Chapter 9

Internet Control Message Protocol

Objectives

Upon completion you will be able to:

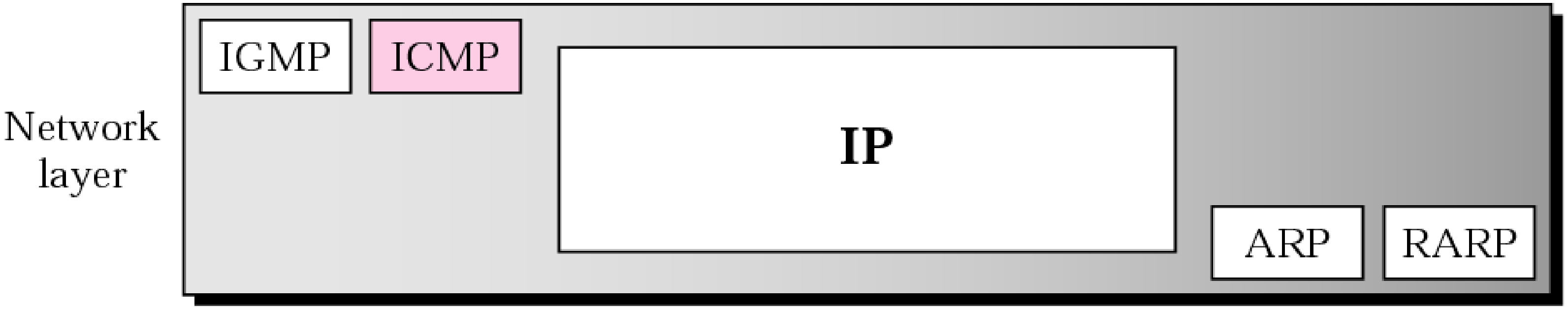
- *Be familiar with the ICMP message format*
- *Know the types of error reporting messages*
- *Know the types of query messages*
- *Be able to calculate the ICMP checksum*
- *Know how to use the ping and traceroute commands*
- *Understand the modules and interactions of an ICMP package*

ICMP encapsulation

IP has two deficiencies:

1. No error control
2. Lack of assistance mechanisms

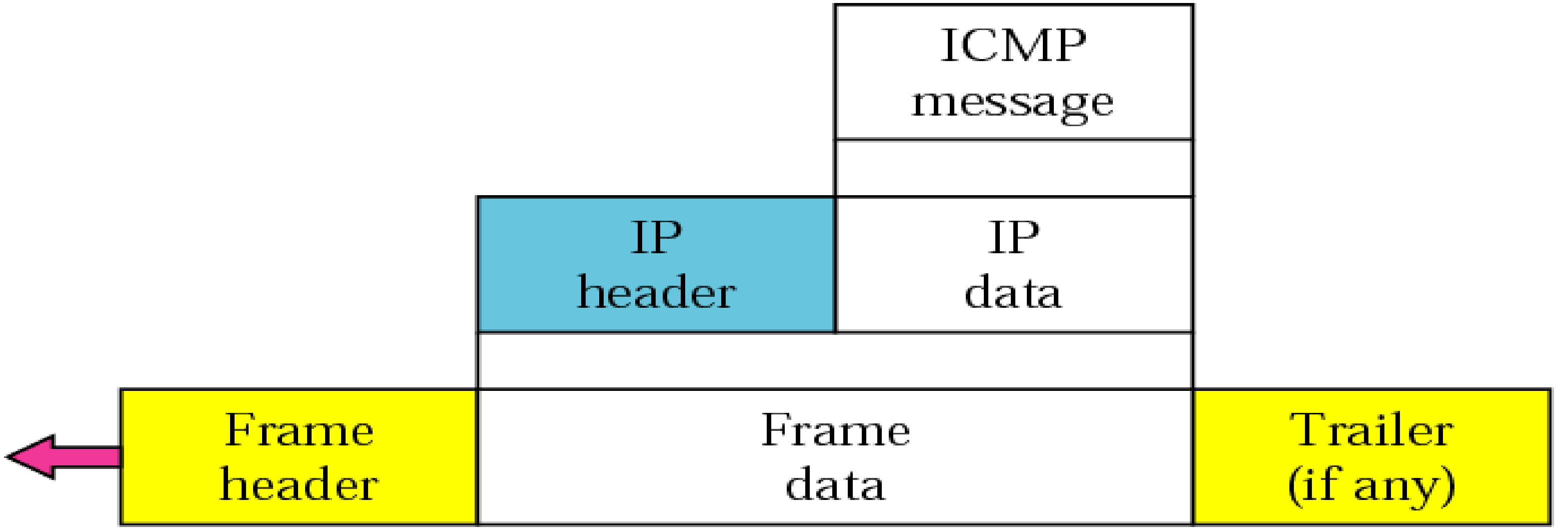
ICMP is companion of IP protocol



ICMP encapsulation

ICMP Messages are first encapsulated in IP Datagrams before going to lower layer.

Value of Protocol field in IP Datagram set to 1, to indicate the data is ICMP message.



9.1 TYPES OF MESSAGES

ICMP messages are divided into error-reporting messages and query messages.

The error-reporting messages report problems that a router or a host (destination) may encounter.

The query messages get specific information from a router or another host. Ex. Nodes can discover their neighbors. (Traceroute)

Figure 9.3 *ICMP messages*

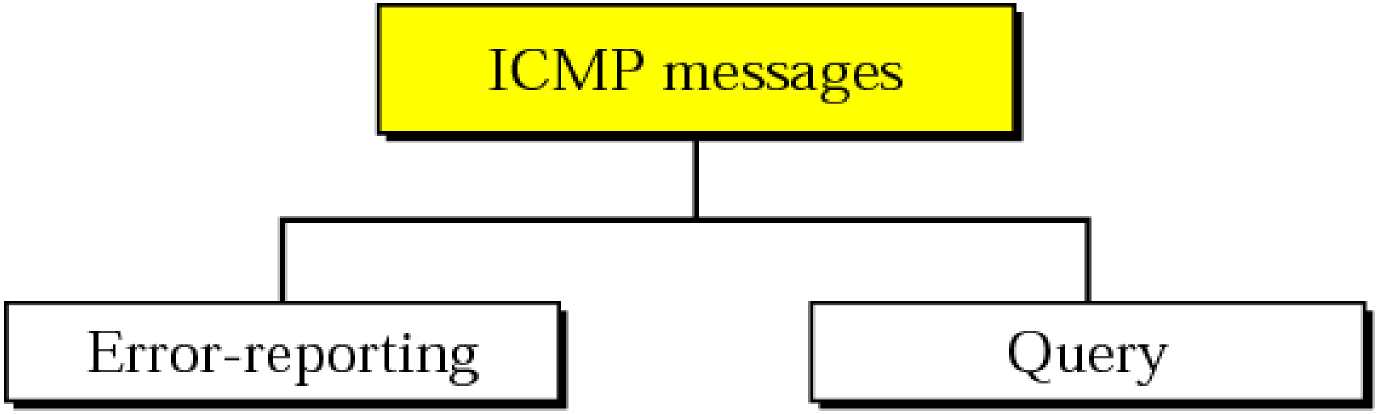
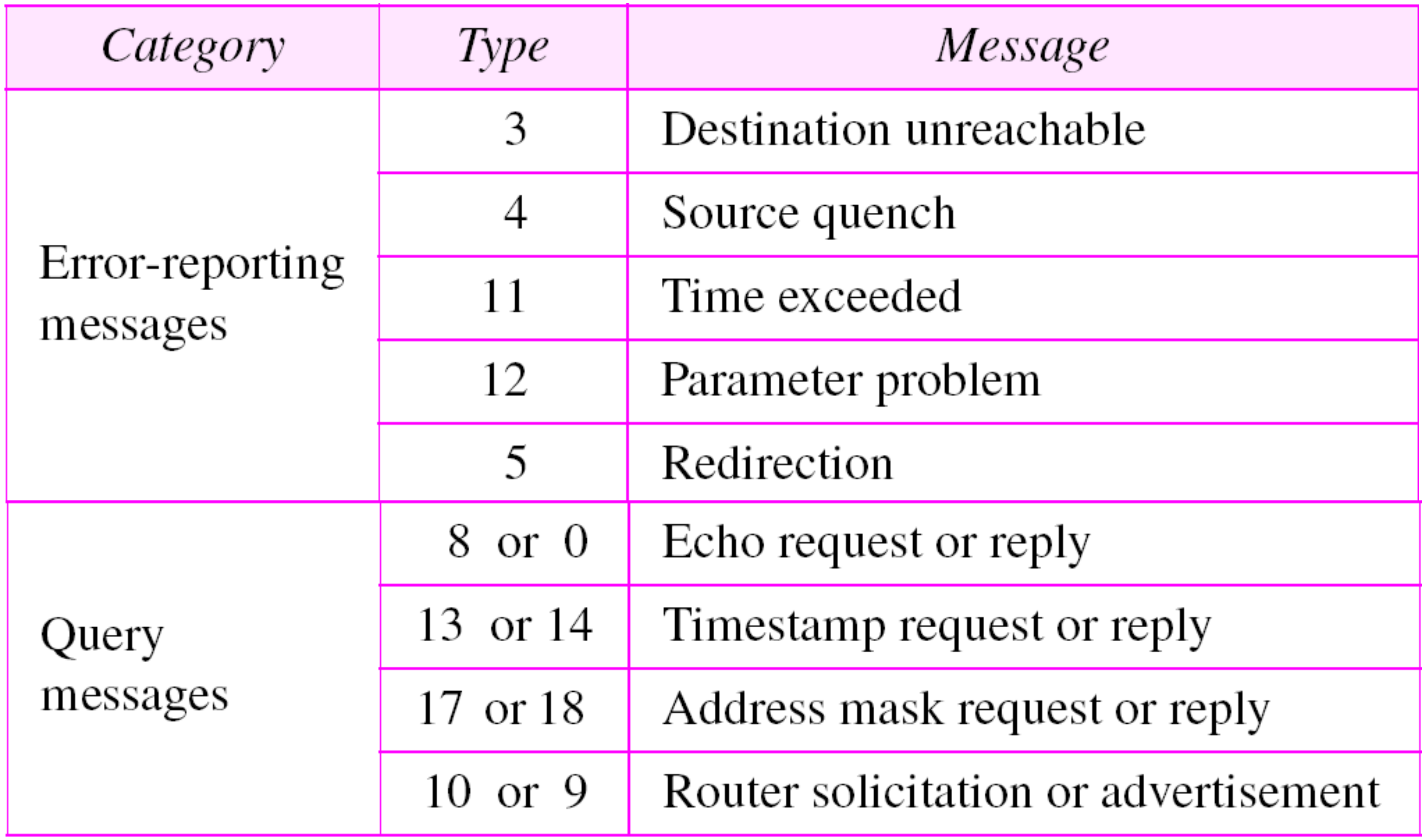


Table 9.1 ICMP messages in each category



Category	*Type*	*Message*
Error-reporting messages	3	Destination unreachable
	4	Source quench
	11	Time exceeded
	12	Parameter problem
	5	Redirection
Query messages	8 or 0	Echo request or reply
	13 or 14	Timestamp request or reply
	17 or 18	Address mask request or reply
	10 or 9	Router solicitation or advertisement

9.2 MESSAGE FORMAT

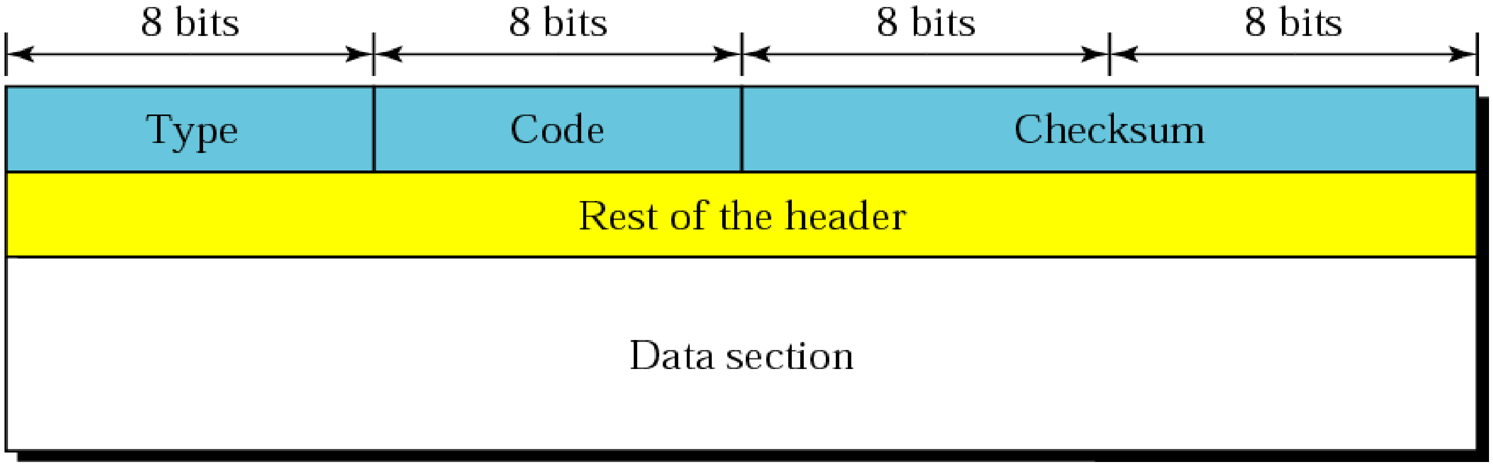
An ICMP message has an 8-byte header and a variable-size data section.

Although the general format of the header is different for each message type, the first 4 bytes are common to all.

General format of ICMP messages

Type = Defines type of ICMP message
Code = Specifies the reason for particular message type.

Rest of header is specific for each message type.



8 bits
8 bits
8 bits
8 bits
Type
Code
Checksum
Rest of the header
Data section

9.3 ERROR REPORTING

IP, as an unreliable protocol, is not concerned with error checking and error control. ICMP was designed, in part, to compensate for this shortcoming. ICMP does not correct errors, it simply reports them.

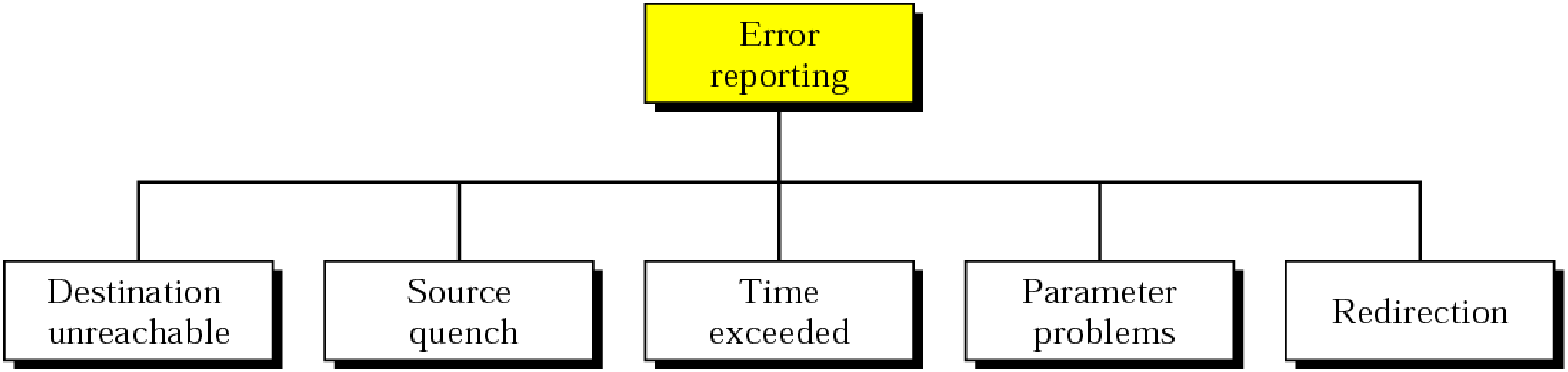
The topics discussed in this section include:

Destination Unreachable
Source Quench
Time Exceeded
Parameter Problem
Redirection

Note:

ICMP always reports error messages to the original source.

Error-reporting messages (Five Types)



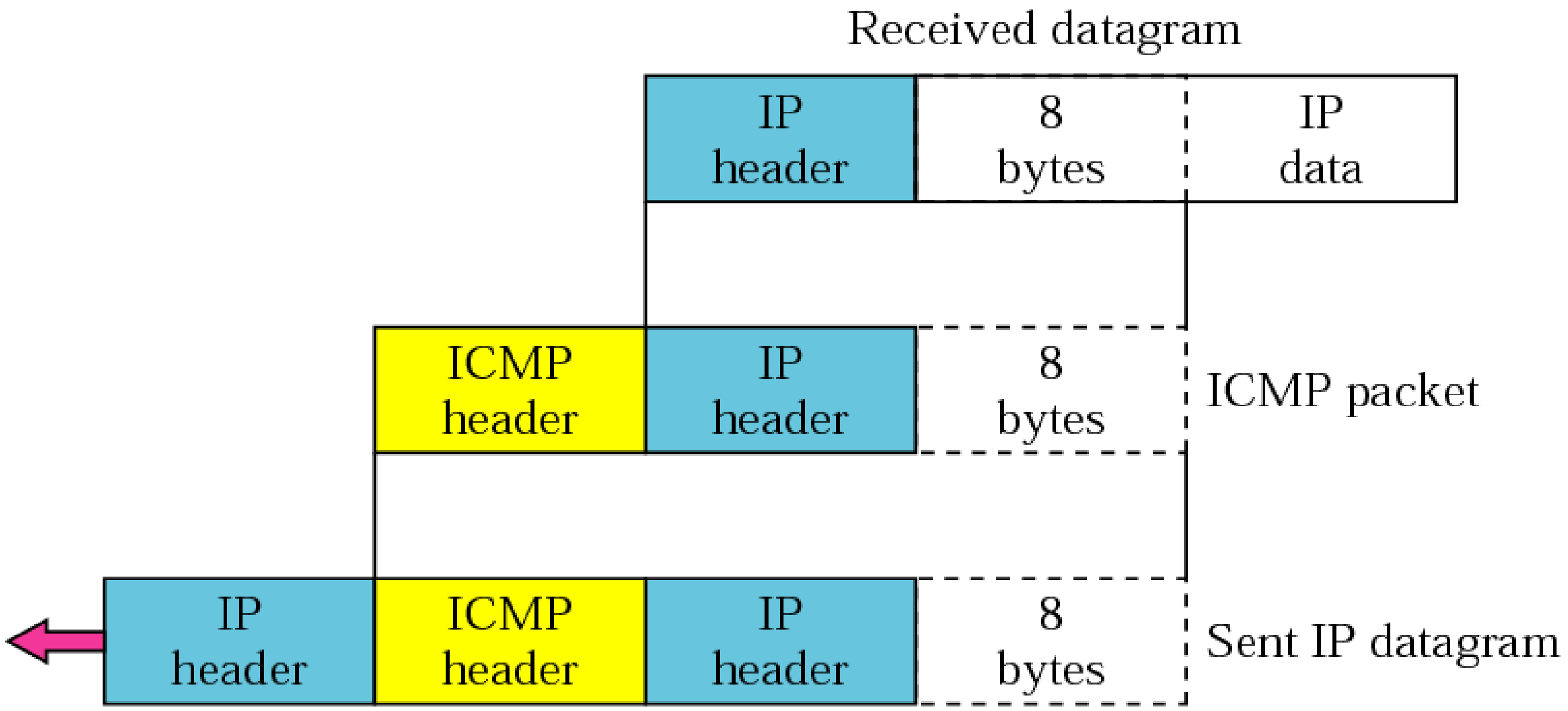
Note:

The following are important points about ICMP error messages:

- ***No ICMP error message will be generated in response to a datagram carrying an ICMP error message.***
- ***No ICMP error message will be generated for a fragmented datagram that is not the first fragment.***
- ***No ICMP error message will be generated for a datagram having a multicast address.***
- ***No ICMP error message will be generated for a datagram having a special address such as 127.0.0.0 or 0.0.0.0.***

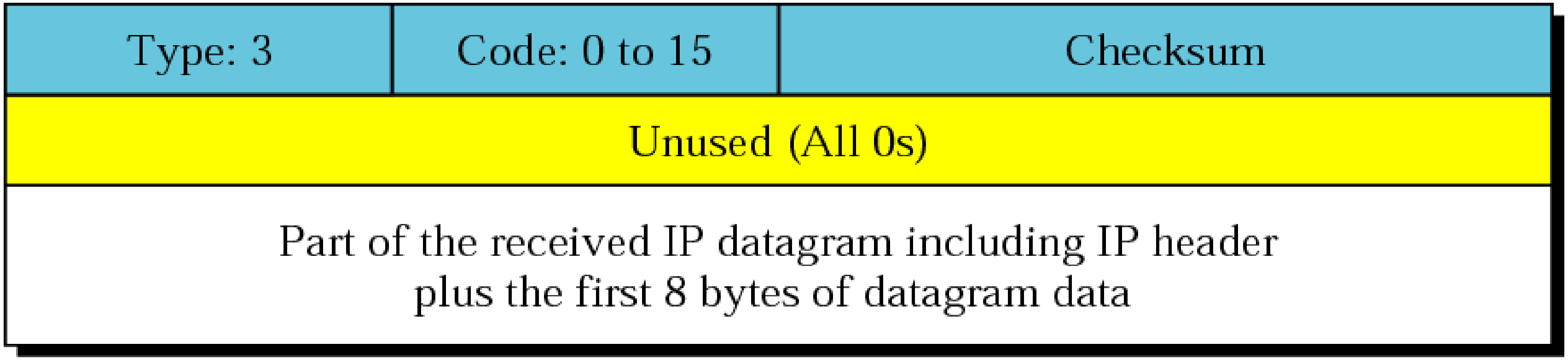
Figure 9.6 ***Contents of data field for the error messages***

ICMP Forms an error packet, which is then encapsulated in an IP Datagram.



Destination-unreachable format

Code field specifies the reason for discarding the datagram.



Type: 3	Code: 0 to 15	Checksum
Unused (All 0s)		
Part of the received IP datagram including IP header plus the first 8 bytes of datagram data		

Destination-unreachable format Codes:

Code 0: Network unreachable due to h/w failure.
Code 1: Host is unreachable
Code 2: Protocol is unreachable
Code 3: Port is unreachable.
Code 4: Fragmentation required but DF is set.
Code 5: Source routing can not be accomplished.
Code 6: Destination network is unknown.
Code 7: Destination host unknown.
Code 8: Source host is isolated
Code 9: Comm. With dst. Network administratively prohibited.
Code 10: Comm. with dst. host administratively prohibited.
Code 11: Network unreachable for specified type of service.
Code 12: Host is unreachable for specified type of service.
.
.

Note:

Destination-unreachable messages with codes 2 or 3 can be created only by the destination host.

Other destination-unreachable messages can be created only by routers.

Note:

A router cannot detect all problems that prevent the delivery of a packet.

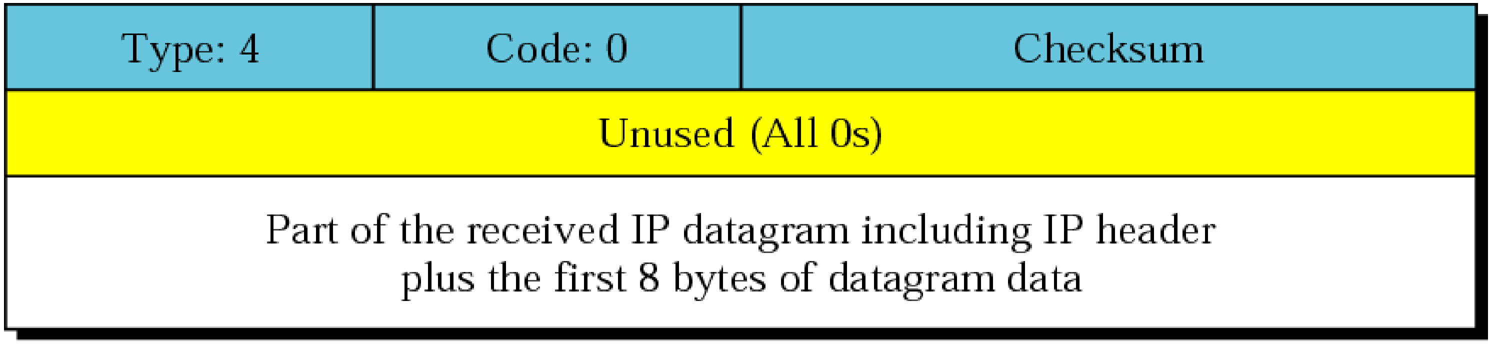
Note:

There is no flow-control mechanism in the IP protocol.

Figure 9.8 *Source-quench format*

A source-quench message informs the source that a datagram has been discarded due to congestion in a router or the destination host.

The source must slow down the sending of datagrams until the congestion is relieved.



Type: 4

Code: 0

Checksum

Unused (All 0s)

Part of the received IP datagram including IP header plus the first 8 bytes of datagram data

Note:

One source-quench message is sent for each datagram that is discarded due to congestion.

Note:

Whenever a router decrements a datagram with a time-to-live value to zero, it discards the datagram and sends a time-exceeded message to the original source.

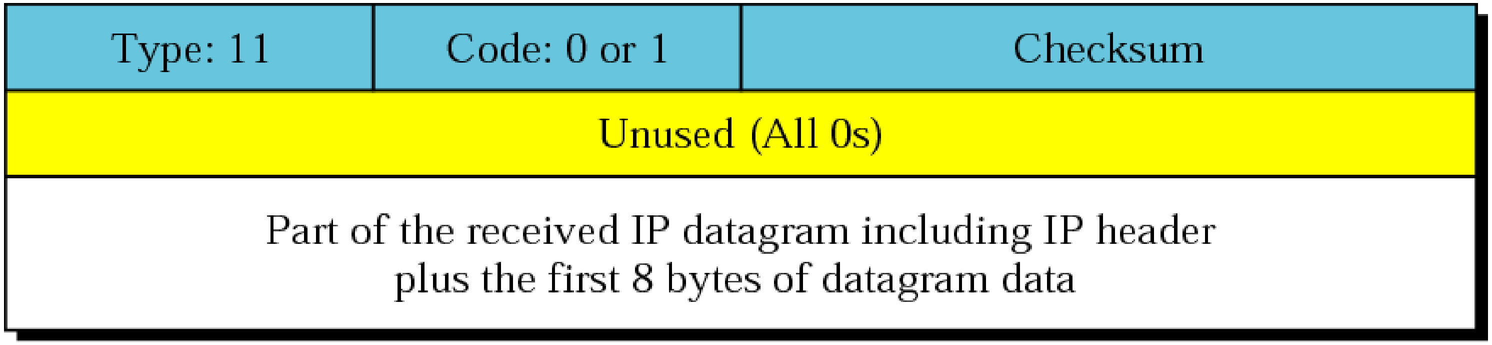
Note:

When the final destination does not receive all of the fragments in a set time, it discards the received fragments and sends a time-exceeded message to the original source.

Note:

In a time-exceeded message, code 0 is used only by routers to show that the value of the time-to-live field is zero. Code 1 is used only by the destination host to show that not all of the fragments have arrived within a set time.

Figure 9.9 *Time-exceeded message format*



Type: 11

Code: 0 or 1

Checksum

Unused (All 0s)

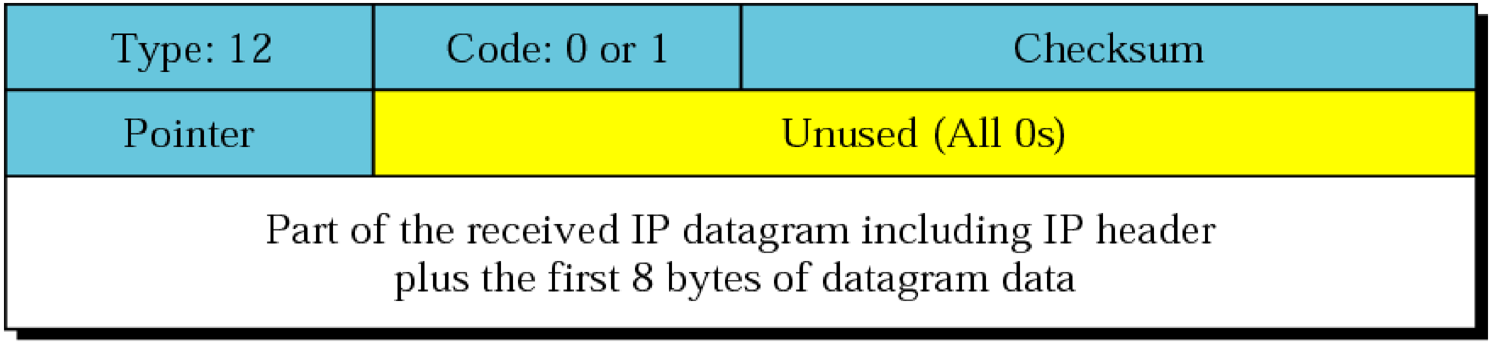
Part of the received IP datagram including IP header
plus the first 8 bytes of datagram data

Note:

A parameter-problem message can be created by a router or the destination host.

Figure 9.10 *Parameter-problem message format*

Code 0: If any ambiguity in the header part
Code 1: The required part of an option is missing. (Pointer Not used in this case).



Type: 12	Code: 0 or 1	Checksum
Pointer	Unused (All 0s)	
Part of the received IP datagram including IP header plus the first 8 bytes of datagram data		

Figure 9.11 *Redirection concept*

A host usually starts with a small routing table that is gradually augmented and updated. One of the tools to accomplish this is the redirection message.

A redirection message is sent from a router to a host on the same local network.

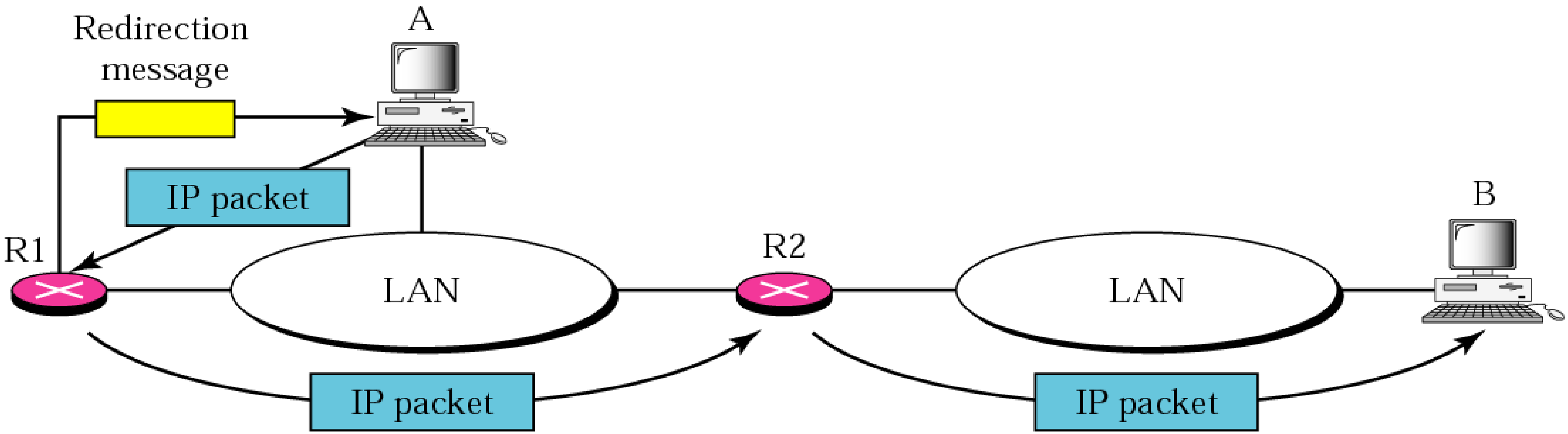
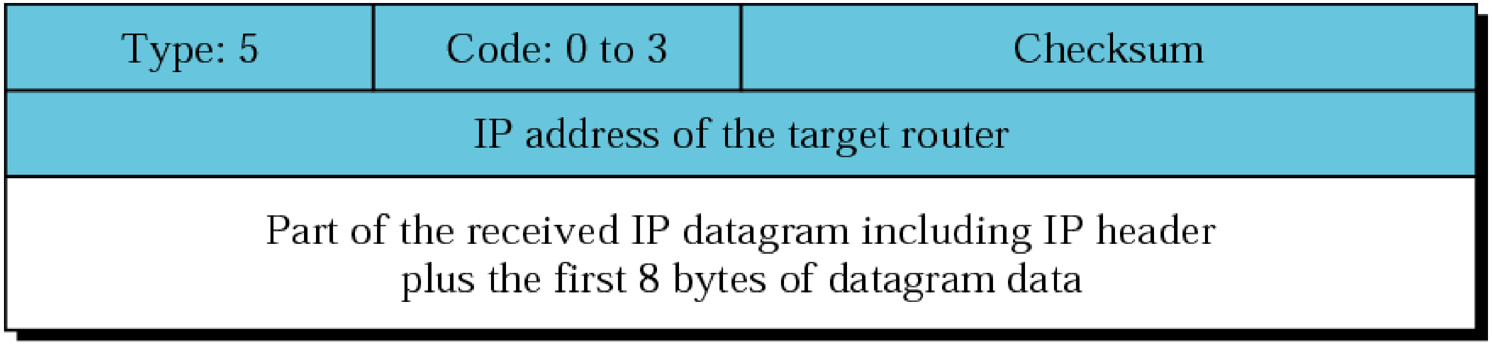


Figure 9.12 *Redirection message format*



Type: 5

Code: 0 to 3

Checksum

IP address of the target router

Part of the received IP datagram including IP header
plus the first 8 bytes of datagram data

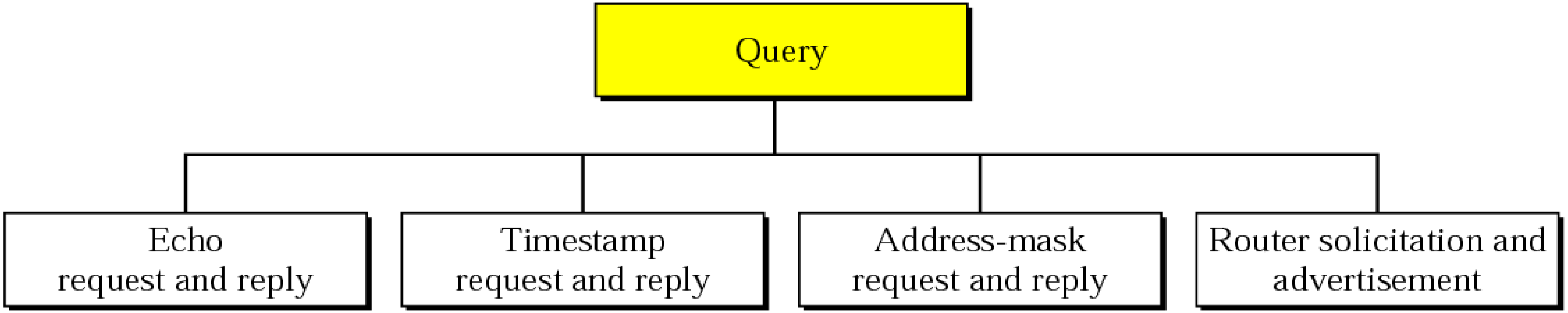
9.4 QUERY

ICMP can also diagnose some network problems through the query messages, a group of four different pairs of messages. In this type of ICMP message, a node sends a message that is answered in a specific format by the destination node.

The topics discussed in this section include:

Echo Request and Reply
Timestamp Request and Reply
Address-Mask Request and Reply
Router Solicitation and Advertisement

Figure 9.13 *Query messages*


Query
Echo request and reply
Timestamp request and reply
Address-mask request and reply
Router solicitation and advertisement

Note:

An echo-request message can be sent by a host or router. An echo-reply message is sent by the host or router which receives an echo-request message.

Echo-request and echo-reply messages can be used by network managers to check the operation of the IP protocol.

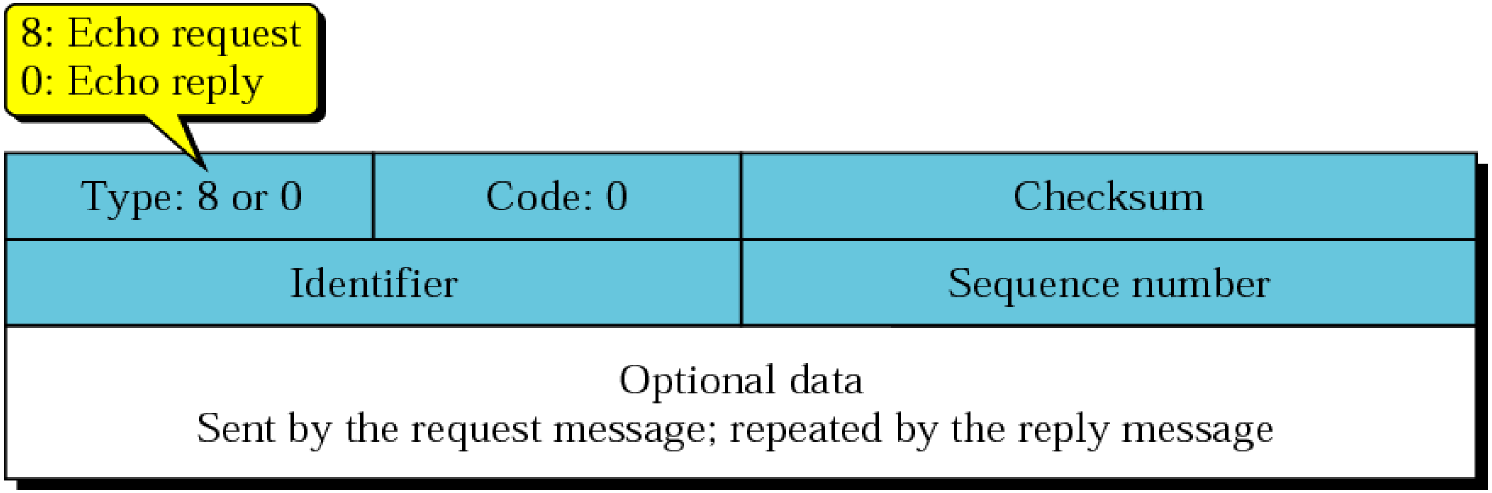
Note:

Echo-request and echo-reply messages can test the reachability of a host. This is usually done by invoking the ping command.

Echo-request and echo-reply messages

Used to determine if there is communication at IP Level. Echo-Reply is the proof that the intermediate routers are receiving, processing, and forwarding IP Datagrams.

Identifier is same as process id.



8: Echo request
0: Echo reply

Type: 8 or 0	Code: 0	Checksum
Identifier		Sequence number
Optional data Sent by the request message; repeated by the reply message		

Timestamp-request and timestamp-reply message format

1. Used to determine Round-trip time needed for an IP Datagram to travel between source and destination.

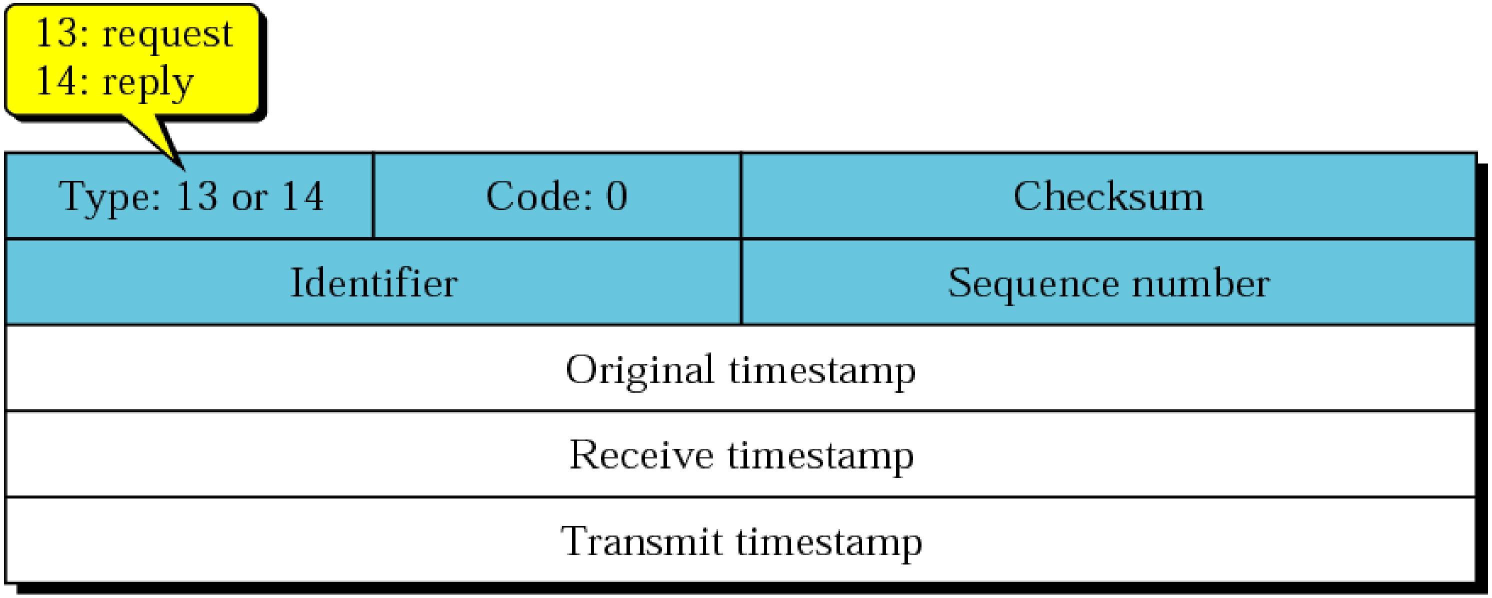
2. Field is 32-bit long. Time measured in milliseconds From midnight.

3.Source fills *original timestamps* field.

4. Receiver fills Receive timestamps and Transmit Timestamps field.

Timestamp-request and timestamp-reply message format

Sending time = receive timestamp – original timestamp
receiving time = returned time – transmit timestamp
RTT = sending time + receiving time



13: request
14: reply

Type: 13 or 14	Code: 0	Checksum

Identifier	Sequence number

Original timestamp

Receive timestamp

Transmit timestamp

Note:

Timestamp-request and timestamp-reply messages can be used to calculate the round-trip time between a source and a destination machine even if their clocks are not synchronized.

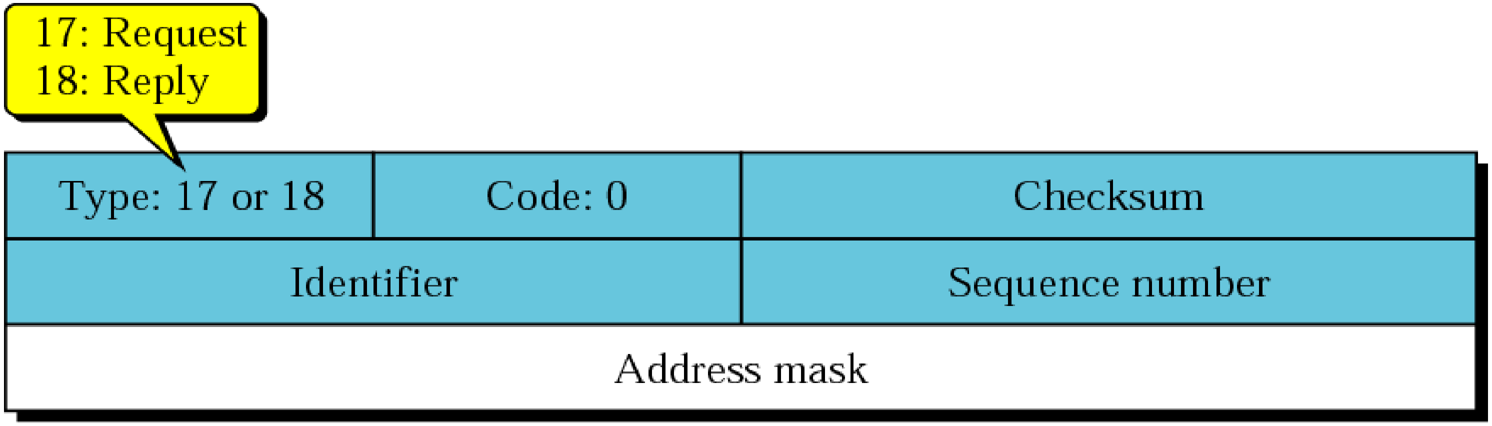
Note:

The timestamp-request and timestamp-reply messages can be used to synchronize two clocks in two machines if the exact one-way time duration is known.

Mask-request and mask-reply message format

To obtain the mask, host may send request message to router Address mask field is 0 in request message.

Masking needed for diskless stations at boot time.



17: Request
18: Reply

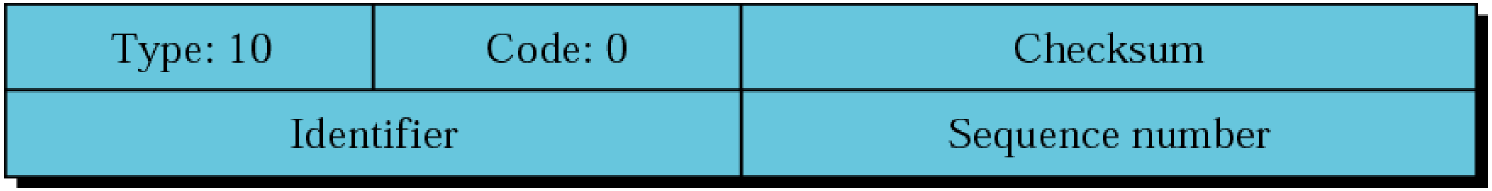
Type: 17 or 18	Code: 0	Checksum
Identifier		Sequence number
Address mask		

Router-solicitation message format

To know if the routers and alive and functioning.

Hosts broadcast router solicitation message, routers reply with routers advertisement message.

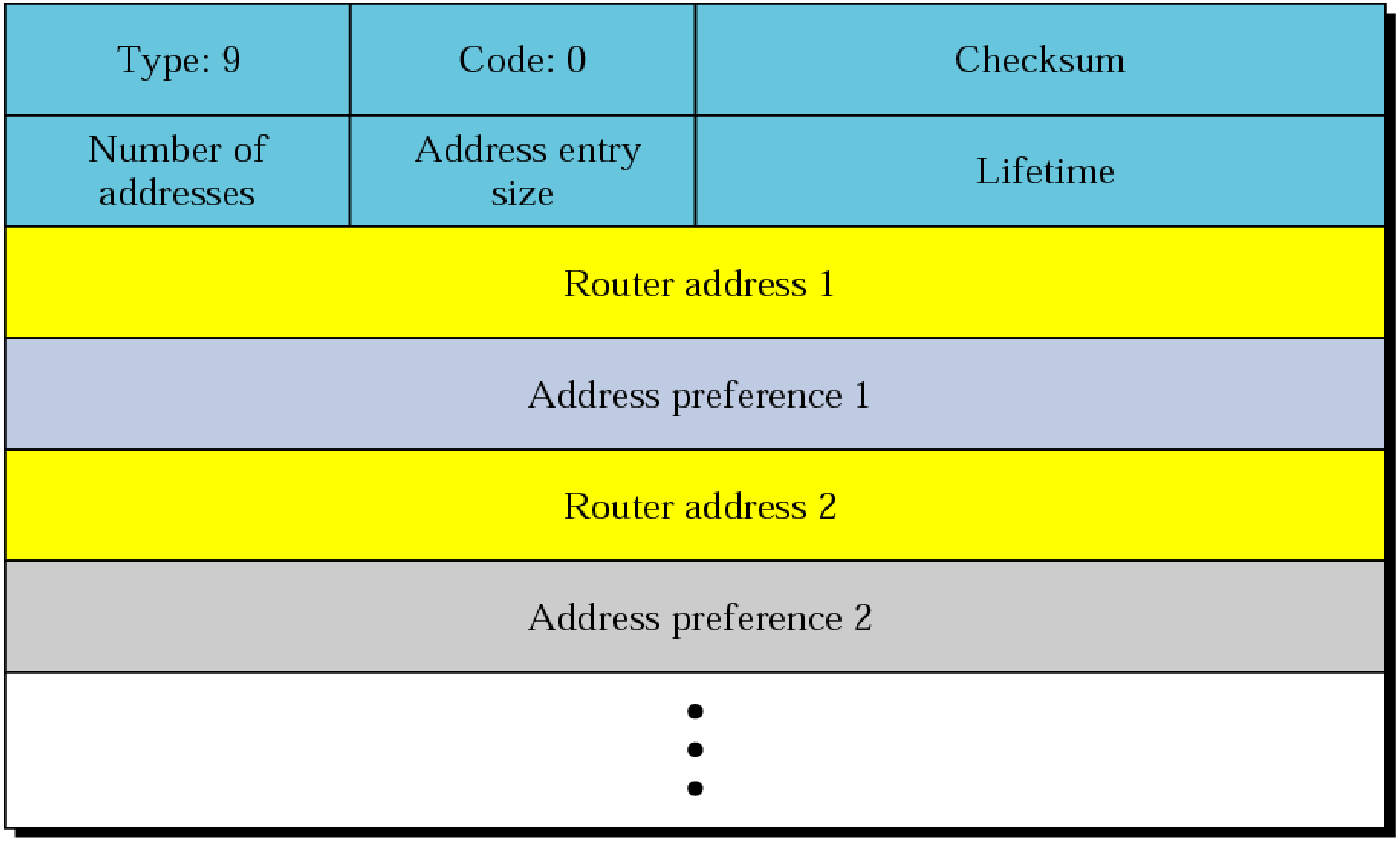
Router announces presence of all routers on the network of which it is aware via router advertisement message.



Type: 10	Code: 0	Checksum
Identifier		Sequence number

Figure 9.18 ***Router-advertisement message format***

Address preference level zero is default router.



9.5 CHECKSUM

In ICMP the checksum is calculated over the entire message (header and data).

The topics discussed in this section include:

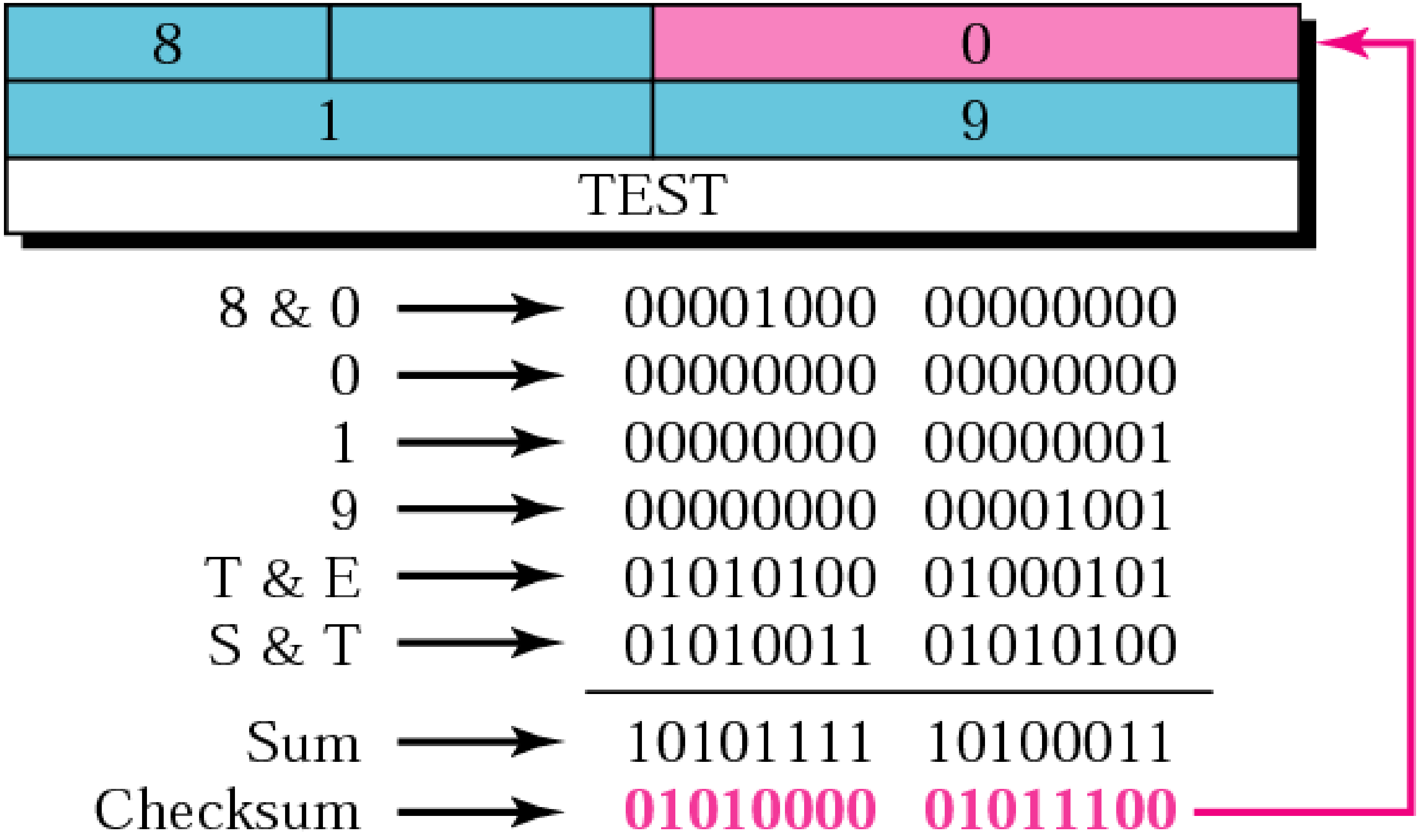
Checksum Calculation
Checksum Testing

Example 1

Figure 9.19 shows an example of checksum calculation for a simple echo-request message (see Figure 9.14). We randomly chose the identifier to be 1 and the sequence number to be 9. The message is divided into 16-bit (2-byte) words. The words are added together and the sum is complemented. Now the sender can put this value in the checksum field.

See Next Slide

Figure 9.19 *Example of checksum calculation*



8 & 0 → 00001000 00000000
0 → 00000000 00000000
1 → 00000000 00000001
9 → 00000000 00001001
T & E → 01010100 01000101
S & T → 01010011 01010100

Sum → 10101111 10100011
Checksum → **01010000 01011100**

9.6 DEBUGGING TOOLS

We introduce two tools that use ICMP for debugging: ping and traceroute.

The topics discussed in this section include:

Ping
Traceroute

Example 2

We use the ping program to test the server fhda.edu. The result is shown below:

ICMP data 56 bytes + ICMP Header 8 bytes + IP Header 20 bytes = 84 Bytes.

```
$ ping fhda.edu
PING fhda.edu (153.18.8.1) 56 (84) bytes of data.
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=0 ttl=62 time=1.91 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=1 ttl=62 time=2.04 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=2 ttl=62 time=1.90 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=3 ttl=62 time=1.97 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=4 ttl=62 time=1.93 ms
```

Example 2 (Continued)

```
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=5 ttl=62 time=2.00 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=6 ttl=62 time=1.94 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=7 ttl=62 time=1.94 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=8 ttl=62 time=1.97 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=9 ttl=62 time=1.89 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=10 ttl=62 time=1.98 ms

--- fhda.edu ping statistics ---
11 packets transmitted, 11 received, 0% packet loss, time 10103ms

rtt min/avg/max = 1.899/1.955/2.041 ms
```

Example 3

For the this example, we want to know if the adelphia.net mail server is alive and running. The result is shown below:

```
$ ping mail.adelphia.net
PING mail.adelphia.net (68.168.78.100) 56(84) bytes of data.
64 bytes from mail.adelphia.net (68.168.78.100): icmp_seq=0 ttl=48 time=85.4 ms
64 bytes from mail.adelphia.net (68.168.78.100): icmp_seq=1 ttl=48 time=84.6 ms
64 bytes from mail.adelphia.net (68.168.78.100): icmp_seq=2 ttl=48 time=84.9 ms
64 bytes from mail.adelphia.net (68.168.78.100): icmp_seq=3 ttl=48 time=84.3 ms
64 bytes from mail.adelphia.net (68.168.78.100): icmp_seq=4 ttl=48 time=84.5 ms
```

See Next Slide

Example 3 (Continued)

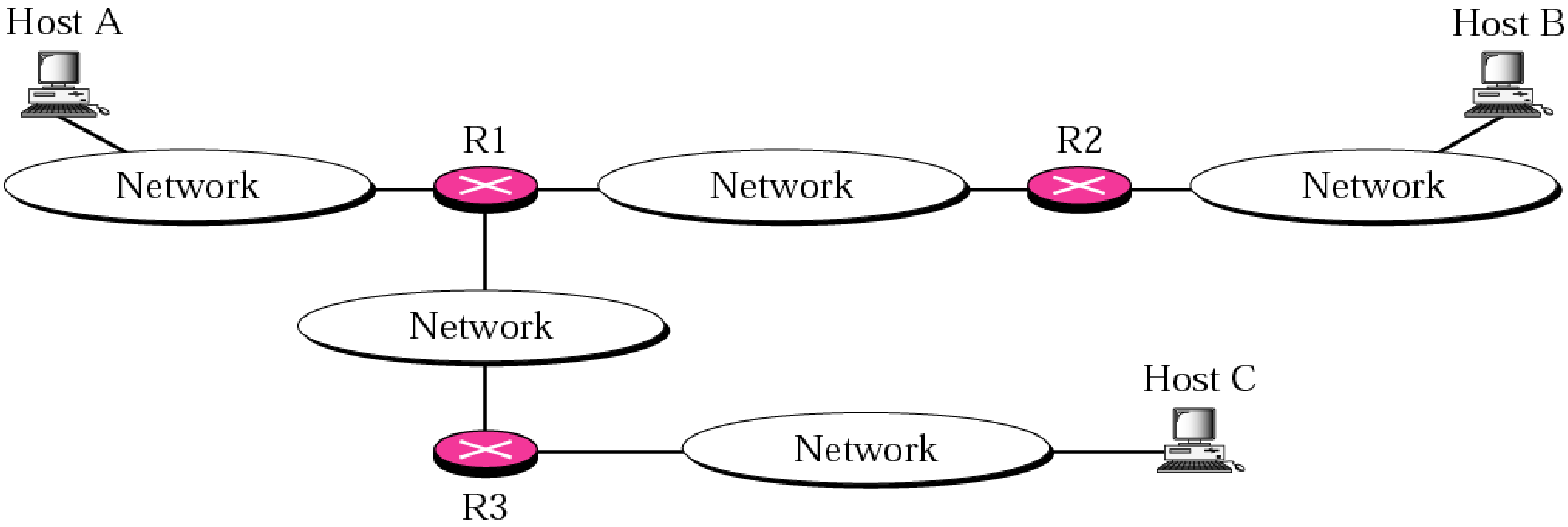
```
64 bytes from mail.adelphia.net (68.168.78.100): icmp_seq=5 ttl=48 time=84.7 ms
64 bytes from mail.adelphia.net (68.168.78.100): icmp_seq=6 ttl=48 time=84.6 ms
64 bytes from mail.adelphia.net (68.168.78.100): icmp_seq=7 ttl=48 time=84.7 ms
64 bytes from mail.adelphia.net (68.168.78.100): icmp_seq=8 ttl=48 time=84.4 ms
64 bytes from mail.adelphia.net (68.168.78.100): icmp_seq=9 ttl=48 time=84.2 ms
64 bytes from mail.adelphia.net (68.168.78.100): icmp_seq=10 ttl=48 time=84.9 ms
64 bytes from mail.adelphia.net (68.168.78.100): icmp_seq=11 ttl=48 time=84.6 ms
64 bytes from mail.adelphia.net (68.168.78.100): icmp_seq=12 ttl=48 time=84.5 ms

--- mail.adelphia.net ping statistics ---

14 packets transmitted, 13 received, 7% packet loss, time 13129ms
rtt min/avg/max/mdev = 84.207/84.694/85.469
```

Figure 9.20 *The traceroute program operation*

Traceroute Command:
1. Uses two ICMP messages, time exceeded and destination unreachable.
2. Uses UDP datagram with TTL 1, router R1 responds with Time Exceeded message.
3. Same router for 3 times for better RTT calculations.



Example 4

We use the traceroute program to find the route from the computer voyager.deanza.edu to the server fhda.edu. The following shows the result:

```
$ traceroute fhda.edu
traceroute to fhda.edu (153.18.8.1), 30 hops max, 38 byte packets
1 Dcore.fhda.edu (153.18.31.254) 0.995 ms 0.899 ms 0.878 ms
2 Dbackup.fhda.edu (153.18.251.4) 1.039 ms 1.064 ms 1.083 ms
3 tiptoe.fhda.edu (153.18.8.1) 1.797 ms 1.642 ms 1.757 ms
```

See Next Slide

Example 4 (Continued)

The un-numbered line after the command shows that the destination is 153.18.8.1. The TTL value is 30 hops. The packet contains 38 bytes: 20 bytes of IP header, 8 bytes of UDP header, and 10 bytes of application data. The application data is used by traceroute to keep track of the packets.

The first line shows the first router visited. The router is named Dcore.fhda.edu with IP address 153.18.31.254. The first round trip time was 0.995 milliseconds, the second was 0.899 milliseconds, and the third was 0.878 milliseconds.

The second line shows the second router visited. The router is named Dbackup.fhda.edu with IP address 153.18.251.4. The three round trip times are also shown.

The third line shows the destination host. We know that this is the destination host because there are no more lines. The destination host is the server fhda.edu, but it is named tiptoe. fhda.edu with the IP address 153.18.8.1. The three round trip times are also shown.

Example 5

In this example, we trace a longer route, the route to xerox.com

```
$ traceroute xerox.com
traceroute to xerox.com (13.1.64.93), 30 hops max, 38 byte packets
1 Dcore.fhda.edu (153.18.31.254) 0.622 ms 0.891 ms 0.875 ms
2 Ddmz.fhda.edu (153.18.251.40) 2.132 ms 2.266 ms 2.094 ms
...

18 alpha.Xerox.COM (13.1.64.93) 11.172 ms 11.048 ms 10.922 ms
```

Here there are 17 hops between source and destination. Note that some round trip times look unusual. It could be that a router is too busy to process the packet immediately.

Example 6

An interesting point is that a host can send a traceroute packet to itself. This can be done by specifying the host as the destination. The packet goes to the loopback address as we expect.

```
$ traceroute voyager.deanza.edu
traceroute to voyager.deanza.edu (127.0.0.1), 30 hops max, 38 byte packets
1 voyager (127.0.0.1) 0.178 ms 0.086 ms 0.055 ms
```

Example 7

Finally, we use the traceroute program to find the route between fhda.edu and mhhe.com (McGraw-Hill server). We notice that we cannot find the whole route. When traceroute does not receive a response within 5 seconds, it prints an asterisk to signify a problem, and then tries the next hop..

```
$ traceroute mhhe.com
traceroute to mhhe.com (198.45.24.104), 30 hops max, 38 byte packets
1 Dcore.fhda.edu (153.18.31.254) 1.025 ms 0.892 ms 0.880 ms
2 Ddmz.fhda.edu (153.18.251.40) 2.141 ms 2.159 ms 2.103 ms
3 Cinic.fhda.edu (153.18.253.126) 2.159 ms 2.050 ms 1.992 ms
   ...
16 * * *
17 * * *
...............
```

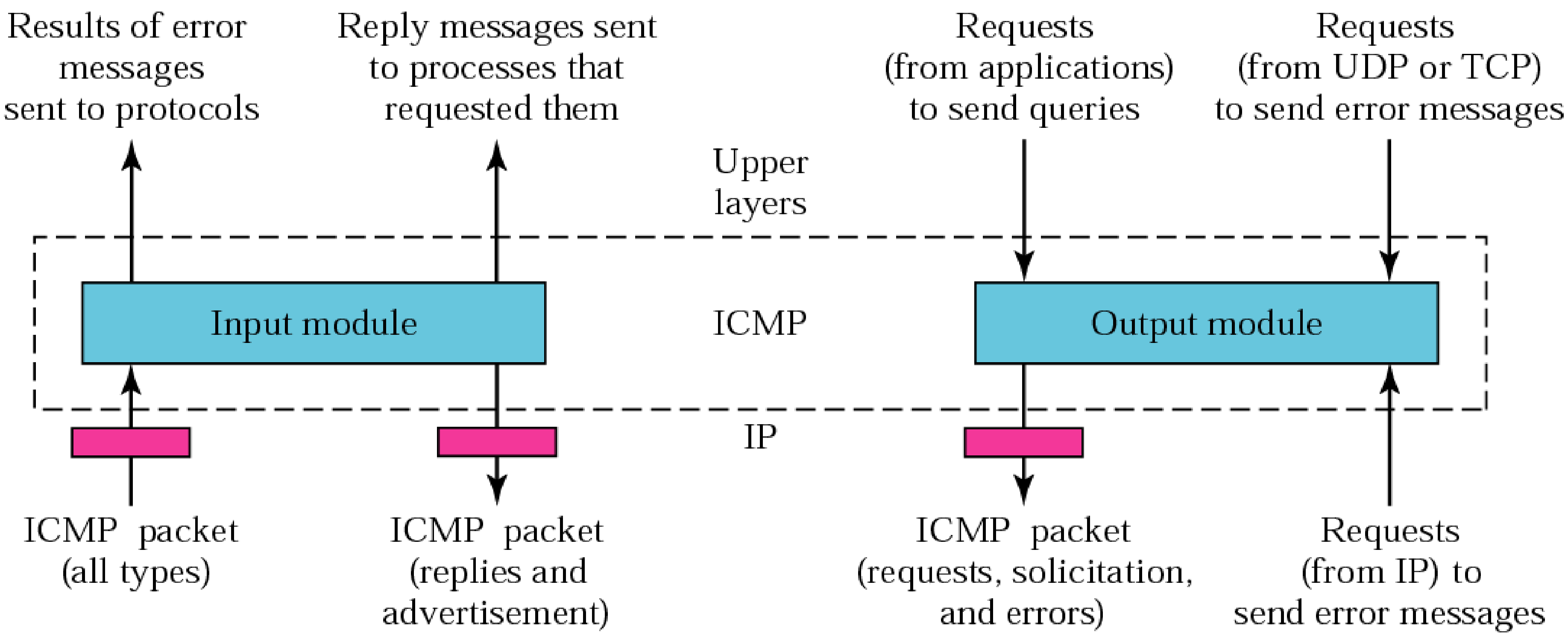
9.7 ICMP PACKAGE

To give an idea of how ICMP can handle the sending and receiving of ICMP messages, we present our version of an ICMP package made of two modules: an input module and an output module.

The topics discussed in this section include:

Input Module
Output Module

Figure 9.21 *ICMP package*


Results of error messages sent to protocols
Reply messages sent to processes that requested them
Requests (from applications) to send queries
Requests (from UDP or TCP) to send error messages
Upper layers
Input module
ICMP
Output module
IP
ICMP packet (all types)
ICMP packet (replies and advertisement)
ICMP packet (requests, solicitation, and errors)
Requests (from IP) to send error messages