CHAPTER 3

NETWORK LAYER/INTERNET PROTOCOLS

You will learn about the following in this chapter:

- IP operation, fields and functions
- Fragmentation and reassembly of datagrams
- ICMP messages and meanings

IP

IP (Internet Protocol) does most of the work in the TCP/IP protocol suite. All protocols and applications within the TCP/IP suite run on top of IP and utilize it for logical Network layer addressing and transmission of datagrams between internet hosts. IP maps to the Internet layer of the DoD and to the Network layer of the OSI models. ICMP (Internet Control Message Protocol) is considered an integral part of IP and uses IP for its datagram delivery; we will discuss ICMP later in this chapter. Figure 3.1 shows how IP maps to the DoD and OSI models.

How Do I Buy a Ticket on the IP Train?

The phrase ”runs on top of” might sound as if an application or protocol is buying a ticket to ride on an IP train. This term is not restricted to IP. In reality, it is industry jargon used to describe any upper-layer protocol or application coming down the OSI model and utilizes another lower-layer protocol’s features (in this case IP at the Network layer).

IP provides an unreliable, connectionless datagram delivery service, which means that IP does not guarantee that an IP datagram successfully reaches its destination; rather it provides best effort of delivery, which means it sends it out and hopes it gets there. IP simply adds logical source and destination network layer addresses and delivers the datagram relying on other layers to guarantee it gets to its destination. If there is a problem with delivery, IP relies on ICMP to send messages when it encounters an error. When IP encounters an error in delivery, it simply trashes the datagram, causing an ICMP message to be sent to the source host detailing what kind of delivery problem occurred. IP relies on upper layers to provide reliability; for example, TCP (which will be discussed in more detail in Chapter 8, “Transmission Control Protocol (TCP)”).

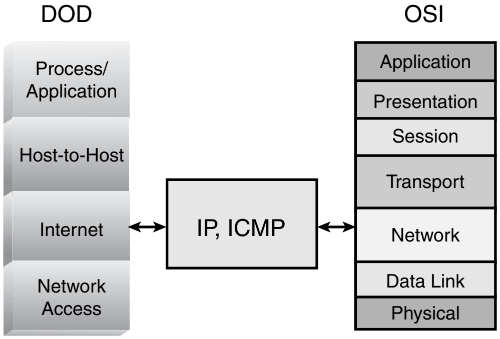


FIGURE 3.1
IP provides logical addressing and connectionless delivery of datagrams for all protocols within the TCP/IP suite.

The Internet Protocol’s primary function is logical network layer addressing of hosts and delivery of information in the form of datagrams between hosts. IP addressing is discussed in detail in Chapter 2, “IP Addressing.” IP also performs other important functions such as fragmentation and reassembly, which are necessary when datagrams are too large to be sent by a source host and must be broken up into smaller datagrams. Because IP is connectionless it does not require a connection between hosts. It does not sequence, acknowledge, or control the flow of data between hosts. IP treats each datagram as a separate entity; it merely addresses the datagram and sends it out, hoping it reaches the destination.

IP receives a stream of data from UDP or TCP, breaks up this information into chunks, and addresses and packages each piece as a datagram, which then can be sent to a destination host across the network. Routers and routing protocols determine the path selection between a source and destination, which we discuss in more detail in Chapters 5, “IP Routing” and 6, “Routing Protocols.”

RFC 791

RFC 791 defines IP, and its fields and functionality. In this chapter we will look at an IP header and its fields and examine the other protocols that reside on the Internet layer. We will discuss IP routing in Chapters 5 and 6.

IP Header

Figure 3.2 shows the format of an IP datagram as defined by RFC 791. An IP header contains a minimum of 20 bytes, unless options are present. Figure 3.3 shows an IP header as seen through a Sniffer protocol analyzer. We will describe each field after the figures.

FIGURE 3.2
The Internet Protocol header provides for identification of logical source and destination network addresses.

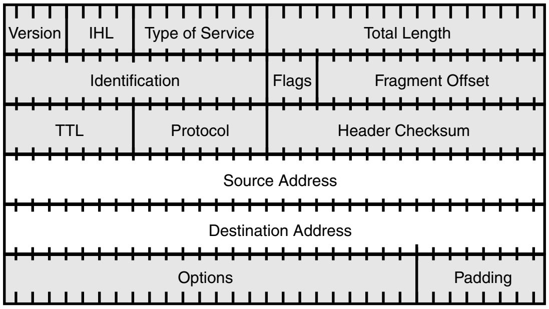
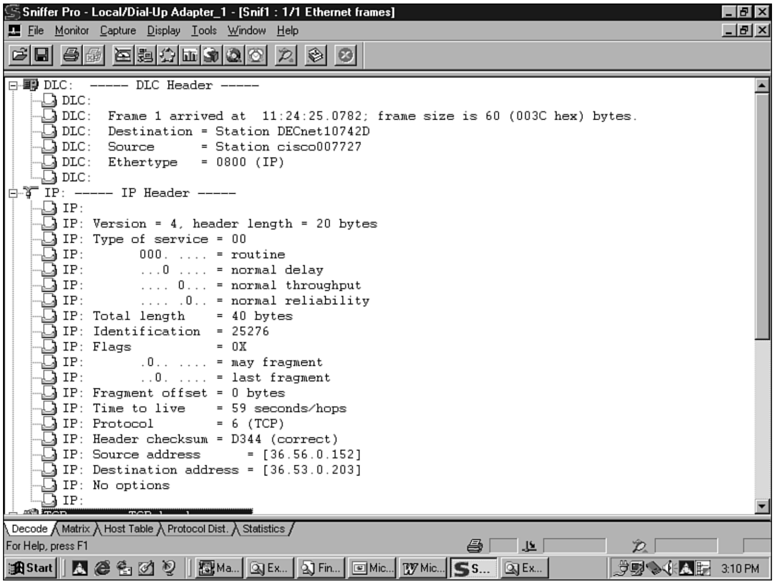


FIGURE 3.3
Note the Ethertype value contained within the DLC (Data Link Control) header states that protocol type 0x0800 or IP is the protocol being carried within this frame.


Sniffer Pro - Local/Dial-Up Adapter_1 - [Snif1 : 1/1 Ethernet frames]
File Monitor Capture Display Tools Window Help
Decode
Matrix
Host Table
Protocol Dist.
Statistics
For Help, press F1
Start
Ma...
Ex...
Fin...
Mic...
Mic...
S...
Ex...
3:10 PM

```
DLC:  ----- DLC Header -----
  DLC:
  DLC:  Frame 1 arrived at  11:24:25.0782; frame size is 60 (003C hex) bytes.
  DLC:  Destination = Station DECnet10742D
  DLC:  Source      = Station cisco007727
  DLC:  Ethertype   = 0800 (IP)
  DLC:
IP: ----- IP Header -----
  IP:
  IP: Version = 4, header length = 20 bytes
  IP: Type of service = 00
  IP:       000. .... = routine
  IP:       ...0 .... = normal delay
  IP:       .... 0... = normal throughput
  IP:       .... .0.. = normal reliability
  IP: Total length    = 40 bytes
  IP: Identification  = 25276
  IP: Flags           = 0X
  IP:       .0.. .... = may fragment
  IP:       ..0. .... = last fragment
  IP: Fragment offset = 0 bytes
  IP: Time to live    = 59 seconds/hops
  IP: Protocol        = 6 (TCP)
  IP: Header checksum = D344 (correct)
  IP: Source address      = [36.56.0.152]
  IP: Destination address = [36.53.0.203]
  IP: No options
  IP:
```

Take a look at the IP header in Figure 3.3. The first field is the version type, which should always be version 4, the official standard. This is followed by the header length, indicating the size of the IP header as 20 bytes. Type of service (ToS) values follow. Most often the ToS value, as in this case, will have a value of zero because ToS is seldom used. However, this trend is changing as ToS becomes more understood and more vendors implement it successfully in their products. New applications capable of setting these bits to influence a router's routing of datagrams are emerging. This allows them to request from routers a certain level of ToS service for transmitted data.

The total length of the datagram is 40 bytes, which includes the IP header and data being carried within the frame. The IP ID value given to this datagram is 25276. Note that in the Flags field the datagram can be fragmented if necessary, and that this is the last fragment. Because the fragment offset has a value of zero, we can deduce that this is the first, last, and only fragment within the stream.

The TTL value currently is set to 59 seconds, which is the amount of time left for this datagram to live on the internetwork. The next protocol being carried within the frame is TCP. IP uses the checksum value to identify frame damage. The sending host's logical IP address is 36.56.0.152 and the destination host's address is 36.53.0.203.

Version

The value within the 4-bit version field identifies the version of IP being implemented. This value will always be version 4. It is the most current and popular version of IP.

Internet Header Length

The 4-bit header length field identifies the size of the IP header in 32-bit words. All IP headers must have a minimum of 20 bytes in size unless additional options are implemented and specified within the option field, which we will discuss later in the "Type of Service" section of this chapter.

Type of Service

The ToS bits (8 total bits) within the IP header can influence the path datagrams take when being forwarded by routers from source to destination. ToS bits allow upper applications and processes to indicate the type of quality or service they require from a router. Until recently, the use of the ToS bits has been nonexistent. However, many companies now implement them to facilitate more intelligent path selection. If ToS has not been implemented, this field will have a value of zero. Because the use of ToS is uncommon, zero is an expected value. RFC's 1340 and 1349 describe the specific use and functions of these bits. The following explains the various ToS bits.

Bits 0–2: Precedence Bits

Bits 0–2, known as precedence bits, when used in the following combinations mean different things, and typically only the government uses these implementations to define the importance of a datagram. This value uses the options field within the IP header, discussed later in this chapter, to further describe the type of precedence requested. The precedence value typically describes the type of security levels requested defined by the Defense Intelligence Agency. Table 3.1 describes the various precedence bits set.

TABLE 3.1 Bits 0–2 = Precedence

Bit Position	Description
000	Routine information
001	Priority information
010	Immediate delivery
011	Flash

TABLE 3.1 Continued

Bit Position	Description
100	Flash override
101	Critical information
110	Internetwork control
111	Network control

Until recently most applications did not support or use precedence bits. However, they comply with governmental network implementations, which require multilevel security functions. No further discussion of these bits will follow. Precedence is described in further detail in Chapter 8. The organization that chooses to implement these bit must specifically define the precedence bits, and their use and meanings.

The next three ToS bits are the most commonly used for influencing traffic patterns.

Bit 3

Bit 3 can have one of two values:

- 0 = Normal delay
- 1 = Low delay

Delay is based on the end-to-end propagation delay of data transmitted over a link. When multiple paths exist to a destination, an application can direct routers en route to select the path with the least delay—a faster path.

Bit 4

Bit 4 can have one of two values:

- 0 = Normal throughput
- 1 = High throughput

When an application requires a path to a destination that offers high throughput rates, it sets the throughput bit to 1. Routers forward traffic along paths that support the highest possible data rate, measured as the bandwidth capacity of a link between source and destination.

Bit 5

Bit 5 can have one of two values:

- 0 = Normal reliability
- 1 = High reliability

Routers measure a link's reliability by the number of errors encountered and lost datagrams it experiences when forwarding or receiving across an interface. If multiple paths exist and one appears to be more reliable than the other, a router forwards traffic for applications, setting the reliability bit to 1 across this interface. Critical applications that cannot tolerate data loss may request this type of service.

Bits 6 and 7 (Reserved)

Bits 6 and 7 are reserved and not currently used.

Upper-layer applications or processes can request for routers to deliver their data along paths that meet their service requirements. For ToS delivery to work all routers and routing protocols within the path from source to destination must understand, and be configured to forward datagrams based on the ToS designation. Not all routing protocols understand these bits. The following routing protocols understand ToS:

- OSPF
- EIGRP
- IGRP
- BGP

The following protocols do not understand ToS:

- RIP v1
- RIP v2

Although a routing protocol might have the capability to understand and act upon these bits, it still requires configuration. If someone has not configured the router to support ToS, it simply ignores this information and forwards the datagram the best it can. Let's look at an example of how ToS works. When an application requests datagrams to be sent through a low delay path, it sets bit 3 to a value of 1. Routers along the path attempt to send datagrams across links offering faster transmissions, such as 100Mb Ethernet versus slower WAN links. Keep in mind that delay is measured by the round-trip propagation delay across the link. Therefore, a link with a lower delay would be the preferred path.

High throughput translates to high-capacity links, which have the capability to carry larger amounts of data in a shorter time frame. This is useful for large file transfers. Routers measure the capacity of a link in terms of bandwidth, which is the transfer rate in bits across the interface. For example, an Ethernet 100Mbps link carries 100 million bits per second, which is faster than a 10Mbps (10 million bit per second) interface. If reliability is the objective, such as the case in an application performing critical processes or requiring security, an application can request a more reliable transmission path by setting bit 5 to a value of 1. This might be a transaction-based processing application, which requires access to a company's fault-tolerant backbone.

It is very important to understand that by setting these bits, you change the way routers route traffic. Without a thorough understanding of the types of protocols, applications, traffic flows and protocol timers within your network, this manipulation could have catastrophic results. It is imperative that you baseline your network thoroughly before attempting to implement them. When properly used, they can dramatically increase the performance of your network and network applications.

Total Length

The 2-byte total length field within the IP header defines the length in bytes of the entire IP datagram. This value includes the IP header and data being carried within the datagram.

Identification

The sender gives each IP datagram an ID value prior to transmission, which is found in the 2-byte identification field. This ID uniquely identifies a datagram or stream of datagrams. The destination host uses this ID to reassemble the datagrams received. When a source host's IP process receives a large stream of contiguous data from UDP or TCP for datagram packaging, it breaks up this stream (fragmentation) when it receives a packet that is too large for the transmission medium. IP then assigns all datagrams belonging to the stream the same ID.

When transmitted from source to destination, datagrams can take different paths with widely varying characteristics, causing them to arrive out of order. The destination host uses this ID to recognize that all the datagrams belong to a stream. It then reassembles them in the correct order based on the fragment offset value, which we will explain later in this section.

Flags

Bit 0 = Reserved. Must be a value of zero.

Bit 1 = Can have one of two values:

- 0 = May Fragment
- 1 = Don't Fragment

The 3-bit flags field is used on hosts and gateways for fragmentation purposes. If a host or a gateway supports fragmentation, it can break a stream of data into smaller pieces before transmission. If it does not support fragmentation (the don't-fragment bit is set), the host or gateway cannot fragment the stream. Typically, the sending host has the responsibility of performing fragmentation. The destination host reassembles the datagrams into the original stream before passing it up to the upper layer (TCP or UDP) for processing. However, this is not always the case.

When a source host sends a datagram that reaches a segment and is too large to be forwarded, the gateway performs the fragmentation, breaking the datagram into smaller units acceptable for the media. Having intervening gateways perform fragmentation of datagrams in transit is not a good idea. Routers forced to perform this function require additional resources and add unnecessary overhead and latency to the delivery of the datagram. Because different underlying network architectures support varying frame sizes, find out what the lowest value MTU (Maximum Transmission Unit) size is for your network and configure your hosts and gateways to support it. For example, Ethernet has a maximum frame size of 1518 bytes, whereas Token-Ring frames might range from 4,500 bytes to 17,800 bytes.

The DF bit has another use. Some implementations use the DF bit to dynamically discover the MTU size of the network end to end. If intervening routers have this bit enabled, when an end host attempts to send a datagram that is larger than the next segment along the transmission path, the router will not forward the frame. Instead, the router drops the datagram and kicks back to the source host an ICMP message indicating the datagram is too large and has exceeded the maximum segment size. The host then can use this information to resize its next datagram. This process will continue until the sending host has discovered the proper size to send, allowing intervening routers to simply forward datagrams without performing fragmentation en route to its destination.

Bit 3 = Can have one of two values:

- 0 = Last Fragment
- 1 = More Fragments

The Last or More Fragments bit indicates whether this is the last datagram within a stream or more datagrams are to follow. If there is only one datagram this bit will be zero, indicating that this is the first and the last, meaning it is the only one. When a destination host receives a datagram, it notes the ID value and checks whether this bit is a one or zero, indicating that this is the last datagram or more are expected. If it expects more, this host holds the datagrams in memory until all others with the same ID arrive and the stream is reassembled and passed up to the appropriate upper-layer protocol for processing. By matching the IDs and referencing the last or more bit, the host knows when to stop expecting future datagrams within this stream and when to start reassembling them.

Fragment Offset

The sending host uses the 13-bit fragment offset value to identify the position of the datagram with the stream being sent—the order in which this datagram belongs when more than one is sent with the same IP identification. The sending host always assigns the first datagram with an offset value of zero. It assigns the second, third, or more datagrams with a number based on the MTU size. The receiver uses the fragment offset value to put the datagrams within the same stream back together upon reception or detect that one is missing within the stream.

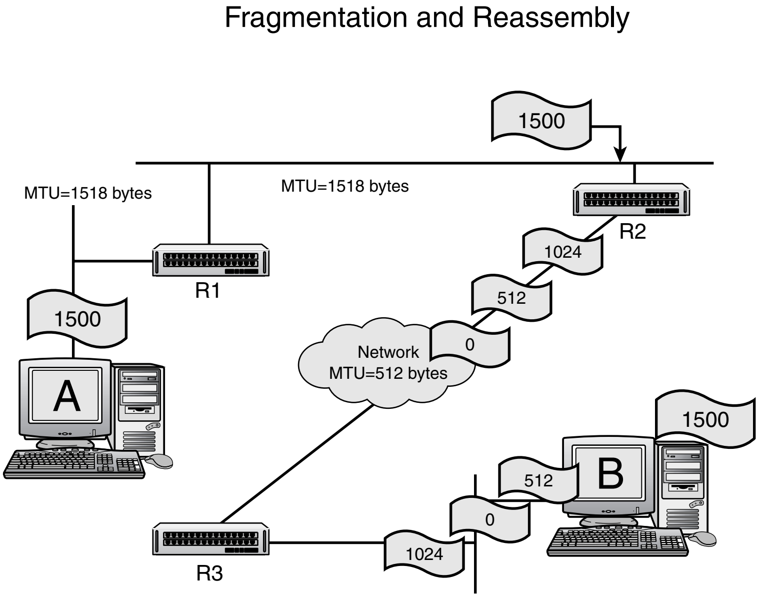
For example, if a sending host has three datagrams to send that belong to a stream, it goes through this process:

1. Assigns each of these datagrams the same ID.
2. Sets the More Fragments bit to a one on the first two datagrams to indicate that these are not the only datagrams in this stream and that more are to follow.
3. Sets the Last Fragment bit on the final datagram in the stream.
4. Each datagrams offset bears an offset value identifying where this datagram belongs in the stream.

Figure 3.4 shows an example of a sending host and a receiving host using the fragment offset value to determine in what order datagrams belong. In Figure 3.4 Station A wants to communicate with Station B and sends a single datagram containing 1500 bytes of information on the wire. Note that the local segment attached to segment A can accept a maximum transmission unit (MTU) size of 1,518 bytes, which is sufficient to support this frame size. The datagram makes it through Router 1, which forwards it on to Router 2 as is. Note that the segment between Router 1 and 2 also accepts an MTU of 1,518.

In Figure 3.4 the WAN segment between Routers 2 and 3 will not accept a segment larger than 512 bytes, so Router 2 must break up the original datagram sent by Host A to fit this datagram on the wire; this is called fragmentation. There now are three datagrams forwarded across this link to Router 3 and ultimately to their final destination, Host B. Once these datagrams reach Host B, this host performs the reassembly prior to sending them up to the upper-layer process. Host B (the receiving host) uses the IP identification field, Last and More Fragments bit, and the fragment offset to piece the datagram back together.

FIGURE 3.4
In this figure, Host A sends a 512-byte datagram, which is too large for the WAN. Router 2 breaks up the datagram (fragments) into smaller chunks, forwarding them. The destination host is responsible for reassembly.



Using Figure 3.4 as an example, the receiving host expects to find and reassemble the datagrams in the following order:

1. The first datagram in a stream always has an offset value of zero, indicating the first one in the stream.
2. The second datagram has an offset value of 512.
3. The third datagram has an offset value of 1024.

The receiver stores datagrams in its memory buffer waiting for messages within the stream to arrive. However, it does not always receive the datagrams in order. For example, if the second datagram arrives first with offset 512, the receiver knows to expect several others because

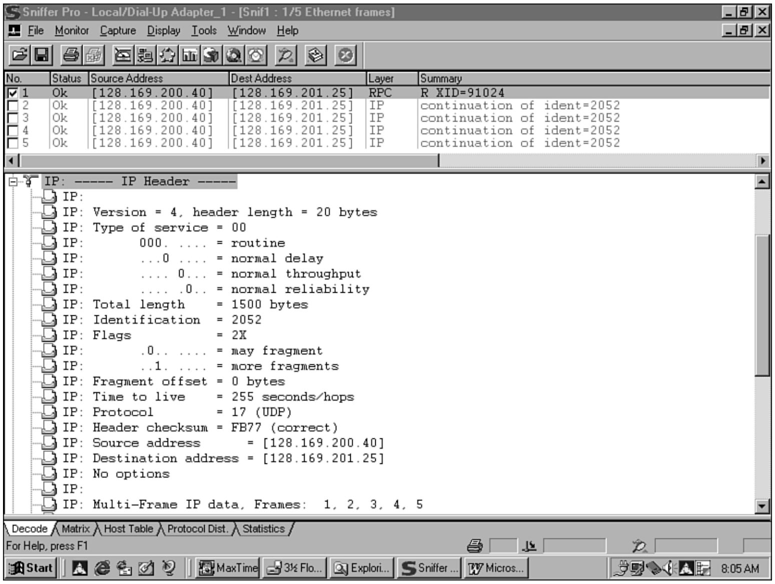
- The sending host has set More Fragments bit in the second datagram.
- This is not the Last Fragment.
- The receiver has not received a datagram with the same ID containing fragment offset zero, which indicates the beginning.

Because it has received a datagram with the More Fragments bit set and has not received a datagram with a Last Fragment bit set, it knows to expect more datagrams. Once the receiver gets all the fragments within the stream, it can put them in the correct order and pass them up to the upper-layer application for processing.

Figures 3.5, 3.6, and 3.7 depict another example of fragmentation and reassembly as seen through a Sniffer. Figure 3.5 details the first datagram within the stream; Figure 3.6 details the second datagram within the stream and Figure 3.7 details the last datagram within the stream. In Figure 3.5 notice that in frame 1 (highlighted in the detail pane), the IP ID value of this frame (frame 1) and all other frames (frames 2–5) shown in the summary pane bear the same value of 2052, shown as "continuation of indent=2052." This means that frames 1–5 all belong to stream 2052.

Note that the sending host has set the More Fragments bit within the first frame, indicating the this datagram has more fragments or datagrams within this stream. In other words, this is not the last datagram. Also note that the sending host set the fragment offset to zero, indicating the first fragment in series 2502.

FIGURE 3.5
The More fragments bit is set indicating more datagrams will follow. The fragment offset set to zero indicates this is the first datagram within the stream.



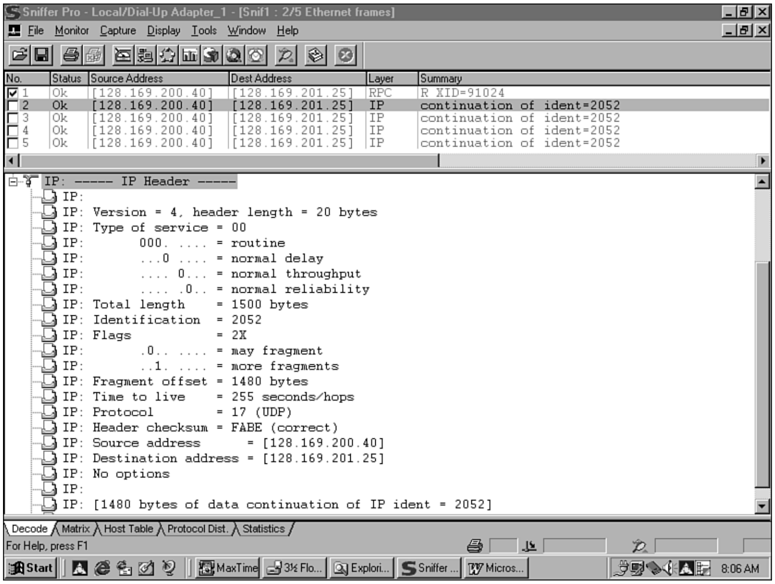
Sniffer Pro - Local/Dial-Up Adapter_1 - [Snif1 : 1/5 Ethernet frames]

```
No.   Status Source Address    Dest Address      Layer Summary
☑ 1   Ok     [128.169.200.40]  [128.169.201.25]  RPC   R XID=91024
□ 2   Ok     [128.169.200.40]  [128.169.201.25]  IP    continuation of ident=2052
□ 3   Ok     [128.169.200.40]  [128.169.201.25]  IP    continuation of ident=2052
□ 4   Ok     [128.169.200.40]  [128.169.201.25]  IP    continuation of ident=2052
□ 5   Ok     [128.169.200.40]  [128.169.201.25]  IP    continuation of ident=2052
```

```
IP: ----- IP Header -----
  IP:
  IP: Version = 4, header length = 20 bytes
  IP: Type of service = 00
  IP:       000. .... = routine
  IP:       ...0 .... = normal delay
  IP:       .... 0... = normal throughput
  IP:       .... .0.. = normal reliability
  IP: Total length    = 1500 bytes
  IP: Identification  = 2052
  IP: Flags           = 2X
  IP:       .0.. .... = may fragment
  IP:       ..1. .... = more fragments
  IP: Fragment offset = 0 bytes
  IP: Time to live    = 255 seconds/hops
  IP: Protocol        = 17 (UDP)
  IP: Header checksum = FB77 (correct)
  IP: Source address      = [128.169.200.40]
  IP: Destination address = [128.169.201.25]
  IP: No options
  IP:
  IP: Multi-Frame IP data, Frames:  1, 2, 3, 4, 5
```

Figure 3.6 shows the second fragment in stream 2502. Note that the More Fragments bit is set, indicating that more datagrams follow this one. Note that the fragment offset is 1480, which indicates this is not the first datagram (because it is not zero). Because the last fragment bit is not set, we know it is not the last but somewhere in the middle (in this case second). The destination host uses this information to put the fragments in the stream in the right order when it receives all the fragments within the stream.

FIGURE 3.6
The fragment offset is used by the receiving host to reassemble datagrams.



Sniffer Pro - Local/Dial-Up Adapter_1 - [Snif1 : 2/5 Ethernet frames]
File Monitor Capture Display Tools Window Help
Decode Matrix Host Table Protocol Dist. Statistics
For Help, press F1
Start MaxTime 3½ Flo... Explori... Sniffer ... Micros...
8:06 AM

```
No.   Status Source Address    Dest Address      Layer Summary
☑ 1   Ok     [128.169.200.40]  [128.169.201.25]  RPC   R XID=91024
□ 2   Ok     [128.169.200.40]  [128.169.201.25]  IP    continuation of ident=2052
□ 3   Ok     [128.169.200.40]  [128.169.201.25]  IP    continuation of ident=2052
□ 4   Ok     [128.169.200.40]  [128.169.201.25]  IP    continuation of ident=2052
□ 5   Ok     [128.169.200.40]  [128.169.201.25]  IP    continuation of ident=2052
```

```
IP: ----- IP Header -----
  IP:
  IP: Version = 4, header length = 20 bytes
  IP: Type of service = 00
  IP:       000. .... = routine
  IP:       ...0 .... = normal delay
  IP:       .... 0... = normal throughput
  IP:       .... .0.. = normal reliability
  IP: Total length    = 1500 bytes
  IP: Identification  = 2052
  IP: Flags           = 2X
  IP:       .0.. .... = may fragment
  IP:       ..1. .... = more fragments
  IP: Fragment offset = 1480 bytes
  IP: Time to live    = 255 seconds/hops
  IP: Protocol        = 17 (UDP)
  IP: Header checksum = FABE (correct)
  IP: Source address      = [128.169.200.40]
  IP: Destination address = [128.169.201.25]
  IP: No options
  IP:
  IP: [1480 bytes of data continuation of IP ident = 2052]
```

Figure 3.7 skips fragment 3 and shows the final fragment in the stream. Note that this fragment belongs to the stream because it has the identification value of 2502 like all the other fragments. We know this is the last fragment because Last Fragment is set. The fragment offset of 7400 is higher than the previous fragments so the destination host knows this is the last fragment and starts at the offset.

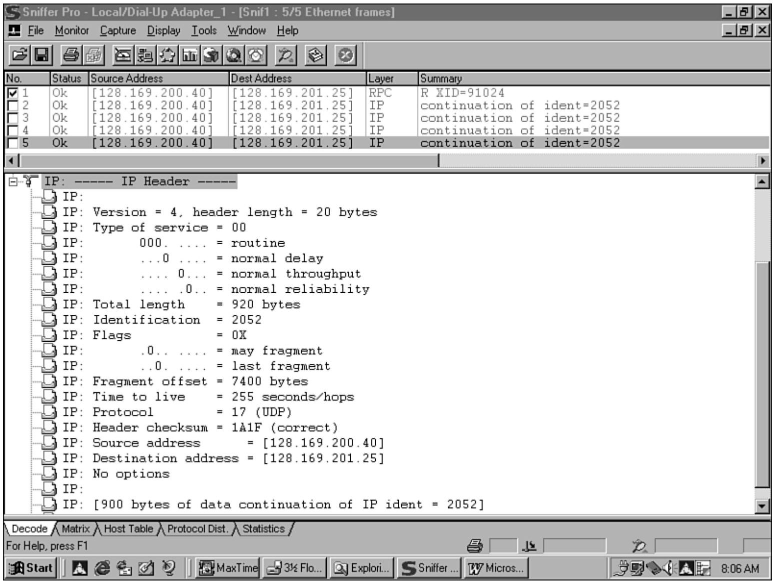
Time To Live

All devices processing a datagram decrement the 1-byte TTL value, which is measured in seconds. This value has two main purposes:

- To define the maximum time a datagram may live on the Internet prior to being discarded.
- To ensure that undeliverable datagrams also are discarded.

Before routers existed on the Internet, to ultimately remove a lost datagram from the network it was necessary to recognize and limit the time it traversed the network. The TTL timer was used for this purpose. It remains today and is used for various purposes, such as tracing a route to a destination host or network, which we will discuss later in this chapter. Basically, whenever a device such as a gateway processes a datagram, it must decrement this value by at least one before forwarding it. Each TTL value indicates a value of one second, which is more time than it should take for a router to process a datagram. This value can typically be equated to a hop count indicating the datagram has passed through *x* number of routers by determining how many TTL units this value has been decremented.

FIGURE 3.7
Note the Last fragment bit is set indicating to the receiver that this is the last datagram within the stream.


Sniffer Pro - Local/Dial-Up Adapter_1 - [Snif1 : 5/5 Ethernet frames]
File Monitor Capture Display Tools Window Help
Decode Matrix Host Table Protocol Dist. Statistics
For Help, press F1
Start
MaxTime
3½ Flo...
Explori...
Sniffer ...
Micros...
8:06 AM

```
No.  Status Source Address    Dest Address       Layer Summary
☑ 1  Ok     [128.169.200.40]  [128.169.201.25]   RPC   R XID=91024
☐ 2  Ok     [128.169.200.40]  [128.169.201.25]   IP    continuation of ident=2052
☐ 3  Ok     [128.169.200.40]  [128.169.201.25]   IP    continuation of ident=2052
☐ 4  Ok     [128.169.200.40]  [128.169.201.25]   IP    continuation of ident=2052
☐ 5  Ok     [128.169.200.40]  [128.169.201.25]   IP    continuation of ident=2052
```

```
IP: ----- IP Header -----
   IP:
   IP: Version = 4, header length = 20 bytes
   IP: Type of service = 00
   IP:       000. .... = routine
   IP:       ...0 .... = normal delay
   IP:       .... 0... = normal throughput
   IP:       .... .0.. = normal reliability
   IP: Total length    = 920 bytes
   IP: Identification  = 2052
   IP: Flags           = 0X
   IP:       .0.. .... = may fragment
   IP:       ..0. .... = last fragment
   IP: Fragment offset = 7400 bytes
   IP: Time to live    = 255 seconds/hops
   IP: Protocol        = 17 (UDP)
   IP: Header checksum = 1A1F (correct)
   IP: Source address      = [128.169.200.40]
   IP: Destination address = [128.169.201.25]
   IP: No options
   IP:
   IP: [900 bytes of data continuation of IP ident = 2052]
```

A network administrator can configure the starting TTL value but this value cannot exceed 255 (seconds). The general rule is that a device may not forward a datagram with a TTL value of 1 or less. If a datagram has lived on the wire for 255 seconds and has not reached its ultimate destination, it has lived too long on the network and should be removed. When this occurs, the device that the TTL expired on sends an ICMP message back to the source indicating that the datagram TTL time has exceeded. We will discuss ICMP messages later in this chapter.

Protocol

The 1-byte protocol field contains a value that specifies the next protocol expected within the datagram. This value will contain one of two values:

- 06 (TCP)
- 17 (UDP)

IP uses these values to identify to which upper-layer protocol to hand the information when it receives it.

Header Checksum

Because IP is a connectionless protocol, it does not implement any type of error correction mechanism, such as sequencing and acknowledgments. IP simply sends datagrams out, addresses them, and hopes they reach their destination. Therefore, IP uses a simple checksum, contained in this 2-byte field, to verify the integrity of the IP header and the data being carried to ensure that nothing has happened to the bits while in transit between source and destination hosts. Intervening devices must recalculate and verify this checksum value along the way because some of the fields within the IP header change while in transit (for example, time to live). If at any point in time a device deems this value invalid due to damage, it trashes the datagram without sending a message to the source. IP relies on upper-layer protocols to detect the loss of a datagram and recover from it (retransmit).

Source Address

The sending host places its logical Network layer (IP address) address within this 4-byte field for identification purposes.

Destination Address

This 4-bit field identifies the recipient's logical network layer address. This logical 32-bit IP address identifies the destination network and host.

Options

Hosts or gateways can implement optional parameters; when used, this variable-length field defines them here. However, options do not have to exist within a datagram; if implemented, all hosts and gateways must recognize and support their implementation. This could include the use of a security option as specified within by the precedence value used in the ToS field.

Padding

IP uses this variable-length field only when an IP header does not end on a 32-bit boundary. Because the IP header is expressed in 32-bit words, padding is used to ensure that this happens.

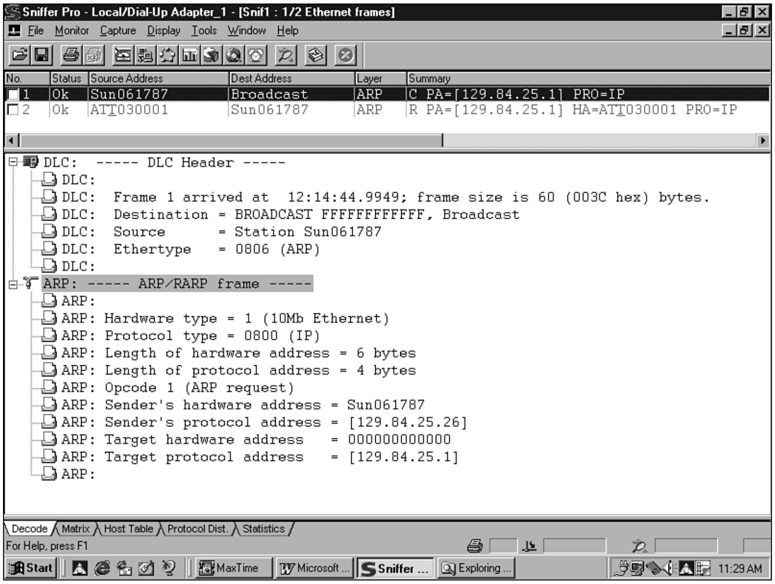
ICMP

End hosts and gateways (routers) use the Internet Control Message Protocol (ICMP), defined by RFC 792, as a control, messaging, and diagnostic protocol. ICMP exists at the Network layer of the OSI and Internet layer of the DoD models. (See Figure 3.1 to view where ICMP falls within the DoD and OSI models.) Although ICMP resides at the same layer as IP, ICMP is considered an integral part of the IP protocol. As such, it utilizes the services of IP for its delivery of messages. Figure 3.8 shows an ICMP echo message encapsulated within an IP datagram.

There are many types of ICMP messages; the ICMP echo request probably is the most common. You can use the echo request as a diagnostic tool to check connectivity between end hosts. We discuss all ICMP types in more detail later in the chapter. Note the ICMP protocol type in the IP equals one (ICMP).

Echo Request and Reply

The term *echo request and reply* describes messages you can use to test network connectivity. You can do this by using the Ping utility. The Ping program uses these ICMP echo request and reply messages. An echo request is like you shouting to a network, "Hello, are you there?" What bounces back, be it negative or positive (connectivity or no connectivity), is the echo reply. Just as with an echo, you always receive a reply.


Sniffer Pro - Local/Dial-Up Adapter_1 - [Snif1 : 1/2 Ethernet frames]
File Monitor Capture Display Tools Window Help
Decode Matrix Host Table Protocol Dist. Statistics
For Help, press F1
Start MaxTime Microsoft ... Sniffer ... Exploring ...
11:29 AM

```
No. Status Source Address  Dest Address  Layer Summary
1   Ok     Sun061787       Broadcast     ARP   C PA=[129.84.25.1] PRO=IP
2   Ok     ATT030001       Sun061787     ARP   R PA=[129.84.25.1] HA=ATT030001 PRO=IP

DLC:  ----- DLC Header -----
  DLC:
  DLC:  Frame 1 arrived at  12:14:44.9949; frame size is 60 (003C hex) bytes.
  DLC:  Destination = BROADCAST FFFFFFFFFFFF, Broadcast
  DLC:  Source      = Station Sun061787
  DLC:  Ethertype   = 0806 (ARP)
  DLC:
ARP: ----- ARP/RARP frame -----
  ARP:
  ARP: Hardware type = 1 (10Mb Ethernet)
  ARP: Protocol type = 0800 (IP)
  ARP: Length of hardware address = 6 bytes
  ARP: Length of protocol address = 4 bytes
  ARP: Opcode 1 (ARP request)
  ARP: Sender's hardware address = Sun061787
  ARP: Sender's protocol address = [129.84.25.26]
  ARP: Target hardware address   = 000000000000
  ARP: Target protocol address   = [129.84.25.1]
  ARP:
```

FIGURE 3.8
The ICMP echo message is used to verify network layer communications between hosts.

Destination hosts and gateways on occasion need to inform a source host of delivery problems, test connectivity to that host, ask for transmission to slow down, and so on. ICMP has a total of 15 different messages identified by the value contained within its type field used to inform a source host. These messages allow a source host to learn and recover from some (not all) of the problems that can occur on an internetwork. Although these messages inform a host of problems, ICMP does not guarantee a solution to these problems. Like IP, ICMP is a connectionless protocol. Hosts or gateways can send unsolicited ICMP control or diagnostic messages. ICMP uses many types of messages for different purposes.

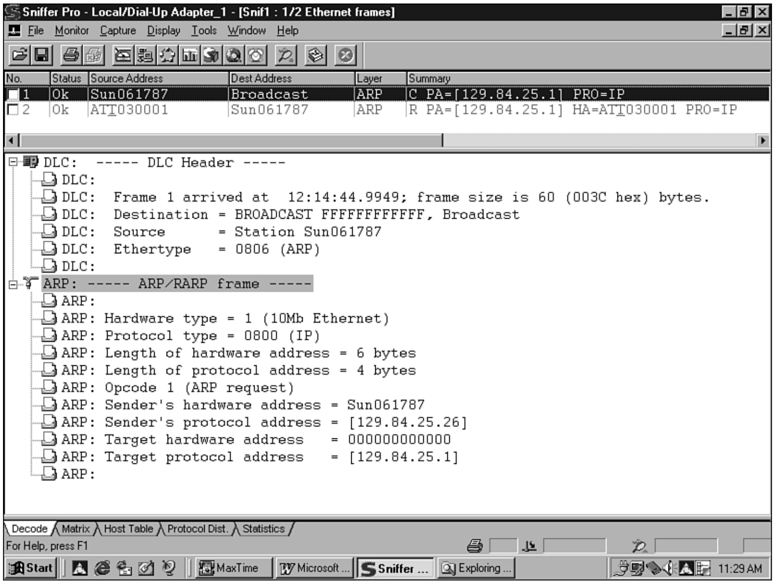
You might have used the familiar Ping utility (ICMP message types 0 and 8). Ping allows a user to send a sonar-like ping from one host to another to verify connectivity. The Ping utility utilizes the services of ICMP messages to perform this task. When a user executes the Ping command specifying a remote host's name or address, the host receives a series of ICMP messages, known as ***echo requests***. The receiving host in turn responds to each of these messages using the ICMP Echo Reply message type 0. We discuss Ping and the other various message types and their purposes later in this chapter.

ICMP Header and Message Formats

IP identifies ICMP messages contained within an IP datagram with protocol type 1. Figure 3.9 shows the general format of an ICMP echo message. The first four bytes (1-byte type field, 1-byte code field, and 2-byte checksum) have the same format for all message types. All other fields and information contained within the ICMP header vary depending on the message type being sent. We describe the various types, codes, and the checksum later in this chapter.

Hosts and gateways use ICMP as a messaging, control, and diagnostic protocol to alert a host of problems or test connectivity. Note the protocol value contained within the IP header is 1, indicating this is an ICMP message. To determine the type of ICMP message, look in the ICMP header at the type field. Once you determine the type of ICMP message, you can use the code field to further identify the purpose of the message. The type field identifies the particular ICMP messages. Some ICMP messages use different values in the code field to further specify the error condition. The checksum field covers the entire ICMP message.

FIGURE 3.9
ICMP messages have a registered value of 1. The type field within the ICMP header identifies the type of ICMP message being sent.



```
No.   Status  Source Address   Dest Address   Layer  Summary
□ 1   Ok      Sun061787        Broadcast      ARP    C PA=[129.84.25.1] PRO=IP
□ 2   Ok      ATT030001        Sun061787      ARP    R PA=[129.84.25.1] HA=ATT030001 PRO=IP
```

```
DLC:  ----- DLC Header -----
  DLC:
  DLC:  Frame 1 arrived at  12:14:44.9949; frame size is 60 (003C hex) bytes.
  DLC:  Destination = BROADCAST FFFFFFFFFFFF, Broadcast
  DLC:  Source      = Station Sun061787
  DLC:  Ethertype   = 0806 (ARP)
  DLC:
ARP: ----- ARP/RARP frame -----
  ARP:
  ARP: Hardware type = 1 (10Mb Ethernet)
  ARP: Protocol type = 0800 (IP)
  ARP: Length of hardware address = 6 bytes
  ARP: Length of protocol address = 4 bytes
  ARP: Opcode 1 (ARP request)
  ARP: Sender's hardware address = Sun061787
  ARP: Sender's protocol address = [129.84.25.26]
  ARP: Target hardware address   = 000000000000
  ARP: Target protocol address   = [129.84.25.1]
  ARP:
```

Codes

Many of the type fields contain more specific information about the error condition identified by a code value. ICMP messages have two types of codes:

- Query
- Error

Queries contain no additional information because they merely ask for information and will show a value of 0 in the code field. ICMP uses the following queries:

- Type 0 = Echo Reply
- Type 8 = Echo Request
- Type 9 = Router Advertisement
- Type 10 = Router Solicitation
- Type 13 = Timestamp Request
- Type 14 = Timestamp Reply
- Type 15 = Information Request (obsolete)
- Type 16 = Information Reply (obsolete)
- Type 17 = Address Mask Request
- Type 18 = Address Mask Reply

Error messages give specific information and will have varying values that further describe conditions. Error messages always include a copy of the offending IP header and up to 8 bytes of the data that caused the host or gateway to send the error message. The source host uses this information to identify and fix the problem reported via the ICMP error message. ICMP uses the following error messages:

- Type 3 = Destination Unreachable
- Type 4 = Source Quench
- Type 5 = Redirect
- Type 11 = Time Exceeded
- Type 12 = Parameter Problems

Table 3.1 lists all of the ICMP codes along with the ICMP message types. We will discuss error codes and the various ICMP message types to which they pertain later.

Checksum

The checksum verifies the validity of the ICMP header. The sending host performs the initial checksum calculation and places the results in this field. The receiving host performs the same calculations to assure that it does not receive data damaged in transit. If the checksum values do not match, it trashes the datagram.

Identifier

The user on the source host can set this optional value to match sent echo requests with received replies.

Sequence Number

The user on the source host can set this optional value to match sent echo requests with received replies.

ICMP Message Types

The type field identifies the type of the message sent by the host or gateway. Many of the type fields contain more specific information about the error condition. Table 3.2 lists the ICMP message types.

TABLE 3.2 ICMP Message Types

Type	Description ICMP Message Types
0	Echo Reply (Ping Reply, used with Type 8, Ping Request)
3	Destination Unreachable
4	Source Quench
5	Redirect
8	Echo Request (Ping Request, used with Type 0, Ping Reply)
9	Router Advertisement (Used with Type 9)
10	Router Solicitation (Used with Type 10)
11	Time Exceeded
12	Parameter Problem
13	Timestamp Request (Used with Type 14)
14	Timestamp Reply (Used with Type 13)
15	Information Request (obsolete) (Used with Type 16)
16	Information Reply (obsolete) (Used with Type 15)
17	Address Mask Request (Used with Type 17)
18	Address Mask Reply (Used with Type 18)

Because each of the ICMP message headers vary depending on which one is sent, we will discuss each type separately, identifying the corresponding code fields, if applicable.

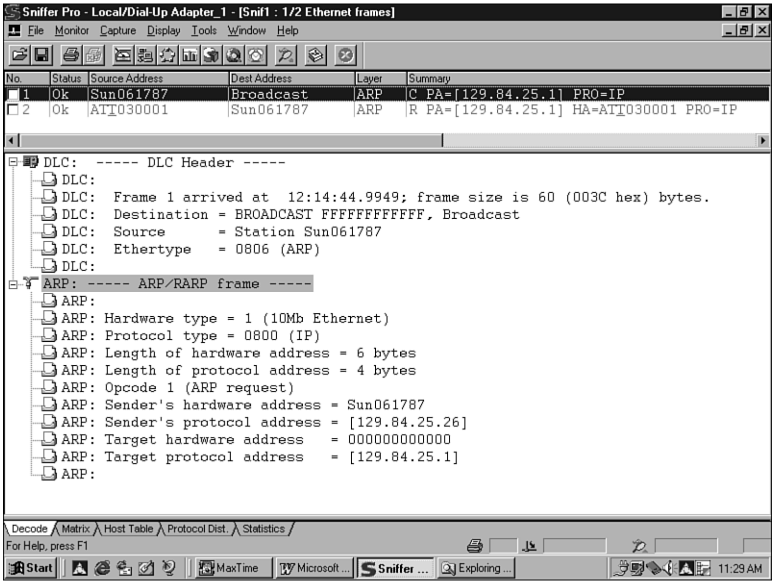
Ping: Echo Request and Reply—Types 8 and 0

We discuss the ICMP Echo Request Type 8 and Echo Reply Type 0 because ICMP uses these messages in tandem. Remote hosts use these two message types to test connectivity. As previously mentioned, the user executes the Ping utility, initiating the generation of ICMP echo requests with the expectation that the destination host sends a corresponding echo reply. Upon successful receipt of the replies to the echo requests, the messages do the following:

- Indicate a successful test.
- Assume that a valid communication path between the hosts exists.
- Assume the end host works through the Network layer.

Figure 3.10 shows an example of an echo request; Figure 3.11 shows an example of an echo reply. In Frame 1 host 36.53.0.202 sends an echo request to test the connectivity with host 36.21.0.1. Note the detail pane indicates a type 8 value stating this is an echo request. The ID value of 52743 and the sequence number of 57098 are optionally included to provide a reasonable match with the echo reply. In Frame 2 host 36.53.0.202 returns the echo reply to host 36.21.0.1. The type code 0 indicates this is a reply and the previous ID and sequence number values used in the echo request frame match.

FIGURE 3.10
This is an example of an echo request and reply generated as a result of the Ping Utility.


Sniffer Pro - Local/Dial-Up Adapter_1 - [Snif1 : 1/2 Ethernet frames]
File Monitor Capture Display Tools Window Help
Decode Matrix Host Table Protocol Dist. Statistics
For Help, press F1
Start MaxTime Microsoft ... Sniffer ... Exploring ... 11:29 AM

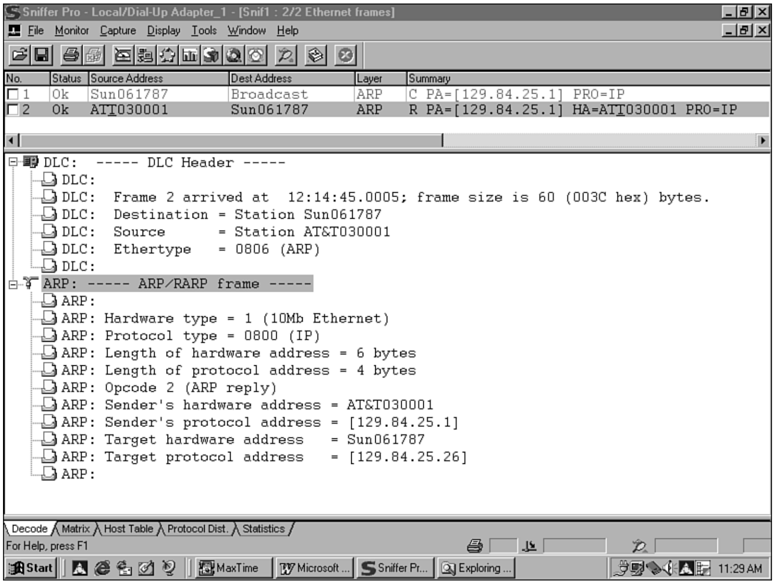
```
No.  Status Source Address Dest Address  Layer Summary
1    Ok     Sun061787      Broadcast     ARP   C PA=[129.84.25.1] PRO=IP
2    Ok     ATT030001      Sun061787     ARP   R PA=[129.84.25.1] HA=ATT030001 PRO=IP
```

```
DLC:  ----- DLC Header -----
  DLC:
  DLC:  Frame 1 arrived at  12:14:44.9949; frame size is 60 (003C hex) bytes.
  DLC:  Destination = BROADCAST FFFFFFFFFFFF, Broadcast
  DLC:  Source      = Station Sun061787
  DLC:  Ethertype   = 0806 (ARP)
  DLC:
ARP: ----- ARP/RARP frame -----
  ARP:
  ARP: Hardware type = 1 (10Mb Ethernet)
  ARP: Protocol type = 0800 (IP)
  ARP: Length of hardware address = 6 bytes
  ARP: Length of protocol address = 4 bytes
  ARP: Opcode 1 (ARP request)
  ARP: Sender's hardware address = Sun061787
  ARP: Sender's protocol address = [129.84.25.26]
  ARP: Target hardware address   = 000000000000
  ARP: Target protocol address   = [129.84.25.1]
  ARP:
```

Destination Unreachable—Type 3

ICMP Type 3 message Destination Unreachable alerts a source host of delivery problems encountered while trying to reach the destination. Note that a destination host sends only code types 2 and 3; a router can send all codes. Destination Unreachable uses several code values to further describe the function of the ICMP message being sent. Each code type describes a different delivery problem encountered, as shown here:

FIGURE 3.11
This is an ICMP echo reply message sent in response to a previously received echo request.


Sniffer Pro - Local/Dial-Up Adapter_1 - [Snif1 : 2/2 Ethernet frames]
File Monitor Capture Display Tools Window Help
Decode Matrix Host Table Protocol Dist. Statistics
For Help, press F1
Start
MaxTime
Microsoft ...
Sniffer Pr...
Exploring ...
11:29 AM

```
No.  Status  Source Address  Dest Address  Layer  Summary
1    Ok      Sun061787       Broadcast     ARP    C PA=[129.84.25.1] PRO=IP
2    Ok      ATT030001       Sun061787     ARP    R PA=[129.84.25.1] HA=ATT030001 PRO=IP
```

```
DLC:  ----- DLC Header -----
  DLC:
  DLC:  Frame 2 arrived at  12:14:45.0005; frame size is 60 (003C hex) bytes.
  DLC:  Destination = Station Sun061787
  DLC:  Source      = Station AT&T030001
  DLC:  Ethertype   = 0806 (ARP)
  DLC:
ARP: ----- ARP/RARP frame -----
  ARP:
  ARP: Hardware type = 1 (10Mb Ethernet)
  ARP: Protocol type = 0800 (IP)
  ARP: Length of hardware address = 6 bytes
  ARP: Length of protocol address = 4 bytes
  ARP: Opcode 2 (ARP reply)
  ARP: Sender's hardware address = AT&T030001
  ARP: Sender's protocol address = [129.84.25.1]
  ARP: Target hardware address   = Sun061787
  ARP: Target protocol address   = [129.84.25.26]
  ARP:
```

0 = Network Unreachable

This message indicates that the router cannot find the destination network (does not exist or has failed) or has no route to this network. In other words, the router cannot deliver or forward an IP datagram to the destination network. This could be the result of a network that is beyond the maximum distance limitation for the routing protocol in use and is therefore considered unreachable (too far). When a client attempts to connect to a host on a network that is unreachable, a gateway generates this message to alert the source host of the problem. You can think of this message as the gateway saying to the sending host, “The street you are trying to locate is not found or is too far to reach.”

1 = Host Unreachable

The host unreachable message alerts the sending host that the destination host requested cannot be found. This could happen because this host has been turned off or does not exist. You can think of this message as the gateway saying to the sending host, “I found the street you were looking for, but the house you are trying to find is not there.”

2 = Protocol Unreachable

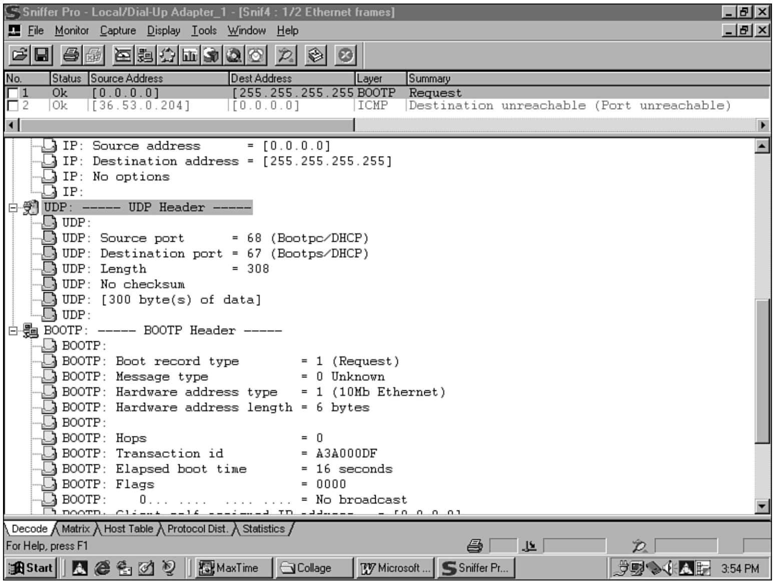
Protocol unreachable indicates that the Transport layer protocol (UDP or TCP) is not available. The destination host or an intervening gateway might send this message. You can think of this message as saying, “The transport layer protocol you are attempting to communicate with is not active on this host.”

3 = Port Unreachable

A port unreachable message indicates that the process or application the source host is attempting to establish a connection with is not active on the destination host. Typically this type of message is sent when an application has not been started or has failed on this host. The destination host or an intervening gateway might send this message. You can think of this message as saying, "The process or application you are attempting to communicate with is not active on this host," or, "I found the street, I found the house, the lights were on, but no one was home."

Figure 3.12 shows a request being sent from a BOOTP client looking for a BOOTP server. Figure 3.13 shows an example of an ICMP destination port unreachable message generated because the router or gateway could not find the BOOTP server, or the server was unavailable. We discuss BOOTP in Chapter 4, "Address Resolution," and UDP in Chapter 9, "User Datagram Protocol (UDP)."

FIGURE 3.12
In frame one, highlighted in the summary pane and show in the detail pane, we see a BOOTP client, "UDP port=68," sending a broadcast to all hosts using UDP port 67, which identifies a BOOTP Server process requesting an IP address.


Sniffer Pro - Local/Dial-Up Adapter_1 - [Snif4 : 1/2 Ethernet frames]
File Monitor Capture Display Tools Window Help
Decode Matrix Host Table Protocol Dist. Statistics
For Help, press F1
Start MaxTime Collage Microsoft ... Sniffer Pr...
3:54 PM

```
No.  Status Source Address   Dest Address        Layer  Summary
□1   Ok     [0.0.0.0]        [255.255.255.255 BOOTP  Request
□2   Ok     [36.53.0.204]    [0.0.0.0]           ICMP   Destination unreachable (Port unreachable)
```

```
    IP: Source address      = [0.0.0.0]
    IP: Destination address = [255.255.255.255]
    IP: No options
    IP:
⊟ UDP: ----- UDP Header -----
    UDP:
    UDP: Source port      = 68 (Bootpc/DHCP)
    UDP: Destination port = 67 (Bootps/DHCP)
    UDP: Length           = 308
    UDP: No checksum
    UDP: [300 byte(s) of data]
    UDP:
⊟ BOOTP: ----- BOOTP Header -----
    BOOTP:
    BOOTP: Boot record type        = 1 (Request)
    BOOTP: Message type            = 0 Unknown
    BOOTP: Hardware address type   = 1 (10Mb Ethernet)
    BOOTP: Hardware address length = 6 bytes
    BOOTP:
    BOOTP: Hops                    = 0
    BOOTP: Transaction id          = A3A000DF
    BOOTP: Elapsed boot time       = 16 seconds
    BOOTP: Flags                   = 0000
    BOOTP:    0... ....  .... .... = No broadcast
    BOOTP: Client self-assigned IP address    = [0.0.0.0]
```

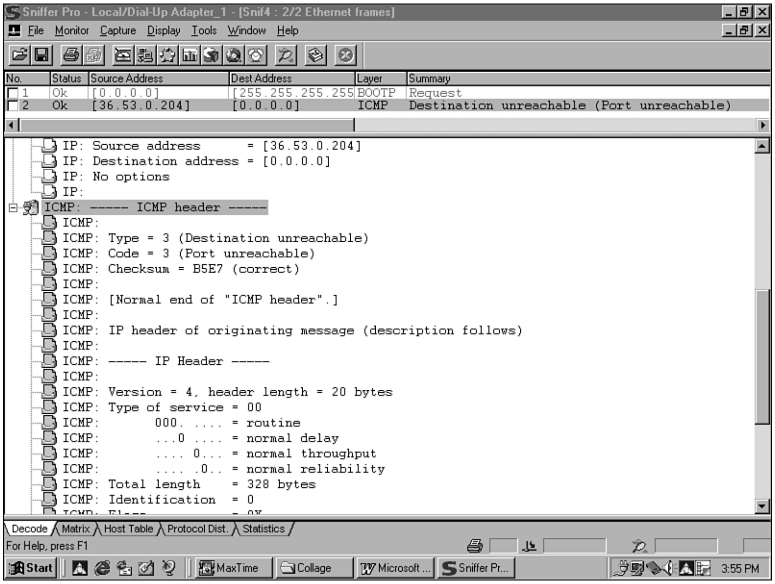
In Figure 3.13 note that the gateway has added a copy of the offending IP header within the ICMP header that caused the error from frame one. By including a copy of the offending IP header, the source might be able to use this information to correct the problem that resulted in this ICMP message being sent.

4 = Fragmentation is needed, but don't-fragment bit set

This message occurs when a router receives a datagram that requires fragmentation, but the router has the DF (don't-fragment) flag turned on. We discussed fragmentation earlier in the chapter. If you recall, the sending host generally has the responsibility of fragmentation. The receiver has the responsibility of reassembly.

However, when a router cannot forward a datagram because it is too big, if allowed the router might fragment the datagram further before transmitting it to an attached segment. If the router has the DF bit set, this will not happen and the router will trash the datagram. It then generates a message to alert the sender of this action by sending a Type 3, Code 4 message. The fragmentation bit also can determine the maximum packet size or MTU that hosts can transmit end to end along the communication path.

FIGURE 3.13
In frame two, highlighted in the summary pane and shown in the detail pane, we see an ICMP message being sent by a gateway (36.53.0.204) stating the previous request failed because the port request (68) is not active and therefore unreachable. As you can see in the detail pane, immediately following the IP header is the ICMP header.


Sniffer Pro - Local/Dial-Up Adapter_1 - [Snif4 : 2/2 Ethernet frames]
File Monitor Capture Display Tools Window Help

```
No.  Status Source Address   Dest Address       Layer  Summary
□ 1  Ok     [0.0.0.0]        [255.255.255.255   BOOTP  Request
□ 2  Ok     [36.53.0.204]    [0.0.0.0]          ICMP   Destination unreachable (Port unreachable)
```

```
   IP: Source address      = [36.53.0.204]
   IP: Destination address = [0.0.0.0]
   IP: No options
   IP:
ICMP: ----- ICMP header -----
   ICMP:
   ICMP: Type = 3 (Destination unreachable)
   ICMP: Code = 3 (Port unreachable)
   ICMP: Checksum = B5E7 (correct)
   ICMP:
   ICMP: [Normal end of "ICMP header".]
   ICMP:
   ICMP: IP header of originating message (description follows)
   ICMP:
   ICMP: ----- IP Header -----
   ICMP:
   ICMP: Version = 4, header length = 20 bytes
   ICMP: Type of service = 00
   ICMP:       000. .... = routine
   ICMP:       ...0 .... = normal delay
   ICMP:       .... 0... = normal throughput
   ICMP:       .... .0.. = normal reliability
   ICMP: Total length    = 328 bytes
   ICMP: Identification  = 0
```

Hosts can use the ICMP messages sent by routers to resize datagrams, dynamically adjusting to the needs of the network. This allows the host to determine the smallest MTU path to a destination.

5 = Source Route Failed

The message occurs if a router encounters a next hop in the source route that does not reside on a directly connected network.

6 = Destination Network Unknown

This message occurs when a router receives an IP datagram that it cannot deliver or forward to a particular network because it is unknown.

7 = Destination Host Unknown

This message occurs when a router receives an IP datagram that it cannot deliver or forward to a particular host because it is unknown.

8 = Source Host Isolated (obsolete)

9 = Destination Network Administratively Prohibited

This message occurs when a router receives an IP datagram that it cannot deliver or forward to a particular network because it is not allowed. Access to this network has been prohibited.

10 = Destination Host Administratively Prohibited

This message occurs when a router receives an IP datagram that it cannot deliver or forward to a particular host because it is not allowed. Access to this host has been prohibited.

11 = Network Unreachable for ToS

This message occurs when a router receives an IP datagram that it cannot deliver or forward to a particular network because the ToS requested is not available.

12 = Host Unreachable for ToS

This message occurs when a router receives an IP datagram that it cannot deliver or forward to a particular host because the ToS requested is not available.

13 = Communication Administratively Prohibited by Filtering

This message occurs when a router receives an IP datagram that it cannot deliver or forward to a particular host because it is not allowed. An administratively configured filter has prohibited access to this process or application.

14 = Host Precedence Violation

This message occurs when a router receives an IP datagram that it cannot deliver or forward to a particular host because the precedence level requested does not match, and is not accepted or is invalid. This could be a source host attempting to access a high security host without the necessary security clearance values.

15 = Precedence Cutoff in Effect

This message rarely occurs. However, you will receive this message when a packet is dropped by the cutoff function.

Precedence Handling For All Routers

Routers must accept and route incoming traffic of all precedence levels normally, unless you have configured it to do otherwise. If you want to learn more about precedence and Destination Unreachable messages 14 and 15, please refer to RFC 1812, 5.3.3.3, "Precedence Handling for All Routers."

Source Quench—Type 4

A receiving host generates this message when it cannot process datagrams at the speed requested due to a lack of memory or internal resources. This message serves as a simple flow control mechanism that a receiving host can utilize to alert a sender to slow down its transmission of data. When the source host receives this message, it must pass this information on to the upper-layer process, such as TCP, which then must control the flow of the application's datastream. A router generates this message when, in the process of forwarding datagrams, it has run low on buffers and cannot queue the datagram for delivery.

Redirect—Type 5

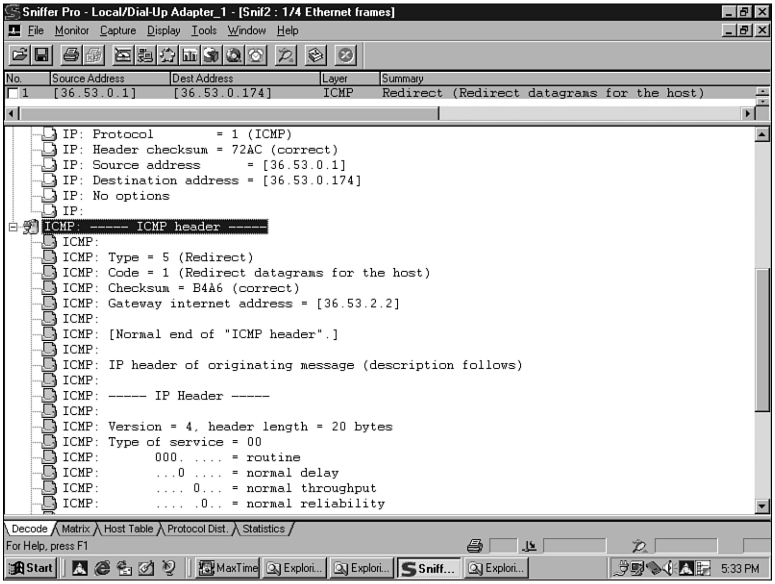
A router sends a redirect error to the sender of an IP datagram when the sender should have sent the datagram to a different router or directly to an end host (if the end host is local). The message assists the sending host to direct a misdirected datagram to a gateway or host. This alert does not guarantee proper delivery; the sending host has to correct the problem if possible.

Only gateways generate redirect messages to inform source hosts of misguided datagrams. Note that a gateway receiving a misdirected frame does not trash the offending datagram if it can forward it. The gateway forwards the frame, sends an alert message to the source, and hopes the source host will properly direct future frames to the designated host or gateway indicated in the message. ICMP redirect messages alert source hosts when a datagram has been misdirected and should be resent. Four redirect error codes can occur:

1. 0 = Redirect for Network
2. 1 = Redirect for Host
3. 2 = Redirect for Type-of-Service and Network
4. 3 = Redirect for Type-of-Service and Host

Figure 3.14 shows an example of a ICMP redirect message. In this example, a gateway (36.53.0.1) alerts host (36.53.0.174) that it should be sending future datagrams to the following gateway internet address (36.53.2.2). This alert message also includes a copy of the offending IP header for the source host's inspection.

FIGURE 3.14
ICMP redirect messages are sent by gateways to hosts alerting them of messages that have been misdirected.


Sniffer Pro - Local/Dial-Up Adapter_1 - [Snif2 : 1/4 Ethernet frames]
File Monitor Capture Display Tools Window Help
Decode Matrix Host Table Protocol Dist. Statistics
For Help, press F1
Start MaxTime Explori... Explori... Sniff... Explori... 5:33 PM

```
No.  Source Address  Dest Address    Layer  Summary
1    [36.53.0.1]     [36.53.0.174]   ICMP   Redirect (Redirect datagrams for the host)
```

```
IP: Protocol        = 1 (ICMP)
IP: Header checksum = 72AC (correct)
IP: Source address      = [36.53.0.1]
IP: Destination address = [36.53.0.174]
IP: No options
IP:
ICMP: ----- ICMP header -----
ICMP:
ICMP: Type = 5 (Redirect)
ICMP: Code = 1 (Redirect datagrams for the host)
ICMP: Checksum = B4A6 (correct)
ICMP: Gateway internet address = [36.53.2.2]
ICMP:
ICMP: [Normal end of "ICMP header".]
ICMP:
ICMP: IP header of originating message (description follows)
ICMP:
ICMP: ----- IP Header -----
ICMP:
ICMP: Version = 4, header length = 20 bytes
ICMP: Type of service = 00
ICMP:       000. .... = routine
ICMP:       ...0 .... = normal delay
ICMP:       .... 0... = normal throughput
ICMP:       .... .0.. = normal reliability
```

Router Advertisement and Solicitation—Types 9 and 10

Rather than initializing a routing table with static routes specified in configuration files, you can use the router ICMP advertisement and solicitation messages. After bootstrapping, a host can transmit a broadcast or multicast a solicitation message to which a router or routers responds with a router advertisement. This allows communicating hosts to learn of available routes dynamically and update their routing tables. We will discuss routing in more detail in Chapters 5 and 6.

Time Exceeded—Type 11

The time exceeded message occurs when a router receives a datagram with a TTL (Time To Live) of 0 or 1. IP uses the TTL field to prevent infinite routing loops. A router cannot forward a datagram that has a TTL of 0 or 1. Instead, it trashes the datagram and sends a time exceeded message. Two different time exceeded error codes can occur:

1. 0 = Time-To-Live Equals 0 During Transit
2. 1 = Time-To-Live Equals 0 During Reassembly

Note that a router cannot forward a datagram with a TTL of 0 or 1 both during transit or reassembly.

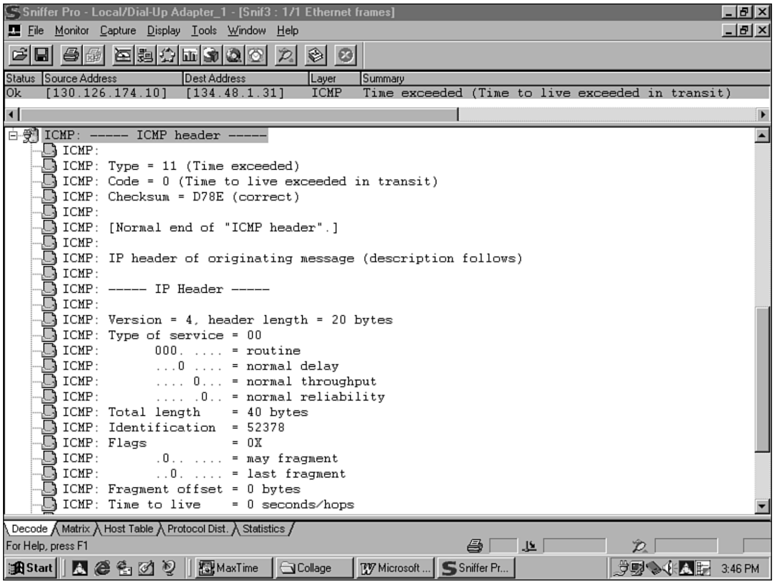
As previously mentioned in the IP section of this chapter, the TTL timer is measured in seconds and originally was used before the existence of routers to guarantee that a datagram did not live on the Internet forever. Each gateway processing a datagram reduces this value by at least one if it takes longer to process and forward the datagram. When this value expires, the gateway trashes the datagram and sends a message back to the sender notifying the host of the situation.

The traceroute utility also uses the TTL value to discover the path or route to a destination host or network. Upon execution of the traceroute command, the initial ICMP message is sent out with a TTL value of 1 set in the IP header. You can use the traceroute program to determine, or rather trace, the path to a destination. Traceroute accomplishes this by sending a sequence of datagrams with the TTL set to 1, 2, and so on. It then uses the ICMP Time Exceeded messages like a trail of breadcrumbs to trace the routers along the path. We will provide you with examples later in this section.

As you might recall from earlier in this chapter, when a router receives a datagram with a TTL of zero, it trashes the datagram and returns an ICMP time exceeded message to the source. This message allows the host to learn of the first router in the path to the destination. Figure 3.15 shows an ICMP message generated as a result of a TTL expiration.

As shown in the figure, ICMP message type 11 alerts a source host of a TTL expiration. Code 0 identifies the reason for the expiration as time to live being exceeded while the datagram was in transit. This message also includes a copy of the original datagram header that caused the error to assist the source host in correcting the problem. Within the offending header contained within the ICMP message, you can see that the "TTL value = 0 seconds/hops," which is why the original datagram was trashed.

FIGURE 3.15
The ICMP time exceeded message is sent when the TTL timer expires.


Sniffer Pro - Local/Dial-Up Adapter_1 - [Snif3 : 1/1 Ethernet frames]
File Monitor Capture Display Tools Window Help
Decode Matrix Host Table Protocol Dist. Statistics
For Help, press F1
Start MaxTime Collage Microsoft ... Sniffer Pr...
3:46 PM

```
Status Source Address     Dest Address    Layer  Summary
Ok     [130.126.174.10]   [134.48.1.31]   ICMP   Time exceeded (Time to live exceeded in transit)
```

```
ICMP: ----- ICMP header -----
  ICMP:
  ICMP: Type = 11 (Time exceeded)
  ICMP: Code = 0 (Time to live exceeded in transit)
  ICMP: Checksum = D78E (correct)
  ICMP:
  ICMP: [Normal end of "ICMP header".]
  ICMP:
  ICMP: IP header of originating message (description follows)
  ICMP:
  ICMP: ----- IP Header -----
  ICMP:
  ICMP: Version = 4, header length = 20 bytes
  ICMP: Type of service = 00
  ICMP:       000. .... = routine
  ICMP:       ...0 .... = normal delay
  ICMP:       .... 0... = normal throughput
  ICMP:       .... .0.. = normal reliability
  ICMP: Total length    = 40 bytes
  ICMP: Identification  = 52378
  ICMP: Flags           = 0X
  ICMP:       .0.. .... = may fragment
  ICMP:       ..0. .... = last fragment
  ICMP: Fragment offset = 0 bytes
  ICMP: Time to live    = 0 seconds/hops
```

Now the source host sends a new ICMP trace with a TTL value of 2, which allows this datagram to be forwarded by the first router (which decrements the value by one) and reaches the next router in the path with a TTL of one. This router must trash the frame and send back an ICMP time exceeded. This process continues until the path to the destination network or host is fully discovered or deemed unreachable. As you can see, traceroute is another useful troubleshooting tool, typically used in conjunction with other utilities such as the Ping utility to test connectivity between two hosts.

Tip

Both the Ping and traceroute utilities can help you when troubleshooting.

Parameter Problem—Type 12

The parameter problem message indicates that a host or gateway received and could not interpret an invalid or misunderstood parameter. A host or gateway also can send this message when no other ICMP message covering the problem can be used to alert the sending host. In this respect, it is a *catchall message*. In most cases this message indicates some type of implementation error occurred, perhaps because of vendor incompatibility issues. A host or gateway will not send this message unless it trashes the datagram containing the parameter problem.

Two parameter problem error messages can occur:

1. **0 = IP Header Bad (catchall error0)**

   A host or gateway sends this error to indicate a general implementation error of an unspecific nature.

2. **1 = Required Option Missing**

   The host or gateway expected a specific option, but the sender did not send it.

Timestamp Request and Reply—Types 13 and 14

Timestamp request and reply messages work in tandem. You have the option of using timestamps. When used, a timestamp request permits a system to query another for the current time. It expects a recommended value returned to be the number of milliseconds since midnight, Coordinated Universal Time. This message provides millisecond resolution, considered a beneficial feature when compared to other means of obtaining time from another host who provides resolution in seconds. The two systems compare the three timestamps and use RTT to adjust the sender's or receiver's time if necessary. Note that most systems set the transmit and receive time as the same value.

The process for time resolution goes as follows:

1. The requestor stamps the originate time and sends the query.
2. The replying system stamps the receive time when it receives the query.
3. The replying system stamps the transmit time when it sends the reply to the query.

Information Request and Reply—Types 15 and 16

Although ICMP messages list information request and reply as a potential ICMP message type, they actually do not occur; thus they are obsolete. A host can request information such as to what network it was attached.

Address Mask Request and Reply—Types 17 and 18

Address mask request and reply messages work in tandem. Although we rarely use this message today, its original design supported the function of dynamically obtaining a subnet mask. Hosts can use the ICMP address mask request to acquire subnet masks during bootstrap from a remote host. However, problems can occur when using ICMP to receive a mask if a host gives an incorrect mask from an external source. If the external source does not give a response, the source host must assume a classful mask (that the network is not subnetted).

Summary

IP is the workhorse of the Network layer within the TCP/IP suite. All protocols and applications utilize IP for logical Network layer addressing and transmission of datagrams between internet hosts. IP provides an unreliable, connectionless datagram delivery service and uses ICMP to send messages when it encounters an error.

End host and routers use ICMP as a control, messaging, and diagnostic tool. ICMP utilizes IP to deliver its messages and is considered an integral part of IP. ICMP messages notify a host of problems. Although ICMP does not offer a solution to these problems, it can provide enough information for a source host to solve some of the problems that might occur in the internetwork. The most popular ICMP message is the echo request and reply. Utilizing the Ping utility, these messages allow you to test connectivity between end hosts.

Review Questions

1. Which Network layer protocol is responsible for fragmentation and reassembly of datagrams?
2. A user would like to test connectivity between two remote hosts, so the user executes Ping. Which two ICMP message types are used to accomplish this test?
3. What does IP do when it receives data from UDP or TCP?
4. How many minimum bytes are there in an IP header and what fields are contained within that header?
5. What are the ToS bits within the IP header used for?
6. What two field values does the destination host use to ensure that it reassembles datagrams in the correct order?
7. What type of ICMP message is Destination Unreachable, and what does it mean if you receive a Destination Unreachable ICMP message?
8. What does a 0 (zero) error code mean when you have a type 3, Destination Unreachable ICMP message?
9. What does a 4 error code mean when you have a type 3, Destination Unreachable ICMP message?
10. What type of ICMP message is Time Exceeded and what does it mean if you receive a Time Exceeded ICMP message?