CHAPTER

Internet Control Message Protocol

7

What You Will Learn

In this chapter, you will learn about ICMP messages, their types, and (in many cases) the codes used in each type. We'll look at which ICMP messages are routinely blocked at firewalls and which are essential for proper device operation.

You will learn about the common ping utility for determining device accessibility ("reachability") on an IP network. We'll discuss the mechanics of both ping and traceroute, and use several ping examples to illustrate ICMP on the network.

The only function of the IP layer is to provide addressing for and route the IP packet. That's all. Once an IP packet has been dealt with, the IP layer just looks for the next packet. But IP is a connectionless, "best effort," or "unreliable" method of packet delivery. The terms "best effort" and "unreliable" often make it sound like IP is casual about the delivery of packets, which is why they are in quotes so that no one takes them too literally. IP's best effort is usually just fine, given the low error rates on modern transports, and it is mostly unreliable with regard to a lack of guarantees, as has been pointed out. Besides, there is nothing wrong with letting other layers, such as the TCP segments or the Ethernet frames, have the major responsibility for error detection and correction.

This is not to say that IP should be oblivious to errors. The network layer, in its ubiquitous and key position at the heart of the protocol stack, should know about packet errors and is in a good position to let layers above know what's going on (although IP lets the upper layers decide what to do about the condition).

And there's plenty that can still go wrong, and not just with regard to bit errors. A packet might wander the router cloud until the TTL field hits zero. A destination server might be down. A destination server might no longer *exist*. The "do not fragment" bit might forbid fragmentation when it is needed to send a packet, stopping the routing process cold. In all of these situations, the sender should be informed of the condition.

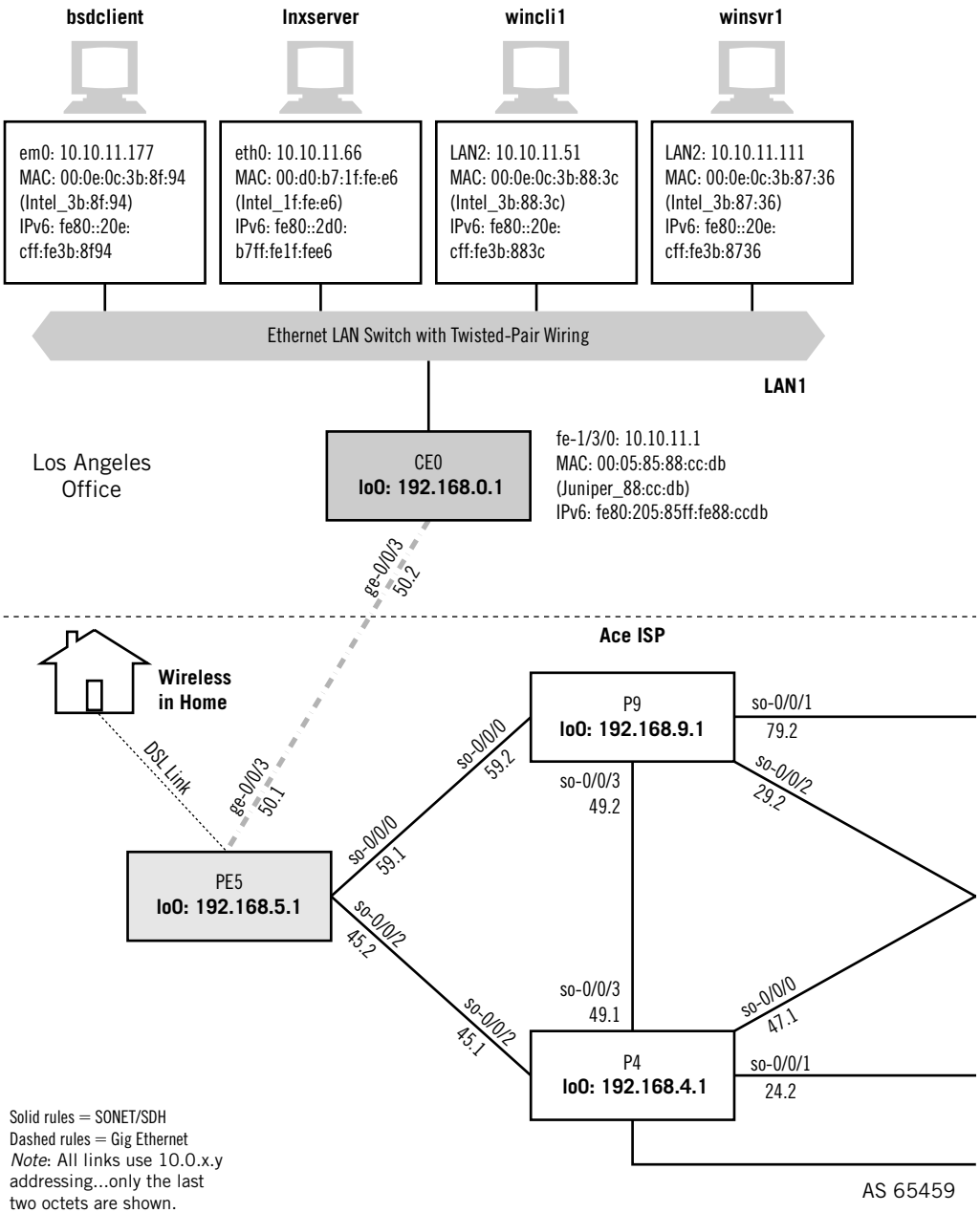
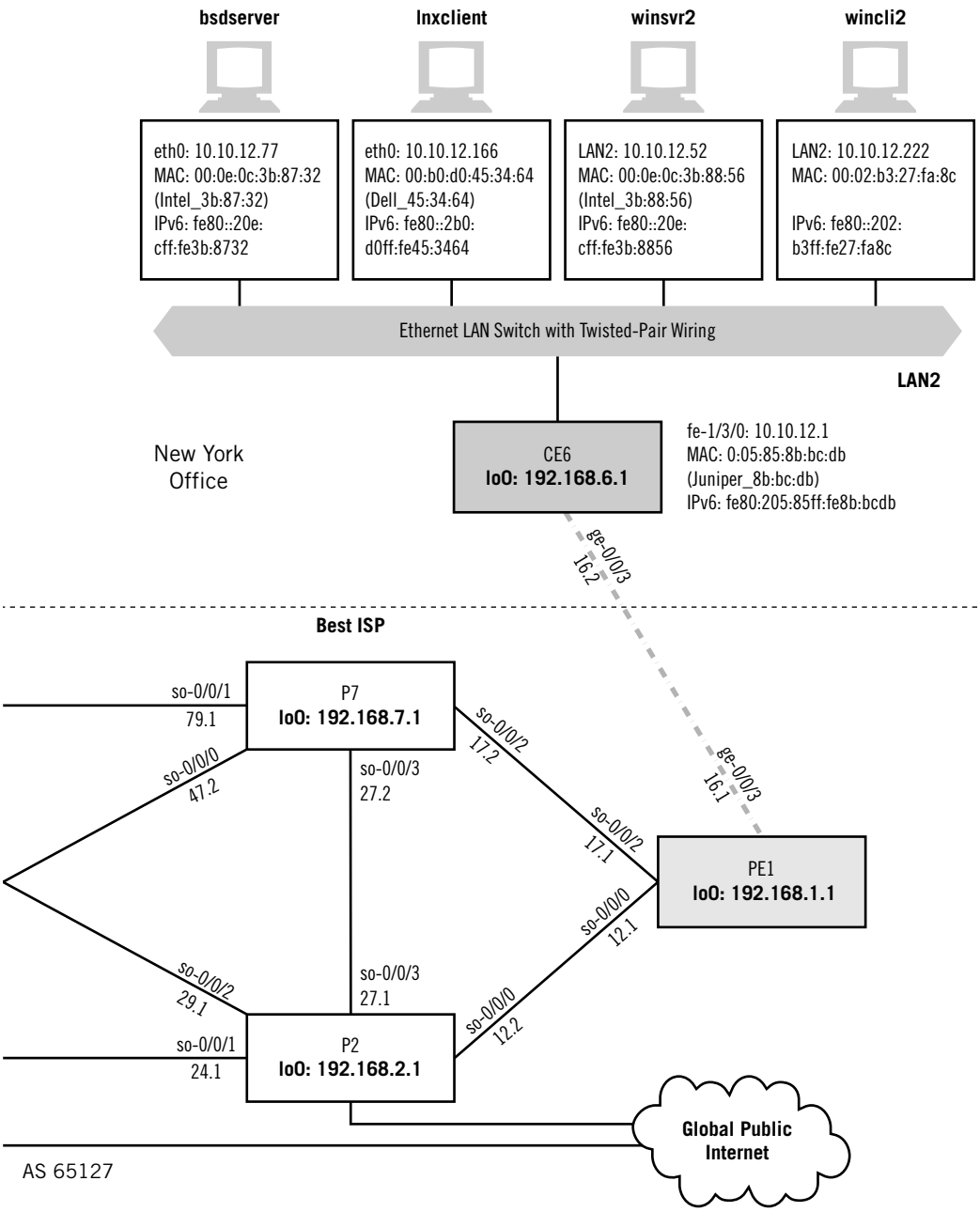


FIGURE 7.1

ICMP is used on all devices on the Illustrated Network, routers, and hosts. In this chapter, we'll work with the hosts on the LANs.


bsdserver
lnxclient
winsvr2
wincli2
eth0: 10.10.12.77
MAC: 00:0e:0c:3b:87:32
(Intel_3b:87:32)
IPv6: fe80::20e:
cff:fe3b:8732
eth0: 10.10.12.166
MAC: 00:b0:d0:45:34:64
(Dell_45:34:64)
IPv6: fe80::2b0:
d0ff:fe45:3464
LAN2: 10.10.12.52
MAC: 00:0e:0c:3b:88:56
(Intel_3b:88:56)
IPv6: fe80::20e:
cff:fe3b:8856
LAN2: 10.10.12.222
MAC: 00:02:b3:27:fa:8c
IPv6: fe80::202:
b3ff:fe27:fa8c
Ethernet LAN Switch with Twisted-Pair Wiring
LAN2
New York
Office
CE6
lo0: 192.168.6.1
fe-1/3/0: 10.10.12.1
MAC: 0:05:85:8b:bc:db
(Juniper_8b:bc:db)
IPv6: fe80:205:85ff:fe8b:bcdb
ge-0/0/3
16.2
Best ISP
so-0/0/1
79.1
P7
lo0: 192.168.7.1
so-0/0/2
17.2
so-0/0/0
47.2
so-0/0/3
27.2
ge-0/0/3
16.1
so-0/0/2
17.1
PE1
lo0: 192.168.1.1
so-0/0/0
12.1
so-0/0/2
29.1
so-0/0/3
27.1
so-0/0/0
12.2
so-0/0/1
24.1
P2
lo0: 192.168.2.1
Global Public
Internet
AS 65127

Without error condition feedback from the network, the natural response to an unexpected result (in this case, a reply) is to simply repeat the original message. Sometimes this might work, especially if the condition is transient, but semipermanent or permanent error conditions must be reported to the source. Otherwise, repetitive sending might result in an endless error loop, and certainly adds unnecessary traffic loads to the network.

This chapter explores aspects of IP's built-in error reporting protocol, the Internet Control Message Protocol (ICMP). Note that ICMP does not deal with "error messages," but "*control* messages," a better term to cover all of the roles that have evolved for ICMP. We'll start by looking at one indispensable utility used on all TCP/IP network: ping. We'll be using the same LAN-based hosts as in the previous chapter, as shown in Figure 7.1.

ICMP AND PING

The easiest way to look at ICMP on the Illustrated Network is with ping and traceroute. Both utilities have been used before in this book, but because traceroute will be used again in the chapters on routing, this chapter will use ICMP and ping.

The ping utility is just a way to "bounce" packets off a target device and see if it is there—that is, it has the IP address that was provided, is powered on, and alive. The device might still not function in the correct way (i.e., the router might not be routing properly), but at least the device is present and accounted for. It is routine to ping a newly installed device, host, router, or anything else, just to see if it responds. If it doesn't, network administrators have a place to start troubleshooting.

Let's use ping from the `lnxclient` to the `bsdserver`, both on LAN2 to start exploring ICMP. Windows XP only sends four pings by default, but Unix systems will just keep going until stopped with `^C` (which is what was done here).

```
[root@lnxclient admin]# ping 10.10.12.77
PING 10.10.12.77 (10.10.12.77) 56(84) bytes of data.
64 bytes from 10.10.12.77: icmp_seq=1 ttl=64 time=0.549 ms
64 bytes from 10.10.12.77: icmp_seq=2 ttl=64 time=0.169 ms
64 bytes from 10.10.12.77: icmp_seq=3 ttl=64 time=0.171 ms
64 bytes from 10.10.12.77: icmp_seq=4 ttl=64 time=0.187 ms
64 bytes from 10.10.12.77: icmp_seq=5 ttl=64 time=0.216 ms
^C
--- 10.10.12.77 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 3996ms
rtt min/avg/max/mdev = 0.169/0.258/0.549/0.146 ms
[root@lnxclient admin]#
```

The output shows the ICMP sequence numbers and round-trip time (`rtt`) for the group in terms of minimum, average, maximum, and even the maximum deviation from the mean. We do not have DNS on the network, so we have to use IP addresses. Most

ping implementations will accept host names, and some (such as Cisco routers) will even do a reverse DNS lookup when given an IP address and report the host name in the result. This can be very helpful when an IP address is entered incorrectly or assigned to a different device than anticipated.

We can look at the ICMP packets used with ping in more detail. Let's use both LANs this time, and ping from `wincl1` (`10.10.11.51`) on LAN1 to `wincli2` (`10.10.12.222`) on LAN2. With XP, we won't have to worry about stopping the sequence.

```
C:\Documents and Settings\Owner> ping 10.10.12.222

Pinging 10.10.12.222 with 32 bytes of data:

Reply from 10.10.12.222: bytes=32 time<1ms TTL=126
Reply from 10.10.12.222: bytes=32 time<1ms TTL=126
Reply from 10.10.12.222: bytes=32 time<1ms TTL=126
Reply from 10.10.12.222: bytes=32 time<1ms TTL=126

Ping statistics for 10.10.12.222:
   Packets: Sent = 4, Received = 4, Lost = 0 (0% less),
Approximate round-trip times in milliseconds:
   Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Due to the way the Windows operating systems handle timing, it's not unusual to have RTTs of 0.

What does this group of packets look like at the target? Figure 7.2 shows us.

We can see that the four pings are accomplished with eight packets sent over the network. Look at the last column in the upper part of the figure. Ping employs messages in request–reply pairs using the ICMP protocol. An Echo request is sent out which basically tells the receiver to "send an ICMP Echo message back to me, okay?" Once the reply is received, the next request is sent, statistics compiled as the procedure goes along, and so on.

The details of Frame 1 show that the ICMP message is carried directly inside an IP packet (and then Ethernet II frame). But ICMP is not often shown as a transport layer protocol. That would make ICMP function at the same level as things like TCP and UDP, and this is simply not true. ICMP, as we will find, is concerned with network layer problems, so portraying ICMP as a type of special protocol associated with IP is not really a mistake.

So technically, because IPv4 packets carry ICMP messages as protocol number 1, ICMP is as valid a layer above IP as TCP or UDP or any other of the 200 or so defined IP protocols that can be carried inside IP packets. But because every IP implementation must include ICMP (and IPv6 has ICMPv6), it makes sense to bundle ICMP and IP together. This also implies that ICMP messages do not report their own errors.

What if no reply is received by the source of a ping? The source then times out and another ICMP Echo request message is sent. Naturally, no statistics can be generated, and we get a "host unreachable" message in most cases. We can force a timeout simply by trying to ping a nonexistent address (this could also be the result of a simple typo).

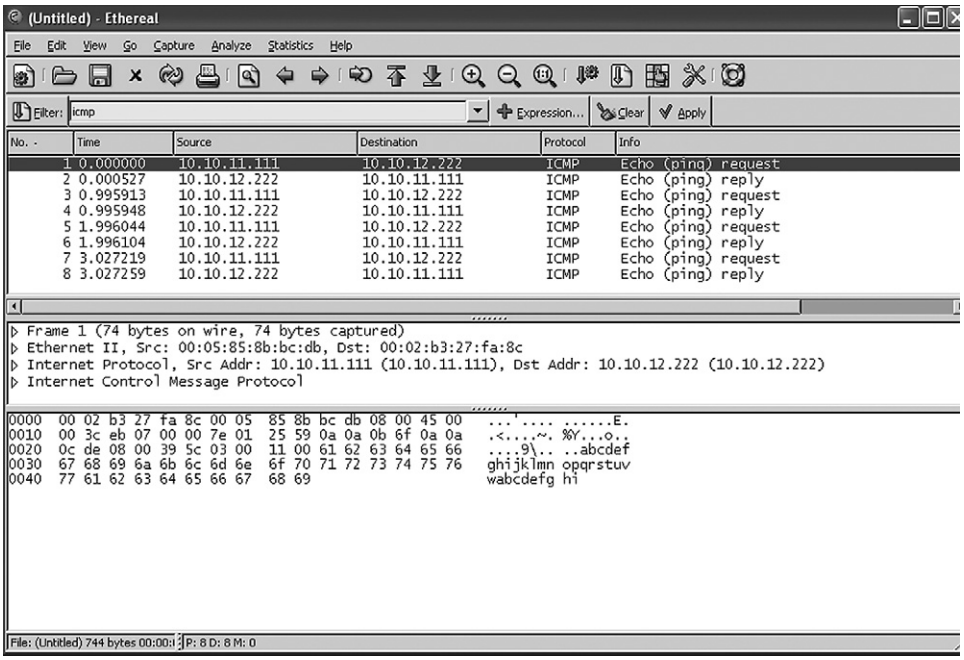


FIGURE 7.2

Ping ICMP requests and replies showing details of the ping echo request in the middle pane. Note that the content of the packet is the ICMP message, not TCP or UDP.

```
[root@lnxclient admin]# ping 10.10.12.55
PING 10.10.12.55 (10.10.12.55) 56(84) bytes of data.
From 10.10.12.166 icmp_seq=1 Destination Host Unreachable
From 10.10.12.166 icmp_seq=2 Destination Host Unreachable
From 10.10.12.166 icmp_seq=3 Destination Host Unreachable

--- 10.10.12.55 ping statistics ---
3 packets transmitted, 0 received, +3 errors, 100% packet loss, time 5022ms,
pipe 3
[root@lnxclient admin]#
```

Many ping implementations report either "unreachable" or "unknown" errors. The *unreachable* report implies that the target was once known to the source and reachable, but isn't "reachable" at the moment. The *unknown* report implies that the source has never heard of the target address or port. However, unreachable reports are often returned by a host source pinging a new device, which obviously should be unknown.

Most network people treat both error condition reports the same way: Something is just plain wrong.

Ping remains the first choice for checking connectivity on the Internet, between hosts, and between host and router. On LANs, the first troubleshooting step is "can you ping it?" If you cannot, there's no sense of going further. If you can, and things like applications still do not function as expected, at least the troubleshooting process can continue productively.

Firewalls sometimes screen out ICMP messages in the name of security. In these cases, even a failed ping does not prove that a device is not working properly. However, diagnostics become more complex, although not impossible. Of course, screening out *all* ICMP messages from a site usually also eliminates correct error reporting and proper operation of the device. After we list the ICMP message types, we'll discuss which ICMP messages are essential.

Ping works with IPv6, too. On most Unix hosts, it's called `ping6`. When used with the special IPv6 multicast address `ff02::1`, the `%em0` addition probes for the IPv6 address of every interface on the LAN, a form of forced *neighbor discovery* in IPv6. Here's what it looks like on LAN2 when run from the `bsdserver`.

```
bsdserver# ping6 ff02::1%em0
PING6(56=40+8+8 bytes) fe80::20e:cff:fe3b:8732%em0 -> ff02::1%em0
16 bytes from fe80::20e:cff:fe3b:8732%em0, icmp_seq=0 hlim=64 time=0.154 ms
16 bytes from fe80::202:b3ff:fe27:fa8c%em0, icmp_seq=0 hlim=128 time=0.575
ms(DUP!)
16 bytes from fe80::5:85ff:fe8b:bcdb%em0, icmp_seq=0 hlim=64 time=1.192
ms(DUP!)
16 bytes from fe80::20e:cff:fe3b:8856%em0, icmp_seq=0 hlim=64 time=0.097
ms(DUP!)
^C

--- ff02::1%em0 ping6 statistics ---
1 packets transmitted, 1 packets received, +3 duplicates, 0% packet loss
round-trip min/avg/max/std-dev = 0.071/2.520/39.406/8.950 ms
bsdserver#
```

All four systems on LAN2 are listed, except for `lnxclient`, which does not have an IPv6 address. But hosts `winsrv2` (`fe80::20e:cff:fe3b:8856`), `wincli2` (`fe80::202:b3ff:fe27:fa8c`), router `TP6` (`fe80::5:85ff:fe8b:bcdb`), and even `bsdserver` (`fe80::20e:cff:fe3b:8732`) itself have all replied. Oddly, the Windows XP client replies with a hop limit of 128.

IPv6 traffic (and ICMPv6) is also visible to Ethereal, so we can explore the format of these packets a little further. Figure 7.3 shows how the exchange of the `ping6 ff02::1%em0` packets looks like from `wincli2` when run from `bsdserver`. Note that this only captures the exchange of packets that `wincli2` processes.

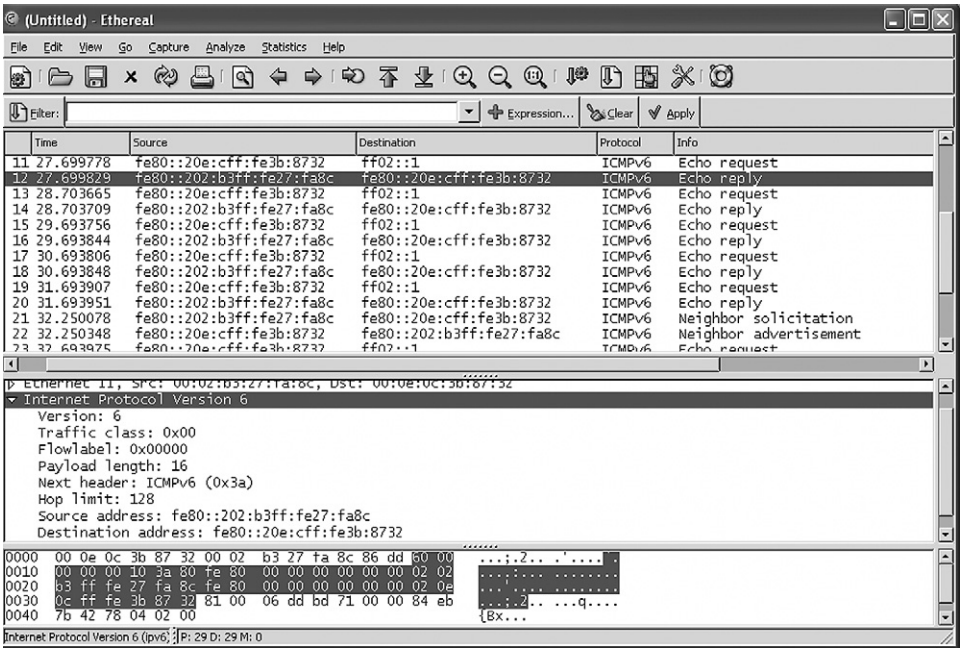


FIGURE 7.3

ICMPv6 capture showing the ICMPv6 echo reply message from `wincli2`. The header details are shown in the middle pane.

IPv6 uses its own version of ICMP, called (not surprisingly) ICMPv6. The ICMPv6 Echo reply message, sent in response to the ping to multicast group `ff02::1`, is highlighted in the figure. From the source address, we can tell this is from `wincli2`. We looked at the details of the IPv6 header in the last chapter. Note that the hop limit is 128 in the reply, and that the protocol number for ICMP is `0x3a` (58 decimal).

THE ICMP MESSAGE FORMAT

ICMP is usually considered to be part of the IP layer itself, and that is how ICMP is presented here. Hosts are supposed to set the IPv4 packet header TOS field to 0 if the packet carries an ICMP message, and routers are supposed to set the precedence field to 6 or 7.

Figure 7.4 shows the format of two ICMP messages. All ICMP messages start with the same three fields: an 8-bit Type and Code, followed by a 16-bit Checksum. Then, depending on the value of the Type, the details of what follows varies. So to be more informative, a second ICMP message is shown. The second message displays the format used for a very common network condition, Destination Unreachable, which we saw earlier.

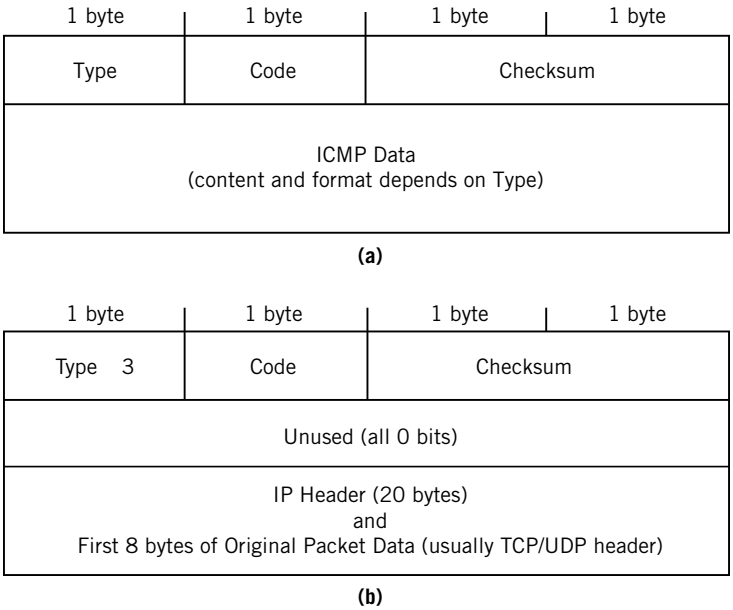


FIGURE 7.4

ICMP message format, showing how a specific message such as Destination Unreachable uses the fields following the initial three. (a) General format of ICMP message. (b) Format of Destination Unreachable ICMP message.

Destinations on a TCP/IP network can be unreachable for a number of reasons. The host could be down, or have a new IP address that is not yet known to all systems. The destination's Internet name could have been typed incorrectly (but still maps to an existing IP address), the only link to the site could have failed, and so on.

ICMP Message Fields

The fields that appear in all ICMP messages follow:

Type—This 8-bit field defines the major purpose of the ICMP message. Most indicate error conditions, but two of the most common type values, 8 and 0, mean Echo Request and Echo Reply, respectively. A Type value of 3 means Destination Unreachable. All Types determine the format of the rest of the ICMP message beyond the first three fields.

Code—This 8-bit field gives additional information about the condition in the Type field. This is often not necessary, and many Types have only a Code = 0 defined. Other Types have many Code values defined to allow the source to

focus on the real problem. For example, Destination Unreachable (Type = 3) has 16 codes (0–15) defined.

Checksum—This is the same type of checksum as used for the IP packet header. This points out that ICMP, although considered part of IP itself, is really just as much a separate layer as anything else in TCP/IP and so must provide for its own error checking.

ICMP Types and Codes

There are about 40 defined ICMP message types, and message types 41 through 255 are reserved for future use. Only a handful of the types have more than a Code value of 0 defined, but these are the more important ICMP message types.

There are two major categories of ICMP messages: error messages (reports that do not expect a response) and queries (messages sent with the expectation of a matching response). Some others do not fall neatly into either category. The structure of the fields following the checksum depends on the type of ICMP message. These two formats are shown in Figure 7.5.

Note that the Destination Unreachable format shown in Figure 7.4 is an ICMP error message and does not generate a reply. The fields that appear following the initial three in the ICMP Destination Unreachable message are very common.

Unused—This 32-bit field must be set to all 0 bits for Destination Unreachable, but in other ICMP messages it is often used as a sequence number to allow requests and responses to be coordinated by senders and receivers.

IP Header and More—The last 28 bytes of the ICMP Destination Unreachable message consist of the original IP header (usually 20 bytes, but can be up to 60 bytes) and the first 8 bytes of the segment inside the packet. Usually, this includes the ports used by the TCP or UDP segment. This practice allows senders to realize exactly what field value is objectionable. It's one thing to say "Port unreachable," but better to say "Hey! The port in the UDP segment you sent, which is port 6735, can't be reached here right now..."

Usually, the error messages have the all-zero unused byte followed by the 28-byte header and packet data, but not always. Identifiers track Query message request/response pairs, and the sequence numbers help sort out queries sent by the same process (the process identifier, the PID, is often the ICMP Query identifier in Unix systems).

The suite of the 40 ICMP message types can be implemented by hosts or routers. Some of the types are mandatory, some are optional, some are for experimental use, and some are obsolete. In some cases, specifications explicitly state that hosts or routers be able to transmit and receive (process) ICMP messages, but not in all cases.

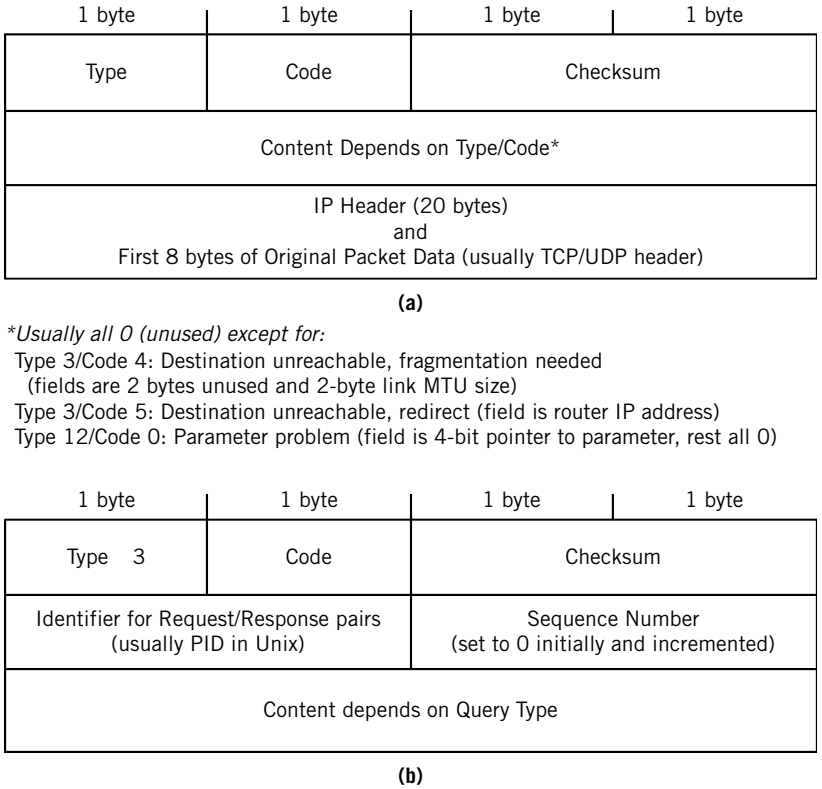


FIGURE 7.5

ICMP error and query messages. Note that error messages include the IP header that generated the error. (a) ICMP error message. (b) ICMP query message.

Let's take a look at what the specifications say about ICMP messages. First, we'll look at error messages, and then query messages, and then all the rest.

ICMP Error Messages

ICMP Error messages report semipermanent network conditions. The five ICMP error messages are displayed in Table 7.1, which shows how routers and hosts should handle each type.

Time-exceeded errors result from TTL expiration (Code = 0) or when fragments cannot be completed quickly enough at a receiver (Code = 1). Parameter problems are usually sent in regard to IP options. The codes are for a bad IP header (0), missing a required option field (1), or a bad length (2).

Which of these message types are essential to device operation and should not be blocked? Generally, the Destination Unreachable is essential (it is used by traceroute), and used in MTU path calculations. Of the others, the Redirect message is most often

Table 7.1 ICMP Error Messages

Type	Meaning	Codes	Data	Router Sends	Router Receives	Host Sends	Host Receives
3	Destination Unreachable	0–15	IP hdr + 8 bytes	M	M	M	M
4	Source Quench	0	IP hdr + 8 bytes	Obs	Obs	Obs	Obs
5	Redirect	0–3	IP hdr + 8 bytes	M	M	Opt	Opt
11	Time Exceeded	0–1	IP hdr + 8 bytes	M	M	Opt	Opt
12	Parameter Problem	0–2	IP hdr + 8 bytes	M	M	M	M

Obs, obsolete; Opt, optional; M, mandatory.

Table 7.2 ICMP Destination Unreachable Codes

Code	Meaning
0	Network is unreachable (the router's links to it might have failed).
1	Host is unreachable (the router can't reach the host; it might be turned off).
2	Requested protocol is unreachable (the process might not be running on the host).
3	Port is unreachable (the remote application might not be running on the host).
4	Fragmentation needed at router but DF flag is set (used for path MTU determination).
5	Source route has failed (source route path might go through down link or router).
6	Destination network is unknown (different than Code = 0; router can't find it).
7	Destination host is unknown (different than Code = 1; router can't find host).
8	Source host is isolated (source host is not allowed to send onto the network).
9	Communication with this network is administratively forbidden (due to firewall).
10	Communication with this host is administratively forbidden (due to firewall).
11	Network is unreachable with specified Type of Service (router can't forward).
12	Host is unreachable with specified Type of Service (router can't forward).
13	Communication administratively prohibited (by route filtering).
14	Host precedence violation (the first-hop router does not support this precedence).
15	Precedence cut-off in effect (requested precedence too low for router network).

blocked, because it does just as it says, that is, it tells another device to send packets somewhere else.

Many ICMP errors are Destination Unreachable errors. The 16 codes for this error type and their meanings are shown in Table 7.2, which includes a likely cause for the condition.

The precedence bits are in the TOS field of the IPv4 packet header, and are distinct from the TOS bits themselves (and are almost universally ignored anyway).

ICMP Query Messages

ICMP Query messages are used to question conditions on the network. These messages are used in pairs, and each request anticipates a response. The 10 ICMP Query messages are listed in Table 7.3, which shows how routers and hosts should handle each type.

These ICMP messages in Table 7.3 allow routers and hosts to query for timestamp, address mask, and domain name information. Echo requests and replies have special uses described in the section of this chapter on ping.

Table 7.3 ICMP Query Messages

Type	Meaning	Codes	Data	Router Sends	Router Receives	Host Sends	Host Receives
0	Echo reply	0	Varies	M	M	M	M
8	Echo request	0	Varies	M	M	M	M
13	Timestamp request	0	12 bytes	Opt	Opt	Opt	Opt
14	Timestamp reply	0	12 bytes	Opt	Opt	Opt	Opt
15	Information request	0	0 bytes	Obs	Obs	Obs	Obs
16	Information reply	0	0 bytes	Obs	Obs	Obs	Obs
17	Mask request	0	4 bytes	M	M	Opt	Opt
18	Mask reply	0	4 bytes	M	M	Opt	Opt
37	Domain name request	0	0 bytes	M	M	M	M
38	Domain name reply	0	0 bytes	M	M	M	M

Obs, obsolete; Opt, optional; M, mandatory.

Which of these should be allowed to pass through firewalls? Sites most often allow Echo messages (used by ping), although some allow only incoming Echo replies but not Echo requests (which allows my devices to ping yours, but not the other way around). The timestamp reply is also used by `traceroute`, and if these messages are blocked, asterisks (*) appear instead of times in the `traceroute` report (we'll look at `traceroute` operation in detail in Chapter 9).

Table 7.4 Other ICMP Query Messages

Type	Meaning	Codes	Data	Router Sends	Router Receives	Host Sends	Host Receives
1	Unassigned	NA	NA	NA	NA	NA	NA
2	Unassigned	NA	NA	NA	NA	NA	NA
6	Alternate host address	0	(4 bytes)	(Prohibited)	(Prohibited)	Opt	Opt
9	Router advertisement	0	Varies	M	Opt	Prohibited	Opt
10	Router solicitation	0	0 bytes	M	M	Opt	Opt
19	Reserved–security	NA	NA	NA	NA	NA	NA
20–29	Reserved–robustness	NA	NA	NA	NA	NA	NA
30	Traceroute	0–1	Varies	Opt	Opt	M	M
31	Datagram conversion error	0–11	Varies	?	?	?	?
32	Mobile host redirect	0	Varies	Opt	Opt	Opt	Opt
33	IPv6 where-are-you	0	?	Opt	Opt	Opt	Opt
34	IPv6 I-am-here	0	?	Opt	Opt	Opt	Opt
35	Mobile registration request	0, 16	Varies	Opt	Opt	Opt	Opt
36	Mobile registration reply	0, 16	Varies	Opt	Opt	Opt	Opt
39	SKIP	0	Varies	Opt	Opt	Opt	Opt
40	Photurius	0–3	Varies	Exp	Exp	Exp	Exp

Exp, expired; Obs, obsolete; Opt, optional; M, mandatory; NA, not applicable.

Other ICMP Messages

Some ICMP messages do not fall neatly into either the error or query category. These messages are typically used in specialized circumstances. The other 25 ICMP messages are listed in Table 7.4, again showing how routers and hosts should handle each type.

The messages displayed in Table 7.4 are less intuitive than others. Many of the other messages are relatively new, apply to special circumstances, and not much has been published about their use.

Very little has been written on the use of the alternate host address message and the table is filled in with more suggestions than anything else. Router advertisement and solicitation messages are defined in RFC 1256 as part of "neighbor discovery" for IPv4 and a way around network administrators needing to know local router addresses.

The traceroute message was introduced in RFC 1393 and was supposed to be a more formal way to perform a traceroute, but never really caught on. RFC 1393 describes an alternate traceroute method that uses a single packet with an IP header Traceroute option field and uses the answering ICMP Type = 30 messages from routers to gather the same information while using far fewer messages. However, support for this method is not mandatory on routers, making this form of traceroute problematic.

Datagram conversion errors are part of the "Next Generation Internet" protocol using 64-bit addresses described in RFC 1475 and occurring when packets cannot be converted to the new format. The mobile-related messages (32, 36, and 37) are part of Mobile IP (or "IP Mobility"). SKIP is the Simply Key Management for Internet Protocols and is used for Internet security. So is Photurius, an experimental aspect of IPSec that has four codes: one reserved (0), one for an unknown IPSec Security Parameter Index (SPI, 1), one for failed authentication (2), and one for failed decryption (3).

SENDING ICMP MESSAGES

Few TCP/IP protocols have been the subject of as much tinkering and add-on functionality as ICMP. The original specification of ICMP was in RFC 792 and refined in RFC 1122 (Host Network Requirements) and RFC 1812 (Router Requirements). RFC 1191 added path MTU discovery functions to ICMP, RFC 1256 added router discovery, and RFC 1393 extended traceroute functions with a special message type not often used.

But at heart, ICMP is a collection of predefined messages to indicate very specific conditions. If the sender of a packet receives an ICMP message that involves ICMP itself (the query messages), then ICMP deals with it directly. Otherwise, other protocols are notified. (Unreachable ports are reported to UDP, which lacks the segment tracking that TCP has, and so forth.) The precise response of an application to an ICMP message can vary, but usually the error is reported to the user so that corrective action (even if it's just "Stop doing that!") can be taken.

When ICMP Must Be Sent

Systems that detect a packet error and discard the packet may or may not send an ICMP message back to the originating host. Usually it depends on whether the error is transient or semipermanent.

Things like invalid checksums are ignored in TCP/IP, because these are considered to be transient failures that should not persist. The philosophy is that if the data are important, the sender will simply resend. Transient errors are unlikely to repeatedly manifest themselves in a chain of packets, and thus do not indicate a network-wide problem.

However, semipermanent errors such as invalid IP addresses need to be reported to the originator. These are fundamental problems with the network or in the way that the application is trying to use the network. The sender must either stop or change the content of the packets.

It is important to realize that the presence of many ICMP messages on a network does not mean that things are not working well, nor does the lack of ICMP messages mean that the network is working fine.

Most users see only a handful of ICMP message types, especially those used for ping and `traceroute`, such as the Time Exceeded, Timestamp Reply, Destination Unreachable, and Echo messages.

When ICMP Must *Not* Be Sent

ICMP also establishes situations when ICMP messages must *not* be sent. Transients like checksum errors or intermittent link-level failures are clear examples, but ICMP goes further than this. Generally, error messages should not be sent if they will generate more network traffic and add little new information to what is obvious to the sender.

For example, RFC 1122 says that ICMP error message should never be sent if a receiver gets the following:

- ICMP error message (e.g., errors in ICMP checksums should not be reported as errors)
- Internet Group Management Protocol (IGMP) message (IGMP is for multicast, and multicast traffic tends to multiply exponentially on the network, and one error could trigger many error messages)
- Packet with a broadcast or multicast destination address (another traffic-oriented rule)
- Link-layer frame with broadcast or multicast address
- Packet with a special source address (all zeros, loopback, and so on)
- Any fragment other than the first fragment of a fragmented packet

PING

Most people who know little about how TCP/IP works usually know of the ICMP-based application known as ping. The original metaphor was the "ping" of a naval sonar unit. Ping is a simple Echo query-and-response ICMP message that is used to see if another

device is up and reachable over the network. A successful ping means that network administrators looking at problems can relax a great deal: The network routers on the path and at least two hosts are running just fine.

Ping implementations and the parameters supported vary greatly among operating systems and routers (most routers support ping). Some only send four packets and quit, unless told to send more. Others send constantly until told to stop. The parameters can usually set many of the IPv4 packet header fields such as TTL, TOS, and so on to specific values.

Usually, Unix versions use the PID as the Identifier field in the ping message, but Linux increments this based on application calls. Unix ping messages are usually 56 bytes long, but Windows implementations use only 32 bytes. The payload of the ping message echoed back to the sender typically consists of an 8-byte timestamp and a fill pattern. The timestamp can be used to roughly calculate round-trip delays through the network (in milliseconds).

Ping has some quirks that users should be aware of. First, small pings (maybe 56 or 64 bytes in the packet) often work fine, while larger pings with more realistic payload sizes do not go through reliably. That's what users care about—the network is struggling with real data packets. Seeing a small ping getting through reliably is not always helpful.

Also, the round-trip times are not often vital information. You expect round-trip times to go up as packet sizes increase, and that's typically what is observed. The same is true if the network is heavily loaded. But this is a relative, not absolute, observation. Only when round-trip times are longer than expected, or if they vary by huge amounts, is there an indication that something is wrong.

Part of the reason that round-trip times are not reliable is that routers (in particular) and even hosts might process ICMP Echo requests at a lower priority than other traffic. In fact, in many router architectures, ICMP message processing requires a trip to the control-plane processor, while transit traffic is forwarded in the forwarding-plane hardware.

We'll be using ping extensively in many chapters in this book.

TRACEROUTE

Traceroute is not an ICMP-based network utility in the same sense that ping is. However, because traceroute uses ICMP messages to perform its functions, and for many people the next step after ping is traceroute, this is the place to discuss this utility. We'll use traceroute heavily in Chapter 9 and throughout the rest of the book.

After ping has been used to verify that an IP address is reachable over the network, the next logical step is to determine *how* the packets make their way to the destination and back. In other words, we would like to trace the route from source to destination (the reverse path is normally the same). Yes, IP networks route around failures and routing tables can change, but paths are usually stable on the order of hours if not days when things are not going completely haywire. Of course, paths might also simply be asymmetric, yet stable, so it is not only path changes that are challenging for traceroute interpretation.

Traceroute implementations vary even more than those for ping. Some have graphical displays and use other Internet utilities to display location and administrative information about the routers and networks uncovered. This in turn has made many network administrators so nervous that they routinely block traceroute ICMP messages with firewalls or route filters to hide topology details. In fairness, the Internet is no longer a teaching tool or good place to explore the limits of knowledge, and there are so many disruptive or even malicious people on the Internet, that a certain amount of anxiety is completely understandable (which is why a network such as the one used for this book makes so much sense).

On Unix-based systems, traceroute often sends a sequence of three UDP packets (a typical default is three) to an invalid port on another host (this number starts at 33434). The utility can also use ICMP Echo requests, which is what the Windows version does. Some versions even use TCP (a utility called `tcptraceroute`).

Whatever the type of packet, the TTL field is initially set to 1 in the three packet set, so the first router along the path should generate an ICMP Time Exceeded message to the sender. The round-trip delay in the timestamp field and IP address of the router is recorded by the sender and another set of packets is sent, this time with the TTL set to 2. These packets are discarded by the second router, and another ICMP message is sent back. The process is repeated until the destination host is reached and the host returns a Destination Port Unreachable message, or until a firewall is encountered that blocks the ICMP messages or unsolicited UDP traffic. (These messages mimic port scans and are sometimes blocked, as mentioned earlier in this chapter.)

The end result should be a list of the routers on the path from source to destination (or the firewall) that also records round-trip delays. In some cases (sometimes many cases), some routers will not respond to the TTL "timeout" with an ICMP message, but simply silently discard the offending packet. If the packet does not return within the timeout window (Cisco routers use a default timeout of 2 seconds), most traceroute implementations indicate this with an asterisk (*) or some other placeholder and just keep going, trying to reach the next router. (The appearance of the asterisk does *not* necessarily mean that the packet was lost.)

One nagging traceroute issue is the number of messages exchanged over the network needed to reveal fairly basic information. RFC 1393 describes an alternate traceroute method that uses a single packet with an IP header Traceroute option field and uses the answering ICMP Type = 30 messages from routers to gather the same information while using far fewer messages. However, support for this method is not mandatory on routers.

We'll use traceroute a lot in many of the chapters of this book too.

PATH MTU

ICMP messages also play a role in path MTU discovery. We've already mentioned the MTU as a critical link parameter determined by the maximum frame size. Packets, including all headers, that fit inside the smallest frame size on the path from source to

destination do not have to be fragmented and do not incur any of the penalties that fragmentation involves.

But tuning the path MTU size to packet size has another network benefit: This practice maximizes throughput and minimizes the overhead required to move large messages from system to system. Overhead bytes are those that do no useful work in terms of data transfer, but are necessary for the data transfer to take place at all.

Consider a data transfer using 68-byte MTUs, once the smallest size possible. If usual IP and TCP headers are used, which are 20 bytes each, they will take up 40 bytes of the packet, leaving only 28 bytes for data. So a whopping 59% (40/68) of the packet is made of overhead. And a minimum of 35,715 packets need to be sent, routed, and processed to transfer every megabyte of data. Bumping this MTU size up to 576 bytes (a typical default value and the functional minimum for IPv4) cuts the overhead down to about 7% (40/576) and requires only 1866 packets per megabyte of data, about 5% of the previous number of packets.

Using the typical Internet frame size of 1500, the overhead shrinks to about 2.5% and the number of packets required for a megabyte of data becomes a respectable 685. Larger MTUs have proportional benefits. (It is sometimes pointed out that bigger packets are not always more efficient; they can add delay for smaller units of traffic, a phenomenon often called "serial delay," and on high bit error links, larger packets almost guarantee that a bit error requiring a resend will occur during frame transmission. On older, more error-prone networks, throughput shrank to zero as packet size grew.)

The 576-btye MTU size was selected as a compromise between latency ("delay") and throughput for modems and low-speed serial SLIP implementation. This is directly related to the serialization delay discussed below. And use of an MTU size smaller than 512 precludes the use of the Dynamic Host Configuration Protocol (DHCP).

Now, TCP can adjust this message size, no matter what the default, but UDP traffic, which is growing, cannot. Of course, every link from host to router to router to host can have a different MTU size. That is what path MTU discovery is all about. It works via the following:

- Setting the DF flag in the IP header to 1 (don't fragment)
- Sending a large packet to the destination to which the path MTU is being determined
- Seeing if any router responds with an ICMP Destination Unreachable message with Code = 4 (fragmentation required but don't fragment bit is set)
- Repeating the first three steps with a smaller packet size

The process stops when a message is received from the destination host, showing that a path MTU of this size works. Again, paths are fluid on TCP/IP router networks, but they are remarkably stable considering all that can go wrong. By the way, it is assumed that the path MTU for outbound packets is the same as the path MTU size for inbound packets, but this is *not* true just often enough to make the process unnecessarily haphazard.

Table 7.5 Path MTU Plateaus for Various Network Link Types	
Plateau Size in Bytes	**Description**
65535	Maximum MTU and packet size
32000	A value established "just in case"
17914	16-Mbps IBM token ring LANs
8166	IEEE 802.4 token bus LANs
4352	FDDI (100 Mbps fiber rings)
2314	Wireless IEEE 802.11b native frame (often "adjusted" to 1492)
2002	4-Mbps IEEE 802.5 token ring (recommended value)
1492	IEEE 802.3 LANs (also used in 802.2)
1006	SLIP
508	Arcnet (proprietary LAN from Datapoint)
296	Some point-to-point links use this value
68	Minimum MTU size

The path MTU "seed" or probe size and adjustment steps are not randomly chosen. A series of "plateaus" representing common link MTU limits has been established. Some of these are shown in Table 7.5.

In practice, as important as the path MTU size is, little is often done about the MTU size except to change the default to 1500 bytes if the default value is less (it usually is). This is because most networks that hold the source and destination networks are Ethernet LANs that do not support 9000-byte jumbo frames. Between routers, WAN links typically support larger MTU sizes (around 4500 bytes or larger), but that does no good if the end system can only handle 1500-byte frames. However, WAN links with MTUs greater than 1500 bytes allow the use of tunnel encapsulation of 1500-byte MTU packets without the need for fragmentation, so the larger MTU is not actually wasted.

ICMPv6

A funny thing happened to ICMP on its way to IPv6. It didn't work. ICMP, now officially called ICMPv4, is built around the IPv4 packet header and things that could go wrong with it. And not only is the IPv6 packet header different, as well as many fields and address sizes, but many functions added to IPv4 that affected ICMPv4 were scattered in separate RFCs and implementation varied. These functions are systematized in ICMPv6.

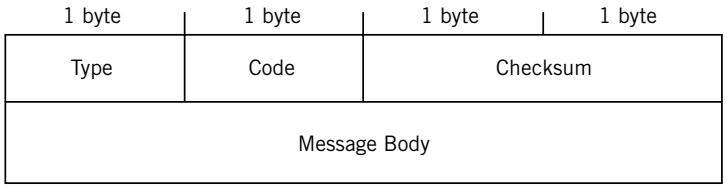
ICMPv6 makes some major changes to ICMPv4:

- New ICMPv6 messages and procedures replace ARPs.
- There are ICMPv6 messages to help with automatic address configuration.

- Path MTU discovery is automatic, and a new Packet Too Big message is sent to the source for over-large packets because IPv6 routers do not fragment.
- There is no Source Quench in ICMPv6 (it is obsolete in ICMPv4, but still exists).
- IGMP for multicast is included in ICMPv6.
- ICMPv6 helps detect nonfunctioning routers and inactive partner hosts.
- ICMPv6 is so different that it now has its own IP protocol number. IPv6 uses the next header value of 58 for ICMPv6 messages.

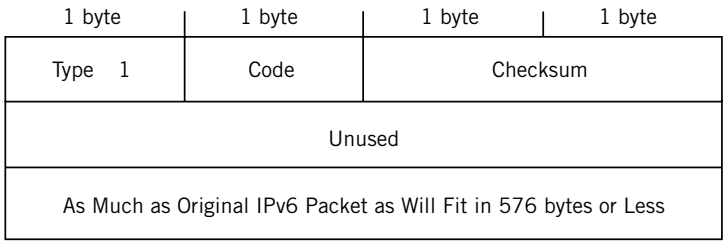
Basic ICMPv6 Messages

The general ICMPv6 message format is similar to ICMPv4, but somewhat simpler. The structure of a generic ICMPv6 message and the common Destination Unreachable message are shown in Figure 7.6. ICMPv6 error messages are in the range 0 to 127. Some of the most common are shown in the figure as well.



(a)

Basic ICMPv6 Type field values:
1 Destination Unreachable
2 Packet Too Big
3 Time Exceeded
4 Parameter Problem
5 Redirect
128 Echo Request
129 Echo Reply



(b)

FIGURE 7.6

ICMPv6 message formats, which can be compared to the IPv4 versions in Figure 7.4. (a) Generic ICMPv6 message format. (b) ICMPv6 Destination Unreachable message.

Table 7.6 Destination Unreachable Codes for ICMPv6	
Code	**Meaning**
0	No route to destination
1	Communication with destination administratively prohibited
2	Next destination in the IPv6 Routing header is not a neighbor, and this is a strict route (routing headers are not currently supported)
3	Address unreachable
4	Port unreachable

Destination Unreachable

In ICMPv6, the Destination Unreachable message type is Type = 1. The codes that can be compared to Table 7.2 IPv4 codes number only five and are listed in Table 7.6.

Packet Too Big

A router sends an ICMPv6 Packet Too Big message to the source when the packet is bigger than the MTU for the next-hop link. The next-hop link's MTU size is reported in the message. In ICMPv4, this type of information was supplied in the Destination Unreachable message. The format of the Packet Too Big message is shown in Figure 7.7.

Time Exceeded

An ICMPv6 Time Exceeded message is sent by a router when the Hop Limit field of the IPv6 header reaches 0 (ICMPv6 Code = 0) or when the receiver's fragment reassembly timeout (senders can still fragment under IPv6) has expired (ICMPv6 Code = 1). The

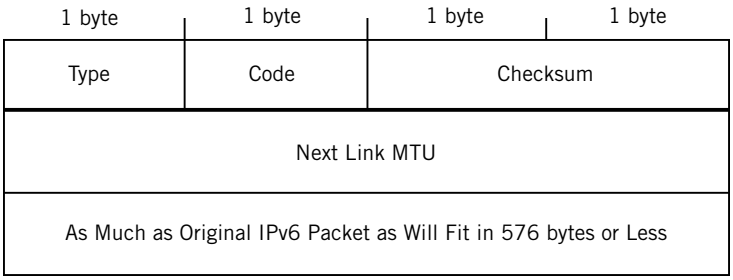


FIGURE 7.7

ICMPv6 Packet Too Big format, showing details of the fields used.

Table 7.7 Parameter Problem Codes and Meanings

Code	Meaning
0	Erroneous header field encountered
1	Unrecognized next header type encountered
2	Unrecognized IPv6 option encountered

format is the same as for the ICMPv6 Destination Unreachable message, except that the Type is 3.

Parameter Problem

As in ICMPv4, an ICMPv6 Parameter Problem message is sent by a host or router that cannot process a packet due to a header field problem. The codes are listed in Table 7.7.

Echo Request and Reply

Under IPv6, ping becomes "pingv6" (the name is not important) and uses ICMPv6 Echo Request and Reply messages, but with Type = 128 used for requests and Type = 129 used for replies.

Neighbor Discovery and Autoconfiguration

ICMPv6 provides a number of neighbor discovery functions that help with:

- Location of routers
- IPv6 parameter configuration
- Location of local hosts
- Neighbor unreachability detection
- Automatic address configuration and duplicate detection

These ICMPv6 functions use the following message types:

Router Solicitation Type = 133 messages are sent by a host to ask neighbor routers to make their presence known and provide link and Internet parameters, similar to the ICMPv4 Router Solicitations. The message is sent to the all-router link-local IPv6 multicast address.

Router Advertisement Type = 134 messages are sent periodically by every router and in response to a host's Router Solicitation, similar to the ICMPv4 Router Advertisements. The message is sent either to the all-nodes IPv6 multicast address (unsolicited) or to the querying host (solicited).

Neighbor Solicitation Type = 135 messages are used, as ARP in IPv4, to find the link-layer address of a neighbor, verify the neighbor is still reachable with the cached entry, or check that no other node has this IPv6 address. These messages also detect unresponsive neighbors.

Neighbor Advertisement Type = 136 messages are sent in response to Neighbor Solicitation messages and resemble the ARP response. Nodes can also announce changes in link-layer addresses by sending unsolicited.

Neighbor Advertisements. Redirect Type = 137 messages perform the same role as the ICMPv4 redirect.

Routers and Neighbor Discovery

IPv6 routers provide their hosts with basic configuration and parameter information using Router Advertisement messages sent to the all-hosts link-local IPv6 multicast address. Hosts do not have to wait for these periodic router messages and can send a Router Solicitation message at startup. This reply is sent to the host's link-local address.

Each router will supply data that includes the following:

- Link-layer router address
- MTU for any links that have variable MTUs
- List of all prefixes and lengths used on the LAN (the specification says "link")
- Prefixes that a host can use to create its addresses
- Default Hop Limit value to use on packets
- Values for miscellaneous timers
- Location of a DHCP server where the host should fetch more information

Note that the Router Advertisement (RA) will indicate the availability of a DHCP server for stateless configuration (RA option O), or the requirement to perform stateful configuration (RA option M). The location of the DHCPv6 server is not specified, merely that it's available and what the requirements are for use.

Interface Addresses

Each IPv6 interfaces has a *list* of addresses and prefixes associated with it, including a unique link-local address. In theory, this should allow LANs to easily migrate from one ISP to another simply by changing prefixes and allowing the older prefix to age-out of the host. In practice, migration between IPv6 service providers is not as simple. DNS entries do not just "flop over," and host and router configuration (and firewalls!) have static configuration parameters. The point is that router advertisements assign a lifetime, which must be refreshed, to advertised prefixes. This also makes it easier to move hosts from LAN to LAN.

Each host can use some of the prefixes and lengths advertised by the routers (if they are flagged for this use) to construct host addresses. A private (ULA local) or global address can be constructed by appending a unique interface identifier to the advertised prefix and added to the list of the host's IPv6 addresses.

Router advertisements can also direct a host to a DHCP server that can assign addresses chosen by a network administrator.

Neighbor Solicitation and Advertisement

One of the problems with ARP in IPv4 was that it was essentially a frame-level protocol that did not fit in well with the IP layer at all. In IPv6, "ARPs" are ICMPv6 messages. ICMPv6 packets can be handled easily at the IPv6 layer, and can be authenticated and even encrypted with IPSec techniques.

In addition to finding neighbor link-layer addresses, the Neighbor Solicitation and Advertisement messages are used to find "dead" routers and partner hosts, and detect duplicate IPv6 addresses.

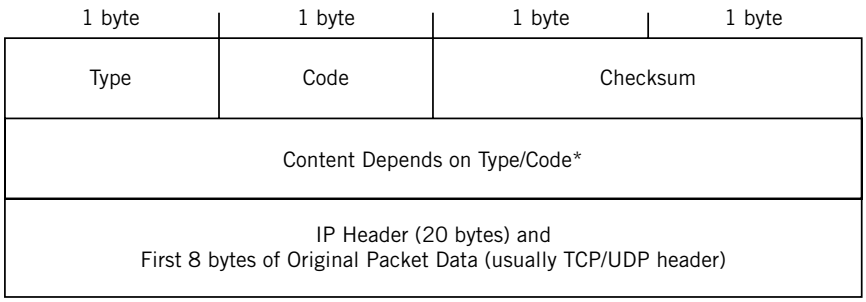
Neighbor Solicitation messages are sent to the solicited-node IPv6 multicast address, which is formed by appending the last 3 bytes of an IPv6 link-local address to a multicast prefix. The use of the multicast address cuts down on the number of hosts that has to pay attention to the "ARP" message (in fact, only the target system should process the request). The sender also includes its own link-layer address with the message.

Duplicate IP addresses are always a problem. Before a system can claim an IPv6 address or any other address *not* constructed by adding a link-local address to a prefix, the system sends a Neighbor Solicitation message asking whether any neighbor already has that IPv6 address. This message uses the special IPv6 Unspecified Source address as the source address, because you can't ask about a source address by using the source address! If the address is in use, the response is multicast to inform all devices. Addresses that are manually assigned are tested in the same fashion.

Dead routers and hosts are detected by a sending unicast Router and Neighbor Solicitation message to the device in question.

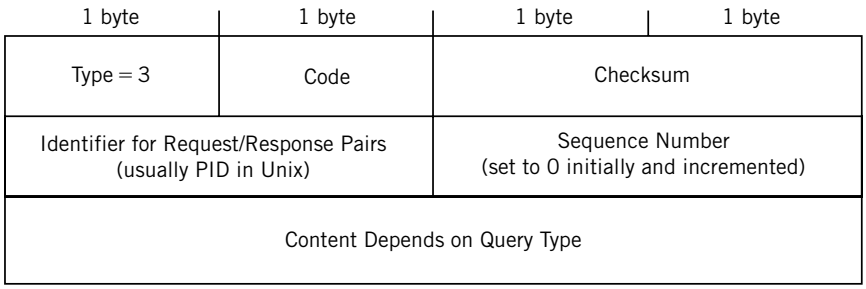
QUESTIONS FOR READERS

Figure 7.8 shows some of the concepts discussed in this chapter and can be used to help you answer the following questions.



(a)

**Usually all 0 (unused) except for:*
Type 3/Code 4: Destination unreachable, fragmentation needed
(fields are 2 bytes unused and 2-byte link MTU size)
Type 3/Code 5: Destination unreachable, redirect (field is router IP address)
Type 12/Code 0: Parameter problem (field is 4-bit pointer to parameter, rest all 0)



(b)

FIGURE 7.8

ICMP error and query messages in general. (a) Error message. (b) Query message.

1. How many types of error-reporting messages are there in ICMP? How many pairs of query messages are there in ICMP?
2. Which pair of ICMP messages can be used to obtain the subnet mask?
3. Which kind of ICMP message notifies a host that there is a problem in the packet header?
4. Which fields are used for the ICMP checksum calculation?
5. A ping sent to IP address `10.10.12.77` (the address assigned to `bsdserver`) on LAN2 is successful. Later, it turns out that the `bsdserver` was powered off for maintenance at the time. What could have happened?