Chapter

4

The Current Ethernet Specifications

THE FOLLOWING COMPTIA NETWORK+ EXAM OBJECTIVES ARE COVERED IN THIS CHAPTER:

✓ **2.6 Categorize LAN technology types and properties**

- Types:
  - Ethernet
  - 10BaseT
  - 100BaseTX
  - 100BaseFX
  - 1000BaseT
  - 1000BaseX
  - 10GBaseSR
  - 10GBaseLR
  - 10GBaseER
  - 10GBaseSW
  - 10GBaseLW
  - 10GBaseEW
  - 10GBaseT
- Properties
  - CSMA/CD
  - Broadcast
  - Collision
  - Bonding
  - Speed
  - Distance

Before we move on and explore networking devices, the TCP/IP and DoD models, IP addressing, subnetting, and routing in the upcoming chapters, you've got to understand the big picture of LANs and learn the answer to the key questions: "How is Ethernet used in today's networks, and what are Media Access Control (MAC) addresses and how are they used?"

This chapter will answer those questions and more. I'll not only discuss the basics of Ethernet and the way MAC addresses are used on an Ethernet LAN, but I'll also cover the protocols used with Ethernet at the Data Link layer as well. You'll also learn about the various Ethernet specifications.

So now, let's get started with the fundamentals of connecting two hosts together.

For up-to-the-minute updates for this chapter, please see `www.lammle.com` or `www.sybex.com/go/comptianetwork+studyguide`.

Network Basics

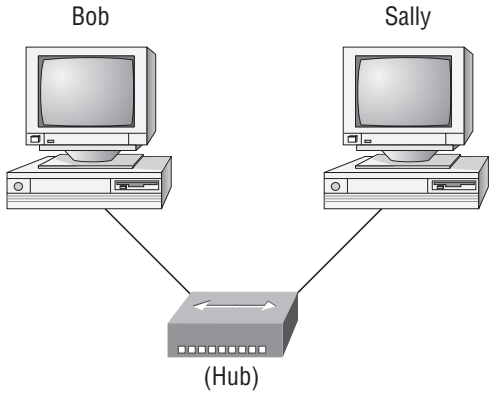
Networks and networking have grown exponentially over the last 20 years—understandably so. They've had to evolve at light speed just to keep up with huge increases in basic mission-critical user needs ranging from sharing data and printers to more advanced demands like videoconferencing. Unless everyone who needs to share network resources is located in the same office area (an increasingly uncommon situation), the challenge is to connect the sometimes large number of relevant networks together so all users can share the networks' wealth.

As I started to discuss in Chapter 1, "Introduction to Networks," let's take a look at how communication happens on a basic local area network (LAN). Starting with Figure 4.1, you get a picture of a basic LAN network that's connected together using an Ethernet connection to a hub. This network is actually one collision domain and one broadcast domain, but don't stress if you have no idea what this means—I'm going to talk about both collision and broadcast domains in depth in Chapter 5, "Networking Devices."

Okay, about Figure 1.1… How would you say the PC named Bob communicates with the PC named Sally? Well, they're both on the same LAN connected with a multiport repeater (a hub). So does Bob just send out a data message, "Hey Sally, you there?" or does Bob use Sally's IP address and put things more like, "Hey 192.168.0.3, are you there?" I hope you picked the IP address option, but even if you did, the news is still bad—both answers are wrong! Why?

Because Bob is actually going to use Sally's MAC address (known as a *hardware address*), which is burned right into the network card of Sally's PC, to get hold of her.

FIGURE 4.1 The basic network



This is all good, but how does Bob get Sally's MAC address when Bob knows only Sally's name and doesn't even have her IP address? Bob is going to start by using name resolution (hostname to IP address resolution), something that's usually accomplished using Domain Name Service (DNS). And note that if these two hosts are on the same LAN, Bob can just broadcast to Sally asking her for the information (no DNS needed)—welcome to Microsoft Windows, Vista included!

Here's the output from a network analyzer depicting a simple name-resolution process from Bob to Sally:

```
Time      Source     Destination   Protocol   Info
53.892794  192.168.0.2  192.168.0.255  NBNS  Name query NB SALLY<00>
```

As I already mentioned, because the two hosts are on a local LAN, Windows (Bob) will broadcast to resolve the name Sally (the destination 192.168.0.255 is a broadcast address). Let's take a look at the rest of the information:

```
EthernetII,Src:192.168.0.2(00:14:22:be:18:3b),Dst:Broadcast(ff:ff:ff:ff:ff:ff)
```

This output shows that Bob knows his own MAC address and source IP address but not Sally's IP address or MAC address; so, Bob sends a broadcast address of all *f*s for the MAC address (a Data Link layer broadcast) and an IP LAN broadcast of 192.168.0.255. Again, no worries—you're going to learn all about broadcasts in Chapter 6, "Introduction to Internet Protocol (IP)."

Before the name is resolved, the first thing Bob has to do is broadcast on the LAN to get Sally's MAC address so he can communicate to her PC and resolve her name to an IP address:

```
Time      Source     Destination Protocol Info
5.153054  192.168.0.2 Broadcast  ARP Who has 192.168.0.3? Tell 192.168.0.2
```

Next, check out Sally's response:

```
Time      Source     Destination   Protocol  Info
5.153403 192.168.0.3 192.168.0.2  ARP  192.168.0.3 is at 00:0b:db:99:d3:5e
5.53.89317  192.168.0.3 192.168.0.2 NBNS Name query response NB 192.168.0.3
```

Okay, sweet—Bob now has both Sally's IP address and her MAC address. These are both listed as the source address at this point because this information was sent from Sally back to Bob. So, *finally*, Bob has all the goods he needs to communicate with Sally. And just so you know, I'm going to tell you all about Address Resolution Protocol (ARP) and show you exactly how Sally's IP address was resolved to a MAC address a little later in Chapter 6.

By the way, I want you to understand that Sally still had to go through the same resolution processes to communicate back to Bob—sounds crazy, huh? Consider this a welcome to IPv4 and basic networking with Windows—and we haven't even added a router yet.

Ethernet Basics

Ethernet is a contention media-access method that allows all hosts on a network to share the same bandwidth of a link. Ethernet is popular because it's readily scalable, meaning that it's comparatively easy to integrate new technologies, such as Fast Ethernet and Gigabit Ethernet, into an existing network infrastructure. It's also relatively simple to implement in the first place, and with it, troubleshooting is reasonably straightforward. Ethernet uses both Data Link and Physical layer specifications, and this section of the chapter will give you both the Data Link layer and Physical layer information you need to effectively implement, troubleshoot, and maintain an Ethernet network.

Collision Domain

The term *collision domain* is an Ethernet term that refers to a particular network scenario wherein one device sends a packet out on a network segment, thereby forcing every other device on that same physical network segment to pay attention to it. This is bad because if two devices on one physical segment transmit at the same time, a *collision event*—a situation where each device's digital signals interfere with another on the wire—occurs and forces the devices to retransmit later. Collisions have a dramatically negative effect on network performance, so they're definitely something we want to avoid!

The situation I just described is typically found in a hub environment where each host segment connects to a hub that represents only one collision domain and one broadcast domain. This begs the question, "What's a broadcast domain?"

Broadcast Domain

Here's that answer… A *broadcast domain* refers to the set of all devices on a network segment that hear all the broadcasts sent on that segment.

Even though a broadcast domain is typically a boundary delimited by physical media like switches and repeaters, it can also reference a logical division of a network segment where all hosts can reach each other via a Data Link layer (hardware address) broadcast.

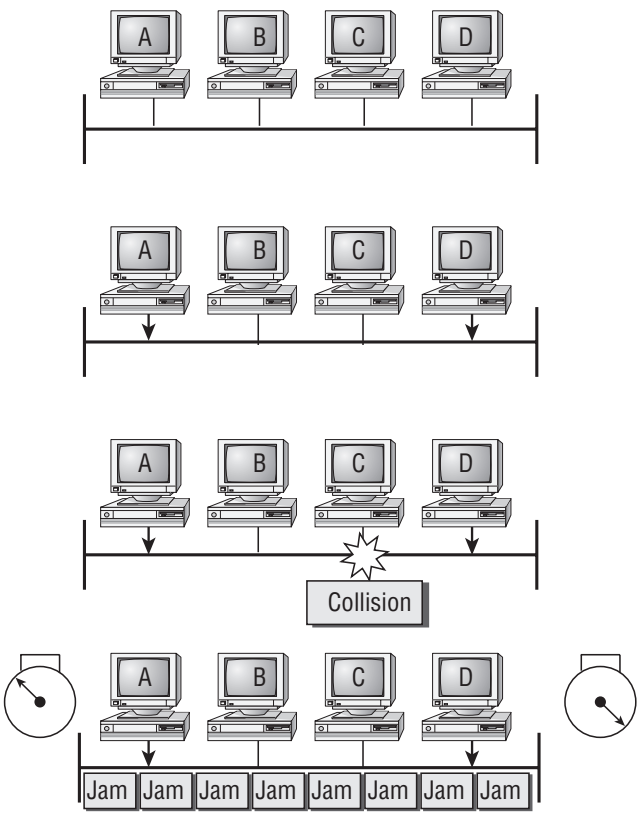
That's the basic story, but rest assured that I'll be delving deeper into the skinny on collision and broadcast domains a bit later in Chapter 5.

CSMA/CD

Ethernet networking uses *Carrier Sense Multiple Access with Collision Detection (CSMA/CD)*, a protocol that helps devices share the bandwidth evenly without having two devices transmit at the same time on the network medium. CSMA/CD was created to overcome the problem of those collisions that occur when packets are transmitted simultaneously from different hosts. And trust me—good collision management is crucial, because when a host transmits in a CSMA/CD network, all the other hosts on the network receive and examine that transmission. Only bridges and routers can effectively prevent a transmission from propagating throughout the entire network.

So, how does the CSMA/CD protocol work? Let's start by taking a look at Figure 4.2.

FIGURE 4.2 CSMA/CD



Carrier Sense Multiple Access with Collision Detection (CSMA/CD)

When a host wants to transmit over the network, it first checks for the presence of a digital signal on the wire. If all is clear, meaning that no other host is transmitting, the host will then proceed with its transmission. But it doesn't stop there. The transmitting host constantly monitors the wire to make sure no other hosts begin transmitting. If the host detects another signal on the wire, it sends out an extended jam signal that causes all hosts on the segment to stop sending data (think busy signal). The hosts respond to that jam signal by waiting a while before attempting to transmit again. Backoff algorithms determine when the colliding stations can retransmit. If collisions keep occurring after 15 tries, the hosts attempting to transmit will then time out. Pretty clean!

When a collision occurs on an Ethernet LAN, the following happens:

- A jam signal informs all devices that a collision occurred.
- The collision invokes a random backoff algorithm.
- Each device on the Ethernet segment stops transmitting for a short time until the timers expire.
- All hosts have equal priority to transmit after the timers have expired.

And following are the effects of having a CSMA/CD network sustaining heavy collisions:

- Delay
- Low throughput
- Congestion

Backoff on an 802.3 network is the retransmission delay that's enforced when a collision occurs. When a collision occurs, a host will resume transmission after the forced time delay has expired. After this backoff delay period has expired, all stations have equal priority to transmit data.

In the following sections, I'm going to cover Ethernet in detail at both the Data Link layer (Layer 2) and the Physical layer (Layer 1).

Half- and Full-Duplex Ethernet

Just so you know, half-duplex Ethernet is defined in the original 802.3 Ethernet specification. Basically, when you run half duplex, you're using only one wire pair with a digital signal either transmitting or receiving. This really isn't all that different from full duplex because you can both transmit and receive—you just don't get to do that at the same time running half duplex like you can if you're running full duplex.

Here's how it works: If a host hears a digital signal, it uses the CSMA/CD protocol to help prevent collisions and to permit retransmitting if a collision does occur. Half-duplex Ethernet—typically 10Base-T—is only about 30 to 40 percent efficient because a large 10Base-T network will usually provide only 3 to 4Mbps at most. Although it's true that 100Mbps Ethernet can and sometimes does run half duplex, it's just not very common to find that happening these days.

In contrast, full-duplex Ethernet uses two pairs of wires at the same time instead of one measly wire pair like half duplex employs. Plus, full duplex uses a point-to-point connection between the transmitter of the sending device and the receiver of the receiving device. This means that with full-duplex data transfer, you not only get faster data-transfer speeds, but you also get collision-prevention too—sweet!

You don't need to worry about collisions because now it's like a freeway with multiple lanes instead of the single-lane road provided by half duplex. Full-duplex Ethernet is supposed to offer 100 percent efficiency in both directions—for example, you can get 20Mbps with a 10Mbps Ethernet running full duplex or 200Mbps for Fast Ethernet. But this rate is something known as an *aggregate rate*, which translates as "you're supposed to get" 100 percent efficiency. No guarantees, in networking as in life.

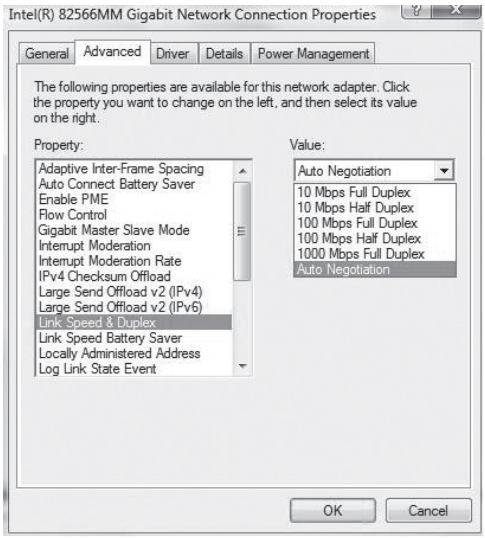
Full-duplex Ethernet can be used in many situations; here are some examples:

- With a connection from a switch to a host
- With a connection from a switch to a switch
- With a connection from a host to a host using a crossover cable

You can run full duplex with just about any device except a hub.

You may be wondering: If it's capable of all that speed, why wouldn't it deliver? Well, when a full-duplex Ethernet port is powered on, it first connects to the remote end and then negotiates with the other end of the Fast Ethernet link. This is called an *auto-detect mechanism*. This mechanism first decides on the exchange capability, which means it checks to see if it can run at 10, 100, or even 1000Mbps. It then checks to see if it can run full duplex; and if it can't, it will run half duplex instead.

Hosts usually auto-detect both the Mbps and the duplex type available (the default setting), but you can manually set both the speed and duplex type on the network interface card (NIC) card, as shown in the following graphic:



It is pretty rare these days to go into a NIC configuration on a host and change these settings, but this example shows that you can do that if you want.

Remember that half-duplex Ethernet shares a collision domain and provides a lower effective throughput than full-duplex Ethernet, which typically has a private collision domain and a higher effective throughput.

Last, remember these important points:

- There are no collisions in full-duplex mode.
- A dedicated switch port is required for each full-duplex host.
- The host network card and the switch port must be capable of operating in full-duplex mode.

Now let's take a look at how Ethernet works at the Data Link layer.

Ethernet at the Data Link Layer

Ethernet at the Data Link layer is responsible for Ethernet addressing, commonly referred to as *hardware addressing* or *MAC addressing*. Ethernet is also responsible for framing packets received from the Network layer and preparing them for transmission on the local network through the Ethernet contention media-access method.

Ethernet MAC addresses are made up of hexadecimal addresses. So before I discuss MAC addresses, let's start by talking about binary, decimal, and hexadecimal addresses and how to convert one to another.

Binary to Decimal and Hexadecimal Conversion

Understanding the differences between binary, decimal, and hexadecimal numbers and how to convert one format into the other is very important before we move to discussing the TCP/IP protocol stack and IP addressing in Chapter 6 and Chapter 7, "IP Addressing."

So let's get started with binary numbering. It's pretty simple, really. Each digit used is limited to either a 1 (one) or a 0 (zero), and each digit is called 1 bit (short for *bi*nary digi*t*). Typically, you count either 4 or 8 bits together, with these being referred to as a *nibble* and a *byte*, respectively.

What's interesting about binary numbering is the value represented in a decimal format—the typical decimal format being the base-10 number scheme that we've all used since kindergarten. The binary numbers are placed in a value spot, starting at the right and moving left, with each spot having double the value of the previous spot.

Table 4.1 shows the decimal values of each bit location in a nibble and a byte. Remember, a nibble is 4 bits and a byte is 8 bits.

TABLE 4.1 Binary Values

Nibble Values	Byte Values
8 4 2 1	128 64 32 16 8 4 2 1

What all this means is that if a one digit (1) is placed in a value spot, then the nibble or byte takes on that decimal value and adds it to any other value spots that have a 1. And if a zero (0) is placed in a bit spot, you don't count that value.

Let me clarify things for you—if we have a 1 placed in each spot of our nibble, we then add up 8 + 4 + 2 + 1, to give us a maximum value of 15. Another example for our nibble values is 1010, which means that the 8 bit and the 2 bit are turned on and equal a decimal value of 10. If we have a nibble binary value of 0110, then our decimal value is 6, because the 4 and 2 bits are turned on.

But the byte values can add up to a value that's significantly higher than 15. This is how—if we count every bit as a one (1), then the byte binary value looks like this (remember, 8 bits equal a byte):

11111111

We then count up every bit spot because each is turned on. It looks like this, which demonstrates the maximum value of a byte:

128 + 64 + 32 + 16 + 8 + 4 + 2 + 1 = 255

A binary number can equal plenty of other decimal values. Let's work through a few examples:

10010110

Which bits are on? The 128, 16, 4, and 2 bits are on, so we'll just add them up: 128 + 16 + 4 + 2 = 150.

01101100

Which bits are on? The 64, 32, 8, and 4 bits are on, so we add them up: 64 + 32 + 8 + 4 = 108.

11101000

Which bits are on? The 128, 64, 32, and 8 bits are on, so we add the values: 128 + 64 + 32 + 8 = 232.

You should memorize Table 4.2 before braving the IP sections in Chapters 6 and 7.

TABLE 4.2 Binary-to-Decimal Memorization Chart

Binary Value	Decimal Value
10000000	128
11000000	192
11100000	224

TABLE 4.2 Binary-to-Decimal Memorization Chart *(continued)*

Binary Value	Decimal Value
11110000	240
11111000	248
11111100	252
11111110	254
11111111	255

Hexadecimal addressing is completely different than binary or decimal—it's converted by reading nibbles, not bytes. By using a nibble, we can convert these bits to hex pretty simply. First, understand that the hexadecimal addressing scheme uses only the numbers 0 through 9. And because the numbers 10, 11, 12, and so on can't be used (because they are two-digit numbers), the letters *A*, *B*, *C*, *D*, *E*, and *F* are used to represent 10, 11, 12, 13, 14, and 15, respectively.

Hex is short for *hexadecimal*, which is a numbering system that uses the first six letters of the alphabet (*A* through *F*) to extend beyond the available 10 digits in the decimal system. Hexadecimal has a total of 16 digits.

Table 4.3 shows both the binary value and the decimal value for each hexadecimal digit.

TABLE 4.3 Hex-to-Binary to Decimal Chart

Hexadecimal Value	Binary Value	Decimal Value
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6

TABLE 4.3 Hex-to-Binary to Decimal Chart *(continued)*

Hexadecimal Value	Binary Value	Decimal Value
7	0111	7
8	1000	8
9	1001	9
A	1010	10
B	1011	11
C	1100	12
D	1101	13
E	1110	14
F	1111	15

Did you notice that the first 10 hexadecimal digits (0–9) are the same values as the decimal values? If not, look again. This handy fact makes those values super easy to convert.

So suppose you have something like this: 0x6A. (Some manufacturers put *0x* in front of characters so you know that they're a hex value, while others just give you an "*h*". It doesn't have any other special meaning.) What are the binary and decimal values? To correctly answer that question, all you have to remember is that each hex character is one nibble and two hex characters together make a byte. To figure out the binary value, first put the hex characters into two nibbles and then put them together into a byte. 6 = 0110 and A (which is 10 in hex) = 1010, so the complete byte is 01101010.

To convert from binary to hex, just take the byte and break it into nibbles. Here's how you do that:

Say you have the binary number 01010101. First, break it into nibbles—0101 and 0101—with the value of each nibble being 5 because the 1 and 4 bits are on. This makes the hex answer 0x55. And in decimal format, the binary number is 01010101, which converts to 64 + 16 + 4 + 1 = 85.

Okay, now try another binary number:

11001100

Our answer is 1100 = 12 and 1100 = 12 (therefore, it's converted to CC in hex). The decimal conversion answer is 128 + 64 + 8 + 4 = 204.

One more example, and then we need to get working on the Physical layer. Suppose we're given the following binary number:

10110101

The hex answer is 0xB5, because 1011 converts to B and 0101 converts to 5 in hex value. The decimal equivalent is 128 + 32 + 16 + 4 + 1 = 181.

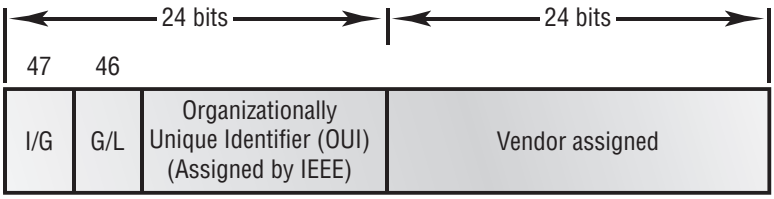
See the Written Lab for more practice with binary/hex/decimal conversion.

Ethernet Addressing

Now that you've got binary to decimal to hexadecimal address conversion down, we can get into how Ethernet addressing works. It uses the *Media Access Control (MAC) address* burned into each and every Ethernet NIC. The MAC, or hardware, address is a 48-bit (6-byte) address written in a hexadecimal format.

Figure 4.3 shows the 48-bit MAC addresses and how the bits are divided.

FIGURE 4.3 Ethernet addressing using MAC addresses



The *organizationally unique identifier (OUI)* is assigned by the Institute of Electrical and Electronics Engineers (IEEE) to an organization. It's composed of 24 bits, or 3 bytes. The organization, in turn, assigns a globally administered address (24 bits, or 3 bytes) that is unique (supposedly—no guarantees) to each and every adapter it manufactures. Look closely at the figure. The high-order bit is the Individual/Group (I/G) bit. When it has a value of 0, we can assume that the address is the MAC address of a device and may well appear in the source portion of the MAC header. When it is a 1, we can assume that the address represents either a broadcast or multicast address in Ethernet or a broadcast.

The next bit is the Global/Local bit (G/L, also known as U/L, where *U* means *universal*). When set to 0, this bit represents a globally administered address (as standardized by the IEEE). When the bit is a 1, it represents a locally governed and administered address. The low-order 24 bits of an Ethernet address represent a locally administered or manufacturer-assigned code. This portion commonly starts with 24 0s for the first card made and continues in order until there are 24 1s for the last (16,777,216th) card made. You'll find that many manufacturers use these same six hex digits as the last six characters of their serial number on the same card.

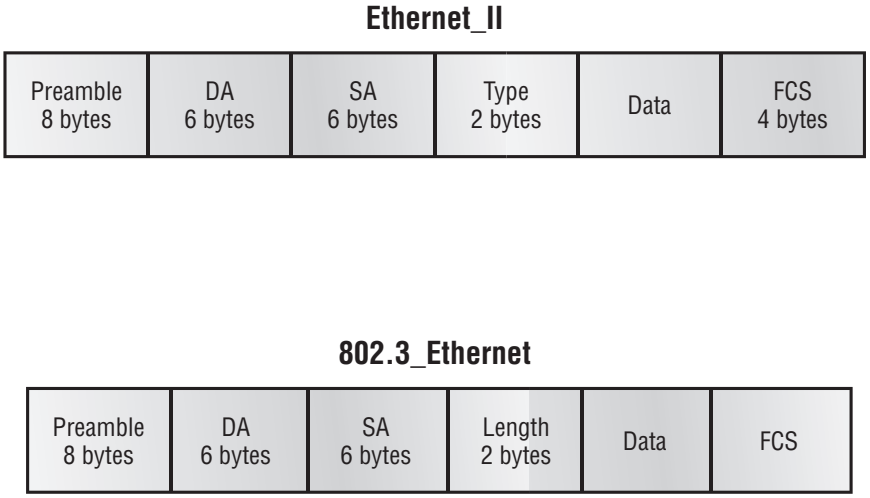
Ethernet Frames

The Data Link layer is responsible for combining bits into bytes and bytes into frames. Frames are used at the Data Link layer to encapsulate packets handed down from the Network layer for transmission on a type of physical media access.

The function of Ethernet stations is to pass data frames between each other using a group of bits known as a MAC frame format. This provides error detection from a cyclic redundancy check (CRC). But remember—this is error detection, not error correction. The 802.3 frames and Ethernet frame are shown in Figure 4.4.

Encapsulating a frame within a different type of frame is called *tunneling*.

FIGURE 4.4 802.3 and Ethernet frame formats



Following are the details of the different fields in the 802.3 and Ethernet frame types:

The following section regarding frame headings and the various types of Ethernet frames are beyond the CompTIA Network+ objectives. Throughout the rest of this book, I will show you screen shots from a network analyzer. It's always good to understand what you are looking at, so I put this section in to help you understand a frame structure.

Preamble An alternating 1,0 pattern provides a 5MHz clock at the start of each packet, which allows the receiving devices to lock the incoming bit stream.

Start Frame Delimiter (SFD)/Synch The preamble is seven octets, and the SFD is one octet (synch). The SFD is 10101011, where the last pair of 1s allows the receiver to come into the alternating 1,0 pattern somewhere in the middle and still sync up and detect the beginning of the data (this field is not shown in the figure).

Destination Address (DA) This transmits a 48-bit value using the least significant bit (LSB) first. The DA is used by receiving stations to determine whether an incoming packet is addressed to a particular host. The DA can be an individual address or a broadcast or

multicast MAC address. Remember that a broadcast is all 1s (or *F*s in hex) and is sent to all devices, but a multicast is sent only to a similar subset of hosts on a network.

Source Address (SA) The SA is a 48-bit MAC address used to identify the transmitting device, and it uses the LSB first. Broadcast and multicast address formats are illegal within the SA field.

Length or Type 802.3 uses a Length field, but the Ethernet frame uses a Type field to identify the Network layer protocol. 802.3 by itself cannot identify the upper-layer routed protocol and must be used with a proprietary LAN—Internetwork Packet Exchange (IPX), for example.

Data This is a packet sent down to the Data Link layer from the Network layer. The size can vary from 64 to 1500 bytes.

Frame Check Sequence (FCS) FCS is a field at the end of the frame that's used to store the CRC.

Okay—let's take a minute to look at some frames caught on our trusty network analyzer. You can see that the following frame has only three fields: Destination, Source, and Type (shown as Protocol Type on this analyzer):

```
Destination:   00:60:f5:00:1f:27
Source:        00:60:f5:00:1f:2c
Protocol Type: 08-00 IP
```

This is an Ethernet_II frame. Notice that the type field is IP, or 08-00 (mostly just referred to as 0x800) in hexadecimal.

The next frame has the same fields, so it must be an Ethernet_II frame too:

```
Destination:   ff:ff:ff:ff:ff:ff Ethernet Broadcast
Source:        02:07:01:22:de:a4
Protocol Type: 08-00 IP
```

Did you notice that this frame was a broadcast? You can tell because the destination hardware address is all 1s in binary, or all *F*s in hexadecimal.

Let's take a look at one more Ethernet_II frame. I'll talk about this next example again when we use IPv6 in Chapter 6, but you can see that the Ethernet frame is the same Ethernet_II frame we use with the IPv4 routed protocol. The difference is that the Type field has 0x86dd when we are carrying IPv6 data; and when we have IPv4 data, we use 0x0800 in the Protocol field:

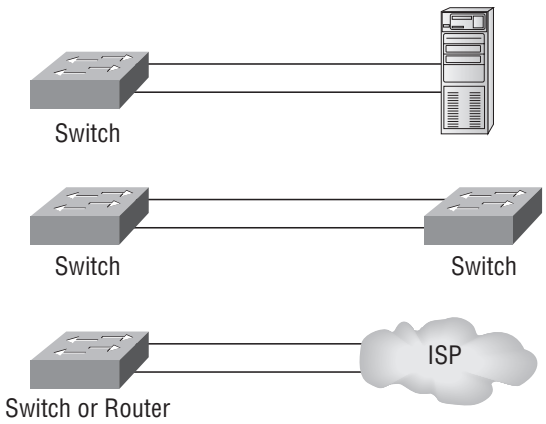
```
Destination: IPv6-Neighbor-Discovery_00:01:00:03 (33:33:00:01:00:03)
Source: Aopen_3e:7f:dd (00:01:80:3e:7f:dd)
Type: IPv6 (0x86dd)
```

This is the beauty of the Ethernet_II frame. Because of the Protocol field, we can run any Network layer routed protocol and it will carry the data because it can identify that particular Network layer protocol.

Channel Bonding

Channel bonding (also known as *Ethernet bonding*) is a computer-networking arrangement where two or more network interfaces on a host are combined for redundancy and/or increased throughput. There are various names for this technology, but Link Aggregation is the most common. Cisco calls this EtherChannel. Figure 4.5 shows some examples of Ethernet channel bonding.

FIGURE 4.5 Ethernet channel bonding example



In Figure 4.5, you can see that bonding can be used to attach multiple connections to a server, between switches, and even for connections to the Internet, providing fault tolerance as well as improved throughput.

Ethernet at the Physical Layer

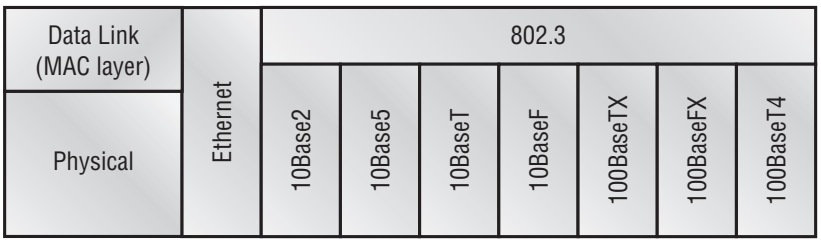
Ethernet was first implemented by a group called DIX (Digital, Intel, and Xerox). They created and implemented the first Ethernet LAN specification, which the IEEE used to create the IEEE 802.3 Committee. This was a 10Mbps network that ran on coax, then on twisted-pair, and finally on fiber physical media.

The IEEE extended the 802.3 Committee to two new committees known as 802.3u (Fast Ethernet) and 802.3ab (Gigabit Ethernet on Category 5+) and then finally 802.3ae (10Gbps over fiber and coax).

Figure 4.6 shows the IEEE 802.3 and original Ethernet Physical layer specifications.

When designing your LAN, it's really important to understand the different types of Ethernet media available to you. Sure, it would be great to run Gigabit Ethernet to each desktop and 10Gbps between switches (and to servers), and although this is just starting to happen, justifying the cost of that network today for most companies would be a pretty hard sell. But if instead, you mix and match the different types of Ethernet media methods currently available, you can come up with a cost-effective network solution that works great.

FIGURE 4.6 Ethernet Physical layer specifications



The Electronic Industries Association and the newer Telecommunications Industry Alliance (EIA/TIA) is the standards body that creates the Physical layer specifications for Ethernet. The EIA/TIA specifies that Ethernet use a *Registered Jack (RJ) connector* with a 4 5 wiring sequence on *unshielded twisted pair (UTP)* cabling (RJ-45). However, the industry is calling this just an 8-pin modular connector.

Each Ethernet cable type that is specified by the EIA/TIA has something known as *inherent attenuation*, which is defined as the loss of signal strength as it travels the length of a cable, and is measured in decibels (dB). The cabling used in corporate and home markets is measured in categories. A higher-quality cable will have a higher-rated category and lower attenuation. For example, Category 5 is better than Category 3 because Category 5 cables have more wire twists per foot and therefore less crosstalk. *Crosstalk* is the unwanted signal interference from adjacent pairs in the cable.

Here are the original IEEE 802.3 standards:

10Base-2 10Mbps baseband technology, up to 185 meters in length. Known as *Thinnet* and can support up to 30 workstations on a single segment. Uses a physical and logical bus with Attachment Unit Interface (AUI) connectors. The 10 means 10Mbps, *Base* means baseband technology—a signaling method for communication on the network—and the 2 means almost 200 meters. 10Base-2 Ethernet cards use BNC (British Naval Connector, Bayonet Neill-Concelman, or Bayonet Nut Connector) and T-connectors to connect to a network.

10Base-5 10Mbps baseband technology, up to 500 meters in length. Known as *Thicknet*. Uses a physical and logical bus with AUI connectors. Up to 2,500 meters with repeaters and 1,024 users for all segments.

10Base-T 10Mbps using Category 3 UTP wiring. Unlike on 10Base-2 and 10Base-5 networks, each device must connect into a hub or switch, and you can have only one host per segment or wire. Uses an RJ-45 connector (8-pin modular connector) with a physical star topology and a logical bus.

Each of the 802.3 standards defines an AUI, which allows a one-bit-at-a-time transfer to the Physical layer from the Data Link media-access method. This allows the MAC address to remain constant but means the Physical layer can support both existing and new technologies. The original AUI interface was a 15-pin connector, which allowed a transceiver (transmitter/receiver) that provided a 15-pin-to-twisted-pair conversion.

There's an issue, though—the AUI interface can't support 100Mbps Ethernet because of the high frequencies involved. So basically, 100Base-T needed a new interface, and

the 802.3u specifications created one called the Media Independent Interface (MII) that provides 100Mbps throughput. The MII uses a *nibble*, which you of course remember is defined as 4 bits. Gigabit Ethernet uses a Gigabit Media Independent Interface (GMII) and transmits 8 bits at a time.

802.3u (Fast Ethernet) is compatible with 802.3 Ethernet because they share the same physical characteristics. Fast Ethernet and Ethernet use the same maximum transmission unit (MTU) and the same MAC mechanisms, and they both preserve the frame format that is used by 10Base-T Ethernet. Basically, Fast Ethernet is just based on an extension to the IEEE 802.3 specification, and because of that, it offers us a speed increase of 10 times 10Base-T.

Here are the expanded IEEE Ethernet 802.3 standards, starting with Fast Ethernet:

100Base-TX (IEEE 802.3u) 100Base-TX, most commonly known as Fast Ethernet, uses EIA/TIA Category 5, 5E, or 6, UTP two-pair wiring. One user per segment; up to 100 meters long (328 feet). It uses an RJ-45 connector with a physical star topology and a logical bus.

100Base-FX (IEEE 802.3u) Uses fiber cabling 62.5/125-micron multimode fiber. Point-to-point topology; up to 412 meters long. It uses ST and SC connectors, which are media-interface connectors.

Ethernet 's implementation over fiber can sometimes be referred to as 100Base-TF, although this isn't an actual standard. It just means that Ethernet technologies are being run over fiber cable.

1000Base-CX (IEEE 802.3z) Copper twisted-pair called twinax (a balanced coaxial pair) that can run only up to 25 meters and uses a special 9-pin connector known as the High Speed Serial Data Connector (HSSDC).

1000Base-T (IEEE 802.3ab) Category 5, four-pair UTP wiring up to 100 meters long (328 feet).

1000Base-SX (IEEE 802.3z) The implementation of Gigabit Ethernet running over multimode fiber-optic cable (instead of copper twisted-pair cable) and using short wavelength laser. Multimode fiber (MMF) using 62.5- and 50-micron core; uses an 850 nanometer (nm) laser and can go up to 220 meters with 62.5-micron, 550 meters with 50-micron.

1000Base-LX (IEEE 802.3z) Single-mode fiber that uses a 9-micron core and 1300 nm laser and can go from 3 km up to 10 km.

10GBase-T 10GBase-T is a standard proposed by the IEEE 802.3an committee to provide 10Gbps connections over conventional UTP cables (Category 5e, 6, or 7 cables). 10GBase-T allows the conventional RJ-45 used for Ethernet LANs. It can support signal transmission at the full 100-meter distance specified for LAN wiring.

10GBase-SR An implementation of 10 Gigabit Ethernet that uses short-wavelength lasers at 850 nm over multimode fiber. It has a maximum transmission distance of between 2 and 300 meters (990 feet), depending on the size and quality of the fiber.

10GBase-LR An implementation of 10 Gigabit Ethernet that uses long-wavelength lasers at 1,310 nm over single-mode fiber. It also has a maximum transmission distance between 2 meters and 10 km (which is 6 miles!), depending on the size and quality of the fiber.

10GBase-ER An implementation of 10 Gigabit Ethernet running over single-mode fiber. It uses extra-long-wavelength lasers at 1,550 nm. It has the longest transmission distances possible of the 10-Gigabit technologies: anywhere from 2 meters up to 40 km, depending on the size and quality of the fiber used.

10GBase-SW 10GBase-SW, as defined by IEEE 802.3ae, is a mode of 10GBase-S for MMF with a 850 nm laser transceiver with a bandwidth of 10Gbps. It can support up to 300 meters of cable length. This media type is designed to connect to SONET equipment.

10GBase-LW 10GBase-LW is a mode of 10GBase-L supporting a link length of 10 km on standard single-mode fiber (SMF) (G.652). This media type is designed to connect to SONET equipment.

10GBase-EW 10GBase-EW is a mode of 10GBase-E supporting a link length of up to 40 km on SMF based on G.652 using optical-wavelength 1550 nm. This media type is designed to connect to SONET equipment.

If you want to implement a network medium that is not susceptible to electromagnetic interference (EMI), fiber-optic cable provides a more secure, long-distance cable that is not susceptible to EMI at high speeds like UTP is.

Table 4.4 summarizes the cable types.

TABLE 4.4 Common Ethernet Cable Types

Ethernet Name	Cable Type	Maximum Speed	Maximum Transmission Distance	Notes
10Base-5	Coax	10Mbps	500 meters per segment	Also called Thicknet, this cable type uses vampire taps to connect devices to cable.
10Base-2	Coax	10Mbps	185 meters per segment	Also called Thinnet, a very popular implementation of Ethernet over coax.
10Base-T	UTP	10Mbps	100 meters per segment	One of the most popular network cabling schemes.
100Base-TX	UTP, STP	100Mbps	100 meters per segment	Two pairs of Category 5 UTP.

TABLE 4.4 Common Ethernet Cable Types *(continued)*

Ethernet Name	Cable Type	Maximum Speed	Maximum Transmission Distance	Notes
10Base-FL	Fiber	10Mbps	Varies (ranges from 500 meters to 2,000 meters)	Ethernet over fiber optics to the desktop.
100Base-FX	MMF	100Mbps	2,000 meters	100Mbps Ethernet over fiber optics.
1000Base-T	UTP	1000Mbps	100 meters	Four pairs of Category 5e or higher.
1000Base-SX	MMF	1000Mbps	550 meters	Uses SC fiber connectors. Max length depends on fiber size.
1000Base-CX	Balanced, shielded copper	1000Mbps	25 meters	Uses a special connector, the HSSDC.
1000Base-LX	MMF and SMF	1000Mbps	550 meters multimode/ 2000 meters single mode	Uses longer wavelength laser than 1000Base-SX. Uses SC and LC connectors.
10GBase-T	UTP	10Gbps	100 meters	Connects to the network like a Fast Ethernet link using UTP.
10GBase-SR	MMF	10Gbps	300 meters	850 nm laser. Max length depends on fiber size and quality.
10GBase-LR	SMF	10Gbps	10 kilometers	1310 nm laser. Max length depends on fiber size and quality.
10GBase-ER	SMF	10Gbps	40 kilometers	1550 nm laser. Max length depends on fiber size and quality.
10GBase-SW	MMF	10Gbps	300 meters	850 nm laser transceiver.

TABLE 4.4 Common Ethernet Cable Types *(continued)*

Ethernet Name	Cable Type	Maximum Speed	Maximum Transmission Distance	Notes
10GBase-LW	SMF	10Gbps	10 kilometers	Typically used with SONET.
10GBase-EW	SMF	10Gbps	40 kilometers	1550 nm optical wavelength.

An advantage of 100Base-FX over 100Base-TX is longer cable runs, however, 100Base-TX is easier to install.

I know there's a lot of information to remember about the various Ethernet and fiber types used in today's networks, but for the CompTIA Network+ exam, you really need to know them. Trust me, I haven't inundated you with unnecessary information!

Armed with the basics covered in the chapter, you're equipped to go to the next level and put Ethernet to work using various network devices. But to ensure you're really ready, read the summary, go over the Exam Essentials and do the Written Lab and Review Questions for this chapter.

Summary

In this chapter, you learned the fundamentals of Ethernet networking, how hosts communicate on a network, as well as how CSMA/CD works in an Ethernet half-duplex network.

I also talked about the differences between half- and full-duplex modes and discussed how Ethernet channel bonding can be used to attach multiple connections between Ethernet devices.

I finished the chapter with a description of the common Ethernet cable types used in today's networks. And by the way, you'd be wise to study that section really well!

Exam Essentials

Understand basic Ethernet communication. Know how hosts use hardware addresses to communicate on an Ethernet LAN.

Understand Ethernet addressing. Know the hexadecimal addressing scheme used to create an Ethernet address.

Understand binary, decimal, and hexadecimal addressing. Know the different addressing types, and also use the Written Lab to practice your conversions.

Understand the basic definition of channel bonding. Know the various ways you can use channel bonding to make your network more resilient and add bandwidth between devices.

Written Lab

1. Convert from decimal IP address to binary format.

Complete the following table to express 192.168.10.15 in binary format.

128	64	32	16	8	4	2	1	Binary

Complete the following table to express 172.16.20.55 in binary format.

128	64	32	16	8	4	2	1	Binary

Complete the following table to express 10.11.12.99 in binary format.

128	64	32	16	8	4	2	1	Binary

2. Convert the following from binary format to decimal IP address.

   Complete the following table to express 11001100.00110011.10101010.01010101 in decimal IP address format.

128	64	32	16	8	4	2	1	Decimal

   Complete the following table to express 11000110.11010011.00111001.11010001 in decimal IP address format.

128	64	32	16	8	4	2	1	Decimal

   Complete the following table to express 10000100.11010010.10111000.10100110 in decimal IP address format.

128	64	32	16	8	4	2	1	Decimal

3. Convert the following from binary format to hexadecimal.

Complete the following table to express 11011000.00011011.00111101.01110110 in hexadecimal.

128	64	32	16	8	4	2	1	Hexadecimal

Complete the following table to express 11001010.11110101.10000011.11101011 in hexadecimal.

128	64	32	16	8	4	2	1	Hexadecimal

Complete the following table to express 10000100.11010010.01000011.10110011 in hexadecimal.

128	64	32	16	8	4	2	1	Hexadecimal

Review Questions

1. On an Ethernet switched network, what address does one host computer use to communicate with another?
   - **A.** IP address
   - **B.** MAC address
   - **C.** Street address
   - **D.** HUB address
2. Which of the following can run full-duplex and achieve 200Mbps with CAT5e cable?
   - **A.** 100Base-F
   - **B.** 100Base-T
   - **C.** 1000Base-F
   - **D.** 1000Base-T
3. How many devices in a collision domain have to listen when a single host talks?
   - **A.** 2
   - **B.** 3
   - **C.** 1
   - **D.** All
4. If you are using a cable medium called 100Base-TF, what does this mean?
   - **A.** That you are running Ethernet over cable
   - **B.** Ethernet over fiber
   - **C.** Ethernet over ThickNet
   - **D.** That you are bundling multiple connections
5. What protocol helps devices share the bandwidth evenly without having two devices transmit at the same time on the network medium?
   - **A.** TCP/IP
   - **B.** CSMA/CD
   - **C.** HTTPS
   - **D.** TFTP
6. What is the maximum distance of 10GBase-SR?
   - **A.** 100 meters (328 feet)
   - **B.** 302 meters (990 feet)
   - **C.** 305 meters (1000 feet)
   - **D.** 1593 km (6 miles)

7. How many wire pairs are used with half duplex?
   - **A.** 2
   - **B.** 1
   - **C.** 4
   - **D.** None of the above

8. How many wire pairs are used with 100Base-T full duplex?
   - **A.** 2
   - **B.** 1
   - **C.** 4
   - **D.** A or C

9. What is the maximum distance of 100GBase-LR?
   - **A.** 1 mile
   - **B.** 3 miles
   - **C.** 6 miles
   - **D.** 25 miles

10. What is the effective total throughput increase with a full-duplex connection?
    - **A.** None
    - **B.** Twice as much
    - **C.** Four times as much
    - **D.** Ten times as much

11. What device can you not use full-duplex communication with?
    - **A.** Host
    - **B.** Hub
    - **C.** Switch
    - **D.** Router

12. What is the decimal equivalent of this binary number: 11000000.10101000.00110000.11110000?
    - **A.** 192.168.48.192
    - **B.** 192.168.48.240
    - **C.** 192.168.64.224
    - **D.** 192.168.32.248

13. Which technology increases the bandwidth for network transmission by joining together multiple connections in one logical connection?
    - **A.** Bonding
    - **B.** VLANs
    - **C.** STP
    - **D.** Traffic Shaping

14. How is the decimal value 10 represented in binary?
    - **A.** 1000
    - **B.** 1001
    - **C.** 1010
    - **D.** 1011

15. What is the decimal value for the binary number 11101000?
    - **A.** 128
    - **B.** 194
    - **C.** 224
    - **D.** 232

16. What is the decimal number 10 in hexadecimal?
    - **A.** 9
    - **B.** A
    - **C.** C
    - **D.** B

17. How many bits is a MAC address?
    - **A.** 16
    - **B.** 32
    - **C.** 48
    - **D.** 64

18. The maximum distance of 1000Base-T is?
    - **A.** 100 Meters (328 feet)
    - **B.** 128 meters (420 feet)
    - **C.** 1000 meters (3280 feet)
    - **D.** 1024 meters (3360 feet)

19. What is the purpose of the Frame Check Sequence (FCS) in an Ethernet frame?

A. Error correction

B. Error detection

C. Error recovery

D. Creating errors

20. What does the Base mean in 100Base-TX?

A. Broadband

B. 100Mbps

C. Baseband

D. Twisted pair at 100Mbps

Answers to Review Questions

1. B. On an Ethernet Network, the MAC address (hardware address) is used for one host to communicate with another.
2. B. 100Base-T uses CAT5e and can run 200Mbps when using full-duplex. 100Base-TX is only CAT-5, not CAT5e
3. D. When one device sends a packet out on a network segment, all other devices on the same physical network segment must wait and let it be transmitted.
4. B. 100Base-TF means that you have an Ethernet over fiber cable implementation.
5. B. Carrier Sense Multiple Access with Collision Detection (CSMA/CD) helps packets that are transmitted simultaneously from different hosts share bandwidth evenly.
6. B. A 10GBase-SR cable can have a maximum distance of 990 feet (302 meters).
7. B. With half duplex, you are using only one wire pair with a digital signal either transmitting or receiving.
8. A. Full-duplex Ethernet uses two pairs of wires at the same time.
9. C. A 10GBase-LR implementation can go a distance of up to 6 miles
10. B. Double up! You can get 20Mbps with a 10Mbps Ethernet running full duplex or 200Mbps for Fast Ethernet.
11. B. Full-duplex communication cannot be used with a hub, because a hub is a half-duplex single communication device. A host, switch and router have the ability to process traffic (frames), whereas a hub is a multiport repeater.
12. B. 11000000 is 192, 10101000 is 168, 00110000 is 48, and 11110000 is 240.
13. A. Bonding can increase bandwidth and provide redundancy for devices that have multiple links connected together.
14. C. Nibble values are 8 + 4 + 2 + 1, giving us a maximum value of 15. If we have a decimal value of 10, that means the 8 bit and the 2 bit are turned on.
15. D. The 128, 64, 32, and 8 bits are on, so just add the values: 128 + 64 + 32 + 8 = 232.
16. B. The first 10 hexadecimal digits (0–9) are the same values as the decimal values. We already know the binary value for the number 10 is 1010—in hex, the number 10 needs to be displayed as a single character. To display double-digit numbers as a single character, we substitute letters. In our example, 10 is A.
17. C. A MAC, or hardware, address is a 48-bit (6-byte) address written in hexadecimal format.

18. A. 100Base-T and 1000Base-T both have a maximum distance of 100 meters, or 328 feet.

19. B . The FCS can detect frames in the sequence by calculating the cyclic redundancy check (CRC), which verifies that all the bits in the frame are unchanged.

20. C. The 100 means 100Mbps. The Base means baseband, which means baseband technology—a signaling method for communication on the network.

Answers to Written Lab

1. Convert from decimal IP address to binary format.

Complete the following table to express 192.168.10.15 in binary format.

Decimal	128	64	32	16	8	4	2	1	Binary
192	1	1	0	0	0	0	0	0	11000000
168	1	0	1	0	1	0	0	0	10101000
10	0	0	0	0	1	0	1	0	00001010
15	0	0	0	0	1	1	1	1	00001111

Complete the following table to express 172.16.20.55 in binary format.

Decimal	128	64	32	16	8	4	2	1	Binary
172	1	0	1	0	1	1	0	0	10101100
16	0	0	0	1	0	0	0	0	00010000
20	0	0	0	1	0	1	0	0	00010100
55	0	0	1	1	0	1	1	1	00110111

Complete the following table to express 10.11.12.99 in binary format.

Decimal	128	64	32	16	8	4	2	1	Binary
10	0	0	0	0	1	0	1	0	00001010
11	0	0	0	0	1	0	1	1	00001011
12	0	0	0	0	1	1	0	0	00001100
99	0	1	1	0	0	0	1	1	01100011

2. Convert the following from binary format to decimal IP address.

Complete the following table to express 11001100.00110011.10101010.01010101 in decimal IP address format.

Binary	128	64	32	16	8	4	2	1	Decimal
11001100	1	1	0	0	1	1	0	0	204
00110011	0	0	1	1	0	0	1	1	51
10101010	1	0	1	0	1	0	1	0	170
01010101	0	1	0	1	0	1	0	1	85

Complete the following table to express 11000110.11010011.00111001.11010001 in decimal IP address format.

Binary	128	64	32	16	8	4	2	1	Decimal
11000110	1	1	0	0	0	1	1	0	198
11010011	1	1	0	1	0	0	1	1	211
00111001	0	0	1	1	1	0	0	1	57
11010001	1	1	0	1	0	0	0	1	209

Complete the following table to express 10000100.11010010.10111000.10100110 in decimal IP address format.

Binary	128	64	32	16	8	4	2	1	Decimal
10000100	1	0	0	0	0	1	0	0	132
11010010	1	1	0	1	0	0	1	0	210
10111000	1	0	1	1	1	0	0	0	184
10100110	1	0	1	0	0	1	1	0	166

3. Convert the following from binary format to hexadecimal.

   Complete the following table to express 11011000.00011011.00111101.01110110 in hexadecimal.

Binary	128	64	32	16	8	4	2	1	Hexadecimal
11011000	1	1	0	1	1	0	0	0	D8
00011011	0	0	0	1	1	0	1	1	1B
00111101	0	0	1	1	1	1	0	1	3D
01110110	0	1	1	1	0	1	1	0	76

   Complete the following table to express 11001010.11110101.10000011.11101011 in hexadecimal.

Binary	128	64	32	16	8	4	2	1	Hexadecimal
11001010	1	1	0	0	1	0	1	0	CA
11110101	1	1	1	1	0	1	0	1	F5
10000011	1	0	0	0	0	0	1	1	83
11101011	1	1	1	0	1	0	1	1	EB

   Complete the following table to express 10000100.11010010.01000011.10110011 in hexadecimal.

Binary	128	64	32	16	8	4	2	1	Hexadecimal
10000100	1	0	0	0	0	1	0	0	84
11010010	1	1	0	1	0	0	1	0	D2
01000011	0	1	0	0	0	0	1	1	43
10110011	1	0	1	1	0	0	1	1	B3