6

ICMP: Internet Control Message Protocol

6.1 Introduction

ICMP is often considered part of the IP layer. It communicates error messages and other conditions that require attention. ICMP messages are usually acted on by either the IP layer or the higher layer protocol (TCP or UDP). Some ICMP messages cause errors to be returned to user processes.

ICMP messages are transmitted within IP datagrams, as shown in Figure 6.1.

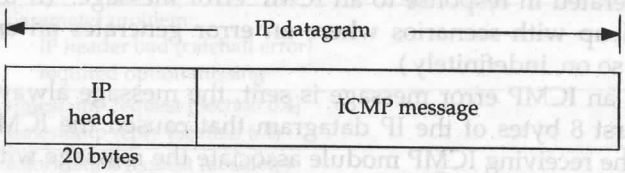

Figure 6.1 ICMP messages encapsulated within an IP datagram.

RFC 792 [Postel 1981b] contains the official specification of ICMP.

Figure 6.2 shows the format of an ICMP message. The first 4 bytes have the same format for all messages, but the remainder differs from one message to the next. We'll show the exact format of each message when we describe it.

There are 15 different values for the *type* field, which identify the particular ICMP message. Some types of ICMP messages then use different values of the *code* field to further specify the condition.

The *checksum* field covers the entire ICMP message. The algorithm used is the same as we described for the IP header checksum in Section 3.2. The ICMP checksum is required.

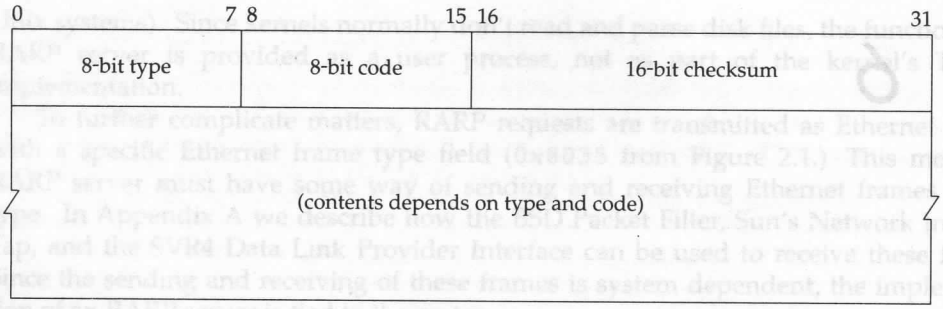

Figure 6.2 ICMP message.

In this chapter we talk about ICMP messages in general and a few in detail: address mask request and reply, timestamp request and reply, and port unreachable. We discuss the echo request and reply messages in detail with the Ping program in Chapter 7, and we discuss the ICMP messages dealing with IP routing in Chapter 9.

6.2 ICMP Message Types

Figure 6.3 lists the different ICMP message types, as determined by the *type* field and *code* field in the ICMP message.

The final two columns in this figure specify whether the ICMP message is a query message or an error message. We need to make this distinction because ICMP error messages are sometimes handled specially. For example, an ICMP error message is never generated in response to an ICMP error message. (If this were not the rule, we could end up with scenarios where an error generates an error, which generates an error, and so on, indefinitely.)

When an ICMP error message is sent, the message always contains the IP header and the first 8 bytes of the IP datagram that caused the ICMP error to be generated. This lets the receiving ICMP module associate the message with one particular protocol (TCP or UDP from the protocol field in the IP header) and one particular user process (from the TCP or UDP port numbers that are in the TCP or UDP header contained in the first 8 bytes of the IP datagram). We'll show an example of this in Section 6.5.

An ICMP error message is never generated in response to

1. An ICMP error message. (An ICMP error message may, however, be generated in response to an ICMP query message.)
2. A datagram destined to an IP broadcast address (Figure 3.9) or an IP multicast address (a class D address, Figure 1.5).
3. A datagram sent as a link-layer broadcast.
4. A fragment other than the first. (We describe fragmentation in Section 11.5.)

type	*code*	Description	Query	Error
0	0	echo reply (Ping reply, Chapter 7)	•	
3		destination unreachable:		
	0	network unreachable (Section 9.3)		•
	1	host unreachable (Section 9.3)		•
	2	protocol unreachable		•
	3	port unreachable (Section 6.5)		•
	4	fragmentation needed but don't-fragment bit set (Section 11.6)		•
	5	source route failed (Section 8.5)		•
	6	destination network unknown		•
	7	destination host unknown		•
	8	source host isolated (obsolete)		•
	9	destination network administratively prohibited		•
	10	destination host administratively prohibited		•
	11	network unreachable for TOS (Section 9.3)		•
	12	host unreachable for TOS (Section 9.3)		•
	13	communication administratively prohibited by filtering		•
	14	host precedence violation		•
	15	precedence cutoff in effect		•
4	0	source quench (elementary flow control, Section 11.11)		•
5		redirect (Section 9.5):		
	0	redirect for network		•
	1	redirect for host		•
	2	redirect for type-of-service and network		•
	3	redirect for type-of-service and host		•
8	0	echo request (Ping request, Chapter 7)	•	
9	0	router advertisement (Section 9.6)	•	
10	0	router solicitation (Section 9.6)	•	
11		time exceeded:		
	0	time-to-live equals 0 during transit (Traceroute, Chapter 8)		•
	1	time-to-live equals 0 during reassembly (Section 11.5)		•
12		parameter problem:		
	0	IP header bad (catchall error)		•
	1	required option missing		•
13	0	timestamp request (Section 6.4)	•	
14	0	timestamp reply (Section 6.4)	•	
15	0	information request (obsolete)	•	
16	0	information reply (obsolete)	•	
17	0	address mask request (Section 6.3)	•	
18	0	address mask reply (Section 6.3)	•	

Figure 6.3 ICMP message types.

5. A datagram whose source address does not define a single host. This means the source address cannot be a zero address, a loopback address, a broadcast address, or a multicast address.

These rules are meant to prevent the *broadcast storms* that have occurred in the past when ICMP errors were sent in response to broadcast packets.

2. Serious timekeepers use the Network Time Protocol (NTP) described in RFC 1305 [Mills 1992]. This protocol uses sophisticated techniques to maintain the clocks for a group of systems on a LAN or WAN to within millisecond accuracy. Anyone interested in precise timekeeping on computers should read this RFC.
3. The Open Software Foundation's (OSF) Distributed Computing Environment (DCE) defines a Distributed Time Service (DTS) that also provides clock synchronization between computers. [Rosenberg, Kenney, and Fisher 1992] provide additional details on this service.
4. Berkeley Unix systems provide the daemon `timed`(8) to synchronize the clocks of systems on a local area network. Unlike NTP and DTS, `timed` does not work across wide area networks.

6.5 ICMP Port Unreachable Error

The last two sections looked at ICMP query messages—the address mask and timestamp queries and replies. We'll now examine an ICMP error message, the port unreachable message, a subcode of the ICMP destination unreachable message, to see the additional information returned in an ICMP error message. We'll watch this using UDP (Chapter 11).

One rule of UDP is that if it receives a UDP datagram and the destination port does not correspond to a port that some process has in use, UDP responds with an ICMP port unreachable. We can force a port unreachable using the TFTP client. (We describe TFTP in Chapter 15.)

The well-known UDP port for the TFTP server to be reading from is 69. But most TFTP client programs allow us to specify a different port using the `connect` command. We use this to specify a port of 8888:

```
bsdi % tftp
tftp> connect svr4 8888          specify the hostname and port number
tftp> get temp.foo               try to fetch a file
Transfer timed out.              about 25 seconds later
tftp> quit
```

The `connect` command saves the name of the host to contact and the port number on that host, for when we later issue the `get` command. After typing the `get` command a UDP datagram is sent to port 8888 on host `svr4`. Figure 6.8 shows the `tcpdump` output for the exchange of packets that takes place.

Before the UDP datagram can be sent to `svr4` an ARP request is sent to determine its hardware address (line 1). The ARP reply (line 2) is returned and then the UDP datagram is sent (line 3). (We have left the ARP request–reply in this `tcpdump` output to remind us that this exchange may be required before the first IP datagram is sent from one host to the other. In future output we'll delete this exchange if it's not relevant to the topic being discussed.)

```
 1   0.0                     arp who-has svr4 tell bsdi
 2   0.002050 (0.0020)       arp reply svr4 is-at 0:0:c0:c2:9b:26

 3   0.002723 (0.0007)       bsdi.2924 > svr4.8888: udp 20
 4   0.006399 (0.0037)       svr4 > bsdi: icmp: svr4 udp port 8888 unreachable

 5   5.000776 (4.9944)       bsdi.2924 > svr4.8888: udp 20
 6   5.004304 (0.0035)       svr4 > bsdi: icmp: svr4 udp port 8888 unreachable

 7  10.000887 (4.9966)       bsdi.2924 > svr4.8888: udp 20
 8  10.004416 (0.0035)       svr4 > bsdi: icmp: svr4 udp port 8888 unreachable

 9  15.001014 (4.9966)       bsdi.2924 > svr4.8888: udp 20
10  15.004574 (0.0036)       svr4 > bsdi: icmp: svr4 udp port 8888 unreachable

11  20.001177 (4.9966)       bsdi.2924 > svr4.8888: udp 20
12  20.004759 (0.0036)       svr4 > bsdi: icmp: svr4 udp port 8888 unreachable
```

Figure 6.8 ICMP port unreachable generated by TFTP request.

An ICMP port unreachable is immediately returned (line 4). But the TFTP client appears to ignore the ICMP message, sending another UDP datagram about 5 seconds later (line 5). This continues three more times before the client gives up.

Notice that the ICMP messages are exchanged between hosts, without a port number designation, while each 20-byte UDP datagram is from a specific port (2924) and to a specific port (8888).

The number 20 at the end of each UDP line is the length of the data in the UDP datagram. In this example 20 is the sum of the TFTP's 2-byte opcode, the 9-byte null terminated name `temp.foo`, and the 9-byte null terminated string `netascii`. (See Figure 15.1 for the details of the TFTP packet layout.)

If we run this same example using the `-e` option of `tcpdump` we see the exact length of each ICMP port unreachable message that's returned to the sender. This length is 70 bytes, and is allocated as shown in Figure 6.9.

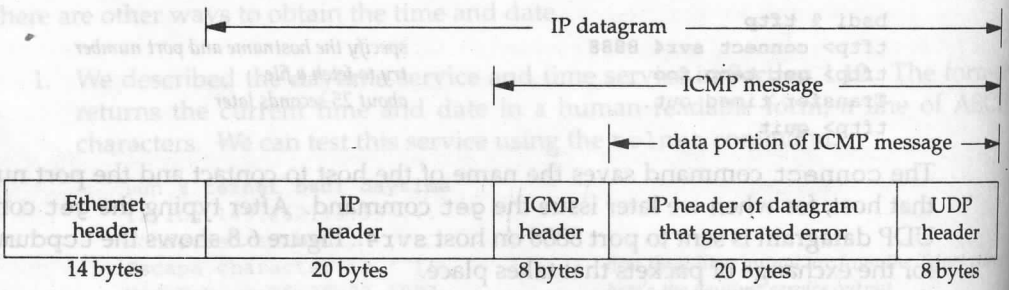

Figure 6.9 ICMP message returned for our "UDP port unreachable" example.

One rule of ICMP is that the ICMP error messages (see the final column of Figure 6.3, p. 71) must include the IP header (including any options) of the datagram that generated the error along with at least the first 8 bytes that followed this IP header. In our example, the first 8 bytes following the IP header contain the UDP header (Figure 11.2).

The important fact is that contained in the UDP header are the source and destination port numbers. It is this destination port number (8888) that caused the ICMP port unreachable to be generated. The source port number (2924) can be used by the system receiving the ICMP error to associate the error with a particular user process (the TFTP client in this example).

One reason the IP header of the datagram that caused the error is sent back is because in this IP header is the protocol field that lets ICMP know how to interpret the 8 bytes that follow (the UDP header in this example). When we look at the TCP header (Figure 17.2) we'll see that the source and destination port numbers are contained in the first 8 bytes of the TCP header.

The general format of the ICMP unreachable messages is shown in Figure 6.10.

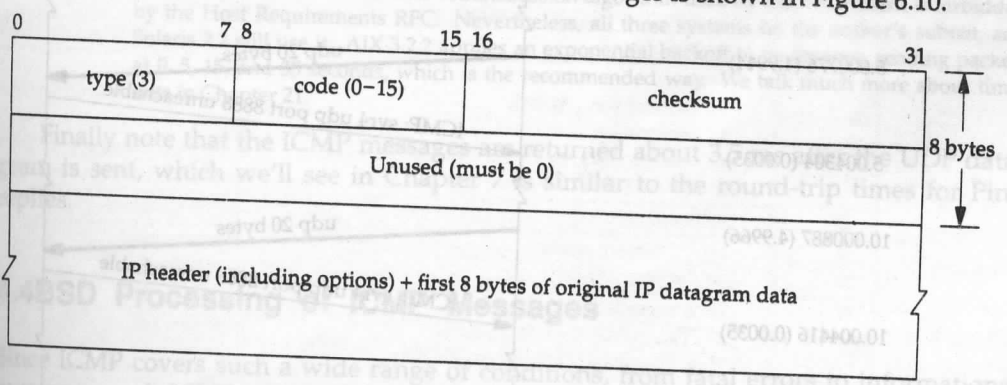

Figure 6.10 ICMP unreachable message.

In Figure 6.3 we noted that there are 16 different ICMP unreachable messages, *codes* 0 through 15. The ICMP port unreachable is *code* 3. Also, although Figure 6.10 indicates that the second 32-bit word in the ICMP message must be 0, the Path MTU Discovery mechanism (Section 2.9) allows a router to place the MTU of the outgoing interface in the low-order 16 bits of this 32-bit value, when *code* equals 4 ("fragmentation needed but the don't fragment bit is set"). We show an example of this error in Section 11.6.

Although the rules of ICMP allow a system to return more than the first 8 bytes of the data portion of the IP datagram that caused the ICMP error, most Berkeley-derived implementations return exactly 8 bytes. The Solaris 2.2 `ip_icmp_return_data_bytes` option returns the first 64 bytes of data by default (Section E.4).