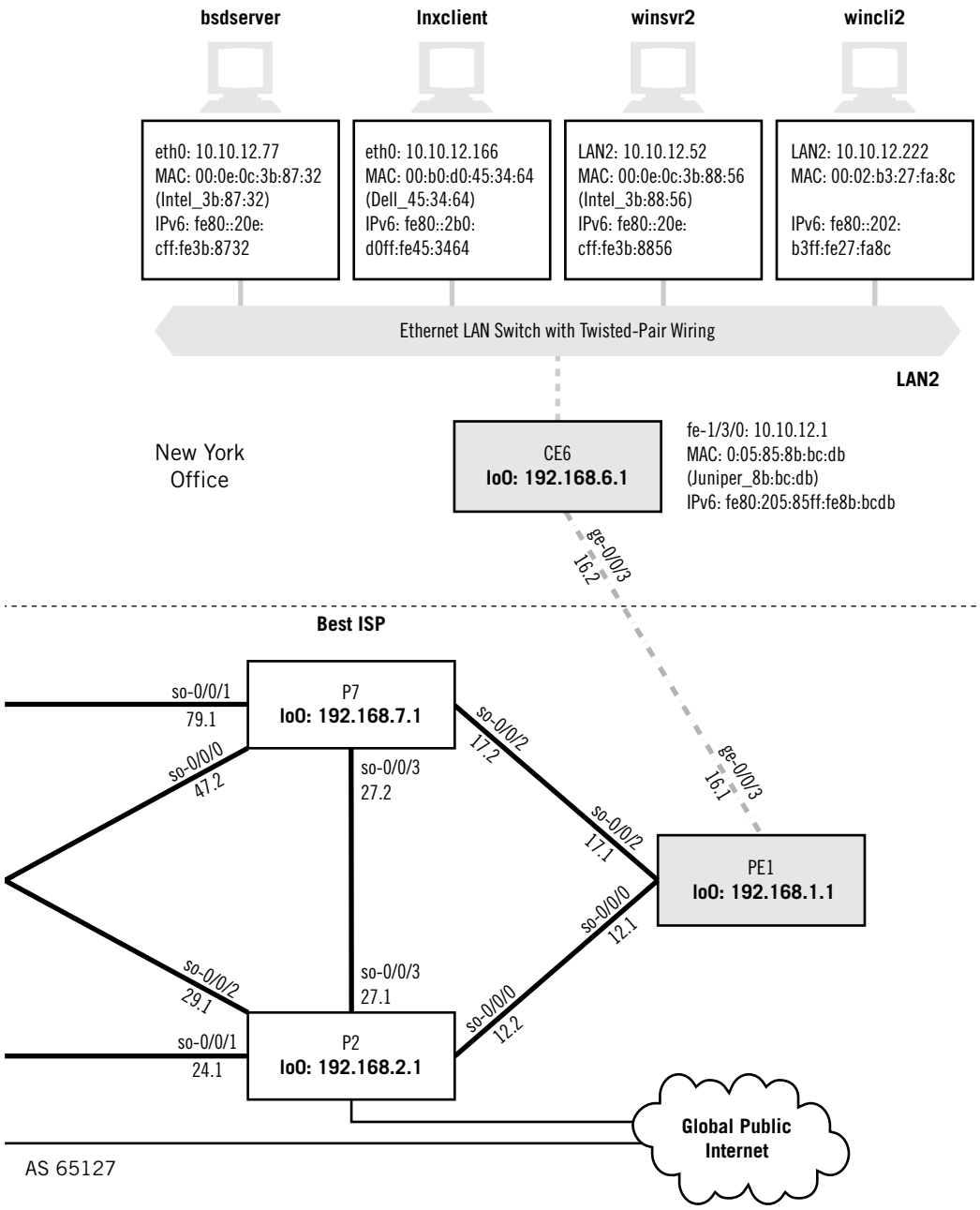


FIGURE 3.1

Connections used on the Illustrated Network. SONET/SDH links are indicated by heavy lines, Ethernet types by dashed lines, and DSL is shown as a dotted line. The home wireless network is not given a distinctive representation.


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

have little to do directly with how TCP/IP protocols or the Internet operates. For example, a full exploration of all the connector types used with fiber-optic cable would take many pages, and yet add little to anyone's understanding of TCP/IP or the Internet. Instead, we will concentrate on the structure of the frames sent on these link types, which *are* often important to TCP/IP, and present some operational details as well.

ILLUSTRATED NETWORK CONNECTIONS

We will start by using Ethereal (Wireshark), the network protocol analyzer introduced in the last chapter, to investigate the connections between systems on the Illustrated Network. It runs on a variety of platforms, including all three used in the Illustrated Network: FreeBSD Unix, Linux, and Windows XP. Ethereal can display real-time packet interpretations and, if desired, also save traffic to files (with a variety of formats) for later analysis or transfer to another system. Ethereal is most helpful when examining all types of Ethernet links. The Ethernet links are shown as dashed lines in Figure 3.1.

The service provider networks' SONET links are shown as heavy solid lines, and the DSL link to the home office is shown as a dotted line. The wireless network inside the home is not given a distinctive representation in the figure. Note that ISPs today typically employ more variety in WAN link types.

Displaying Ethernet Traffic

On the Illustrated Network, all of the clients and servers with detailed information listed are attached to LANs. Let's start our exploration of the links used on the Illustrated Network by using Ethereal both ways to see what kind of *frames* are used on these LANs.

Here is a capture of a small frame to show what the output looks like using tethereal, the text-based version of Ethereal. The example uses the verbose mode (`-V`) to force tethereal to display all packet and frame details. The example shows, highlighted in bold, that Ethernet II frames are used on LAN1.

```
  [root@lnxserver admin]# /usr/sbin/tethereal –V
    Frame 2 (60 bytes on wire, 60 bytes captured)
    Arrival Time: Mar 25, 2008 12:14:36.383610000
    Time delta from previous packet: 0.000443000 seconds
    Time relative to first packet: 0.000591000 seconds
    Frame Number: 2
    Packet Length: 60 bytes
    Capture Length: 60 bytes
Ethernet II, Src: 00:05:85:88:cc:db, Dst: 00:d0:b7:1f:fe:e6
    Destination: 00:d0:b7:1f:fe:e6 (Intel_1f:fe:e6)
    Source: 00:05:85:88:cc:db (Juniper__88:cc:db)
    Type: ARP (0x0806)
    Trailer: 000000000000000000000000000000000...
```

```
Address Resolution Protocol (reply)
    Hardware type: Ethernet (0x0001)
    Protocol type: IP (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (0x0002)
    Sender MAC address: 00:05:85:88:cc:db (Juniper__88:cc:db)
    Sender IP address: 10.10.11.1 (10.10.11.1)
    Target MAC address: 00:d0:b7:1f:fe:e6 (Intel_1f:fe:e6)
    Target IP address: 10.10.11.66 (10.10.11.66)
```

Many details of the packet and frame structure and content will be discussed in later chapters. However, we can see that the source and destination MAC addresses are present in the frame. The source address is `00:05:85:88:cc:db` (the router), and the destination (the Linux server) is `00:d0:b7:1f:fe:e6`. Ethereal even knows which organizations have been assigned the first 24 bits of the 48-bit MAC address (Intel and Juniper Networks). We'll say more about MAC addresses later in this chapter.

Figure 3.2 shows the same packet, and the same information, but in graphical format. Only a small section of the entire window is included. Note how the presence of Ethernet II frames is indicated, parsed on the second line in the middle pane of the window.

Why use text-based output when a graphical version is available? The graphical output shows the raw frame in hex, something the text-based version does not do, and the interpretation of the frame's fields is more concise.

However, the graphical output is not always clearer. In most cases, the graphical representation can be more cluttered, especially when groups of packets are involved. The graphical output only parses one packet at a time on the screen, while a whole string of packets can be parsed with tethereal (but printouts of graphical information can be formatted like tethereal).

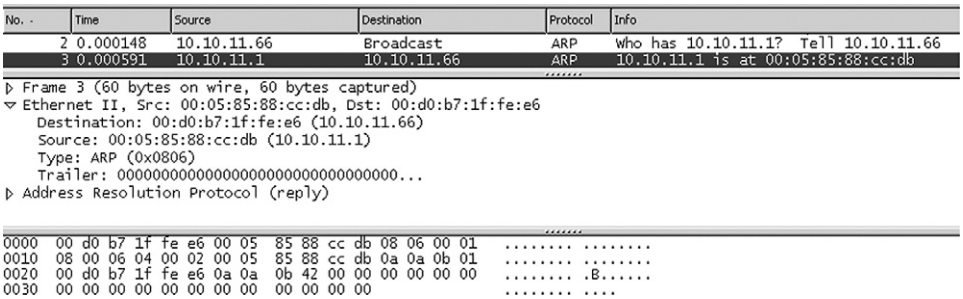


FIGURE 3.2

Graphical interface for Ethereal. There are three main panes. Top to bottom: (1) a digest of the packets header and information, (2) parsed details about frame and packet contents, and (3) the raw frame captured in hexadecimal notation and interpreted in ASCII.

In addition, many network administrators of Internet servers do not install or use a graphical interface, and perform their tasks from a command prompt. If you're not sitting in front of the device, it's more expedient to run the non-GUI version. Tethereal is the only realistic option in these cases. We will use both types of Ethereal in the examples in this book.

In our example network, what about LAN2? Is it also using Ethernet II frames? Let's capture some packets on `bsdserver` to find out.

```
bsdserver# tethereal –V
Capturing on em0
Frame 1 (98 bytes on wire, 98 bytes captured)
    Arrival Time: Mar 25, 2008 13:05:00.263240000
    Time delta from previous packet: 0.000000000 seconds
    Time since reference or first frame: 0.000000000 seconds
    Frame Number: 1
    Packet Length: 98 bytes
    Capture Length: 98 bytes
Ethernet II, Src: 00:0e:0c:3b:87:32, Dst: 00:05:85:8b:bc:db
    Destination: 00:05:85:8b:bc:db (Juniper__8b:bc:db)
    Source: 00:0e:0c:3b:87:32 (Intel_3b:87:32)
    Type: IP (0x0800)
Internet Protocol, Src Addr: 10.10.12.77 (10.10.12.77), Dst Addr: 10.10.12.1
(10.10.12.1)
    Version: 4
    Header length: 20 bytes
    ....
```

Yes, an Ethernet II frame is in use here as well. Even though we're running Ethereal (tethereal) on a different operating system (FreeBSD) instead of on Linux, the output is nearly identical (the differences are due to a slightly different version of Ethereal on the servers). However, LANs are not the only type of connections used on the Illustrated Network.

Displaying SONET Links

What about link types other than Ethernet? ISPs in the United States often use SONET fiber links between routers separated by long distance. In most other parts of the world, SDH is used. SONET was defined initially in the United States, and the specification was adapted, with some changes, for international use by the ITU-T as SDH.

The Illustrated Network uses SONET, not SDH. There are small but important differences between SONET and SDH, but this book will only reference SONET. Line monitoring equipment that allows you to look directly at SONET/SDH frames is expensive and exotic, and not available to most network administrators. So we'll take a different approach: We'll show you the information that's available on a router with a SONET interface. This will show the considerable bandwidth available even in the slowest of SONET links, which runs at 155 Mbps and is the same as the basic SDH speed.

SONET and SDH

The SONET fiber-optic link standard was developed in the United States and is mainly used in places that follow the digital telephony system used in the United States, such as Canada and the Philippines. SDH, on the other hand, is used in places that follow the international standards developed for the digital telephony system in the rest of the world. SDH *must* be used for all international links, even those that link to SONET networks in the United States.

The differences between SONET and SDH transmission frame structures, nomenclature, alarms, and other details are relatively minor. In most cases, equipment can handle SONET/SDH with equal facility.

We can log in to router `CE0` and monitor a SONET interface for a minute or so and see what's going on.

Routers and Users

Usually, network administrators don't let ordinary users casually log in to routers, even edge routers, and poke around. Even if they were allowed to, the ISP's core routers would still remain off limits. But this is *our* network, and we can do as we please, wherever we please.

```
Admin>ssh ce0
adminCE6's password: *********
--- JUNOS 8.4R1.3 built 2007-08-06 06:58:15 UTC
admin@ce0> monitor interface so-0/0/1
R2                                  Seconds: 59             Time: 13:36:05
                                                            Delay: 2/0/3

Interface: so-0/0/1, Enabled, Link is Up
Encapsulation: PPP, Keepalives, Speed: OC3
Traffic statistics:                                         Current delta
  Input bytes:              166207481 (576 bps)                 [2498]
  Output bytes:             171979817 (48 bps)                  [2713]
  Input packets:            2868777 (0 pps)                       [39]
  Output packets:           2869671 (0 pps)                       [39]
Encapsulation statistics:
  Input keepalives:         477607                                [6]
  Output keepalives:        477717                                [7]
  LCP state: Opened
Error statistics:
  Input errors:             0                                     [0]
  Input drops:              0                                     [0]
  Input framing errors:     0                                     [0]
```

```
  Input runts:              0                                        [0]
  Input giants:             0                                        [0]
  Policed discards:         0                                        [0]
  L3 incompletes:           0                                        [0]
  L2 channel errors:        0                                        [0]
  L2 mismatch timeouts:     0                                        [0]
  Carrier transitions:      1                                        [0]
  Output errors:            0                                        [0]
  Output drops:             0                                        [0]
  Aged packets:             0                                        [0]
Active alarms : None
Active defects: None
SONET error counts/seconds:
  LOS count                 1                                        [0]
  LOF count                 1                                        [0]
  SEF count                 3                                        [0]
  ES-S                      1                                        [0]
  SES-S                     1                                        [0]
SONET statistics:
  BIP-B1                    0                                        [0]
  BIP-B2                    0                                        [0]
  REI-L                     0  BIP-B3              Z                 [0]
```

Not much is happening yet on our network in terms of traffic, but the output is still informative. The first column shows cumulative values and the second column shows the change since the last monitor "snapshot" on the link. "Live" traffic during these 59 seconds, in this case mostly a series of *keepalive* packets, is shown in parentheses, both in bytes per second and in packets per second (the example rounds the 39 packets in 59 seconds, or 0.66 packets per second, down to 0 packets per second). The frames carried on the link, listed as encapsulation, belong to a protocol called Point-to-Point Protocol (PPP). Six PPP keepalives have been sent in the 59-second window, and seven have been received (they are exchanged every 10 seconds), adding to the total of more than 477,000 since the link was initialized. The cumulative errors also occurred as the link was initializing itself, and it is reassuring that there are no new errors.

Displaying DSL Links

The Illustrated Network also has a broadband DSL link from an ISP that is used to allow a home office to attach to the router network. This link is shown in red in Figure 3.1. If the permissions are set up correctly, the home user will be able to access network resources on LAN1 and LAN2. DSL links are much faster than ordinary dial-up lines and are always available, just like a leased access line. The DSL link terminates at home in a DSL router (more properly, a *residential gateway*), and the distribution of information to devices in the home can be by wired or wireless LAN.

On the network end of the DSL link, the link terminates at a DSL access multiplexer (DSLAM), typically using IP or ATM technology.

At the user end of the DSL link on the Illustrated Network, the office in the home uses both a wired and a wireless network. This is a common arrangement today: People with laptops can wander, but desktop PCs usually stay put. The wireless network encapsulates packets and sends them to a special device in the home (a wireless access point, often built into a DSL router).

What kind of frames does the DSL link use? That's hard to determine, because the DSL modem is upstream of the DSL router in most cases (sometimes on the side of the house, sometimes closer to the service provider). The wired LAN between DSL router and computer uses the same type of Ethernet frames we saw on LAN1 and LAN2. On a wired LAN, Ethereal will always capture Ethernet II frames, as shown in Figure 3.3.

What can we learn about DSL itself? Well, we can access the DSL router using a Web browser and see what kinds of information are available. Figure 3.4 shows the basic setup screen of the Linksys DSL router (although it's really not doing any real routing, just functioning as a simple gateway between ISP and home LAN).

Because this is a working LAN, I've restored the default names and addresses for this example. The router itself is `WRT54G` (a product designation), and the ISP does not expect only one host to use the DSL link, so no host or domain name is required. We'll talk about the maximum transmission unit (MTU) size later in this chapter. This is set automatically on the link.

The DSL router itself uses IPv4 address 192.168.1.1. We'll talk about what the subnet mask does in Chapter 4. The router hands out IP addresses as needed to devices on the home network, starting with 192.168.1.100, and it uses the Dynamic Host Configuration Protocol (DHCP) to do this. We'll talk about DHCP in Chapter 18.

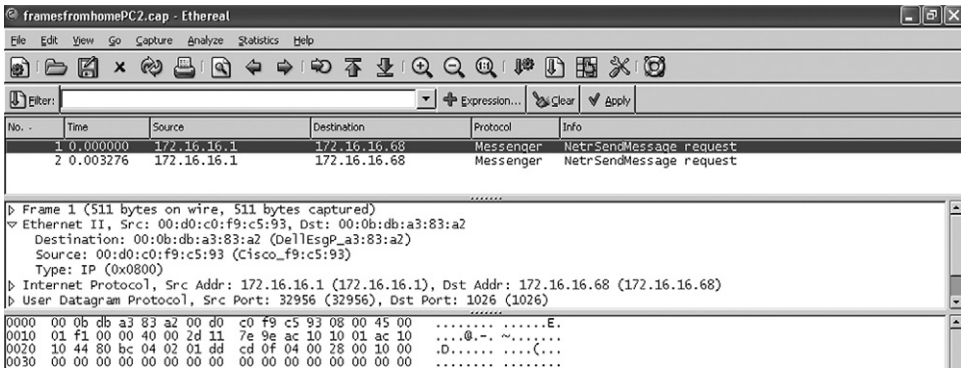


FIGURE 3.3

Ethernet frames on a wired LAN at the end of a DSL link. Capturing raw DSL frame "on the wire" is not frequently done, and is difficult without very expensive and specialized equipment.

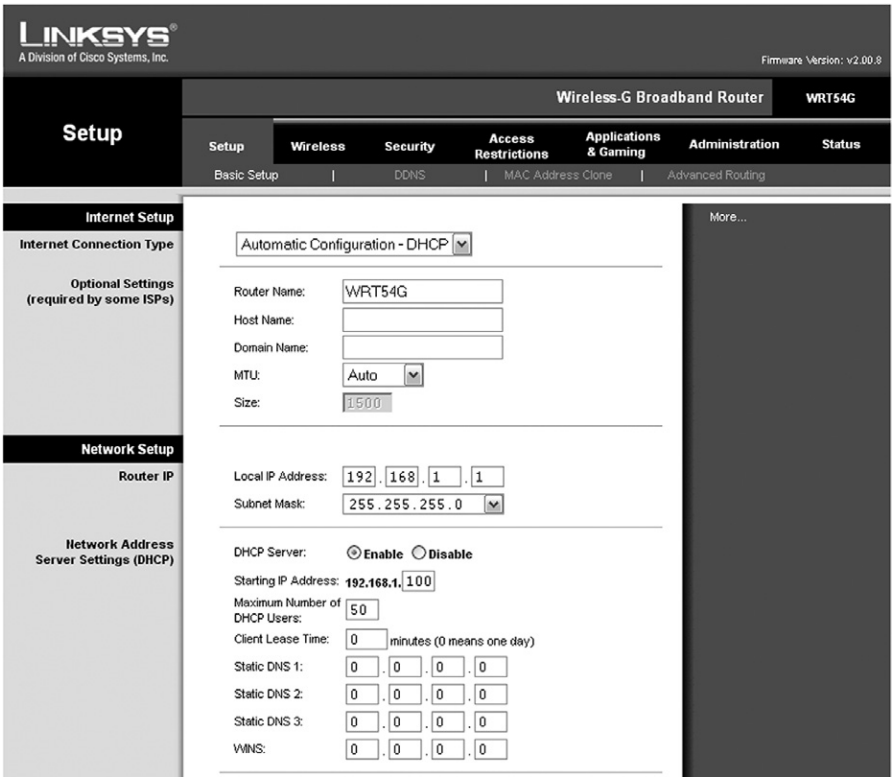


FIGURE 3.4

Basic setup screen for a DSL link. We'll talk about all of these configuration parameters and protocols, such as subnet masks and DHCP, in later chapters.

What kinds of statistics are available on the DSL router? Not much on this model. There are simple incoming and outgoing logs, but these capture only the most basic information about addresses and ports. A small section of the outgoing log is shown in Table 3.1.

These are all Web browser entries that were run with names, not IP addresses (Yahoo is one of them). The table lists the addresses because the residential gateway does not bother to look the names up. However, instead of presenting the port numbers, the log interprets them as a service name (www is port 80 on most servers).

We'll take a more detailed look at DSL later in this chapter. Now, let's take a look at the fourth and last link type used on the Illustrated Network: the four available wireless links used to hook a laptop and printer up to the home office DSL router.

The wireless implementation is a fairly straightforward bridging exercise. A single wireless interface is bridged in software with the Ethernets in the box. The wireless network is a single broadcast/collision domain.

Table 3.1 Outgoing Log Table from DSL Router		
LAN IP	**Destination URL/IP**	**Service/Port Number**
192.168.1.101	202.43.195.13	www
192.168.1.101	64.86.142.99	www
192.168.1.101	202.43.195.52	www
192.168.1.101	64.86.142.120	www

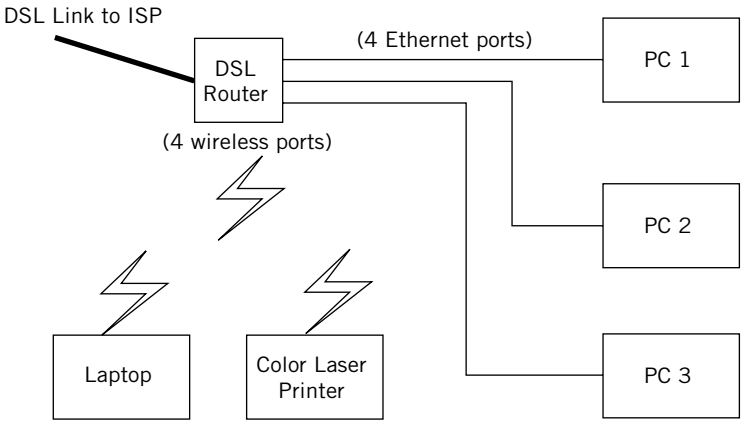


FIGURE 3.5

The home office network for the Illustrated Network. Devices must have either Ethernet ports or wireless interfaces (some have both). Not all printers are network-capable or wireless.

Displaying Wireless Links

The physical arrangement of the home office equipment used on the Illustrated Network is shown in Figure 3.5. In addition to the three wired PCs (used for various equipment configurations), there are two wireless links. One is used by the laptop for mobility, and the other is used to share a color laser printer. The DSL router does not have "ports" in the same sense as wired network devices, but it only supports up to four wireless devices.

The wireless link from the laptop to the DSL router, which uses something called IEEE 802.11g (sometimes called Wireless-G), is a distinct Layer 2 network technology and should not use Ethernet II frames. Let's make sure.

Capturing traffic at the wireless frame level requires special software and special drivers for the wireless network adapter card. The examples in this chapter use information from a wireless packet sniffer called Airopeek NX from Wildpackets.

A sample capture of a data packet and frame from a wireless link is shown in Figure 3.6.

Wireless LANs based on IEEE 802.11 use a distinct frame structure and a complex data link layer protocol. We'll talk about 802.11 shortly, but for now we should just note that the Illustrated Network uses USB-attached wireless NICs, and few wireless sniffers support these types of adapters.

The frame addressing and encapsulation on wireless LANs is much more complicated than Ethernet. Note that the 802.11 MAC frame has *three* distinct MAC addresses, labeled Destination, BSSID, and Source. The wireless LAN has to keep track of source, destination, and wireless access point (Base Station System ID, or BSSID) addresses. Also note that these are not really Ethernet II frames. The frames on the wireless link are structured according to the IEEE 802.2 LLC header. These have "SNAP SAP," indicated by *0xAA*, in the frame, in contrast to Ethernet II frames, which are indicated by *0x01*.

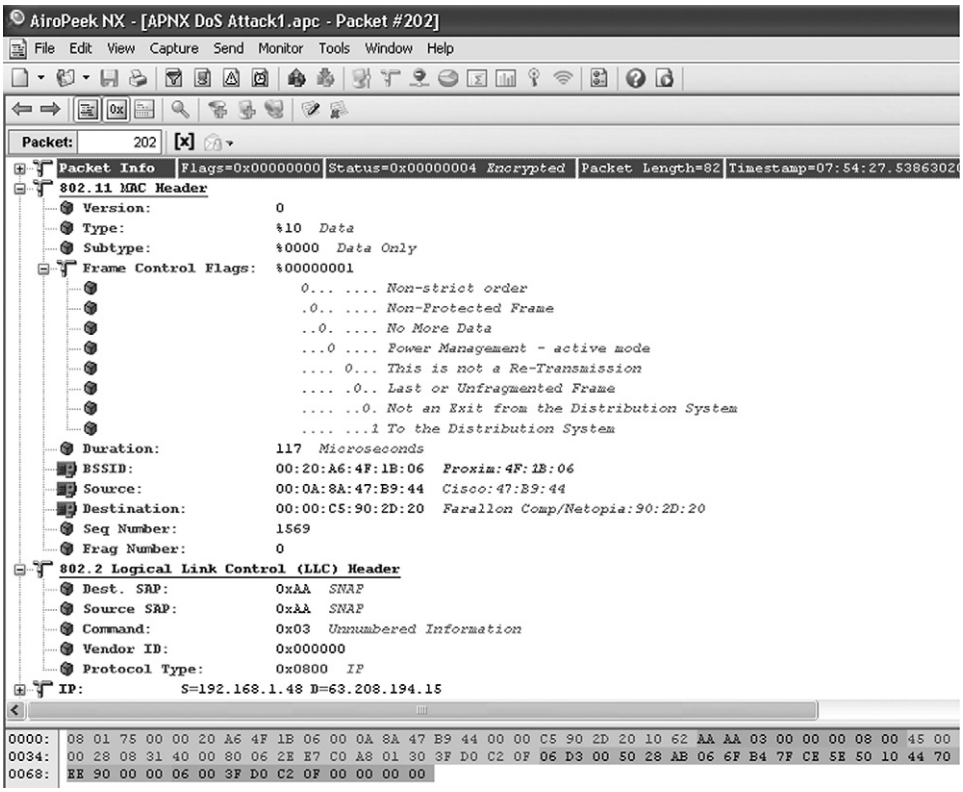


FIGURE 3.6

Data frame and packet on a wireless link. Note that the IEEE 802.11 MAC header is different from the Ethernet in many ways and uses the IEEE 802.2 LLC inside.

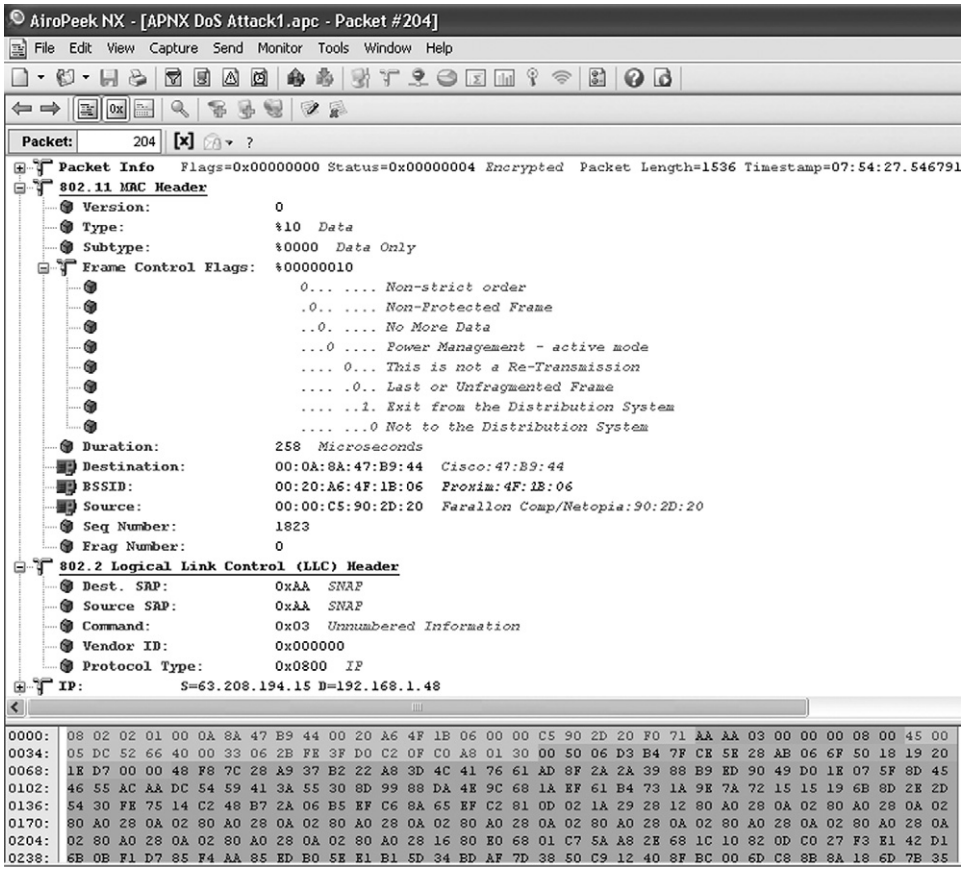


FIGURE 3.7

The next data frame in the sequence, showing how the contents of the address fields shift based on direction and type of wireless frame.

The address fields in 802.11 also "shift" their meaning, as shown in Figure 3.7. The fields are now BSSID, Source, and Destination. This is another capture from Airopeek NX, showing the next data frame sent in the captured exchange. The address fields have different meanings based on whether they are sent *to* the wireless router or are received *from* the wireless router.

Frames and the Link Layer

In summary, we have seen that the connections on the Illustrated Network consist of several types of links. There are wired Ethernet LANs and Gigabit Ethernet links, SONET links and DSL links, and even a wired LAN in the home network. We've looked at some of the frame types that carry information back and forth on the network connections.

There are many more types of frames that can carry IP packets between systems at the data link layer. The rest of this chapter will explore the data link layer in a little more depth.

RFCs and Physical Layers

Internet RFCs usually describe not how the physical (or data link) layers in a TCP/IP network should function, but how to place packets inside data link frames and get them out again at the other end of the link to the adjacent system. It is always good to remember that frames flow between *adjacent* (directly connected or reachable) systems on a network.

THE DATA LINK LAYER

Putting the world of connectors, modems, and electrical digital signal levels of the physical layer aside, let's go right to the data link layer of the TCP/IP protocol stack. It's not that these things are not important to networking; it's just that these things have nothing directly to do with TCP/IP.

The data link layer of TCP/IP takes an IP packet at the source and puts it inside whichever frame structure is used between systems (e.g., an Ethernet frame). The data link layer then passes the frame to the physical layer, which sends the frame as a series of bits over the link itself. At the receiver, the physical and data link layers recover the frame from the arriving sequence of bits and extract the packet. The packet is then passed to the receiving network (IP) layer.

Interfaces for IP packets have been defined for all of the following network types, for both LAN and WAN:

Ethernet—Originally from Digital Equipment Corporation, Intel, and Xerox (sometimes called DIX Ethernet).

IEEE (Institute of Electrical and Electronics Engineers) 802.3—Ethernet-based LANs, including all its variations, such as Gigabit Ethernet.

Synchronous Optical Network, Synchronous Digital Hierarchy (SONET/SDH)—A high-speed, optical WAN transport.

IEEE 802.11 Wireless LANs—Includes any technology, such as WiFi, based on variations of this.

Token Ring—LANs from IBM, the same as IEEE 802.5.

Point-to-Point Protocol (PPP)—This protocol is from the IP developers themselves, and is not limited to carrying IP packets.

X.25—An international standard, public, switched, connection-oriented network protocol.

Frame Relay—An international standard, public, switched, connection-oriented network protocol based on X.25.

Asynchronous Transfer Mode (ATM)—An international standard, public, switched, connection-oriented network protocol based on cells instead of frames.

Fiber Distributed Data Interface (FDDI)—A LAN-like network ring running at 100 Mbps.

Switched Multimegabit Data Services (SMDS)—A high-speed, connectionless, LAN-like, public network service.

Integrated Services Digital Network (ISDN)—A public switched network similar to X.25.

Digital Subscriber Line (DSL)—Based on some older Integrated Services Digital Network (ISDN)–related technologies and used for high-speed Internet access.

Serial Line Interface Protocol (SLIP) and Compressed SLIP (CSLI)—An older way of sending IP packets over a dial-up, asynchronous modem arrangement (also from the IP developers).

Cable Modems (CMODEMs)—A method of sending IP packets over a cable TV infrastructure.

IPoFW IP over Firewire (IEEE 1394)—A popular PC interface for peripheral devices. There are other interfaces as well, such as ARCnet and IEEE 802.4 LANs, but the point is that TCP/IP is not tied to any specific type of network at the lower layers. The TCP/IP protocol stack is very flexible and encompassing, much more so than almost anything else that could be used on a global network.

In the future, this list will get even longer as newer transports for IP packets are standardized and older ones remain (in spite of diminishing interest, standards like these tend to stay in place because no one cares enough to move them to "historic" RFCs). Some of the newer network types that might find their way onto many networks in the future follow:

VDSL—VDSL is a "very-high-speed" form of DSL that uses fiber feeders to reach less than a mile from the home (often called fiber to the neighborhood, or FTTN). Most VDSL service offerings deliver television, telephone, and high-speed Internet access over a unified residential cabling system through a special residential gateway box. On the Illustrated Network, the home office DSL link is actually VDSL, but this service is not as widely available as other forms of DSL.

GE-PONS—These Gigabit Ethernet Passive Optical Network (GE-PONS) nodes are part of a global push toward Fiber to the Home (FTTH), an approach that has been—somewhat ironically—slowed by the popularity of DSL over copper

wires. Based on IEEE 802.3ah standards, this technology can support gigabit speeds in both directions and might take advantage of the popularity of voice over IP (VoIP).

BPL—In some places, high-speed Internet access is provided by the electric utility as part of broadband power line (BPL) technology. Delivered over the same socket as power, BPL services might form a nice adjunct to wireless services, which are hard to cost-justify in sparsely populated areas and over rough terrain.

The advantage of not tying the network layer to any specific type of links at the lower layers is flexibility (IP can run on anything). A new type of network interface can be added without great effort. Also, it makes linking these various network types into an internetwork that much easier.

All TCP/IP implementations must be able to support at least one of the defined interface types. Most implementations of TCP/IP will do fine today with only a handful of interface types, and, as we have seen, Ethernet frames are perhaps the most common of all data-link frame formats for IP packets, especially at the endpoints of the network.

The rest of this chapter provides a closer look at the four link types used on the Illustrated Network, as well as PPP, the major IEFT data-link protocol that we saw used on SONET. The coverage is not intended to be exhaustive, but will be enough to introduce the technologies.

Although all four link types are covered, the coverage is not equal. There is much more information about Ethernet and wireless than SONET or DSL. The main reason is that expensive and exotic line monitoring equipment is needed in order to burrow deep enough in the lower layers of the protocol stacks used in SONET and DSL to show the transmission frames. End users, and even many smaller ISPs, do just fine diagnosing problems on SONET and DSL links with basic Ethernet and IP monitoring tools. Then again, point-to-point links are a bit easier to diagnose than shared media networks. (Is the line protocol up in both directions? Is the distance okay? Is the bit error rate acceptable? Okay, it's not the link layer . . .)

SONET and DSL are distinguished from Ethernet and wireless LANs with regard to addressing. SONET and DSL are point-to-point technologies and use much simpler link-level addressing schemes than LAN technologies. There are only two ends in a point-to-point connection, and you always know which end you are. Anything you send is intended for the other end of the link, and anything you receive comes from the other end as well.

THE EVOLUTION OF ETHERNET

The original Ethernet was developed at the Xerox Palo Alto Research Center (PARC) in the mid-1970s to link the various mainframes and minicomputers that Xerox used in their office park campus environment of close-proximity buildings. The use of WAN

protocols to link all of these buildings did not appeal to Xerox for two reasons. First, an efficient WAN infrastructure would have demanded a mesh of leased telephone lines, which would have been enormously expensive given the number of computers. Second, leased telephone lines did not have the bandwidth (usually these carried only up to 9600 bps, and at most 56 Kbps, in the late 1970s) needed to link the computers.

Their solution was to invent the *local* area network, the LAN. However, Xerox was not interested in actually building hardware and chipsets for their new invention, which was named *Ethernet.* Instead, Bob Metcalf, the Ethernet inventor, left Xerox and recruited two other companies, one to make chipsets for Ethernet and the other to make the hardware components to employ these chipsets. The two companies were chip-maker Intel and computer-maker Digital Equipment Corporation (DEC). Ethernet v1.0 was rolled out in 1980, followed by Ethernet v2.0 in 1982, which fixed some annoying problems in v1.0. This is why, in our examples, Ethereal keeps showing that IP packets are inside Ethernet II frames when they leave and arrive at hosts.

DIX Ethernet, the proprietary version, ran over a single, thick coaxial cable "bus" that snaked through a building or campus. Transmitting and receiving devices (transceivers) were physically clamped to the coaxial cable (with "vampire taps") at predetermined intervals. Transceivers usually had multiple ports for attaching the transceiver cables that led to the actual PC or minicomputer linked by the Ethernet LAN. The whole LAN ran at an aggregate speed of 10 Mbps, an unbelievable rate for the time. But Ethernet had to be fast, because up to 1024 computers could share this single coaxial cable bus to communicate using a media access method known as carrier-sense multiple access with collision detection (CSMA/CD). DIX Ethernet had to be distinguished from all other forms of Ethernet, which were standardized by the IEEE starting in 1984.

The IEEE first standardized a slightly different arrangement for 10-Mbps CSMA/CD LANs (IEEE 802.3) in 1984. Why the IEEE felt compelled to change the proprietary Ethernet technology during the standardization process is somewhat of a puzzle. Some said the IEEE always did this, but around the same time the IEEE essentially rubberstamped IBM's proprietary Token Ring LAN specification as IEEE 802.5. The changes to the hardware of DIX Ethernet were minor. There was no v1.0 support at all (i.e., all IEEE 802.3 LANs were DIX Ethernet v2.0) and the terminology was changed slightly. The DIX transceiver became the IEEE 802.3 "media attachment unit" (MAU), and so on.

However, throughout the 1980s and into the 1990s, as research into network capabilities matured, the IEEE added a number of variations to the original IEEE 802.3 CSMA/CD hardware specification. The original specification became 10Base5 (which meant 10-Mbps transport, using baseband signaling, with a 500-meter LAN segment). This was joined by a number of other variants designed to make LAN implementation more flexible and—especially—less expensive. New IEEE 802.3 variations included 10Base2 (with 200-meter segments over thin coaxial cable), the wildly popular 10BaseT (with hubs instead of segments linked to PCs by up to 100 meters of unshielded twisted-pair copper wire), and versions that ran on fiber-optic cable. Eventually, all of these technologies except those on coaxial cable went first to 100 Mbps (100BaseT), then 1000 Mbps (Gigabit Ethernet), which run over twisted pair for short spans and can use fiber for increasingly long hauls, now in the SONET/SDH ranges.

Today, IEEE 802.3ae 10G-base-er (extended range) LAN physical layer links can span 40 km. Another, "zr," is not standardized, but can stretch the span to 80 km. And interestingly, 10-Gbps Ethernet is back on coaxial cable as "10Gbps cx4."

Ethernet II and IEEE 802.3 Frames

Today, of course, the term "Ethernet" essentially means the same as "IEEE 802.3 LAN." In addition to changing the hardware component names and creating IEEE 802.3 10BaseT, the IEEE also changed the Ethernet *frame* structure for reasons that remain obscure. It was this development that had the most important implication for those implementing the TCP/IP protocol stack on top of Ethernet LANs.

The DIX Ethernet II frame structure was extremely simple. There were fields in the frame header for the source and destination MAC (the upper part of the data link layer, used on LANs) address, a type field to define content (packet) structure, a variable-length data field, and an error-detecting trailer. The source and destination addresses were required for the mutually adjacent systems on a LAN (a point-to-point-oriented data link layer with just a "destination" address would not work on LANs: Who sent this frame?). The type field was required so the recipient software would know the structure of the data inside the frame. That is, the destination NIC could examine the type field and determine if the frame contents were an IP packet, some other type of packet, a control frame, or almost anything else. The destination NIC card could then pass the frame contents to the proper software module (the network layer) for further processing on the frame data contents. The type field value for IP packets was set as `0x0800`, the bit string `00001000 00000000`.

However, the IEEE 802 committee changed the simple DIX Ethernet II frame structure to produce the IEEE 802.3 CSMA/CD frame structure. Gone was the DIX Ethernet II type (often called "Ethertype") field, and in its place was a same-sized `length` field. This action somewhat puzzled observers of LAN technology. DIX Ethernet II frames worked just fine without an explicit length field. The total frame length was determined by the positions of the starting and ending frame delimiters. The data were always after the header and before the trailer. Simple enough for software to figure out.

Now, with IEEE 802.3 it was even easier to figure out the length of a received frame (the software just had to look at the `length` field value). However, it was now impossible for the receiving software to figure out just what the structure of the frame data was by looking only at the frame header. Clearly, a place in the IEEE 802.3 CSMA/CD frame had to be found to put the DIX Ethernet II `type` field, since receivers had to have a way to figure out which software process understood the frame content's data structure. Other protocols did not understand IP packet structures, and vice versa.

The IEEE 802.3 committee "robbed" some bytes from the payload area, bytes which in DIX Ethernet were data bytes. Since the overall length of the frame was already fixed, and this set the length of the frame data to 1500 bytes (the same as in DIX Ethernet), the outcome was to *reduce* the allowed length of the data contents of an IEEE 802.3 frame. A simplified picture of the two frame types indicating the location of the `0x0800` `type` field and the length of the `data` field is shown in Figure 3.8.

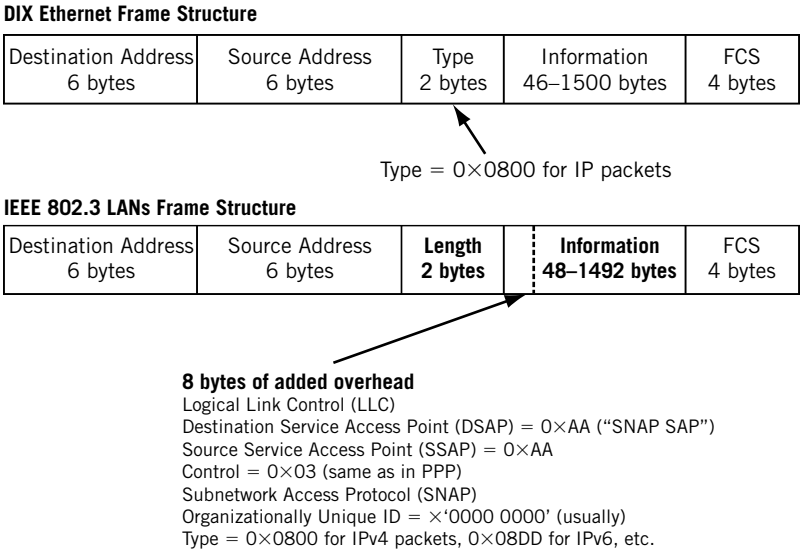


FIGURE 3.8

Types of Ethernet frames. The frames for Gigabit and 10 Gigabit Ethernet differ in detail, but follow the same general structure.

MAC Addresses

The MAC addresses used in 802 LAN frames are all 48 bits (6 bytes) long. The first 24 bits (3 bytes) are assigned by the IEEE to the manufacturer of the NIC (manufacturers pay for them). This is the Organizationally Unique Identifier (OUI). The last 24 bits (3 bytes) are the NIC manufacturer's serial number for that NIC. Some protocol analyzers know the manufacturer's ID (which is not public but seldom suppressed) and display this along with the address. This is how Ethereal displays MAC addresses not only in hex but starting with "Intel_" or "Juniper_."

Note that both frame types use the same, familiar source and destination MAC address, and use a 32-bit (4-byte) frame check sequence (FCS) for frame-level error detection. The FCS used in both cases is a standard, 32-bit cyclical redundancy check (CRC-32). The important difference is that the DIX Ethernet frame indicates information type (frame content) with a 2-byte `type` field (`0x0800` means there is an IPv4 packet inside and `0x86DD` means there is an IPv6 packet inside) and the IEEE 802.3. CSMA/CD frame places this Ethertype field at the end of an additional 8 bytes of overhead called the Subnetwork Access Protocol (SNAP) header. Another 3 bytes are the OUI given to the NIC vendor when they registered with the IEEE, but this field is not always used for that purpose.

The 802.3 frame must subtract these 8 bytes from the IP packet length so that the overall frame length is still the same as for DIX Ethernet II. This is because the maximum length of the frame is universal in almost all forms of Ethernet. The maximum

IEEE 802.3 frame data is 1492 due to the 8 extra bytes needed to represent the `type` field. Any IP packet larger than this will not fit in a single frame, and must *fragment* its payload into more than one frame and have the payload reassembled at the receiver.

That's not all there is to it. LAN implementers and vendors quickly saw that the IEEE 802.3 hardware arrangement was more flexible (and less expensive) than DIX Ethernet. They also saw that the DIX Ethernet II frame structure was simpler and could carry slightly more user data than the complex IEEE 802.3 frame structure. Being practical people, the vendors simply used the flexible IEEE 802.3 hardware with the simple DIX Ethernet II frame structure, creating the mixture that is commonly seen today on most LANs.

Today, just because the *hardware* is IEEE 802.3 compliant (e.g., 100BaseT), does not mean that the frame structure used to carry IP packets is *also* IEEE 802.3 compliant. The frame structure is most likely Ethernet II, as we have seen. (It's worth pointing out that Ethernet frame content *other* than IP usually uses the 802.3 frame format. However, the Illustrated Network is basically an IP-only network.)

THE EVOLUTION OF DSL

IP packet interfaces have been defined for many LAN and WAN network technologies. As soon as a new transport technology reaches the commercial-deployment stage, IP is part of the scheme, if for no other reason than regardless of what is in the middle, TCP/IP in Ethernet frames is at both ends. DSL technologies are a case in point. Originally designed for the "national networks" that would offer everything that the Internet does today, but from the telephone company as part of the Integrated Services Digital Network (ISDN) initiatives of the 1980s, DSL was adapted for "broadband" Internet access when the grand visions of the telephone companies as content providers were reduced to the reality of a restricted role as ISPs and little more. (Even the term "broadband" is a topic of much debate: A working definition is "speeds fast enough to allow users to watch video without getting a headache or becoming disgusted," speeds that keep dropping as video coding and compression techniques become better.)

DSL once included a complete ATM architecture, with little or no TCP/IP. Practical considerations forced service providers to adapt DSLs once again, this time for the real consumer world of Ethernet LANs running TCP/IP. And a tortured adaptation it proved to be. The problem was deeper than just taking an Ethernet frame and mapping it to a DSL frame (even DSL bits are organized into a distinctive transport frame). Users had to be assigned unique IP addresses (not necessary on an isolated LAN), and the issues of bridging versus routing versus switching had to be addressed all over again. This was because linking two LANs (the home user client LAN, even if it had but one PC, and the server LAN) over a WAN link (DSL) was not a trivial task. The server LAN could be the service provider's "home server" or anyplace else the user chose to go on the Internet.

Also, ATM logical links (called permanent virtual circuits, or PVCs) are normally provisioned between the usual local exchange carrier's DSLAM and the Internet access

Networking Visions Today and Yesterday

Today, when anyone can start a Web site with a simple server and provide a service to one and all over the Internet, it is good to remember that things were not always supposed to be this way. Not so long ago, the control of services on a public global network was supposed to be firmly under the control of the service provider. Many of these "fast-packet" networking schemes were promoted by the national telephone companies, from broadband ISDN to ATM to DSL. They all envisioned a network much like the Internet is today, but one with all the servers "in the cloud" owned and operated by the service providers. Anyone wanting to provide a service (such as a video Web site) would have to go to the service provider to make arrangements, and average citizens would probably be unable to break into that tightly controlled and expensive market.

This scheme avoided the risk of controversial Web site content (such as copyrighted material available for download), but with the addition of restrictions and surveillance. Also, the economics for service providers are much different when they control content from when they do not.

Today, ISPs most often provide transport and connectivity between Web sites and servers owned and operated by almost anyone. ISP servers are usually restricted to a small set of services directly related to the ISP, such as email or account management.

provider's aggregation router. This can be very costly because IP generally has much better statistical multiplexing properties and there can be long hauls through the ATM networks before the ATM link is terminated.

The solution was to scrap any useful role for ATM (and any non-TCP/IP infrastructure) except as a passive transport for IP packets. This left ATM without any rationale for existence, because most of the work was done by running PPP over the DSL link between a user LAN and a service provider LAN.

PPP and DSL

Why is PPP used with DSL (and SONET)? The core of the issue is that ISPs needed some kind of *tunneling* protocol. Tunneling occurs when the normal message-packet-frame encapsulation sequence of the layers of a networking protocol suite are violated. When a message is placed inside a packet, then inside a frame, and this frame is placed inside *another* type of frame, this is a tunneling situation. Although many tunneling methods have been standardized at several different TCP/IP layers, tunneling works as long as the tunnel endpoints understand the correct sequence of headers and content (which can also be encrypted for *secure* tunnels).

In DSL, the tunneling protocol had to carry the point-to-point "circuits" from the central networking location to the customer's premises and across the shared media

LAN to the end user device (host). There are many ways to do this, such as using IP-in-IP tunneling, a virtual private network (VPN), or lower level tunneling. ISPs chose PPP as the solution for this role in DSL.

Using PPP made perfect sense. For years, ISPs had used PPP to manage their WAN dial-in users. PPP could easily assign and manage the ISP's IP address space, compartmentalize users for billing purposes, and so on. As a LAN technology, Ethernet had none of those features. PPP also allowed user authentication methods such as RADIUS to be used, methods completely absent on most LAN technologies (if you're on the LAN, it's assumed you belong there).

Of course, keeping PPP meant putting the PPP frame inside the Ethernet frame, a scheme called Point-to-Point Protocol over Ethernet (PPPoE), described in RFC 2516. Since tunneling is just another form of encapsulation, all was well.

PPP is not the only data link layer framing and negotiation procedure (PPP is not a full data link layer specification) from the IETF. Before PPP became popular, the Serial Line Internet Protocol (SLIP) and a closely related protocol using compression (CSLIP, or Compressed SLIP) were used to link individual PCs and workstations not connected by a LAN, but still running TCP/IP, to the Internet over a dial-up, asynchronous analog telephone line with modems. SLIP/CSLIP was also once used to link routers on widely separated TCP/IP networks over asynchronous analog leased telephone lines, again using modems. SLIP/CSLIP is specified in RFC 1055/STD 47.

PPP Framing for Packets

PPP addresses many of the limitations of SLIP, and can run over both asynchronous links (as does SLIP) and synchronous links. PPP provides for more than just a simple frame structure for IP packets. The PPP standard defines management and testing functions for line quality, option negotiation, and so on. PPP is described in RFC 1661, is protocol independent, and is not limited to IP packet transport.

The PPP control signals, known as the PPP Link Control Protocol (LCP), need not be supported, but are strongly recommended to improve performance. Other control information is included by means of a Network Control Protocol (NCP), which defines management procedures for frame content protocols. The NCP even allows protocols other than IP to use the serial link at the same time. The LCP and NCP subprotocols are a distinguishing feature of PPP.

The use of LCP and NCP on a PPP link on a TCP/IP network follows:

- The source PPP system (user) sends a series of LCP messages to configure and test the serial link.
- Both ends exchange LCP messages to establish the link options to be used.
- The source PPP system sends a series of NCP messages to establish the Network Layer protocol (e.g., IP, IPX, etc.).
- IP packets and frames for any other configured protocols are sent across the link.
- NCP and LCP messages are used to close the link down in a graceful and structured manner.

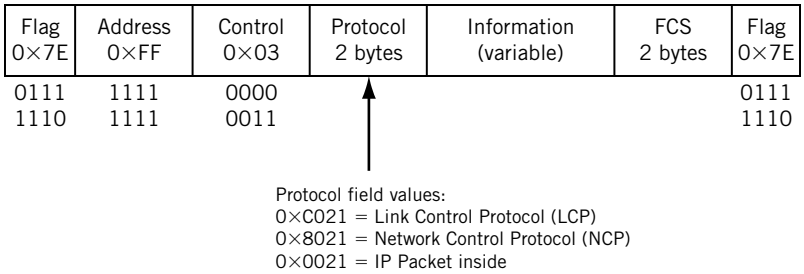


FIGURE 3.9

The PPP frame. The flag bytes (0×7E) essentially form an "idle pattern" on the link that is "interrupted" by frames carrying information.

The benefits are to create a more efficient WAN transport for IP packets. The structure of a PPP frame is shown in Figure 3.9.

The Flag field is `0x7E` (`0111 1110`), as in many other data link layer protocols. The Address field is set to `0xFF` (`1111 1111`), which, by convention, is the "all-stations" or broadcast address. Note that none of the other fields in the Point-to-Point Protocol header have a source address for the frame. Point-to-point links only care about the destination, which is always `0xFF` in PPP and essentially means "any device at the other end of this link that sees this frame." This is one reason why serial interfaces on routers sometimes do not have IP addresses (but many serial interfaces, especially to other routers, have them anyway—this is the only way to make the serial links "visible" to the IP layer and network operations).

The Control field is set to `0x03` (`0000 0011`), which is the Unnumbered Information (UI) format, meaning that there is no sequence numbering in these frames. The UI format is used to indicate that the connectionless IP protocol is in use. The Protocol field identifies the format and use of the content of the PPP frame itself. For LCP messages, the Protocol field has the value `0xC021` (`1100 0000 0010 0001`), for NCP the field has the value `0x8021` (`1000 0000 0010 0001`), and for IP packets the field has the value `0x0021` (`0000 0000 0010 0001`).

Following the header is a variable-length Information field (the IP packet), followed by a PPP frame trailer with a 16-bit, frame check sequence (FCS) for error control, and finally an end-of-frame Flag field.

PPP frames may be compressed, field sizes reduced, and used for many specific tasks, as long as the endpoints agree.

DSL Encapsulation

How are IP packets encapsulated on DSL links? DSL specifications establish a basic DSL frame as the physical level, but IP packets are not placed directly into these frames. IP packets are placed inside PPP frames, and then the PPP frames are encapsulated inside Ethernet frames (this is PPP over Ethernet, or PPPoE). Finally, the Ethernet frames are

placed inside the DSL frames and sent to the DSL Access Module (DSLAM) at the telephone switching office.

Once at the switching office, it might seem straightforward to extract the Ethernet frame and send it on into the "router cloud." But it turns out that almost all DSLAMs are networked together by ATM, a technology once championed by the telephone companies. (Some very old DSLAMs use another telephone company technology known as frame relay.) ATM uses *cells* instead of frames to carry information.

So the network/data-link/physical layer protocol stack used between DSLAMs and service provider routers linked to the Internet usually looks like five layers instead of the expected three:

- IP packet containing user data, which is inside a PPP frame, which is inside an
- Ethernet frame running to the DSL router (PPPoE), which is inside a series of
- ATM cells, which are sent over the physical medium as a series of bits.

We'll take a closer look at frame relay and ATM in a later chapter on public network technologies that can be used to link routers together.

Forms of DSL

Entire books are devoted to the variations of DSL and the DSL protocol stacks used by service providers today. Instead of focusing on all the details of these variations, this section will take a brief look at the variation of DSL that can be used when IP packets make their way from a home PC onto the Internet.

DSL often appears as "xDSL" where the "x" can stand for many different letters. DSL is a modern technology for providing broadband data services over the same twisted-pair (TP), copper telephone lines that provide voice service. DSL services are often called "last-mile" (and sometimes "first-mile") technologies because they are used only for short connections between a telephone switching station and a home or office. DSL is not used between switching stations (SONET is often used there).

DSL is an extension of the Integrated Services Digital Network (ISDN) technology developed by the telephone companies for their own set of combined voice and data services. They operate over short ranges (less than 18 kilofeet) of 24 American Wire Gauge (AWG) voice wire to a telephone central office. DSLs offer much higher speeds than traditional dial-up modems, up to 52 mbps for traffic sent "downstream" to the user and usually from 32 kbps to 1 Mbps from traffic sent "upstream" to the central office. The actual speed is distance limited, dropping off at longer distances.

At the line level, DSLs use one of several sophisticated modulation techniques running in premises DSL router chipsets and DSLAMs at the telephone switching office. These include the following:

- Carrierless Amplitude Modulation (CAP)
- Discrete Multitone Technology (DMT)
- Discrete Wavelet Multitone (DWM)
- Simple Line Code (SLC)
- Multiple Virtual Line (MVL)

DSL can operate in a duplex (symmetrical) fashion, offering the same speeds upstream and downstream. Others, mainly targeted for residential Internet browsing customers, offer higher downstream speeds to handle relatively large server replies to upstream mouse clicks or keystrokes. However, standard VDSL and VDSL2 have much less asymmetry than other methods. For example, 100-Mbps symmetric operation is possible at 0.3 km, and 50 Mbps symmetric at 1 km.

The DSLAMs connect to a high-speed service provider backbone, and then the Internet. DSLAMs aggregate traffic, typically for an ATM network, and then connect to a router network. On the interface to the premises, the DSLAM demultiplexes traffic for individual users and forwards it to the appropriate users.

In order to support traditional voice services, most DSL technologies require a signal filter or "splitter" to be installed on the customer premises to share the twisted-pair wiring. The DSLAM splits the signal off at the central office. Splitterless DSL is very popular, however, in the form of "DSL Lite" or several other names.

In Table 3.2, various types of DSL are compared. The speeds listed are typical, as are the distance (there are many other factors that can limit DSL reach) and services offered.

VDSL requires a fiber-optic feeder system to the immediate neighborhood, but VDSL can provide a full suite of voice, video, and data services. These services include the highest Internet access rates available for residential services, and integration between voice and data services (voice mail alerts, caller ID history, and so on, all on the TV

Table 3.2 Types of DSL

Type	Meaning	Typical Data Rate	Mode	Distance	Applications
IDSL	ISDN DSL	128 Kbps	Duplex	18k ft on 24 AWG TP	ISDN services: voice and data; Internet access
HDSL	High-speed DSL	1.544 to 42.048 Mbps	Duplex	12k ft on 24 AWG TP	T1/E1 service, feeder, WAN access, LAN connections, Internet access
SDSL	Symmetric DSL	1.544 to 2.048 Mbps	Duplex	12k ft on 24 AWG TP	Same as HDSL
ADSL	Asymmetric DSL	1.5 to 6 Mbps 16 to 640 kbps	Down Up	18k ft on 24 AWG TP	Internet access, remote LAN access, some video applications.
DSL Lite (G.Lite)	"Splitterless" ADSL	1.5 to 6 Mbps 16 to 640 kbps	Down Up	18k ft on 24 AWG TP	Same as ADSL, but does not require a premises "splitter" for voice services
VDSL	Very-high-speed DSL	13 to 52 Mbps 1.5 to 2.3 Mbps	Down Up	1k to 4.5k ft depending on speed	Same as ADSL plus full voice and video services, including HDTV

screen). VDSL is used on the Illustrated Network to get packets from the home office's PCs to the ISP's router network (the overall architecture is not very different from DSL in general). From router to router over WAN distances, the Illustrated Network uses a common form of transport for the Internet in the United States: SONET.

THE EVOLUTION OF SONET

SONET is the North American version of the international SDH standard and defines a hierarchy of fast transports delivered on fiber-optic cable. One of the most exciting aspects of SONET when it first appeared around 1990 was the ability to deploy SONET links in self-healing rings, which nearly made outages a thing of the past. (The vast majority of link failures today involve signal "backhoe fade," a euphemism for accidental cable dig-ups.)

Before networks composed almost entirely of fiber-optic cables came along, network errors were a high-priority problem. Protocols such as IP and TCP had extensive error-detection and error-correction (the two are distinct) methods built into their operation, methods that are now quietly considered almost a hindrance in modern networks.

Now, SONET rings do not inherently protect against the common problem of a lack of equipment or route diversity, but at least it's possible. Not all SONET links are on rings, of course. The links on the Illustrated Network are strictly point-to-point.

A Note about Network Errors

Before SONET, almost all WAN links used to link routers were supplied by a telephone company that subscribed to the Bell System standards and practices, even if the phone company was not part of the sprawling AT&T Bell System. In 1984, the Bell System engineering manual named a bit error rate (BER) of 10^{-5} (one error in 100,000 bits sent) as the target for dial-up connections, and put leased lines (because they could be "tuned" through predictable equipment) at 10 times better, or 10^{-6} (one error in every 1,000,000 bits).

SONET/SDH fiber links typically have BERs of 1000 (10^3) to 1 million (10^6) times better than those common in 1984. Since 1000 days is about 3 years, converting a copper link to fiber meant that all the errors seen yesterday are now spread out over the next 3 years (a BER of 10^{-9}) to 3000 years (10^{-12}). LAN error rates, always much lower than those of WANs due to shorter spans and less environmental damage, are in about the same range. Most errors today occur on the modest-length (a kilometer or mile) access links between LAN and WAN to ISP points of presence, and most of those errors are due to intermittently failing or faulty connectors.

The only real alternatives for SONET/SDH high-speed WAN links are newer versions of Ethernet, especially in a metropolitan Ethernet context. The megabit-speed T1 (1.544 Mbps) or E1 (2.048 Mbps) links are used for the local loop. However, even those copper-based circuits are usually serviced by newer technologies and carried over SONET/SDH fiber on the backbone.

How are IP packets carried inside SONET frames? The standard method is called Packet over SONET/SDH (POS). The procedures used in POS are defined in three RFCs:

- RFC1619, PPP over SONET/SDH
- RFC1661, the PPP
- RFC1662, PPP in HDLC-like framing

Packet over SONET/SDH

SONET/SDH frames are not just a substitute for Ethernet or PPP frames. SONET/SDH frames, like T1 and E1 frames, carry unstructured bit information, such as digitized voice telephone calls, and are not usually suitable for direct packet encapsulation. In the case of IP, the packets are placed inside a PPP frame (technically, a type of High-Level Data Link Control ["HDLC-like"] PPP frame with some header fields allowed to vary in HDLC fixed for IP packet payloads). The PPP frame, delimited by a stream of special `0x7E` interframe fill (or "idle" pattern) bits, is then placed into the payload area of the SONET/SDH frame.

Figure 3.10 shows a series of PPP frames inside a SONET frame running at 51.84 Mbps. Although SONET (and SDH) frames are always shown as two-dimensional arrays

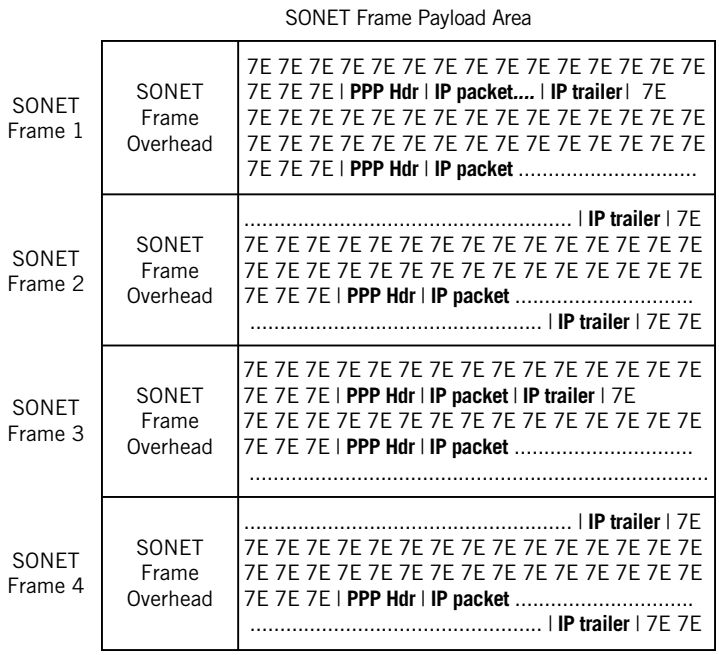


FIGURE 3.10

Packet over SONET, showing how the idle pattern of 0x7E surrounds the PPP frames with IP packets inside.

of bits, the figure is not very accurate. It doesn't show any of the SONET framing bytes, and IP packets are routinely set to around 1500 bytes long, so they would easily fill an entire 774-byte, basic SONET transmission-frame payload area. Even the typical network default maximum IP packet size of 576 bytes is quite large compared to the SONET payload area. However, many packets are not that large, especially acknowledgments.

One other form of transport used on the Illustrated Network is common on IP networks today. Wireless links might some day be more common than anything else.

WIRELESS LANs AND IEEE 802.11

Wireless technologies are the fastest-growing form of link layer for IP packets, whether for cell phones or home office LANs. Cell phone packets are a bit of a challenge, and wireless LANs are evolving rapidly, but this section will focus on wireless LANs, if only because wireless LANs are such a good fit with Ethernet. This section will be a little longer than the others, only because the latest wireless LANs are newer than the previous methods discussed.

The basic components of the IEEE 802.11 wireless LAN architecture are the wireless stations, such as a laptop, and the access point (AP). The AP is not strictly necessary, and a cluster of wireless stations can communicate directly with each other without an AP. This is called an IEEE 802.11 independent, basic service set (IBSS) or ad hoc network. One or more wireless stations form a basic service set (BSS), but if there is only one wireless station in the BSS, an AP is necessary to allow the wireless station to communicate. An AP has both wired and wireless connections, allowing it to be the access "point" between the wireless station and the world. In a typical home wireless network (an arbitrarily low limit), one BSS supports up to four wireless devices, and the AP is bundled with the DSL router or cable modem with the high-speed link for Internet access. (The DSL router or cable modem can have multiple wired connections as well.) In practice, the number of systems you can connect to a given type of AP depends on your performance needs and the traffic mix.

A wireless LAN can have multiple APs, and this arrangement is sometimes called an infrastructure wireless LAN. This type of LAN has more than one BSS, because each AP establishes its own BSS. This is called an extended service set (ESS), and the APs are often wired together with an Ethernet LAN or an Ethernet hub or switch. The three major types of IEEE 802.11 wireless LANs—ad hoc (IBSS), BSS, and ESS—are shown in Figure 3.11.

Wi-Fi

An intended interoperable version of the IEEE 802.11 architecture is known as *Wi-Fi*, a trademark and brand of the Wi-Fi Alliance. It allows users with properly equipped wireless laptops to attach to APs maintained by a service provider in restaurants, bookstores, libraries, and other locations, usually to access the Internet. In some places, especially downtown urban areas, a wireless station can receive a strong signal from two or

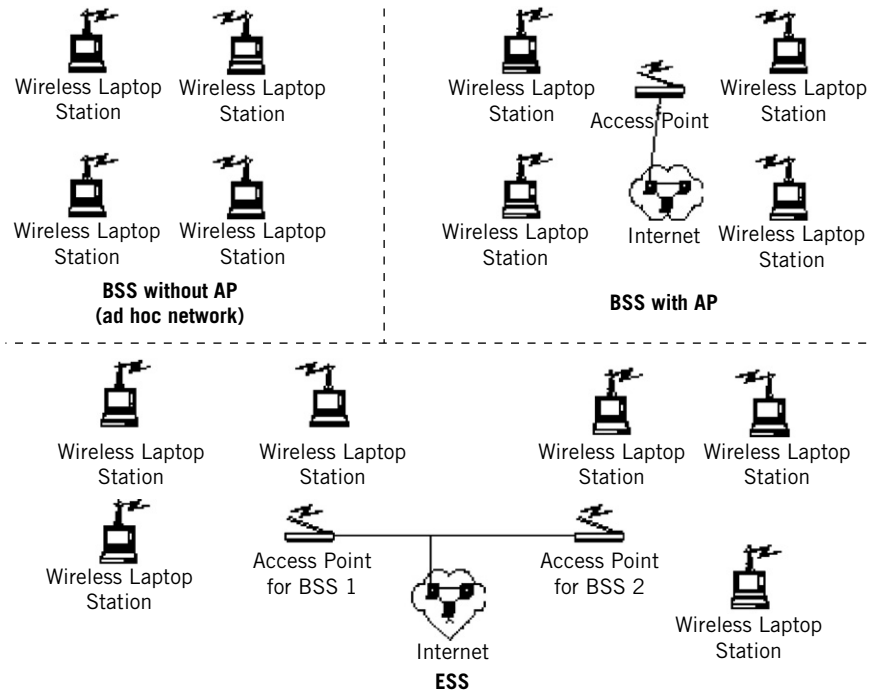


FIGURE 3.11

Wireless LAN architectures. Most home networks are built around an access point built into a DSL router/gateway.

more APs. While a wireless station can belong to more than one BSS through its AP at the same time, this is not helpful when the APs are offering different network addresses (and perhaps prices for attachment). This collection of Wi-Fi networks is sometimes called the "Wi-Fi jungle," and will only become worse as wireless services turn up more and more often in parks, apartment buildings, offices, and so on. How do APs and wireless stations sort themselves out in the Wi-Fi jungle?

If there are APs present, each wireless station in IEEE 802.11 needs to *associate* with an AP before it can send or receive frames. For Internet access, the 802.11 frames contain IP packets, of course. The network administrator for every AP assigns a *Service Set Identifier* (SSID) to the AP, as well as the channels (frequency ranges) that are associated with the AP. The AP has a MAC layer address as well, often called the BSSID.

The AP is required to periodically send out *beacon frames*, each including the AP's SSID and MAC layer address (BSSID), on its wireless channels. These channels are scanned by the wireless station. Some channels might overlap between multiple APs, because the "jungle" has no central control, but (hopefully) there are other channels that do not. In practice, interference between overlapping APs is not a huge problem

in the absence of a high volume of traffic. When you "view available networks" in Windows XP, the display is a list of the SSIDs of all APs in range. To get Internet access, you need to associate your wireless station with *one* of these APs.

After selecting an AP by SSID, the wireless host uses the 802.11 association protocol to join the AP's subnet. The wireless station then uses DHCP to get an IP address, and becomes part of the Internet through the AP.

If the wireless Internet access is not free, or the wireless LAN is intended for restricted use (e.g., tenants in a particular building), the wireless station might have to authenticate itself to the AP. If the pool of users is small and known, the host's MAC address can be used for this purpose, and only certain MAC addresses will receive IP addresses.

Once the user is on the wireless network, many hotels use the *captive portal* form of authentication. The captive portal technique makes the user with a Web browser (HTTP client) to see a special Web page before being granted normal Internet access. The captive portal intercepts all packets regardless of address or port, until the browser is used as a form of authentication device. Once the acceptable use terms are viewed or the payment rates are accepted and arranged, "normal" Internet access is granted for a fixed period of time. It should be noted that captive portals can be used to control wired access as well, and many places (hotel rooms, business centers) use them in this fashion. In many cases, the normal device "firewall" capabilities must be turned off or configured to allow the captive portal Web page to appear.

Another post-access approach employs usernames and passwords—these are popular at coffee shops and other retail establishments. In both cases, there is usually a central *authentication server* used by many APs, and the wireless host communicates with this server using either RADIUS (RFC 2138) or DIAMETER (RFC 3588). Once authenticated, the users' traffic is commonly encrypted to preserve privacy over the airwaves, where signals can usually be picked up easily and without the knowledge of end users.

When accessing the office remotely, even if captive portal or some other method is used, most organizations add something to secure tunneling based on PPTP (Microsoft's Point-to-Point Tunneling Protocol) or PPPoE to run proprietary VPN client software. We've already mentioned PPPoE, and PPTP with VPNs will be explored later in this book.

IEEE 802.11 MAC Layer Protocol

IEEE 802.11 defines two MAC sublayers: the distributed coordination function (DCF) and the point coordination function (PCF). The PCF MAC is optional and runs on top of the DCF MAC, which is mandatory. PCF is used with APs and is very complex, while DCF is simpler and uses a venerable access method known as *carrier sense multiple access with collision avoidance* (CSMA/CA). Note that while Ethernet LANs *detect* collisions between stations sending at the same time with CSMA/CD, wireless LANs *avoid* collisions. Collision *detection* is not appropriate for wireless LANs for a number of reasons, the most important being the *hidden terminal problem*.

To understand the hidden terminal problem, consider the two wireless laptops and AP shown in Figure 3.12. (The problem does not only occur with an AP, but the figure

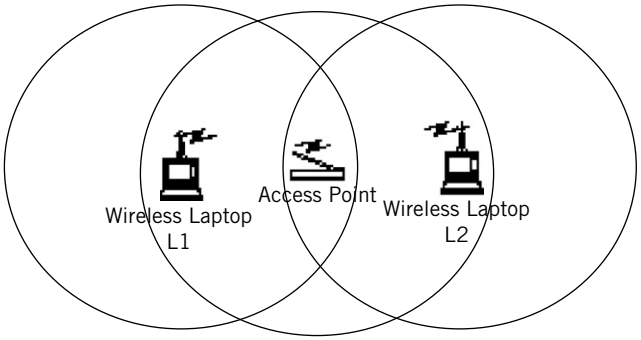


FIGURE 3.12

Hidden terminals on wireless LANs. This can be a problem in larger home networks, and special "LAN extender" devices can be used to prevent the problem.

shows this situation.) Both laptops are within range of the AP, but not of each other (there are many reasons for this, from distance to signal fading). Obviously, if L1 is sending a frame to the AP, L2 could also start sending a frame, because the carrier sensing shows the network as "clear." However, a collision occurs at the AP and both frames have errors, although both L1 and L2 think their frames were sent just fine.

Now, the AP clearly knows what's going on. It just needs a way to tell the wireless stations when it's okay to send (or not). CSMA/CD can use an optional method known as *request to send* (RTS) and *clear to send* (CTS) to avoid these types of undetected collisions. When a sender wants to send a data frame, it must first reserve the channel by sending a short RTS frame to the AP, telling the AP how long it will take to send the data, and receive an acknowledgement frame (ACK) that all went well. If the sender receives a short CTS control frame back, then it can send. Other stations hear the CTS as well, and refrain from sending during this time period.

The way that RTS/CTS works for sending data to an access point is shown in Figure 3.13.

There are two time notations in the figure: DIFS and SIFS. The *distributed inter-frame space* (DIFS) is the amount of time a wireless station waits to send after sensing that the channel is clear. The station waits a bit "just in case" because wireless LANs, unlike Ethernet, do not detect collisions and cease sending, so collisions are very debilitating and must be avoided at all costs. The *short inter-frame spacing* (SIFS) is also used between frames for collision avoidance. There is also a duration timer in all 802.11 frames, measured in microseconds, that tells the other stations how long it will take to send the frame and receive a reply. Stations avoid link access during this time period.

While RTS/CTS does reduce collisions, it also adds delay and reduces the available bandwidth on a channel. In practice, each wireless station sets an RTS threshold so that CTS/RTS is used only when the frame is longer than this value. Many wireless stations set the threshold so high that the value is larger than the maximum frame length, and the RTS/CTS is skipped for all data.

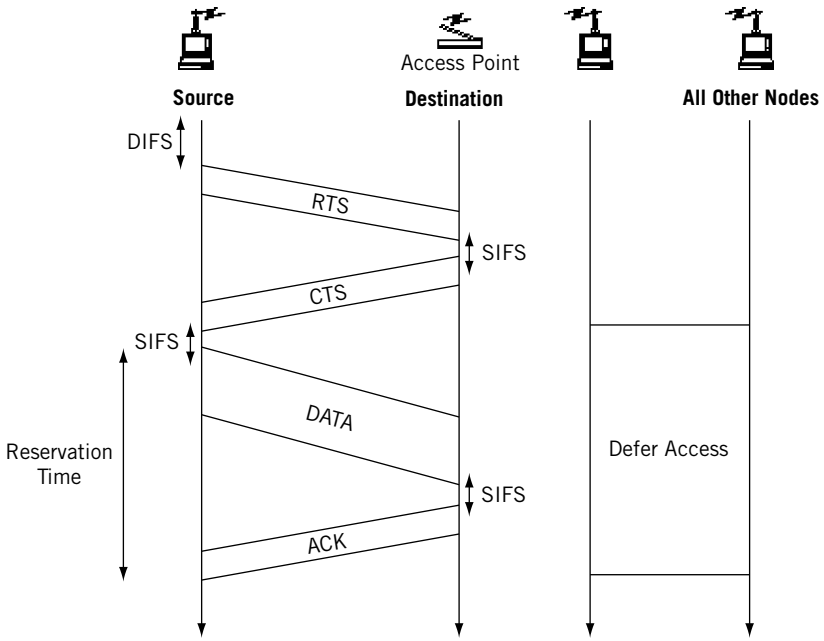


FIGURE 3.13

RTS and CTS in wireless LANs showing how all other nodes must defer access to the medium. The CTS is heard by all other nodes, although this is not detailed in the figure.

Frame Control	Duration	Address 1	Address 2	Address 3	Seq. Control	Address 4	Payload	FCS
2 bytes	2 bytes	6 bytes	6 bytes	6 bytes	2 bytes	6 bytes	0–2312 bytes	4 bytes

FIGURE 3.14

IEEE 802.11 frame structure. Note the potential number of address fields (four) in contrast to the two used in Ethernet II frames.

The IEEE 802.11 Frame

Although the IEEE 802.11 frame shares a lot with the Ethernet frame (which is one reason some packet sniffers can parse wireless frames as if they were Ethernet), there are a number of unique fields in 802.11. There are nine main fields, and the *frame control* (FC) field has 10 fields. The nine major fields of the IEEE 802.11 MAC frame are shown in Figure 3.14. The only fields in the two FC bytes that we will talk about are the *From DS* and *To DS* fields. (In some cases, the first three fields of the 802.11 MAC frame, the version, type, and subtype, are presented separately from the frame control flags, which are all bits.)

Frame control (FC)—This field is 2 bytes long and contains, among other things, two important flag bits: To DS (distribution system) and From DS.

Duration—This byte gives the duration of the transmission in all frame types except one. In one control frame, this "D" byte gives the ID of the frame.

Addresses—There are four possible address fields, each 6 bytes long and structured the same as Ethernet MAC addresses. The fourth field is only present when multiple APs are in use in an ESS. The meaning of each address field depends on the value of the DS flags in the FC field, discussed later.

Sequence control—This 2-byte field gives the sequence number of the frame and is used in flow control.

Payload—This field can be from 0 to 2312 bytes long. Usually it is fewer than 1500 bytes and holds an IP packet, but there are other types of payloads. The precise type and subtype of the content is determined by the content of the FC field.

CRC—The frame cyclical redundancy check is a 4-byte CRC-32, used to determine the nature of the acknowledgement sent.

Why does the wireless frame need to define four address fields? Mainly because the arrangements of wireless stations can be complicated. Is there an AP in the BSS? Is there more than one AP? What type of frame is being sent? Data? Control? Management? The number of address fields present, and what they represent, depend on the answers to these questions.

How do receivers know exactly how many addresses are used and what they represent? That's where the two DS flags in the FC field come in. The meaning of the address fields (and possible presence of the Address 4 field) depends on the values of these two bits. Actually, there are *five* types of MAC addresses used in wireless LANs:

BSSID—This is usually the MAC address of the AP, but it is generated randomly in an IBSS or ad hoc network.

Transmitter Address (TA)—The TA is the MAC address of the individual station that has just sent the frame.

Receiver Address (RA)—The RA is the MAC address of the immediate receiver of the frame. This can be a group or broadcast address.

Source Address (SA)—The SA is the MAC address of the individual station that originated the frame. Due to the possible role played by the AP, the SA is not necessarily the same as the TA.

Destination Address (DA)—The DA is the MAC address of the final destination of the frame, and can also be a group or broadcast as well as an individual station. Again, due to the AP(s), this address might not match the RA.

Table 3.3 DS Bits and Wireless LAN Data Frame Address Fields

Type of Network	From DS	To DS	Address 1	Address 2	Address 3	Address 4
Ad hoc (IBSS)	0	0	DA (= RA)	SA	BSSID	N/A
To AP	0	1	RA (= BSSID)	SA	DA	N/A
From AP	1	0	DA (= RA)	BSSID	SA	N/A
ESS (multiple APs)	1	1	RA	TA	DA	SA

The interplay among these address types and the meaning of the two DS flags for data frames is shown in Table 3.3.

A look back at Figures 3.6 and 3.7 will show that these address patterns are reflected in the screen captures. The last two bits of the frame control flags are the DS bits, which are 01 (To AP) and 10 (From AP), respectively. The Proxima AP is passing the frame between the Cisco and Farallon wireless stations.

The Address 4 field appears only when there are multiple APs. Usually, data frames in a simple BSS with AP use DS bit combinations `01` and `10` to make their way through the AP from one wireless station to another.

QUESTIONS FOR READERS

Figure 3.15 shows some of the concepts discussed in this chapter and can be used to help you answer the following questions.

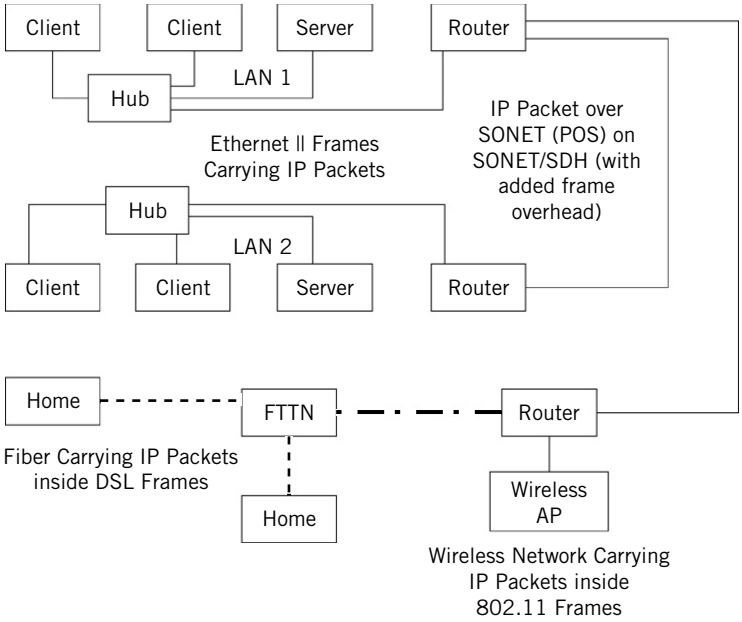


FIGURE 3.15

IP packets are carried in many different types of frames, and some of those frames are tucked inside lower level transmission frames.

1. Both LAN1 and LAN2 use Ethernet II frames. What would happen if frame types on the two LANs were different?
2. SONET/SDH still has its own overhead bytes when IP packets are carried inside the SONET/SDH frames. Why is the SONET/SDH overhead still necessary?
3. What is the captive portal method of wireless access permission and how does it work?
4. Ethernet LANs can extend to metropolitan area distances and perhaps beyond. If Metro Ethernet evolved to remove all distance limits, what are the advantages and disadvantages of *always* using Ethernet frames for IP packets?
5. Why are more than two addresses used in wireless frames in some cases? Which cases require more than two addresses?