Chapter 8

TCP/IP Internetworking I

LEARNING OBJECTIVES

By the end of this chapter, you should be able to:

- Define hierarchical IP addresses, networks and subnets, border and internal routers, and masks.
- Given a routing table and an arriving packet's destination IP address, explain what the router will do with the packet.
- (In a box) Explain the purpose and operation of the Address Resolution Protocol on an Ethernet network.
- Explain the IPv4 packet header fields we did not see in earlier chapters.
- Explain the IPv6 packet header fields and IPv6's use of extension headers.
- Convert a 128-bit IP address into compressed hexadecimal notation.
- Explain TCP fields and session closings.
- Explain why application message fragmentation is not possible with UDP.

INTRODUCTION

Switched networks and wireless networks are governed by Layer 1 and Layer 2 standards. We looked at single network standards in Chapters 5, 6, and 7. In this chapter and the next, we will look at internetworking, which is governed by Layer 3 and Layer 4 standards.

We will only look at TCP/IP internetworking because TCP/IP dominates the work of network professionals at the internet and transport layers. However, real-world routers cannot limit themselves to TCP/IP internetworking. Commercial routers are multiprotocol routers, which can route not only IP packets but also IPX packets, SNA packets, AppleTalk packets, and other types of packets.

M08_PANK4015_10_SE_C08.indd 283 07/08/14 8:17 PM

<table>
<tr><td rowspan="2">5 Application</td><td colspan="3">User Applications</td><td colspan="3">Supervisory Applications</td></tr>
<tr><td>HTTP</td><td>SMTP</td><td>Many Others</td><td>DNS</td><td>Dynamic Routing Protocols</td><td>Many Others</td></tr>
<tr><td>4 Transport</td><td colspan="3">TCP</td><td colspan="3">UDP</td></tr>
<tr><td>3 Internet</td><td colspan="3">IP</td><td colspan="2">ICMP</td><td>ARP</td></tr>
<tr><td>2 Data Link</td><td colspan="6">None: Use OSI Standards</td></tr>
<tr><td>1 Physical</td><td colspan="6">None: Use OSI Standards</td></tr>
</table>

Note: Shaded protocols are discussed in this chapter.

FIGURE 8-1 Major TCP/IP Standards

We looked at the TCP/IP architecture in Chapters 1 and 2. We focused on IP, TCP, and UDP, although we looked at a few other TCP/IP standards. Figure 8-1 shows a few of the many standards the IETF has created within the TCP/IP architecture. Some of the standards are shaded in this figure. We will look at them in this chapter.

IP ROUTING

In this section, we will look at how routers make decisions about forwarding packets—in other words, how a router decides which interface to use to send an arriving packet back out to get it closer to its destination. (In routers, ports are called **interfaces**.)

Router ports are called interfaces.

This forwarding process is called **routing**. Router forwarding decisions are much more complex than the Ethernet switching decisions we saw in Chapter 5. Because of this complexity, routers do more work per arriving packet than switches do per arriving frame. Consequently, routers are more expensive than switches for a given level of traffic. A widely quoted network adage reflects this cost difference: "Switch where you can; route where you must."

When routers forward incoming packets closer to their destination hosts, this is routing.

Test Your Understanding

1. a) What are interfaces? b) What is routing?

Hierarchical IP Addressing

To understand the routing of IP packets, it is necessary to understand IP addresses. In Chapter 1, we saw that IP Version 4 (IPv4) addresses are 32 bits long. However, IP addresses are not simple 32-bit strings. They have internal structure, and this internal structure is important in routing.

Hierarchical Addressing As Figure 8-2 shows, IP addresses are **hierarchical**. They usually consist of three parts that locate a host in progressively smaller parts of the Internet. These are the network, subnet, and host parts. We will see later in this chapter how hierarchical IP addressing simplifies routing tables.

Network Part First, every IP address has a **network part**, which identifies the host's network on the Internet. In this case, *network* is an organizational concept. It is a user organization, such as a manufacturing corporation, or it is an ISP. Whichever organization receives the network part effectively controls part of the Internet.

In IP addressing, network is an organizational concept. A network is an organization that controls part of the Internet. It may be a user organization such as a manufacturing corporation or an ISP.

In Figure 8-2, the network part is 128.171. This is two IP address segments. Each segment is 8 bits long, so the network part for the University of Hawai`i is 16 bits long. All host IP addresses in the university begin with this network part.

Do not get hung up on the network part being 16 bits. This is only an example. Different organizations have different network parts that range from 8 bits to 24 bits in length.

Subnet Part Most large organizations further divide their networks into smaller units called **subnets**. After the network part in an IP address come the bits of the **subnet part**. The subnet part bits specify a particular subnet within the network.

For instance, Figure 8-2 shows that in the IP address 128.171.17.13, the first 16 bits (128.171) correspond to the network part, and the next 8 bits (17) correspond to a subnet on this network. (Subnet 17 is the Shidler College of Business subnet within the University of Hawai`i Network.) All host IP addresses within this subnet begin with 128.171.17.

Again, do not get hung up on the subnet part being 8 bits long. In different organizations, subnet lengths vary widely. Keep clear in your head that the UH Network is only being used as an example.

Host Part The remaining bits in the 32-bit IP address constitute the **host part**, which specifies a particular host on the subnet. In Figure 8-2, the host part is 8 bits long with a segment value of 13. This corresponds to a particular host, 128.171.17.13, on the Shidler College of Business subnet of the University of Hawai`i Network. Again, host parts in different organizations differ in length.

Variable Part Lengths Can you tell just by looking at an IP address which bits correspond to the network, subnet, and host parts? The answer is no. For instance, if you see the IP address 60.47.7.23, you may have an 8-bit network part of 60, an 8-bit subnet part of 47, and a 16-bit host part of 7.23. In fact, parts may not even break conveniently at 8-bit boundaries. The only thing you can tell when looking at an IP address is that it is 32 bits long.

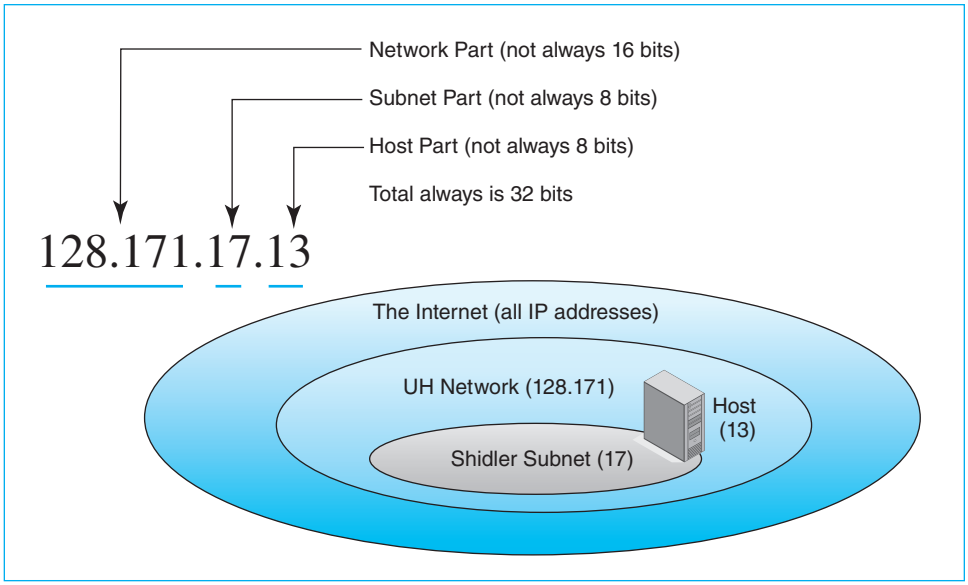


FIGURE 8-2 Hierarchical IPv4 Addresses

Test Your Understanding

2. a) What are the three parts of an IP address? b) How long is each part? c) What is the total length of an IP address? d) In the IP address, 10.11.13.13, what is the network part?

Routers, Networks, and Subnets

Border Routers Connect Different Networks As Figure 8-3 illustrates, networks and subnets are very important in router operation. Here we see a simple site internet. The figure shows that a **border router**'s main job is to connect different networks. This border router connects the 192.168.x.x network within the firm to the 60.x.x.x network of the firm's Internet service provider. Here, the *xs* are the remaining bits of the IP address, so 192.168 and 60 are the network parts of the two networks.

A border router's main job is to connect different networks

Internal Routers Connect Different Subnets The site network also has an internal router. An **internal router**, Figure 8-3 demonstrates, only connects different subnets within a network—in this case, the 192.168.1.x, 192.168.2.x, and 192.168.3.x subnets. Many sites have multiple internal routers to link the site's subnets.

An internal router only connects different subnets within a network.

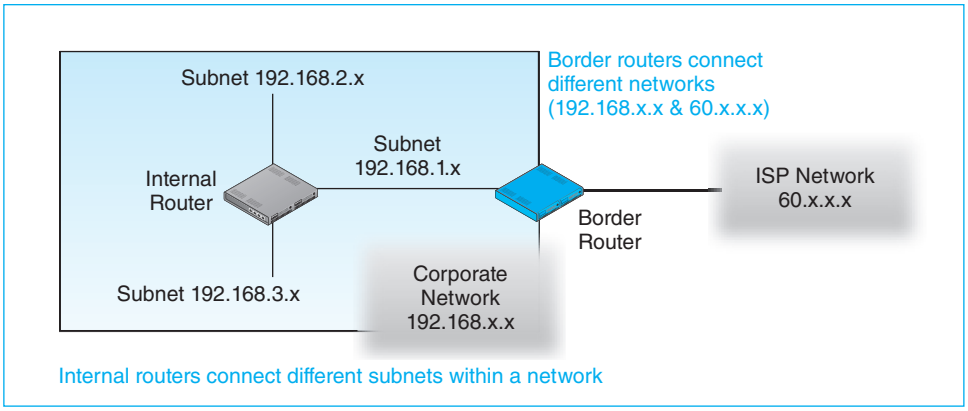


FIGURE 8-3 Border Routers, Networks, and Subnets

Test Your Understanding

3. a) Connecting different networks is the main job of what type of router? b) What type of router only connects different subnets?

Network and Subnet Masks

We have seen that in the University of Hawai`i network, the first 16 bits in IPv4 addresses are the network part, the next 8 are the subnet part, and the final 8 are the host part. However, because the sizes of the network, subnet, and host parts differ, routers need a way to tell the sizes of key parts. The tool that allows them to do this is masks.

32-Bit Strings Figure 8-4 illustrates how masks work. A mask is a string of 32 bits, like an IP address. However, a mask always begins with a series of 1s; this is followed by a series of 0s. The total length of an IP address is always 32 bits, so if a mask begins with twelve 1s, it will end with twenty 0s.

In a network mask, the bits in the network part of the mask are 1s, while the remaining bits are 0s. In subnet masks, the bits of both the network and subnet parts are 1s, and the remaining bits are 0s. We have seen that the University of Hawai`i network part is 16 bits and the subnet part is 8 bits. So the network mask will have sixteen 1s followed by sixteen 0s. The subnet mask will have twenty-four 1s followed by eight 0s.

A mask is a 32-bit string of 1s and 0s.

The mask always has a certain number of initial 1s. The remaining bits are always 0s.

In network masks, the initial 1s correspond to the network part.

In subnet masks, the initial 1s correspond to the network and *subnet parts.*

For example, suppose that the mask is 255.255.0.0. This means that the four 8-bit segments of the mask have the values 255, 255, 0, and 0. In dotted decimal notation

The Problem

There is no way to tell by looking at an IP address what sizes the network, subnet, and host parts are—only that their total is 32 bits

The solution: masks

Series of initial 1s followed by series of final 0s for a total of 32 bits

Example: 255.255.0.0 is sixteen 1s followed by 16 0s

In prefix notation, /16

(Decimal 0 is 8 0s and Decimal 255 is 8 1s)

Result:

Where the mask has 1s, the result is the original bits of the IP address

Where the mask has 0s, the result is 0.

Mask Operation

Network Mask	Dotted Decimal Notation
Destination IP Address	128.171.17.13
Network Mask	255.255. 0. 0
Bits in network part, followed by 0s	128.171. 0 .0

Subnet Mask	Dotted Decimal Notation
Destination IP Address	128.171. 17.13
Subnet Mask	255.255.255. 0
Bits in network part and subnet parts, followed by 0s	128.171. 17. 0

FIGURE 8-4 IP Networks and Subnet Masks

eight 1s is 255 and eight 0s is 0. Therefore, the four segments have, in order, eight 1s, eight 1s, eight 0s, and eight 0s. Putting this together, the mask has sixteen 1s followed by sixteen 0s.

Prefix Notation for Masks Writing 255.255.255.0 is not very difficult, but networking professionals often use a shortcut called prefix notation. The mask 255.255.255.0 is twenty-four 1s followed by eight 0s. In prefix notation, this mask is represented as /24. Do you see the pattern? In **prefix notation**, a mask is represented by a slash followed by the number of initial 1s in the mask. What about 255.0.0.0? Yes, it is /8. Prefix notation is simpler to write than dotted decimal notation. By the way, we call this prefix notation because it focuses on the first part of the mask—the part that is all 1s.

In prefix notation, a mask is represented by a slash followed by the number of initial 1s in the mask.

Another advantage of prefix notation for a mask is that it is simple even if the number of leading 1s is not a multiple of eight. For example, suppose that the mask is eighteen 1s followed by fourteen 0s. The mask in prefix notation is obviously /18. What if you saw this mask in dotted decimal notation, in which would be 255.255.48.0? The first two octets are obviously all 1s. However, you would need your decimal-to-binary calculator to figure out that 48 is 00110000.

Masking IP Addresses Figure 8-4 shows what happens when a mask is applied to an IP address, 128.171.17.13. The mask is 255.255.0.0. Where the mask has 1s, the result is the original bits of the IP address. There are sixteen 1s. This is two octets. So the first two octets of the result would be 128.171. For the remaining sixteen bits, which are 0s, the result of the masking is 0. So the masking result is 128.171.0.0.

Network Masks Network masks, as noted earlier, have 1s in the network part and 0s for remaining bits. If the network mask is 255.255.0.0 and the IP address is 128.171.17.13, then the result of masking is 128.171.0.0. This tells us that 128.171 is the network part.

Subnet Masks For subnet masks, in turn, the initial 1s indicate the number of bits in *both* the network and subnet parts. Therefore, if 128.171 is the network part and 17 is the subnet part, then the subnet mask will be 255.255.255.0 (/24). If you mask 128.171.17.13 with /24, you get 128.171.17.0.[1]

Test Your Understanding

4. a) How many bits are there in a mask? b) What do the 1s in a network mask correspond to in IP addresses? c) What do the 1s in a subnet mask correspond to in IP addresses? d) When a network mask is applied to any IP address on the network, what is the result?
5. a) A mask has eight 1s, followed by 0s. Express this mask in dotted decimal notation. b) Express this mask in prefix notation. c) In prefix notation, a mask is /16. Express this mask in dotted decimal notation. d) Express the mask /18 in dotted decimal notation. (You will need a calculator for this.)

HOW ROUTERS PROCESS PACKETS

Switching versus Routing

In Chapter 5, we saw that Ethernet switching is very simple. Ethernet switches must be organized in a hierarchy. Therefore, there is only a single possible path between any two hosts across the network. When a frame arrives, there is only one possible port to

[1] Why not make the network part 0s and the subnet part 1s instead of making both 1s? Think of a network as a state and a subnet as a city. In the United States, there are two major cities named Portland—one in Maine and the other in Oregon. You cannot just say "Portland" to designate a city. You must give both the state and city. Analogously, there may be many subnet parts with a value of 17, so you must give both the network and subnet parts to designate a specific subnet. Another way to look at it is that if you only had 1s in the subnet part of a subnet mask, you would break the rule that masks must have a number of leading 1s followed by a number of trailing 0s.

use to send the frame back out. Figure 8-5 shows an Ethernet switching table. Because an Ethernet frame can only be sent out one port, each Ethernet address only appears in one row. This row tells the switch which port to use to send the frame back out. This single row can be found quickly, so an Ethernet switch does little work per frame. This makes Ethernet switching fast and therefore inexpensive per frame handled.

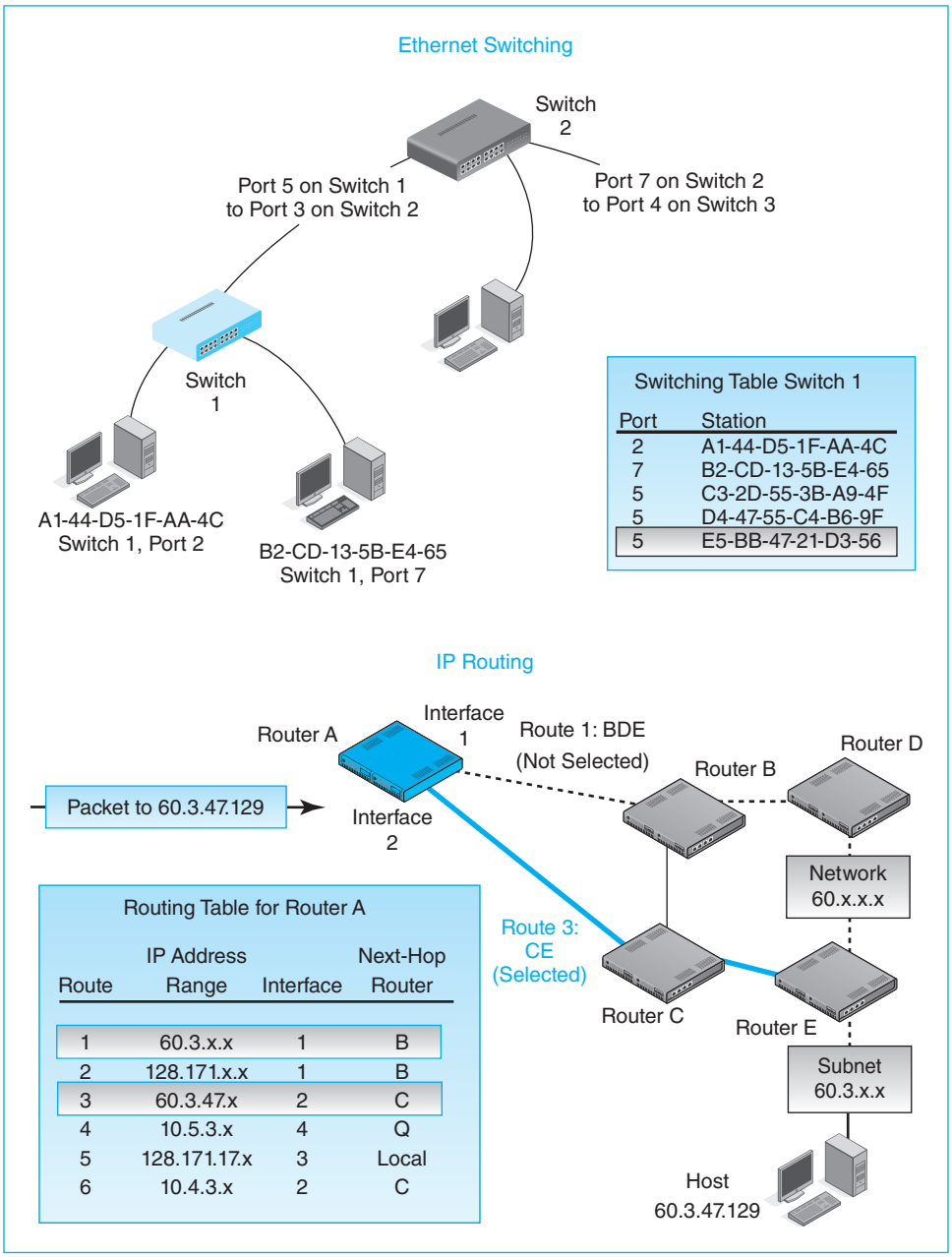


FIGURE 8-5 Ethernet Switching versus IP Routing

Routing

Processing an individual packet and passing it closer to its destination host is called routing

The Routing Table

Each router has a routing table that it uses to make routing decisions

Routing tables have rows

Each row represents a route for a range of IP addresses—often packets going to the same network or subnet

The Routing Decision

Find all row matches

Find the best-match row

Send the frame out, based on information in the row

FIGURE 8-6 The Routing Process (Study Figure)

In contrast, routers are organized in meshes. This gives more reliability because it allows many possible alternative routes between endpoints. Figure 8-5 shows that in a routing table, each row represents an alternative route for a range of IP addresses. Consequently, to **route** (forward) a packet, a router must first find *all* rows representing alternative routes that a particular incoming packet can take. It must then pick the best alternative route from this list. This requires quite a bit of work per packet, making routing more expensive than switching.

Test Your Understanding

6. Why are routing tables more complex than Ethernet switching tables? Give a detailed answer.

Routing Table

Figure 8-7 shows a routing table. It has a number of rows and columns. We will see how a router uses these rows and columns to make the routing decision—what to do with an arriving packet.

Rows Are Routes for All IP Addresses in a Range

In the routing table, each row represents a route for all IP addresses within a range of IP addresses—typically addresses within a particular network or subnet. It does not specify the full route, however; it only specifies the next step in the route (either the next-hop router to handle the packet next or the destination host).

In the routing table, each row represents a route for all IP addresses within a range of IP addresses.

Row	Destination Network or Subnet	Mask (/Prefix)	Metric (Cost)	Interface	Next-Hop Router
1	128.171.0.0	255.255.0.0 (/16)	47	2	G
2	172.30.33.0	255.255.255.0 (/24)	0	1	Local
3	60.168.6.0	255.255.255.0 (/24)	12	2	G
4	123.0.0.0	255.0.0.0 (/8)	33	2	G
5	172.29.8.0	255.255.255.0 (/24)	34	1	F
6	172.40.6.0	255.255.255.0 (/24)	47	3	H
7	128.171.17.0	255.255.255.0 (/24)	55	3	H
8	172.29.8.0	255.255.255.0 (/24)	20	3	H
9	172.12.6.0	255.255.255.0 (/24)	23	1	F
10	172.30.12.0	255.255.255.0 (/24)	9	2	G
11	172.30.12.0	255.255.255.0 (/24)	3	3	H
12	60.168.0.0	255.255.0.0 (/16)	16	2	G
13	0.0.0.0	0.0.0.0 (/0)	5	3	H

FIGURE 8-7 Routing Table

This is important because the routing table does not need a row for each IP address as an Ethernet switching table does. It only needs a row for each group of IP addresses. This means that a router needs many fewer rows than an Ethernet switch would need for the same number of addresses.

However, there are many more IP addresses on the Internet than there are Ethernet addresses in an Ethernet network. Even with rows representing groups of IP addresses, core routers in the Internet backbone still have several hundred thousand rows. In addition, while an Ethernet switch only needs to find a single row for each arriving frame, we will see that routers need to look carefully at *all* rows.

Test Your Understanding

7. a) In a routing table, what does a row represent? b) Do Ethernet switches have a row for each individual Ethernet address? c) Do routers have a row for each individual IP address? d) What is the advantage of the answer to the previous subparts of this question?

Step 1: Finding All Row Matches

We will now see how the router uses its routing table to make routing decisions. The first step is to find which of the rows in the routing table match the destination IP address in an arriving packet. Due to the existence of alternative routes in a router mesh, most packets will match more than one row.

Step 1: Find All Row Matches

The router looks at the destination IP address in an arriving packet

For each row:

Apply the row's mask to the destination IP address in the packet

Compare the result with the row's destination value

If the two match, the row is a match

The router must do this to ALL rows because there may be multiple matches

This step ends with a set of matching rows

Example 1: A Destination IP Address that is NOT in the Range

Destination IP Address of Arriving Packet	60.43.7.8
Apply the (Network) Mask	255.255.0.0
Result of Masking	60.43.0.0
Destination Column Value	128.171.0.0
Does Destination Match the Masking Result?	No
Conclusion	Not a match.

Example 2: A Destination IP Address that IS in the Range

Destination IP Address of Arriving Packet	128.171.17.13
Apply the (Network) Mask	255.255.0.0
Result of Masking	128.171.0.0
Destination Column Value	128.171.0.0
Does Destination Match the Masking Result?	Yes
Conclusion	Row is a match.

Step 2: Find the Best-Match Row

The router examines the matching rows it found in Step 1 to find the best-match row

Basic rule: It selects the row with the longest match (Initial 1s in the row mask)

Tie breaker: If there is a tie on longest match, select among the tie rows based on a metric

For cost metric, choose the row with the lowest metric value

For speed metric, choose the row with the highest metric value

Step 3: Send the Packet Back Out

Send the packet out the interface (router port) designated in the best-match row

Address the packet to the IP address in the next-hop router column

If the address says Local, the destination host is out that interface

Sends the packet to the destination IP address in a frame

FIGURE 8-8 Steps in a Routing Decision (Study Figure)

Row Number Column The first column in Figure 8-7 contains route (row) numbers. Routing tables actually do not have this column. We include it to allow us to refer to specific rows in our discussion. Again, each row specifies a route to a destination.

Row Matches How does the router know which IP addresses match a row? The answer is that it uses the *Destination Network or Subnet* column and the *Mask* column.

Suppose that all IP addresses in the University of Hawai`i (UH) network should match a row. The mask would be the network mask 255.255.0.0, because the UH Network has a 16-bit network part. If this mask is applied to any UH address, the result will be 128.171.0.0. This is the value that will be in the destination column. In fact, this matches Row 1 in Figure 8-7.

Let's see how routers use these two columns in Figure 8-7. Suppose that a packet arrives with the IP address 60.43.7.8. The router will look first at Row 1.

- In this row, the router applies the mask 255.255.0.0 to the arriving packet's destination IP address, 60.43.7.8. The result is 60.43.0.0.
- Next, the router compares the masking result, 60.43.0.0, to the destination value in the row, 128.171.0.0. The two are different, so the row is not a match.

However, suppose that a packet arrives with the IP address 128.171.17.13. Now, the situation is different.

- Again, the router applies the mask 255.255.0.0 in Row 1 to the destination IP address, 128.171.17.13. The result is 128.171.0.0.
- Next, the router compares 128.171.0.0 to the destination value in the row, 128.171.0.0. The two are identical. Therefore, the row is a match.

Mask and Compare This may seem like an odd way to see if a row matches. A human can simply look at 60.43.7.8 and see that it does not match 128.171.0.0. However, routers do not possess human pattern-matching abilities.

While routers cannot do sophisticated pattern recognition, routers (and all computers) have specialized circuitry for doing masking and comparing—the two operations that row matching requires. Thanks to this specialized circuitry, routers can blaze through hundreds of thousands of rows in a tiny fraction of a second.

The Default Row The last row in Figure 8-7 has the destination 0.0.0.0 and the mask 0.0.0.0. This row will match *every* IP address because masking any IP address with 0.0.0.0 will give 0.0.0.0, which is the value in the destination field of Row 13. This row ensures that at least one row will match the destination IP address of every arriving packet. It is called the **default row**. In general, a "default" is something you use if you do not have a more specific choice.

The Need to Look at All Rows Thanks to their mesh topology, internets have many alternative routes. Consequently, a router cannot stop the first time it finds a row match for each arriving packet because there may be a better match further on. A router has to look at each and every row in the routing table to see which match. So far, we have seen what the router does in Row 1 of Figure 8-7. The router then goes on to Row 2 to see if it is a match by masking and comparing. After this, it goes on to Row 3, Row 4, Row 5, and so on, all the way to the final row (Row 13 in Figure 8-7).

Test Your Understanding

8. a) In Row 3 of Figure 8-7, how will a router test if the row matches the IP address 60.168.6.7? Show the calculations in the format given in the text. b) Is the row a

match? c) Why is the last row called the default row? d) Why must a router look at all rows in a routing table? e) Which rows in Figure 8-7 match 172.30.17.6? (Don't forget the default row.) Show your calculations for rows that match. f) Which rows match 60.168.7.32? Show your calculations for rows that match. g) Which rows in Figure 8-7 match 128.171.17.13? (Show your calculations for rows that match.)

Step 2: Selecting the Best-Match Row

List of Matching Rows At the end of Step 1, the mask and compare process, the router has a list of matching rows. For a packet with the destination IP address 128.171.17.13, three rows in Figure 8-7 match. The first is Row 1, as we have already seen. The second is Row 7, with a destination of 128.171.17.0 and a mask of 255.255.255.0. Finally, the default row (Row 13 in this figure) will always be a match. From these, the router must select the best-match row, the row that represents the best route for an IP address.

Basic Rule: Longest Match How does the router decide whether to follow Row 1, Row 7, or Row 13? The answer is that it follows the rule of selecting the **longest match** (the longest number of initial 1s in the mask). Row 1 has a mask of 255.255.0.0, which means that it has a 16-bit match. Row 7, in turn, has the prefix /24, meaning that it has a 24-bit match. Row 13 has a prefix of 0/. Row 7 has the longest match, so the router selects Row 7 as the best match.[2]

By the way, note that the default row always has a prefix of 0/. Consequently, if the default row and other rows are matches, the default row will never be chosen as the best-match row because it will always have the shortest length of match.

Tie-Breaker Rule: Best Metric Value What if two rows tie for the longest length of match? For instance, the destination IP address 172.29.8.112 matches both Row 5 and Row 8 in Figure 8-7. Both have a match length of 24 bits—a tie.

In case of a tie for longest match, the tie-breaker rule is to use the **metric** column, which describes the desirability of a route. For instance, in Figure 8-7, the metric is cost. Row 5 has a cost of 34, while Row 8 has a cost of 20. *Lower cost is better than higher cost*, so the router selects Row 8.

In this case, the row with the *lowest* metric won. However, what would have happened if the metric had been *speed* instead of cost? *More speed is better*, so the router would choose Row 5, with the *higher* speed (34).

Test Your Understanding

9. a) Distinguish between Step 1 and Step 2 in the routing process. b) If any row other than the default row matches an IP address, why will the router never choose the default row? c) Which rows in Figure 8-7 match 128.171.17.13? (Don't forget the default row.) Show your calculations for rows that match. d) Which of these is the best-match row? Justify your answer. e) What rows match 172.40.17.6? Show

[2] Why the longest match rule? The answer is that the closer a route gets a packet to the destination IP address, the better. Row 1 only gets the packet to the UH network, 128.171.x.x, while Row 7 gets the packet all the way to the Shidler College of Business subnet of the University of Hawai`i, 128.171.17.x—the subnet that contains host 128.171.17.13.

your calculations for rows that match. f) Which of these is the best-match row? Justify your answer. g) Which rows match 172.30.12.47? Show your calculations for rows that match. h) Which of these is the best-match row? Justify your answer. i) How would your previous answer change if the metric had been reliability?

Step 3: Sending the Packet Back Out

In Step 1, the router found all rows that matched the destination IP address of the arriving packet. In Step 2, it found the best-match row. Finally, in Step 3, the router sends the packet back out.

Interface Recall that router ports are called interfaces. The fifth column in Figure 8-7 is interface number. If a router selects a row as the best match, the router sends the packet out the interface designated in that row. If Row 1 is selected, the router will send the packet out Interface 2.

Next-Hop Router In a switch, a port connects directly to another switch or to a computer. However, a router interface connects to an entire subnet or network. Therefore, it is not enough to select an interface to send the packet out. It is also necessary to specify *a particular device* on the subnet.

In most cases, the router will send the packet on to another router, called the **next-hop router**. The next-hop router column specifies the router that should receive the packet. It will then be up to that next-hop router to decide what to do next. In Figure 8-7, the next-hop router value is G if Row 1 is selected.[3]

In some cases, however, the destination host will be on the subnet out a particular interface. In that case, the router will send the packet to the destination host instead of to another router. If the next hop is the destination host itself, the next-hop router field will say *local*.

Test Your Understanding

10. a) Distinguish between Step 2 and Step 3 in routing. b) What are router ports called? c) If the router selects Row 13 as the best-match row, what interface will the router send the packet out? d) To what device? e) Why is this router called the default router? (The answer is not in the text.) f) If the router selects Row 2 as the best-match row for packet 172.30.33.6, what interface will the router send the packet out? g) To what device? (Don't say, "the local device.")

Cheating (Decision Caching)

We have discussed what happens when a packet arrives at a router. However, what will the router do if another packet for the same destination IP address arrives immediately afterward? The answer is that the router *should* go through the entire process again. Even if a thousand packets arrive that are going to the same destination IP address, the router should go through the entire three-step process for each of them.

[3] Actually, this column should have the IP address of Router G, rather than its name. However, we include the letter designation rather than the IP address for simplicity of understanding.

As you might expect, a router might cheat, or as it is euphemistically named, cache (remember) the decision it made for a destination IP address. It will then use this decision for successive IP packets going to the same destination. Using a **decision cache** greatly reduces the work that a router will do for each successive packet.

Caching is not prescribed in the Internet Protocol. This is because it is dangerous. The Internet changes constantly as routers come and go and as links between routers

BOX 1

Masking When Masks Do Not Break at 8-Bit Boundaries

All of the masks we have seen up to this point have had their parts broken at 8-bit segment boundaries. For example, at the University of Hawai`i, the network part is 16 bits long, which corresponds to two segments (128.171), the subnet part is 8 bits long (17), and the host part is 8 bits long (13). All of the masks in Figure 8-7 break also at 8-bit segment boundaries.

Masks that break at 8-bit boundaries are easy for humans to read. In general, you can look at a mask in the table and decide if it matches a particular IP address. For instance, if the mask is 255.255.0.0 (/16), and if the destination column value is 128.171.0.0, this definitely matches the IP address 128.171.45.230.

However, masks do not always break at 8-bit boundaries. For example, suppose that a row in the routing table has the destination address 3.136.0.0 and the mask 255.248.0.0. Does the IP address 3.143.12.12 match this row? At first glance, this certainly does not seem to be a match. However, it is.

To see why, look at Figure 8-9. This figure shows the matching analysis when the binary representations are given for each segment. If you follow the masking, you see that the result is a match. When a mask does not break at an 8-bit boundary, you must go back to the raw 32-bit IP address, mask, and destination field values.

Test Your Understanding

11. An arriving packet has the destination IP address 128.171.180.13. Row 86 has the destination value 128.171.160.0. The mask is 255.255.224.0. Does this row match the destination IP address? Show your work. You can use the Windows Calculator if you have a Windows PC. In Windows Vista and earlier versions of Windows, choose scientific when you open the calculator. In the Windows 7 calculator, choose programmer mode.

	Dotted Decimal Notation	Segment 1	Segment 2	Segment 3	Segment 4
IP address	3.143.12.12	00000011	10001111	00001100	00001100
Mask	255.248.0.0	11111111	11111000	00000000	00000000
Result	3.136.0.0	00000011	10001000	00000000	00000000
Destination	3.136.0.0	00000011	10001000	00000000	00000000
Match?	Yes	Yes	Yes	Yes	Yes

FIGURE 8-9 Using a Mask Whose 1s Do Not Break Down at an 8-Bit Boundary

change. Consequently, a cached decision that is used for too long will result in non-optimal routing or even routes that will not work and that will effectively send packets into a black hole.

Test Your Understanding

12. a) What should a router do if it receives several packets going to the same destination IP address? b) How would decision caching speed the routing decision for packets after the first one? c) Why is decision caching dangerous?

BOX 2

The Address Resolution Protocol

The final step in the routing process for each arriving packet is to send the packet back out another interface, to a next-hop router or the destination host.

Address Resolution

To send the packet to a next-hop router or a destination host, the router's interface must place the packet into a frame and send this frame to the next-hop router or destination host. To do this, the interface must know the data link layer address of the destination host. The internet layer process must discover the DLL address of the destination host. This is called **address resolution**.

Address Resolution on an Ethernet LAN with ARP

Figure 8-10 shows the **Address Resolution Protocol (ARP)**, which provides address resolution on Ethernet LANs. There are other address resolution protocols for other subnet technologies.

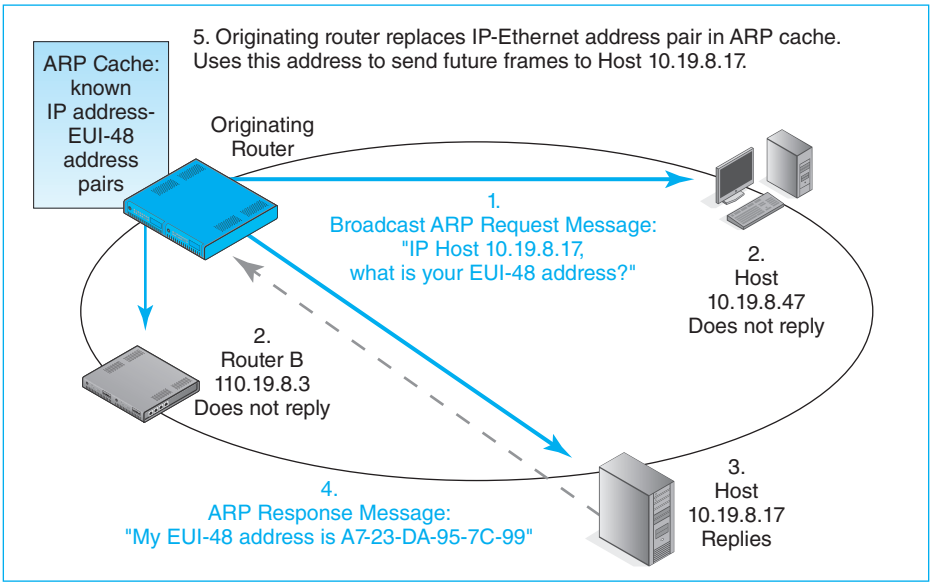


FIGURE 8-10 Address Resolution Protocol (ARP)

(*continued*)

ARP Request Message

Suppose that the router receives an IP packet with destination address 10.19.8.17. Suppose also that the router determines from its routing table that it can deliver the packet to a host on one of its Ethernet subnets.

- First, the router's internet layer process creates an ARP request message that essentially says, "Hey, device with IP address 10.19.8.17, what is your 48-bit EUI-48 address?" The router then broadcasts this ARP packet to all hosts on the subnet.[4]
- Second, the internet layer process on every host examines the ARP request message. If the target IP address is not that of the host, the host's internet layer process ignores the ARP request message. However, the host with IP address 10.19.8.17 composes an ARP response message that includes its EUI-48 address (A7-23-DA-95-7C-99). The target host sends this ARP response message back to the router.
- Third, the router's internet layer process now knows the EUI-48 address associated with IP address 10.19.8.17. It will deliver the packet to that host in a frame addressed to A7-23-DA-95-7C-99.

The ARP Cache

ARP is a time-consuming process, and the router does not want to do it for each arriving packet. Consequently, the internet layer process on the router saves the IP address–data link layer address information in its **ARP cache** (section of memory). Afterward, whenever an arriving packet has the IP address 10.19.8.17, the router looks up the DLL address in the ARP cache.

Using ARP for Next-Hop Routers

We have just looked at how routers use ARP when they deliver packets to destination hosts. A router also needs to know the data link layer destination addresses of next-hop routers so that it can send them packets encapsulated in frames. Routers use ARP to find the DLL destination addresses of both destination hosts and other routers.

ARP Encapsulation: Finally, Another Internet Layer Protocol!

In this book so far, we have only seen a single protocol at the internet layer—the Internet Protocol (IP). However, ARP is also a protocol at the internet layer, and ARP messages are called packets. ARP packets are encapsulated directly in frames, just like IP packets.

Test Your Understanding

13. A router wishes to send an IP packet to a host on its subnet. It knows the host's IP address. a) What else must it know? b) Why must it know it? c) What message will it broadcast? d) Which device will respond to this broadcast message? e) Does a router have to go through the ARP process each time it needs to send a packet to a destination host or to a next-hop router? Explain. f) Is ARP used to find the DLL destination addresses of destination hosts, routers, or both? g) At what layer does the ARP protocol operate? h) Why must client PCs use ARP to transmit packets? The answer is not in the text.

[4] Actually, the router passes the packet down to the data link layer process on the subnet's interface. It tells the data link layer process to broadcast its ARP packet. If the subnet standard is Ethernet, the DLL process places the packet into a frame with the destination Ethernet address FF-FF-FF-FF-FF-FF (forty-eight 1s). This is the Ethernet broadcast address. Switches will send frames with this broadcast address to all stations, and all stations will accept it as they would a frame addressed to their specific Ethernet address.

THE INTERNET PROTOCOL VERSION 4 (IPV4) FIELDS

We have focused on IP routing. However, the Internet Protocol has other properties that networking professionals need to understand.

As noted in Chapter 1, most routers today on the Internet and private internets are governed by the **IP Version 4 (IPv4)** standard. (There were no versions 0 through 3.) We looked at the header checksum, the source IP address, and the destination IP address in the first two chapters. Now we will look at the other fields in the IPv4 header.

The First Row

Figure 8-11 shows the IPv4 packet. Its first 4 bits constitute the **version** field. This field has the value 0100 (binary for 4). This indicates that this is an IPv4 packet. The next field gives the header length, and the last field on the first row gives the total length of the packet.[5]

Between the header and total length fields, two fields govern transmission quality. The **Differentiated Services Control Point** field can be used for priority or other

Bit 0 Bit 31

Version (4 bits) Value is 4 (0100)	Internet Header Length (4 bits)	DSCP (6 bits)	ECN (2 bits)	Total Length (16 bits) Length in octets	
Identification (16 bits) Unique value in each original IP packet				Flags (3 bits)	Fragment Offset (13 bits) Octets from start of original IP fragment's data field
Time to Live (8 bits)		Protocol (8 bits) 1 = ICMP, 6 = TCP, 17 = UDP		Header Checksum (16 bits)	
Source IP Address (32 bits)					
Destination IP Address (32 bits)					
Options (if any)					Padding
Data Field					

DSCP = Differentiated Services Control Point
ECN = Explicit Congestion Notification

FIGURE 8-11 IP Version 4 (IPv4) Packet Syntax

[5] The header length field gives the length of the header in 32-bit units. The length field gives the total length of the IP packet in octets.

quality of service purposes. The **Explicit Congestion Notification** field can be used to reduce the transmission frequency between a pair of hosts to cope with congestion in the transmission system between them.

Test Your Understanding

14. a) What is the main version of the Internet Protocol in use today? b) Which field can be used to specify quality of service? c) How can the ECN field be used?

The Second Row

TCP fragments application messages and sends them in individual packets. This has benefits that we saw in Chapter 1. When IPv4 was created, it was decided to allow routers to fragment packets as well. Although this seemed like a good idea at the time, it led to many problems. Today, operating systems by default tell routers not to fragment IPv4 packets. When IPv6 was developed, packet fragmentation was not allowed. The second row has information that the destination host uses to reassemble fragmented packets. Given the unimportance of IPv4 packet fragmentation, we will ignore the fields in this row.

Test Your Understanding

15. a) Distinguish between application message fragmentation and packet fragmentation. b) Under what circumstances would the identification, flags, and fragment offset fields be used in IP? c) Why did we not study them in detail? d) Does IPv6 allow packet fragmentation?

The Third Row

IP Time to Live (TTL) Field In the early days of the ARPANET, which was the precursor to the Internet, packets that were misaddressed would circulate endlessly among packet switches in search of their nonexistent destinations. To prevent this, IP added a **time to live (TTL)** field that is assigned a value by the source host. Different operating systems have different TTL defaults. Most insert the TTL value 128. Each router along the way decrements (decreases) the TTL field by 1. A router decrementing the TTL to 0 will discard the packet.

IP Protocol Field The **protocol field** tells the contents of the data field. TCP and UDP have protocol values 6 and 17, respectively. If the protocol field value is 1, the IP packet carries an ICMP message in its data field. As we will see later in the chapter, the IP header is a lean mean routing machine with no time for supervisory messages. ICMP is TCP/IP's tool for carrying internet layer supervisory messages. We will look at ICMP at the end of this chapter. After decapsulation, the internet layer process must pass the packet's data field to another process. The protocol field value determines which process should receive the data field.

Test Your Understanding

16. a) What does a router do if it receives a packet with a TTL value of 2? b) What does the next router do? c) What does the protocol field value tell the destination host?

IP Options

The IPv4 header allows options. There are several possible options, and they may come in any order. Some are only read by the destination host. However, a lack of required order means that each router must look at every option to see if it applies. This is time consuming.

Test Your Understanding

17. What problem is caused by the way that IPv4 handles options?

IP VERSION 6 (IPV6)

Outgrowing IPv4

Although IPv4 continues to dominate the Internet's traffic, the Internet Assigned Numbers Authority (IANA) did a poor job distributing IPv4 addresses, and there are no more to distribute. This is forcing more organizations to use IPv6 addresses. Firms that need new IP addresses are forced to apply for blocks of IPv6 addresses. To work with firms that only have IPv6 addresses, other firms must learn to support IPv6.

The most fundamental change in IPv6 is the move from 32-bit addresses to 128-bit addresses. This does not produce merely four times as many addresses. Each additional bit *doubles* the number of addresses. So while there are just under 4.3 billion (4.3×10^9) IPv4 addresses, there are 3.4×10^{38} IPv6 addresses—34 undecillion. To put this in perspective, there are about seven billion people in the world today. For each person, there are 5×10^{28} IPv6 addresses. Even with the Internet of Things, IPv6 will "solve" the address availability problem for many years to come.

Test Your Understanding

18. a) What is the main problem with IPv4 that IPv6 was created to solve? b) How does IPv6 solve this problem?

IPv6

In its 1994 meeting, the IETF decided to create a new version of the Internet Protocol. The IETF called this new version **IP Version 6 (IPv6)**. Over the next few years, the IPv6 standards family grew and matured. It was soon ready to be used, and many network-ing and computer vendors began to build IPv6 into their products.

Organizations soon found that using these new equipment capabilities, however, was a great deal more work than simply turning them on. For many years, few organizations saw the need to make the expensive upgrade to IPv6 because they had enough addresses. In addition, Network Address Translation (NAT) greatly extended the use of existing IP addresses in firms, at the cost of some complexity but at the gain of some security. IPv6 would have the mandatory inclusion of IPsec security functionality, but IPsec was quickly modified to work with IPv4 as well. Seeing no hard business case for upgrading, few companies did. Now that IPv4 addresses are no longer available, however, nearly all companies are beginning to at least plan for the implementation of IPv6, and many have already done so. As we will see in Chapter 9, companies have found that IPv6 implementation is a long and complex process. They need employees who understand this new protocol and other "v6" protocols such as ICMPv6 and DHCPv6.

Test Your Understanding

19. a) What has been holding back the adoption of IPv6? b) What is pushing IPv6 adoption now?

Writing 128-Bit IPv6 Addresses

We write IPv4 addresses in dotted decimal notation—four decimal numbers between 0 and 255 separated by dots. This gives addresses like 128.171.17.13. People can actually remember these addresses.

For the 128-bit addresses of IPv6, we would also like simpler ways to write them, but anything we do will still overload human memory. Consequently, when we write IPv6 addresses for human consumption, we do so to make the writing easier.

A 128-bit IPv6 address is shown in the following example. This is obviously difficult to write and read.

00100000000000010000000001001111111111001010110000000000000000000000000000000000000000000000000001100110100111111000011111001010

Figure 8-12 shows how to simplify this address for human reading and writing. First, IPv6 does not use dotted decimal notation as IPv4 does. Rather, IPv6 uses hexadecimal notation, which we saw in Chapter 5, in the context of Ethernet EUI-48 addresses. Each "nibble" of 4 bits is converted into a hex symbol from 0 through F. A 128-bit IPv6 address, then, would be translated into 32 hex symbols.

In Ethernet, we write hex symbols in pairs, separating each pair with a dash. This gives addresses like A1-B2-C3-D4-E5-F6. In IPv6, in contrast, we group hex symbols in tetrads (groups of four). Each tetrad is called a **field**. An example of a field is *fe56*. Note that we write the hex symbols in *lowercase* when writing IPv6 addresses. Each symbol is still 4 bits, so fe56 represents 16 bits. A full IPv6 address will have eight of these fields, which are separated by *colons* instead of dashes. The following is an IPv6 address written in hexadecimal notation.

128-bit IPv6 Address	0010000000000001000000000100111 1111111000100110000000000000000 0000000000000000000000000000000 1100110100111111100001111110010 10
Convert to hexadecimal notation; divide four-symbol fields by colons.	2001:0027:fe56:0000:0000:0000:cd3f:0fca
Remove leading 0s from each field.	2001:27:fe56::::cd3f:fca
If multiple 0000 fields are shortened, a series of colons can be reduced to a single pair of colons.	2001:27:fe56::cd3f:fca
Only shorten one run of colons. If there are multiple runs of multiple colons, shorten the longest. If two tie for the longest run, shorten only the first.	

FIGURE 8-12 Simplifying 128-bit IPv6 Addresses

2001:0027:fe56:0000:0000:0000:cd3f:0fca

This is still long. Fortunately, there are rules to help us shorten the writing of IPv6 addresses a little. The first is that in each field, *any leading 0s are dropped*. This is easy to understand. If the reader sees *:27:*, this must be *:0027:*. Note that only *leading* 0s are dropped. If trailing 0s or 0s anywhere were dropped, the reader could not know if *:27:* was *:0027:*, *:2700:*, or *0270:*. Dropping leading 0s is also natural because we do that when writing decimal numbers. Here is what the IPv6 address looks like after leading 0s are dropped. This is much shorter.

2001:27:fe56::::cd3f:fca

Note that if a field has four 0s, it will become :: by the first rule. In other words, all four "leading" 0s are dropped, and only the colons remain. A second rule is that if several consecutive fields of zero occur, *one* sequence of *all-zero fields* can be reduced to a simple ::. So if an IPv6 address has a sequence *:0000:0000:0000:*, this can be replaced by ::. This further simplifies our IPv6 address to the following:

2001:27:fe56::cd3f:fca

These two rules reduce length, but applying them has some subrules that appliers must follow.

- First, address simplification should be done whenever possible to the greatest extent possible.
- Second, only a single group of consecutive 0s may be shortened this way in an IP address.
- Third, if there are multiple sequences of all-0 fields, the *longest* group of all-0 fields should be shortened. This just makes sense. One might as well shorten things as much as possible.
- Fourth, if two groups of consecutive 0s tie for the longest number of all-zero groups, the *first* of these groups must be shortened.

These rules can be a little daunting, but it is important to know how to write IPv6 addresses properly. Following these rules means that everyone will write IP addresses the same way. If some people drop some initial 0s while others do not, and if some writers violate other rules, the same IP address will be written in different ways by different people. This will make it more difficult to determine if two written addresses are the same or different. It will also make string searching in databases and configuration fields far more difficult and error prone.

Test Your Understanding

20. a) Are IPv6 addresses written in uppercase or lowercase letters? b) Are IPv6 addresses written with decimal or hexadecimal symbols? c) How many symbols are there in a field? d) How are fields separated? e) How many fields are there in an IPv6 address?

21. a) List the rules for simplifying IPv6 addresses. b) Simplify the following IP address: 2001:0ed2:056b:00d3:000c:abcd:0bcd:0fe0. c) Simplify the following IP address: 2001:0002:0000:0000:0000:abcd:0bcd:0fe0. d) Simplify the following IP address: 2001:0000:0000:00fe:0000:0000:0000:cdef. e) What is the advantage of simplifying IPv6 addresses according to strict rules?

The IPv6 Header

Figure 8-13 shows the IPv6 header. The most obvious difference between the IPv6 and IPv4 headers is that IPv4 headers are usually 20 octets long, while IPv6 headers are 40 octets long. Actually, we will call this the **main IPv6 header** because, as we will see, an IPv6 packet can have multiple headers.

The second difference is that the IPv6 main header, although longer, is simpler than the IPv4 header, with fewer fields for routers and hosts to consider. This relative simplicity means that routers process longer IPv6 headers faster than they can process IPv4 headers.

Version Number Field Both headers begin with a 4-bit **version number** field. For IPv4, the field value is 0100 (four). For IPv6, it is 0110 (six).

Traffic Class and Flow Label Fields The first row of the IPv6 header also contains an 8-bit traffic class field and a 20-bit flow label field.[6] The two fields specify how routing will be handled in terms of priority and other quality of service matters.

The **traffic class field** has two subfields. The six-bit **diffserv (differentiated services)** subfield specifies whether *this particular packet* should be given routine best-effort service, high-priority low-latency service, or some other type of service. The last two bits are for congestion notification.

Bit 0 Bit 31

Version (4 bits) Value is 6 (0110)	Traffic Class (8 bits) Diffserv (6 bits) Congestion Notification (2)	Flow Label (20 bits) Marks a packet as part of a specific flow	
Payload Length (16 bits)		Next Header (8 bits) Name of next header	Hop Limit (8 bits)
Source IP Address (128 bits)			
Destination IP Address (128 bits)			
Next Header or Payload (Data Field)			

FIGURE 8-13 IP Version 6 (IPv6) Packet Syntax

[6] In the original definition of IPv6, these fields were 4 bits and 24 bits, respectively.

The **flow label field** value indicates that the packet is a member of a particular flow. The router has rules that apply to *every packet* in the flow.

Payload Length In IPv6, the **payload length field** gives the length of the packet payload, which is everything beyond the 40-octet main packet header. The payload length field is 16 bits long, so payloads can be up to 65,536 (2^{16}) octets long.

The payload length field gives the length of the packet payload, which is everything beyond the 40-octet main packet header.

Hop Limit Field IPv6 has a **hop limit field** that does the same thing the IPv4 time to live field does. Each router along the way decrements this field's value by 1, and if a router decrements it to 0, the router discards the packet.[7]

No Checksum Field? IPv4 has a header checksum field to check for packet header errors. When IPv4 was created, there was a concern that if packet headers contained errors, they could cause serious problems for the Internet. Experience showed that this concern was groundless. Consequently, IPv6 has no checksum field. The computations needed to check for errors in IPv4 were taxing, even for a 20-octet header. Dropping the checksum field significantly reduces packet handling time on routers.

Test Your Understanding

22. a) How do the version number fields in IPv4 and IPv6 differ? b) What is the general purpose of the diffserv subfield? c) Of the flow label field? d) In IPv6, how can the receiver tell the length of packet? e) Does the payload length field include the lengths of any extension headers in the packet? f) How is the hop limit field used? g) Does IPv6 have a header checksum field? h) What are the implications of this?

Extension Headers

The IPv4 packet has an options field that allows the sender to add options. Few IPv4 packets have options, but each router must check each packet for options, and it must examine each option even if the option does not relate to it. This can cost significant time.

Main Header and Extension Header IPv6 took another approach. As Figure 8-14 shows, the main header can be followed by multiple extension headers. Each extension header has a well-defined purpose, such as providing information for security or mobile operation. Each extension header serves the role that an option does in IPv4.

Next Header Field The headers are daisy chained together based on the **next header** field. The main header's next header field gives the value of the first extension header. That extension header's next header field has the value of the next extension

[7] Internet old-timers know that when IPv4 was created, the time to live value was supposed to be measured in seconds. However, this proved to be unworkable. The value was then interpreted as the maximum number of hops permitted by the packet. The hop limit field name in IPv6 recognizes this.

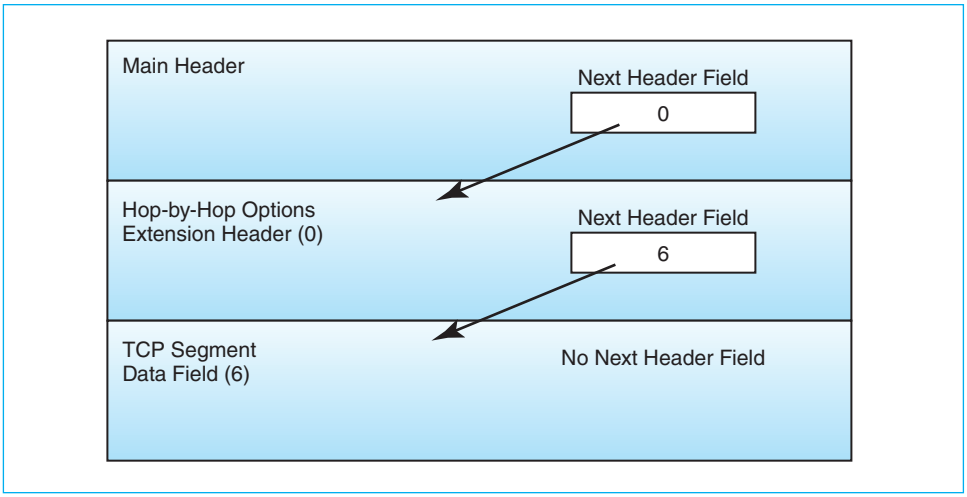


FIGURE 8-14 Main Header and Extension Headers in IPv6

headers. This continues until the last extension header. In the final extension header, the next header field value specifies the data field (TCP, UDP, etc.)

Next Header Values Each extension header has its own next-header field value. This value is placed in the next-header field of the main header or previous extension header. Figure 8-15 shows a few of the extension headers that have been defined for the next-header field, as well as their code values (in parentheses) in next header fields. The full list is much longer. We will discuss only a few of the headers mentioned in the table.

- The **hop-by-hop options** header carries options that must be considered by every router along the packet's route to its destination host. Its importance is indicated by its special extension code, 0. Successive headers usually only have to be dealt with by the destination host. This minimizes the work the router must do on each IPv6 packet.
- The encapsulating security payload (ESP) header (50) is used in IPv6's built-in cryptographic security protocol called IPsec (IP security). We will look at IPsec in the next chapter. Using information in the ESP header, IPsec gives all of

Extension Header Code (Value)	
Extension Header	Upper Layer Messages
Hop-by-Hop Options (0)	TCP (6)
Encapsulating Security Payload Header (50)	UDP (17)
Destination Options (60)	ICMPv6 (58)
Mobility Header (135)	
No Next Header (59)	

FIGURE 8-15 IPv6 Next Header Values

the traditional protections of cryptographic systems that we saw in Chapter 3, including initial authentication and message-by-message encryption for confidentiality, authentication, and message integrity. The term *encapsulating* means that the goal is to protect whatever is carried (encapsulated in) the rest of the packet.[8]

- Sometimes, an IP packet only has a header. In that case, the next header field contains the value 59, to indicate that there is no next header.
- Of course, values are needed to contain the information in IPv6's protocol field. These values indicate the higher-layer message contained in the data field, such as TCP (6) or UDP (17).

Test Your Understanding

23. a) Why is handling options the way that IPv4 does undesirable? b) Why is the approach of using optional extension headers desirable? c) Which header is used by IPsec (IP security)? d) What is usually the only extension header that routers have to consider? e) How does the last extension header before a UDP datagram indicate that the UDP datagram comes next? (You must infer the answer from the text.) f) If you see 59 in the next header field of a header, what will follow this header?

THE TRANSMISSION CONTROL PROTOCOL (TCP)

Fields in TCP/IP Segments

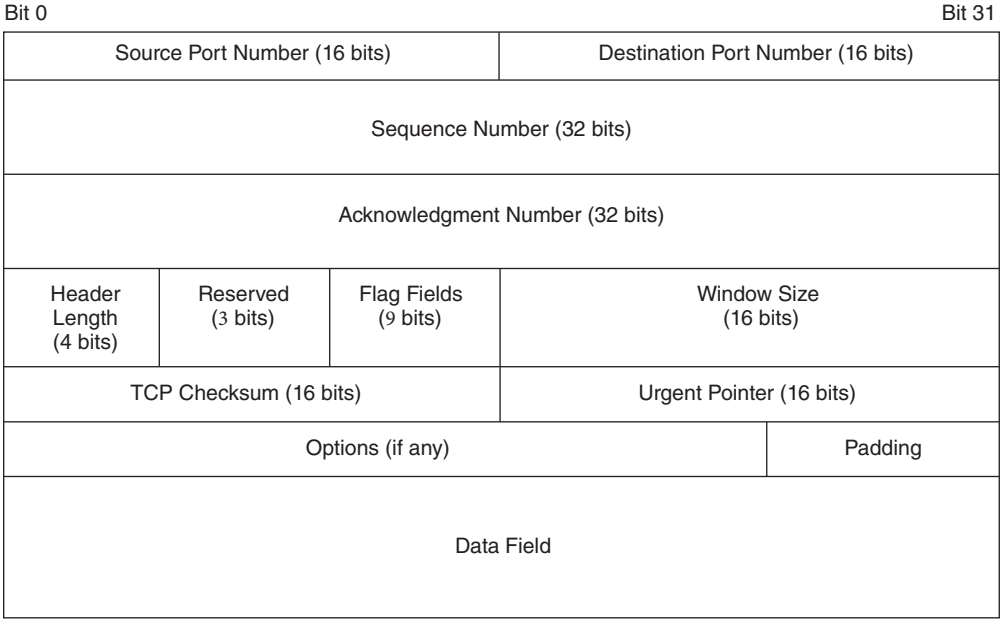
In Chapter 2, we looked briefly at the syntax of TCP messages (segments). In this section, we will look at the syntax of this complex protocol in more depth. When IP was designed, it was made to be a very simple "best effort" protocol (although its routing tables are complex). The IETF left more complex internetwork transmission control tasks to TCP. Consequently, network professionals need to understand TCP very well. Figure 8-16 shows the organization of TCP segments.

Sequence Numbers TCP can handle messages of almost any length. In Chapter 2, we saw that it handles long application messages by fragmenting them into many TCP segments and sending each segment in its own packet. So that the receiver can put the segments back in order, each segment has a **sequence number** that gives its position in the stream of segments. The receiving TCP process puts the segments in order of increasing sequence number, reassembling the full application message. The TCP process then passes the application message up to the correct application process indicated in the port number.[9]

[8] Initially, security was supposed to be a competitive advantage for IPv6 compared to IPv4. However, the IETF quickly made the encapsulating security payload available in IPv4 by allowing its value (50) to appear in the IPv4 protocol field. In addition, while it is often said that the use of security is mandatory in IPv6, the truth is that providing the *capability* for ESP and authentication header security is mandatory. Their *use* is not mandatory. In general, making security an option in IPv4 stole the thunder from IPv6's touted security advantage.

[9] Module A has a detailed discussion of TCP sequence and acknowledgment numbers.

TCP Segment



Flag fields are 1-bit fields. They include SYN, ACK, FIN, and RST.

UDP Datagram

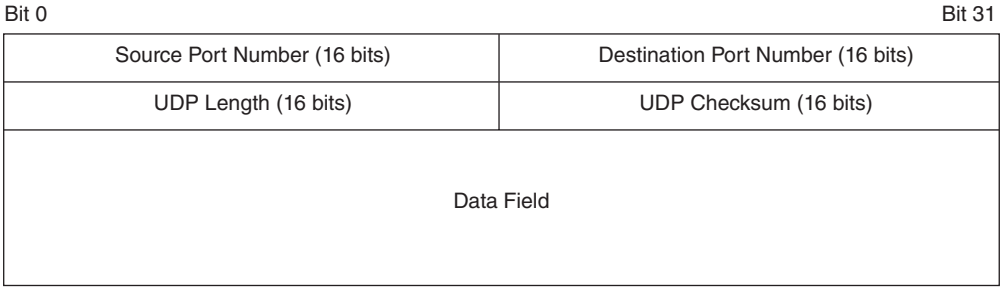


FIGURE 8-16 TCP Segment and UDP Datagram

Acknowledgment Numbers In Chapter 2, we saw that TCP uses **acknowledgments (ACKs)** to achieve reliability. If a transport process receives a TCP segment correctly, it sends back a TCP segment acknowledging the reception. If the sending transport process does not receive an acknowledgment, it transmits the TCP segment again.

The **acknowledgment number field** indicates which segment is being acknowledged. One might expect that if a segment has sequence number X, then the acknowledgment number in the segment that acknowledges it would also be X. As Module A notes, the situation is more complex, but the acknowledgment number is at least related to the sequence number of the segment being acknowledged.

Flag Fields As discussed in Chapter 2, TCP has nine single-bit fields. Single-bit fields are called flag fields, and if they have the value 1, they are said to be **set**. These fields allow the receiving transport process to know the kind of segment it is receiving. We saw several uses of these flag bits in Chapter 2.

- If the ACK bit is set, then the segment acknowledges another segment. If the ACK bit is set, the acknowledgment field must be filled in to indicate which message is being acknowledged.
- If the SYN (synchronization) bit is set, then the segment requests a connection opening.
- If the FIN (finish) bit is set, then the segment requests a normal connection closing.

Openings and Abrupt TCP Closes

In Chapter 2, we saw that TCP is a connection-oriented protocol. Connection-oriented protocols have formal openings and closings. Figure 8-17 recaps these openings and closings.

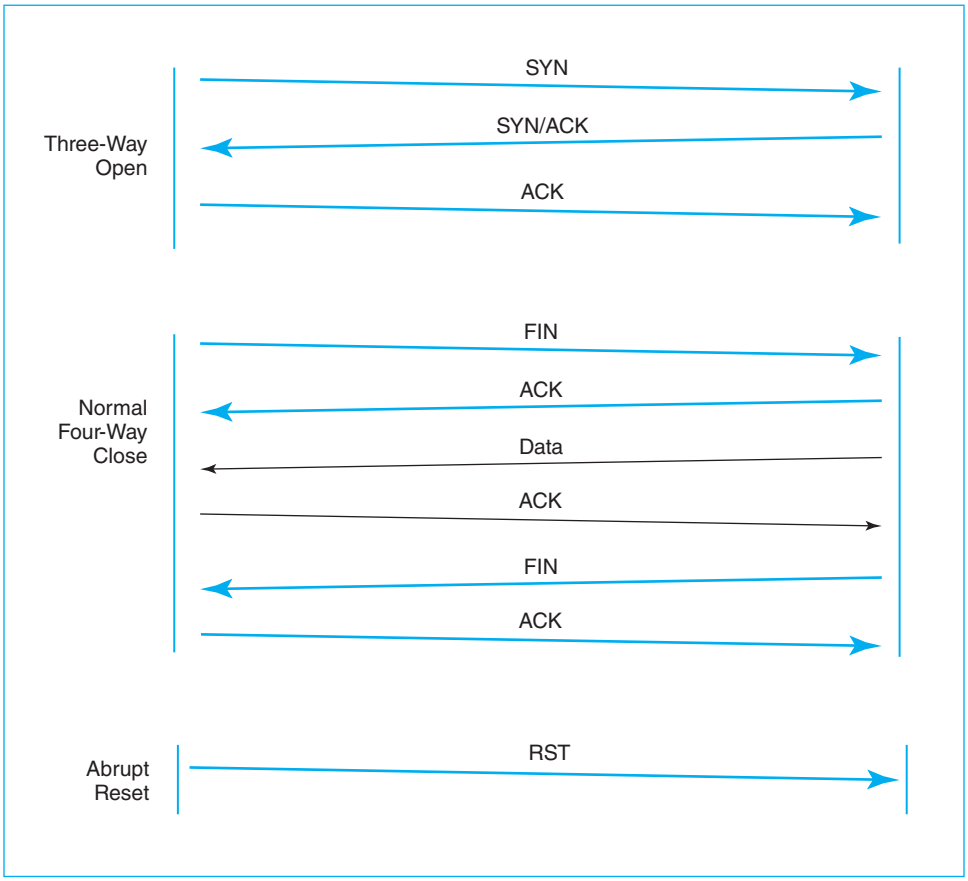


FIGURE 8-17 TCP Session Openings and Closings

In Chapter 2, we looked at *normal* closings. Just as you do not simply hang up on a telephone call when you want to finish talking if you are polite, a normal TCP close consists of two FIN segments, one in each direction, plus their acknowledgments.

However, Figure 8-17 shows that TCP also permits another type of close. This is an abrupt close. Whenever either side wishes to end a conversation, it can simply send a **TCP reset segment**. This is a segment with the **RST** (reset) flag bit set. This may occur if a problem is encountered during a connection, for security reasons, or for any other reason.

Note in Figure 8-17 that an RST segment is not acknowledged. The side that sent the RST segment is not listening any longer, so acknowledging a reset would be as pointless as saying goodbye after someone has hung up on you. The RST segment is one of two segment types that are not acknowledged. As noted in Chapter 2, a segment that is nothing more than acknowledgment (a pure acknowledgment) is not acknowledged because doing so would create an endless loop of acknowledgments.

Test Your Understanding

24. a) For what reason is TCP complex? b) Why is it important for networking professionals to understand TCP? c) What are TCP messages called?

25. a) Why are sequence numbers good? b) What are 1-bit fields called? c) If someone says that a flag field is set, what does this mean? d) If the ACK bit is set, what other field must have a value? e) What is a FIN segment? f) Distinguish between four-way closes and abrupt resets. g) Why is a reset segment not acknowledged?

THE USER DATAGRAM PROTOCOL (UDP)

We saw UDP in Chapter 2. This is a very simple protocol, so the discussion in that chapter is sufficient except for one point. This is the fact that UDP, unlike TCP, cannot do segmentation. This means that an application message must fit into a single UDP datagram. Figure 8-16 shows that the length field in the UDP header is 16 bits long, so the maximum length of the UDP data field, and therefore the maximum length of an application message, is 65,536 octets. UDP messages are called **UDP datagrams**.

UDP cannot do segmentation, so an application message must fit into a single UDP datagram.

Test Your Understanding

26. a) Why can TCP handle long application messages? b) Why can UDP not handle long application messages? c) What is the maximum application message size when UDP is used at the transport layer? d) What are UDP messages called?

	TCP	UDP
Can segment application messages?	Yes	No
Maximum application message size	Unlimited	65,536 octets

FIGURE 8-18 TCP, UDP, and Application Message Length (Study Figure)

CONCLUSION

Synopsis

IPv4 addresses are hierarchical. Their 32 bits usually are divided into a network part, a subnet part, and a host part. All three parts vary in length. A network mask tells what bits are in the network part, while a subnet mask tells what bits are in the total of the network and subnet parts. Masks always begin with a certain number of 1s followed by enough 0s to fill the mask out to 32 bits. For human reading, masks are expressed in dotted decimal notation or prefix notation.

Routers forward packets through an internet. Border routers move packets between the outside world and an internal site network. Internal routers work within sites, moving packets between subnets. Ports in routers are called interfaces. Different interfaces may connect to different types of networks. Most routers are multiprotocol routers, which can handle not only TCP/IP internetworking protocols, but also internetworking protocols from IPX/SPX, SNA, and other architectures. Routers are designed to work in a mesh topology. This creates alternative routes through an internet. Alternative routes are good for reliability. However, the router has to consider the best route for each arriving packet, and this is time consuming and therefore expensive.

To make a routing decision (deciding which interface to use to send an incoming packet back out), a router uses a routing table. Each row in the routing table represents a route to a particular network or subnet. All packets to that network or subnet are governed by the one row. Each row (route) has destination, mask, metric, interface, and next-hop router fields.

If the destination IP address in an arriving packet is in a row's range, that row is a match. After finding all matches in the routing table, the router finds the best-match row on the basis of match length and, in the case of tied match lengths, on metric values. Once a best-match route (row) is selected, the router sends the packet out a particular interface to the next-hop router specified in that row or to the destination host if the destination host is out the interface.

In the examples in the main text, masks broke at 8-bit boundaries, making it easy to specify them with dotted decimal notation. If you read the box "Masking When Masks Do Not Break at 8-Bit Boundaries," you can deal more realistically with the world of masking because masks often do not break at 8-bit boundaries.

If you read the box, "The Address Resolution Protocol (ARP)," you saw that the router must encapsulate the packet in a frame in order to send it out. The frame must be addressed to the DLL address of the host or router to which the packet will be sent. ARP identifies this DLL address.

IP Version 4 has a number of important fields besides the source and destination address fields. The time to live (TTL) field ensures that packets that are misaddressed do not circulate endlessly around the Internet. The protocol field value gives the contents of the data field—ICMP message, TCP segment, UDP datagram, and so forth.

IP Version 6 requires networking and security professionals to understand a new approach to expressing addresses. IPv6 addresses are 128 bits long. For human comprehension, they are expressed in hexadecimal notation. In addition, there are certain rules that compress the hex version of the address to reduce its length.

The IPv6 main header is simpler than the IPv4 header. The IPv6 header saves time on each router by not checking for errors and not allowing packet fragmentation. The IPv6 header has a mechanism for specifying quality-of-service parameters for the packet and for marking a packet as a member of a flow of packets with predefined quality-of-service parameters.

IPv4 has an inelegant method for handling options. A router must read through all options to see which ones are important to it, as opposed to the destination host. IPv6 expresses options as extension headers the hop-by-hop extension header, which every router should read along the way. The router can probably ignore subse-quent extension headers. Reading few if any extension headers reduces processing time, which lowers the cost per packet. [The first is the hop-by-hop] above.

The Transmission Control Protocol (TCP) has sequence numbers that allow the receiving transport process to place arriving TCP segments in order. The TCP header has several flag fields that indicate whether the segment is a SYN, FIN, ACK, or RST segment. Connection openings use a three-step handshake that uses SYN segments. Normal closes involve a four-step message exchange that uses FIN segments. Resets close a connection with a single segment (RST) instead of the normal four.

UDP has a very simple header, with two port number fields, a UDP length field, and a UDP checksum field that is often not used. It is lightweight but unreliable. In this chapter, we learned that UDP has another limitation. It cannot fragment application messages, so application messages must be of limited size.

END-OF-CHAPTER QUESTIONS

Thought Questions

8-1. a) How does the postal service use hierarchical sorting? b) How does this simplify delivery decisions?

8-2. Give a non-network example of hierarchical addressing, and discuss how it reduces the amount of work needed in physical delivery. Do not use any example in the book, the postal service, or the telephone network.

8-3. A client PC has two simultaneous connections to the same webserver application program on a webserver. (Yes, this is possible, and in fact, it is rather common.) What will be different between the TCP segments that the client sends on the two connections? (Hint: Consider all the fields in a TCP segment.)

8-4. A router that has the routing table in Figure 8-7 receives an incoming IPv4 packet. The source IP address in the arriving packet is 10.55.72.234. The destination IP address is 10.4.6.7. The TTL value is 1. The Protocol field value is 6. What will the router do with this packet? (Hint: Consider all the fields in the IP and TCP headers.)

Perspective Questions

8-5. What was the most surprising thing you learned in this chapter?

8-6. What was the most difficult material for you in this chapter?