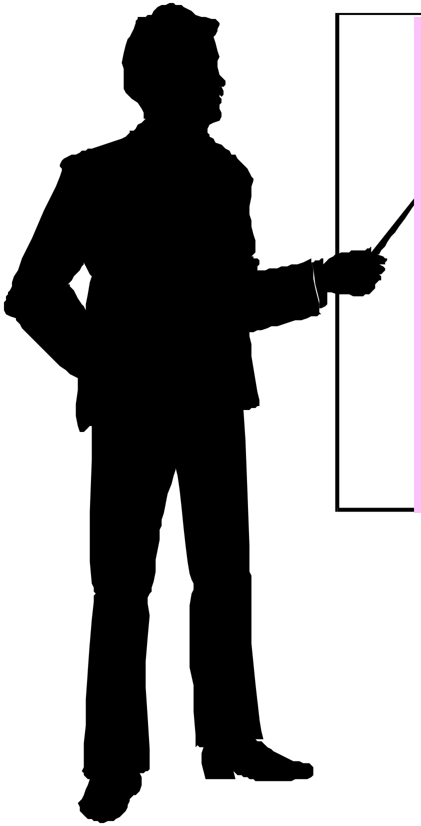


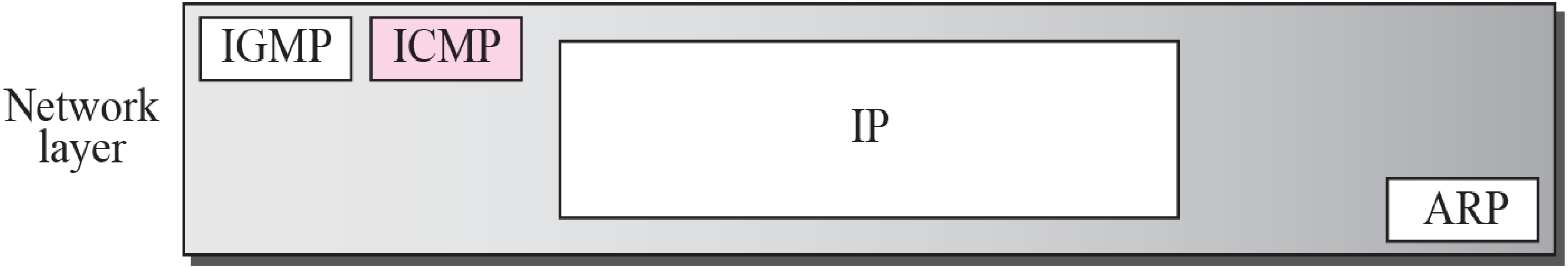
Chapter 9
ICMP

9.1 Introduction

- IP protocol has no error-reporting or error-correcting mechanism
  - When errors occur, no built-in mechanism to notify the original host
- IP protocol also lacks a mechanism for host and management queries
  - A host sometimes needs to determine if a router or another host is alive
  - Network manager needs information from another host and router

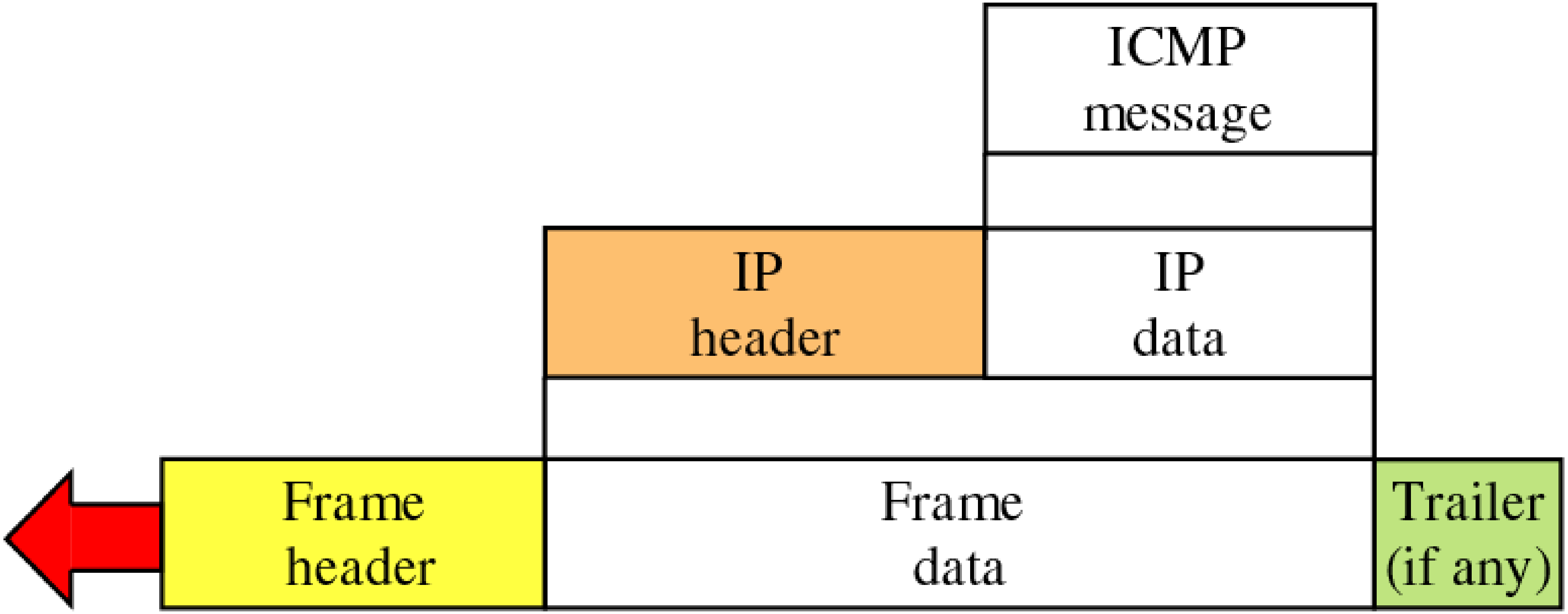
Introduction (cont'd)

- Position of ICMP in the network layer



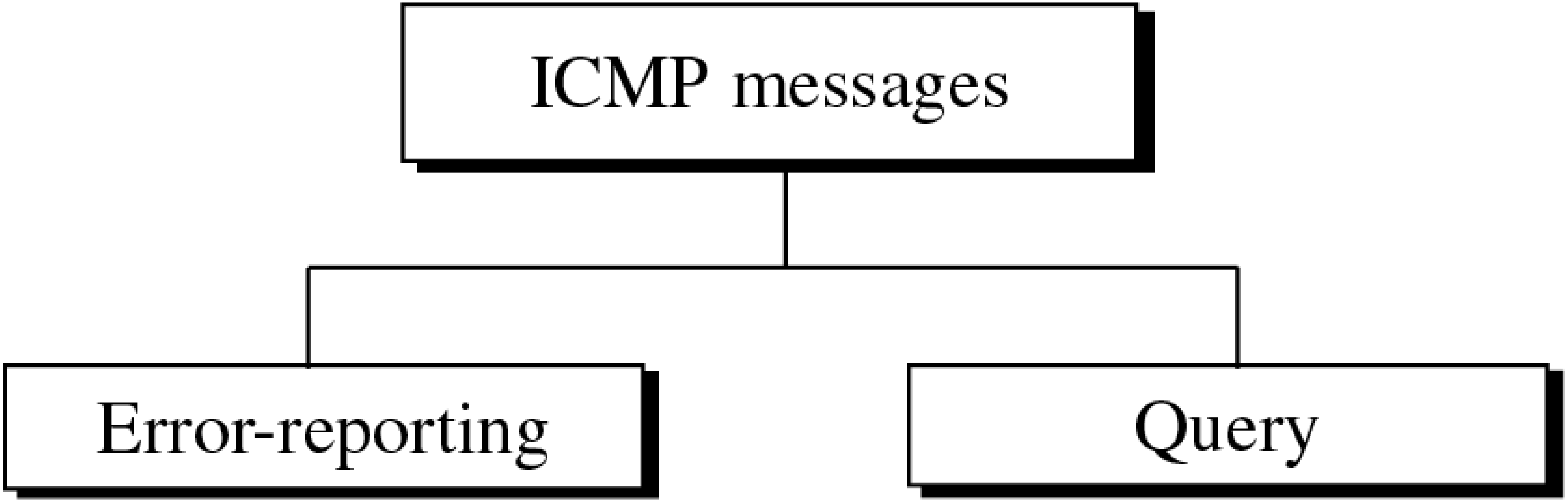
Introduction (cont'd)

- ICMP encapsulation
  - The value of the protocol field in the IP datagram : 1



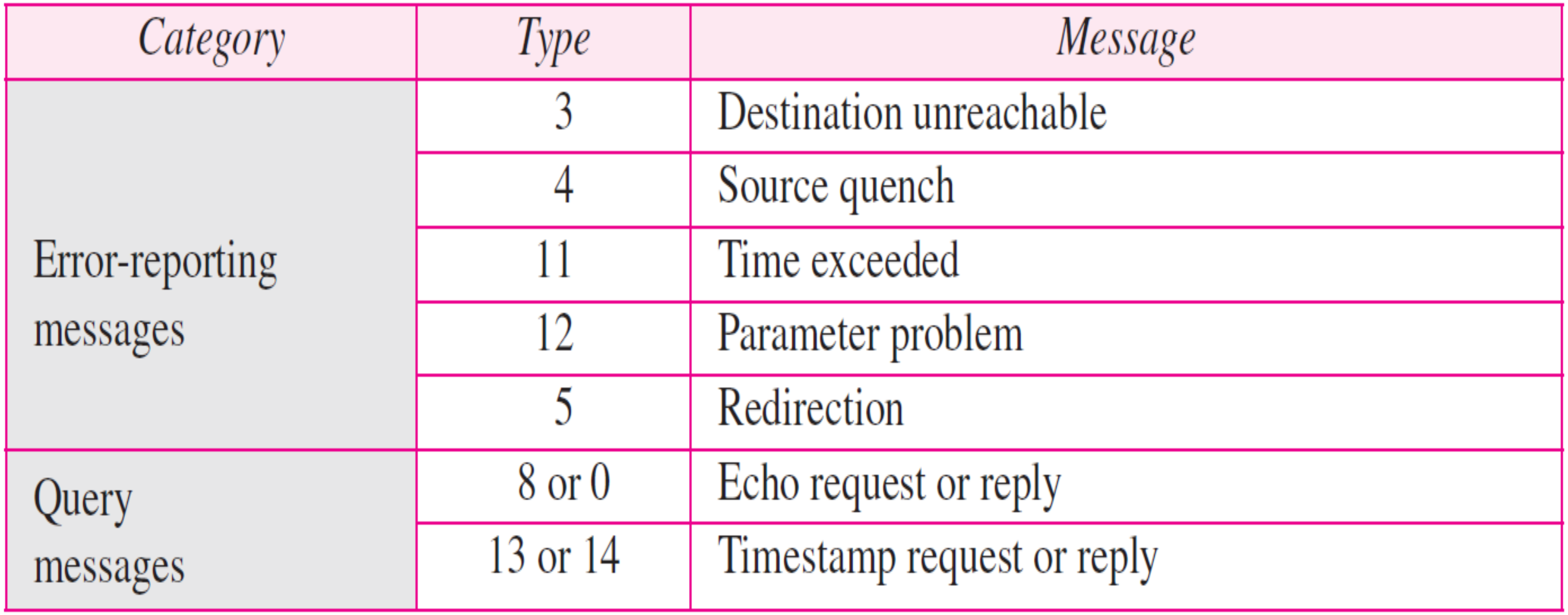
9.2 Types of Message

- Category of ICMP messages



Types of Message (cont’d)

- **ICMP messages**

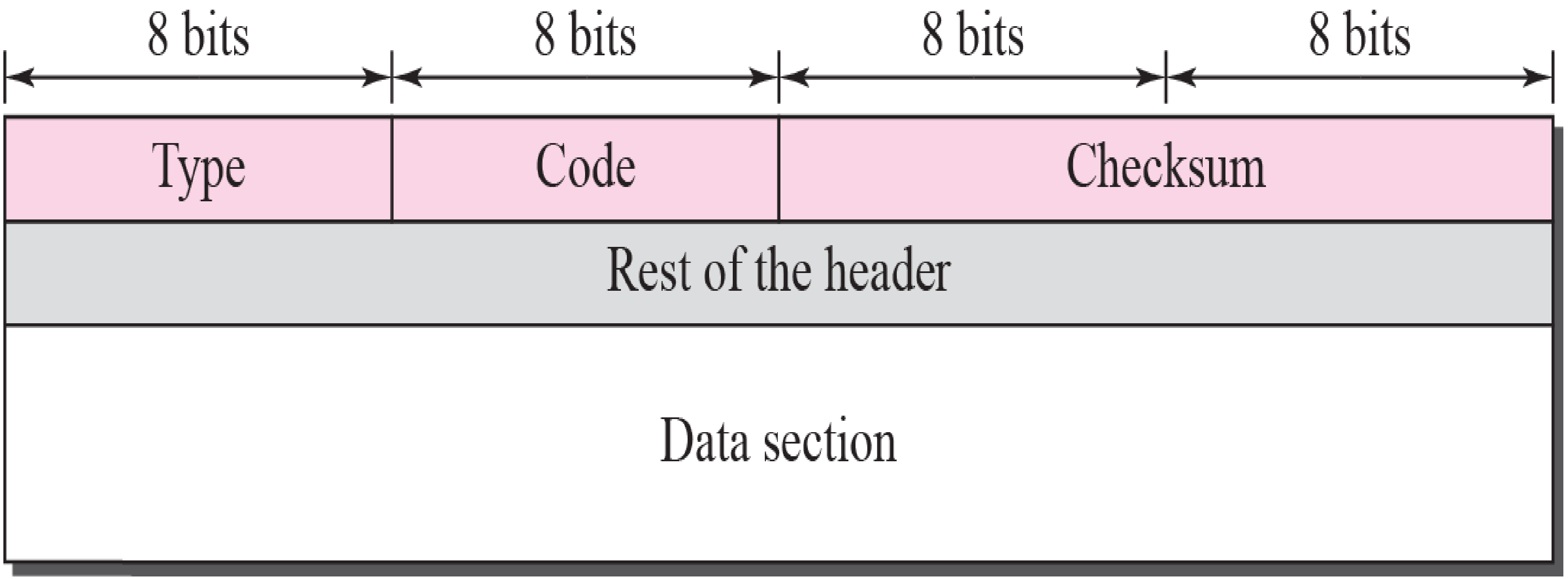


Category	*Type*	*Message*
Error-reporting messages	3	Destination unreachable
	4	Source quench
	11	Time exceeded
	12	Parameter problem
	5	Redirection
Query messages	8 or 0	Echo request or reply
	13 or 14	Timestamp request or reply

Message Format

- Having 8 byte header and variable-size data section
  - ICMP type : defining the type of the message
  - Code field : specifying the reason for the particular message type
  - Checksum field (for header and message)
  - Data section
    - In error message, carrying information for finding the original packet which caused the error
    - In query message, carrying extra information based on the type of the query

Message Format (cont'd)



8 bits

8 bits

8 bits

8 bits

Type

Code

Checksum

Rest of the header

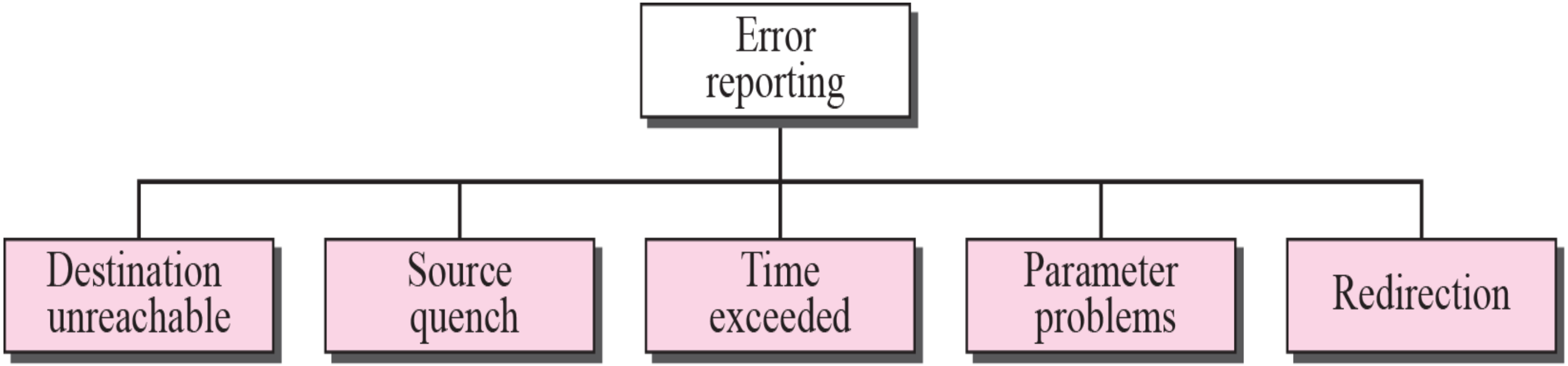
Data section

Error Reporting Message

- ❑ Error checking and control
- ❑ Not correcting errors : it is left to the higher level protocols
- ❑ Always reporting error messages to the original source

Error Reporting Message

- Error-reporting messages



Error Reporting Message (cont'd)

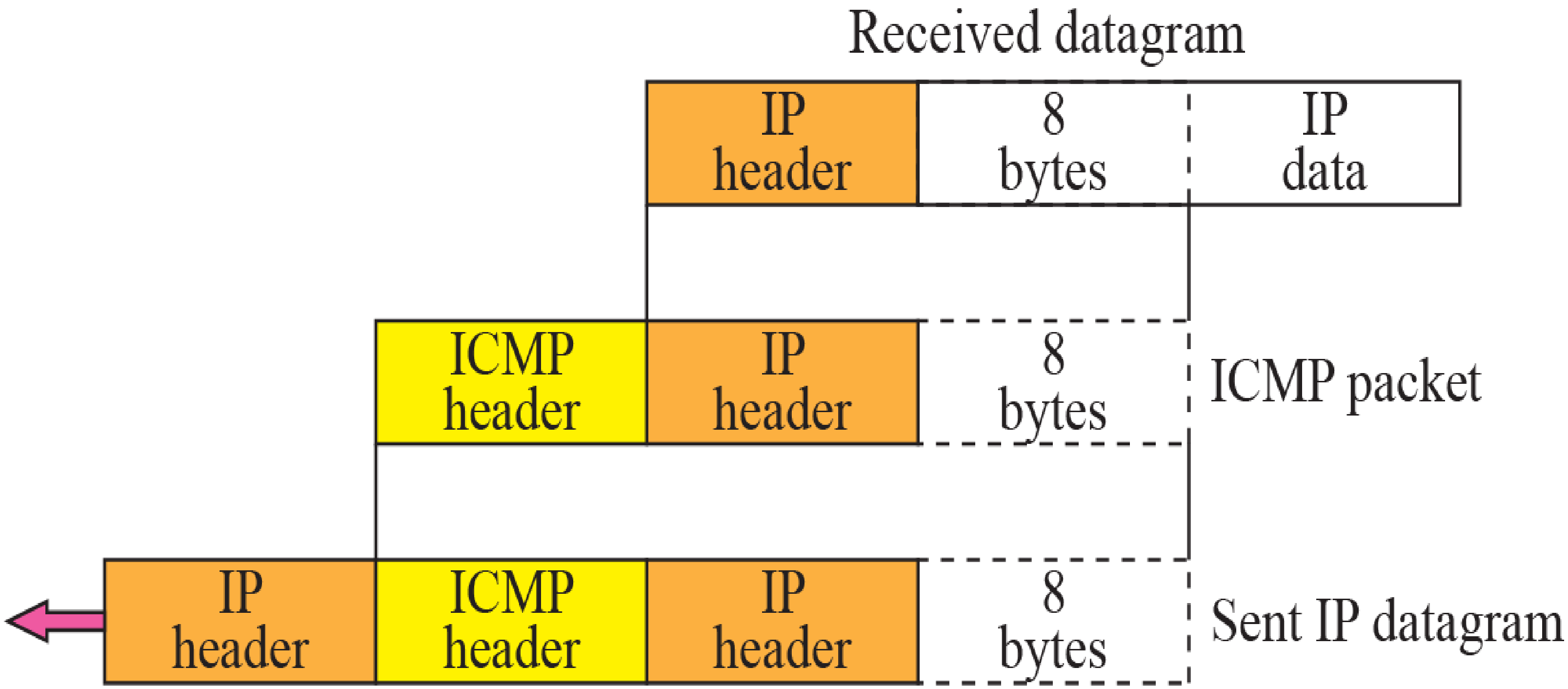
- Important points about ICMP error messages
  - No ICMP error message will be generated in response to a datagram carrying an ICMP error message
  - No ICMP error message will be generated for a fragmented datagram that is not the first fragment
  - No ICMP error message will be generated for a datagram 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

Error Reporting Message (cont'd)

- All error messages
  - containing a data section that includes the IP header of the original datagram + the first 8 bytes of data in that IP datagram
    - 8 bytes of data : port # (UDP and TCP) and sequence # (TCP)
      - Used for informing to the protocols (TCP or UDP) about the error situation

Error Reporting Message (cont'd)

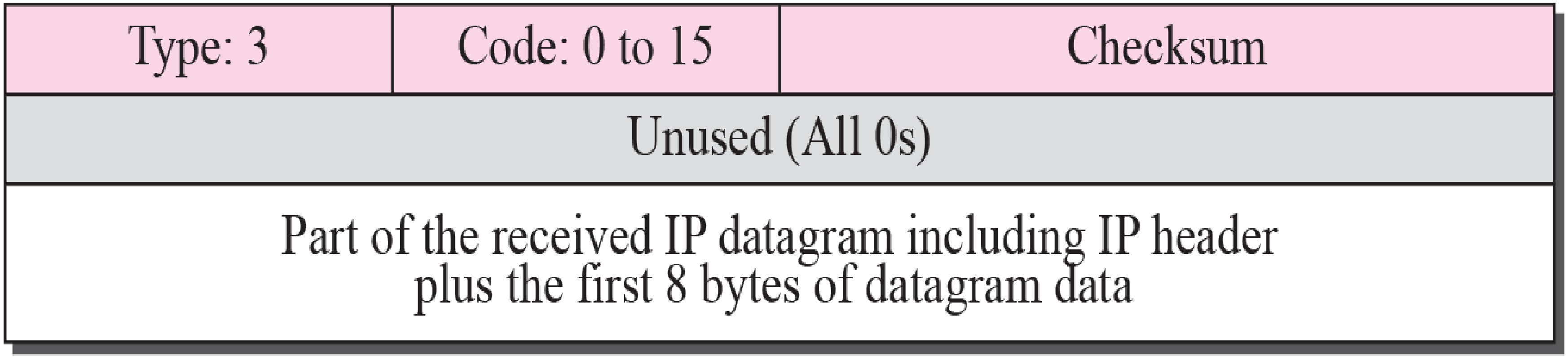
- Contents of data field for the error messages



Error Reporting Message (cont'd)

❑ Destination Unreachable

- When a router cannot route a datagram or a host cannot deliver a datagram, the datagram is discarded.
- Then, the router or the host sends a destination unreachable message back to the source that initiated the datagram.
- Destination unreachable format



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		

Error Reporting Message (cont'd)

- **Code 0 :** network is unreachable, due to hardware failure, can only be generated by a router
- **Code 1 :** host is unreachable, due to hardware failure, can only be generated by a router
- **Code 2 :** protocol such as UDP, TCP or OSPF is not running at the moment.
  - generated only by the destination
- **Code 3 :** the application program (process) that the datagram is destined for is not running at the moment
- **Code 4 :** Fragmentation is required, but the DF (do not fragment) field has been set
- **Code 5 :** Source routing cannot be accomplished
- **Code 6 :** The destination network is unknown.
  - A router has no information about the destination network

Error Reporting Message (cont'd)

- **Code 7 :** The destination host is unknown.
  - the router is unaware of the existence of the destination
- **Code 8 :** The source host is isolated
- **Code 9 :** Communication with the destination network is administratively prohibited
- **Code 10 :** Communication with the destination host is administratively prohibited
- **Code 11 :** the network is unreachable for the specified type of service
- **Code 12 :** The host is unreachable for the specified type of service

Error Reporting Message (cont'd)

- ❑ **Code 13 :** The host is unreachable because the administration has put a filter on it

- ❑ **Code 14 :** The host is unreachable because the host precedence is violated. The requested precedence is not permitted for the destination

- ❑ **Code 15 :** The host is unreachable because its precedence was cut off. This message is generated when the network operators have imposed a minimum level of precedence for the operation of the network

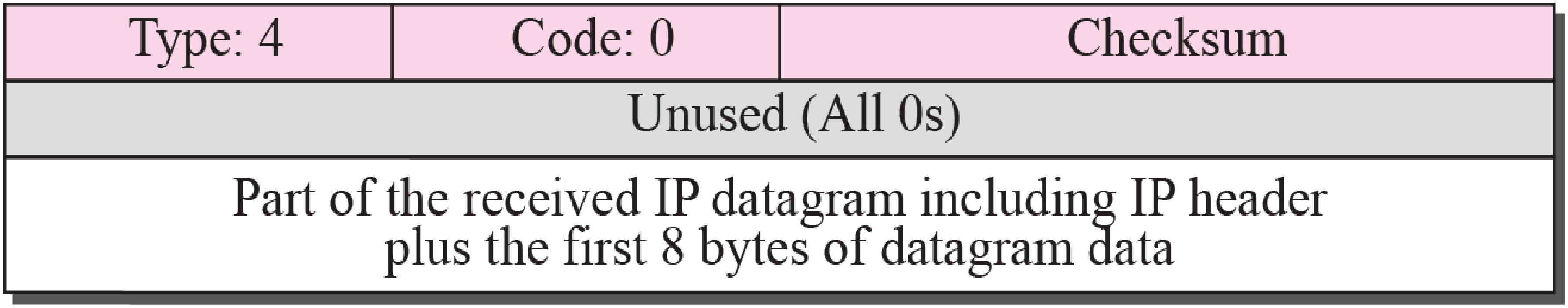
Error Reporting Message (cont'd)

- Destination-unreachable messages with codes 2 or 3 can be created only by the destination host. Other destination-unreachable message can be created only by routers.
- A router can not detect all problems that prevent the delivery of a packet.
  - The case that a datagram is traveling through an Ethernet network.
  - Ethernet does not provide any acknowledgement mechanism.

Error Reporting Message (cont'd)

❑ Source Quench

- is designed to add a kind of flow control to the IP
  - □ IP does not have a flow-control mechanism embedded in the protocol
- when a router or host discards a datagram due to congestion, it sends a source-quench message to the sender of the datagram
  - □ making slow down the sending process



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		

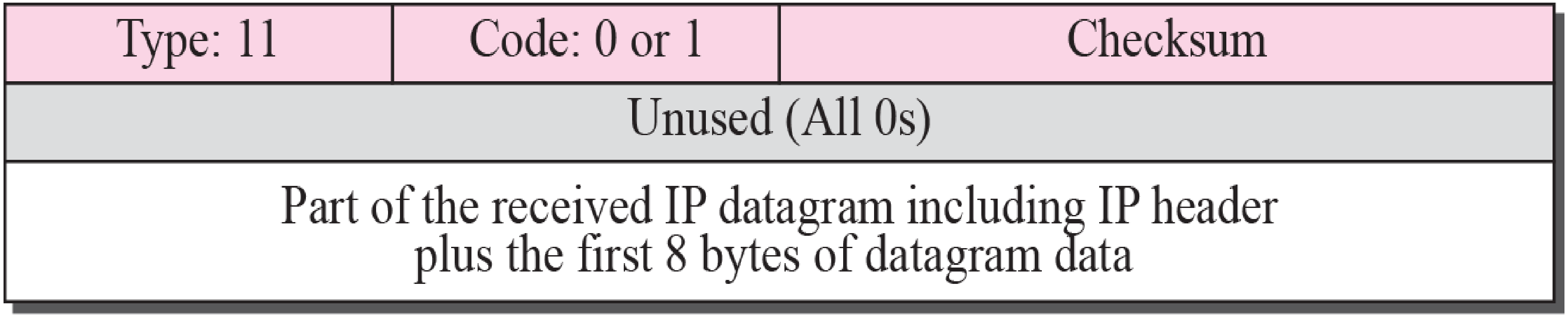
Error Reporting Message (cont'd)

- Time exceeded
  - Whenever a router receives a datagram whose time-to-live field has the value of zero, it discards the datagram and sends a time-exceeded message to the original source
  - 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

Error Reporting Message (cont'd)

- 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

- Time-exceeded message format

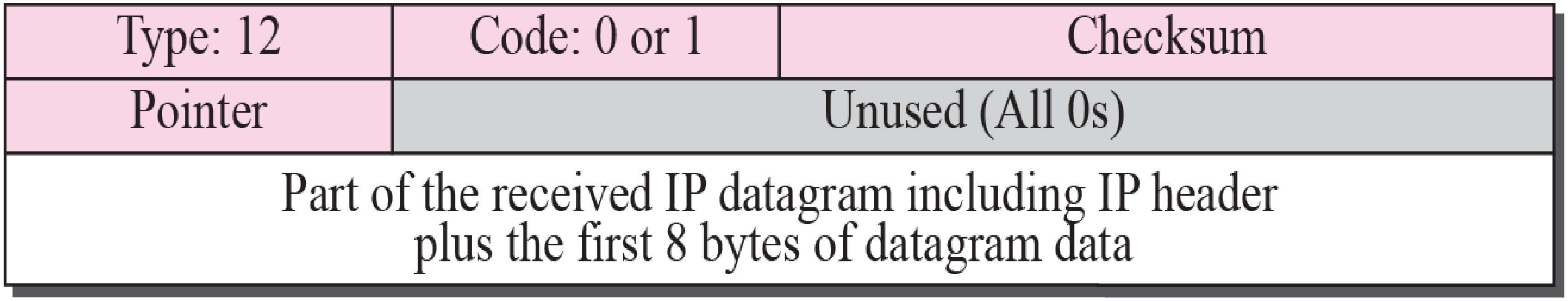


Type: 11	Code: 0 or 1	Checksum
Unused (All 0s)		
Part of the received IP datagram including IP header plus the first 8 bytes of datagram data		

Error Reporting Message (cont'd)

❑ Parameter-problem

- A parameter-problem message caused by ambiguity in the header part can be created by a router or the destination host
- Code 0 : error or ambiguity in one of the header fields
  - □ the value in the pointer field points to the byte with the problem
- Code 1 : the required part of an option is missing. In this case, pointer is not used

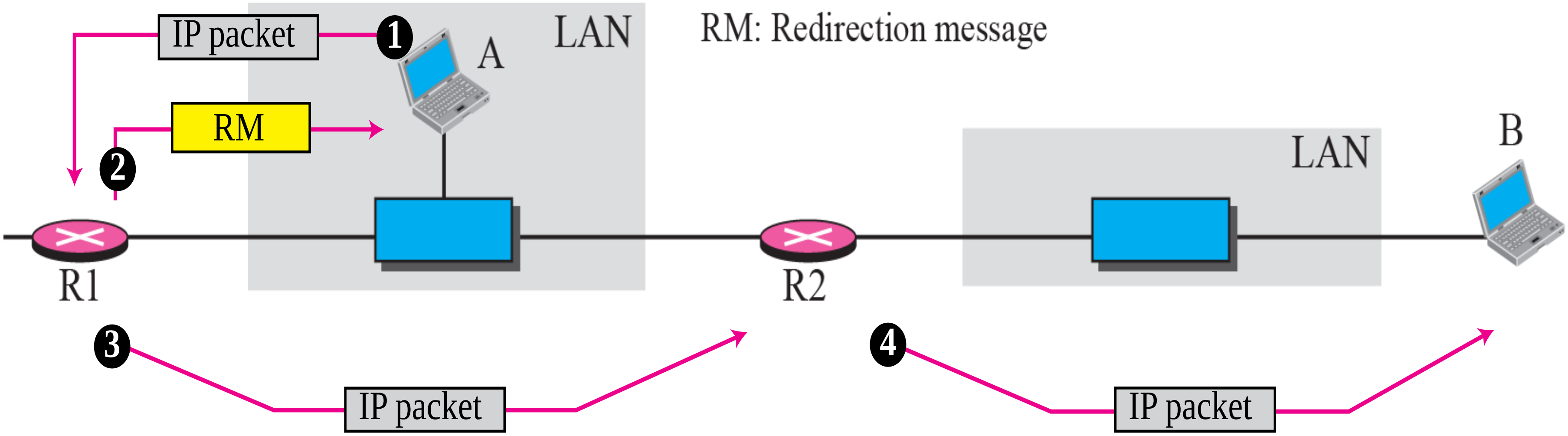


Type: 12	Code: 0 or 1	Checksum
Pointer	Unused (All 0s)	
Part of the received IP datagram including IP header plus the first 8 bytes of datagram data		

Error Reporting Message (cont'd)

❑ Redirection

- 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.



Error Reporting Message (cont'd)

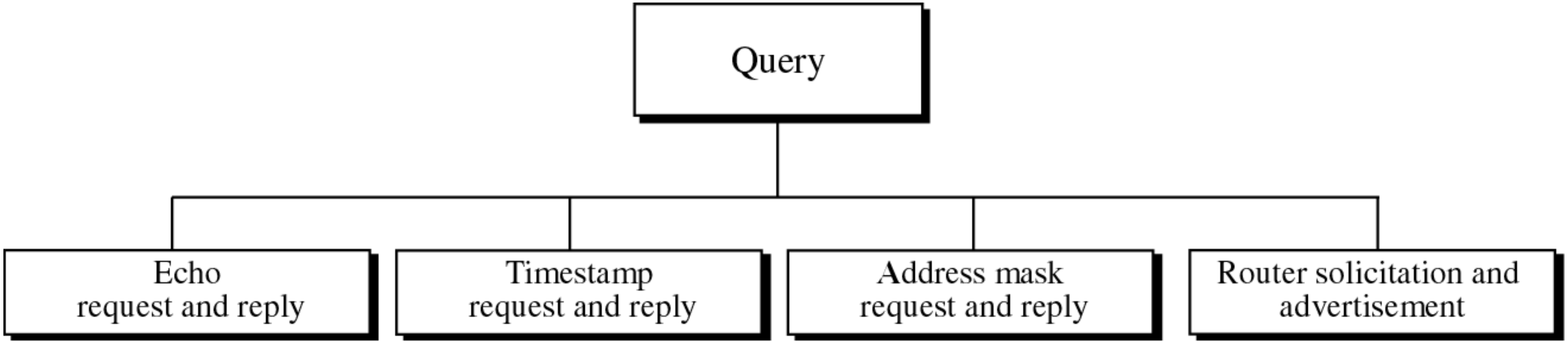
- Redirection message format

<table>
<tr><td>Type: 5</td><td>Code: 0 to 3</td><td>Checksum</td></tr>
<tr><td colspan="3">IP address of the target router</td></tr>
<tr><td colspan="3">Part of the received IP datagram including IP header
plus the first 8 bytes of datagram data</td></tr>
</table>

- **Code 0 : redirection for the network-specific route**
- **Code 1 : redirection for the host-specific route**
- **Code 2 : redirection for network-specific route based on specific type of service**
- **Code 3 : redirection for the host-specific route based on the specified type of service**

Query Message

- Diagnosing some network problems
- 4 different pairs of messages


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

Query Message (cont'd)

- Echo Request and Reply messages
  - designed for diagnostic purpose
  - the combination of echo-request and echo-reply messages determines whether 2 systems (hosts or routers) can communicate with each other
  - 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 message can be used by network managers to check the operation of the IP protocol

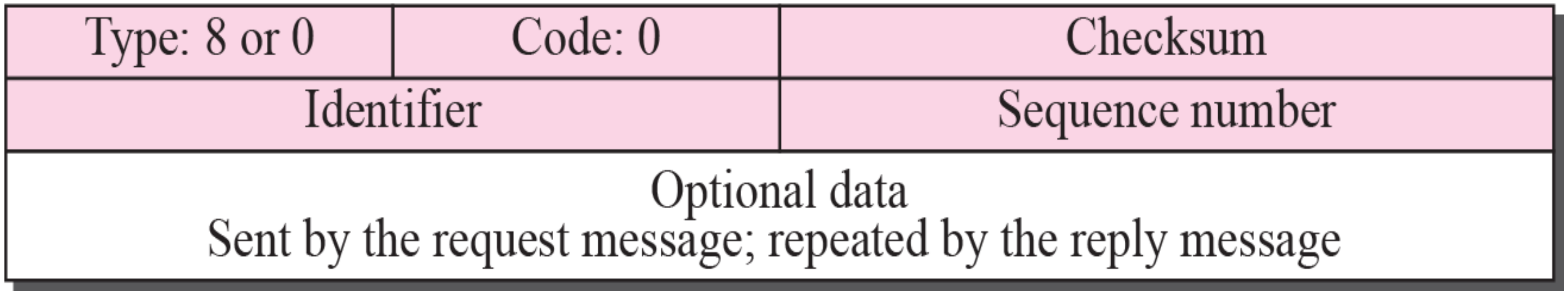
Query Message (cont'd)

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

- ❑ Identifier and sequence number fields are not formally defined by the protocol and can be used by the sender

- ❑ Echo-request and echo-reply message

  - Type 8 : Echo request   - Type 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		

Query Message (cont'd)

- The identifier field
  - defines a group of problems
  - ex) process ID that originated the request
- The sequence number field
  - keeps track of the particular echo request messages sent
- At the user level
  - Invoking the packet Internet groper (ping) command

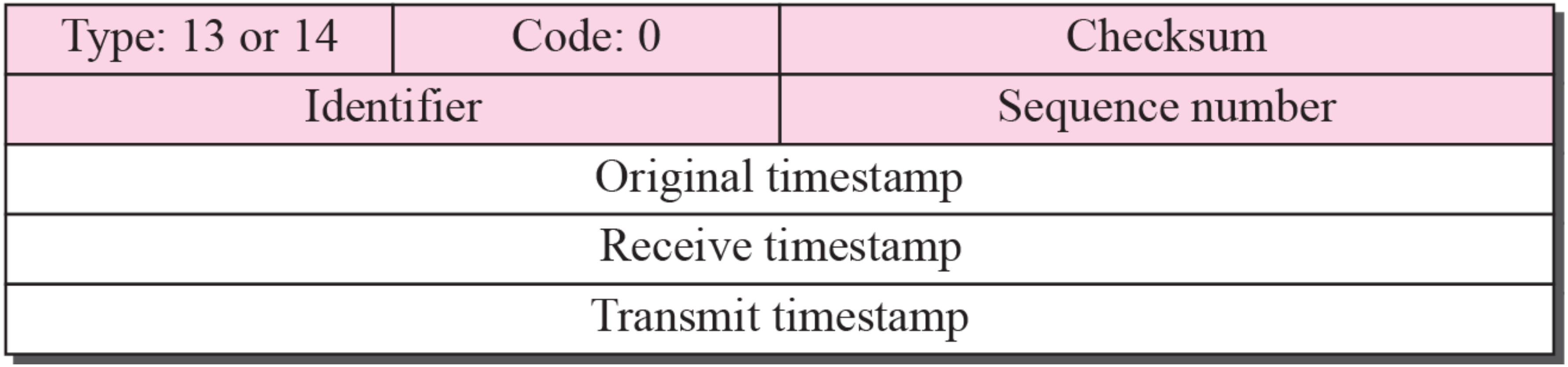
Query Message (cont'd)

- Timestamp Request and Reply
  - 2 machines (routers or hosts) can use the timestamp-request and timestamp-reply messages to determine the round-trip time needed for an IP datagram to travel between them
  - can used to synchronize the clocks in two machines
  - Three timestamp fields are each 32 bits long
    - holding a number representing time measured in milliseconds from midnight in Universal Time
      - Cannot exceed 86,400,000 = 24 x 60 x 60 x 1,000

Query (cont'd)

- Timestamp-request and reply message format

- Type 13 : Request - Type 14 : Reply



<table>
<tr><td>Type: 13 or 14</td><td>Code: 0</td><td>Checksum</td></tr>
<tr><td colspan="2">Identifier</td><td>Sequence number</td></tr>
<tr><td colspan="3">Original timestamp</td></tr>
<tr><td colspan="3">Receive timestamp</td></tr>
<tr><td colspan="3">Transmit timestamp</td></tr>
</table>

- original timestamp field : clock at departure time
- receive timestamp field : at the time the request was received
- transmit timestamp field : at the time the reply message departs

Query Message (cont'd)

- The formulas for computing the one-way or round-trip time required for a datagram to go from a source to a destination and then back again.
  - Sending time = value of receive timestamp – value of original time stamp
  - Receiving time = time the packet returned – value of transmit timestamp
  - Round-trip time = sending time + receiving time

Query Message (cont'd)

- Timestamp-request and timestamp reply message can be used to measure the round-trip time between a source and a destination machine even if their clocks are not synchronized
  - Example
    - □ Value of original timestamp : 46
    - □ Value of receive timestamp : 59
    - □ Value of transmit timestamp : 60
    - □ Time the packet arrived : 67

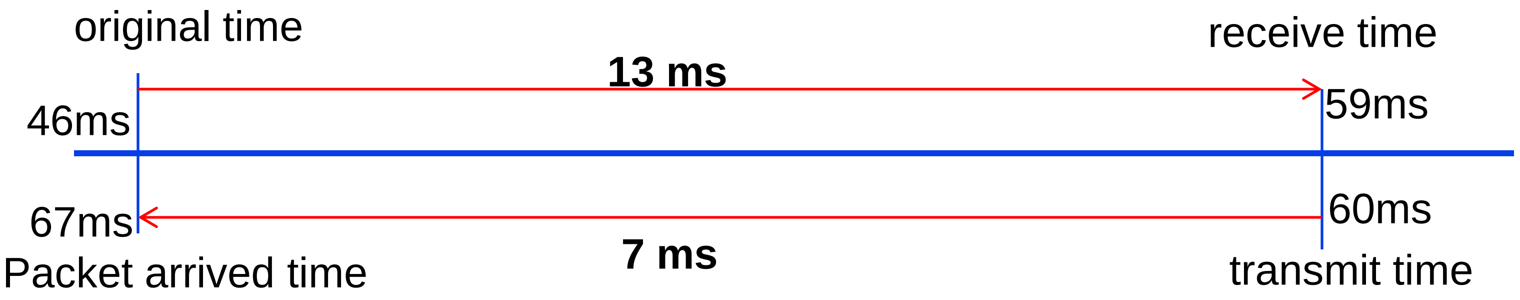
    Sending time = 13 ms

    Receiving time = 7 ms

    Round-trip time = 20 ms

Query Message (cont'd)

- Synchronizing clocks between two machines
  - Time difference = receive timestamp – (original timestamp field + oneway time duration)
  - In previous example,
    - Time difference = 59 – (46 + 10) = 3



Checksum

- Checksum
  - calculating over the entire message (header and data)
- Checksum calculation
  1. Checksum field is set to zero
  2. Sum of all the 16-bit words (header and data) is calculated
  3. Sum is complemented to get the checksum
  4. Checksum is stored in the checksum field

Checksum (cont'd)

- Checksum testing

  1. the sum of all words (header and data) is calculated
  2. the sum is completed
  3. if the result obtained in step 2 is 16 0s, the message is accepted; otherwise, it is rejected.

- Example,

8	0	0
1		9
TEST		

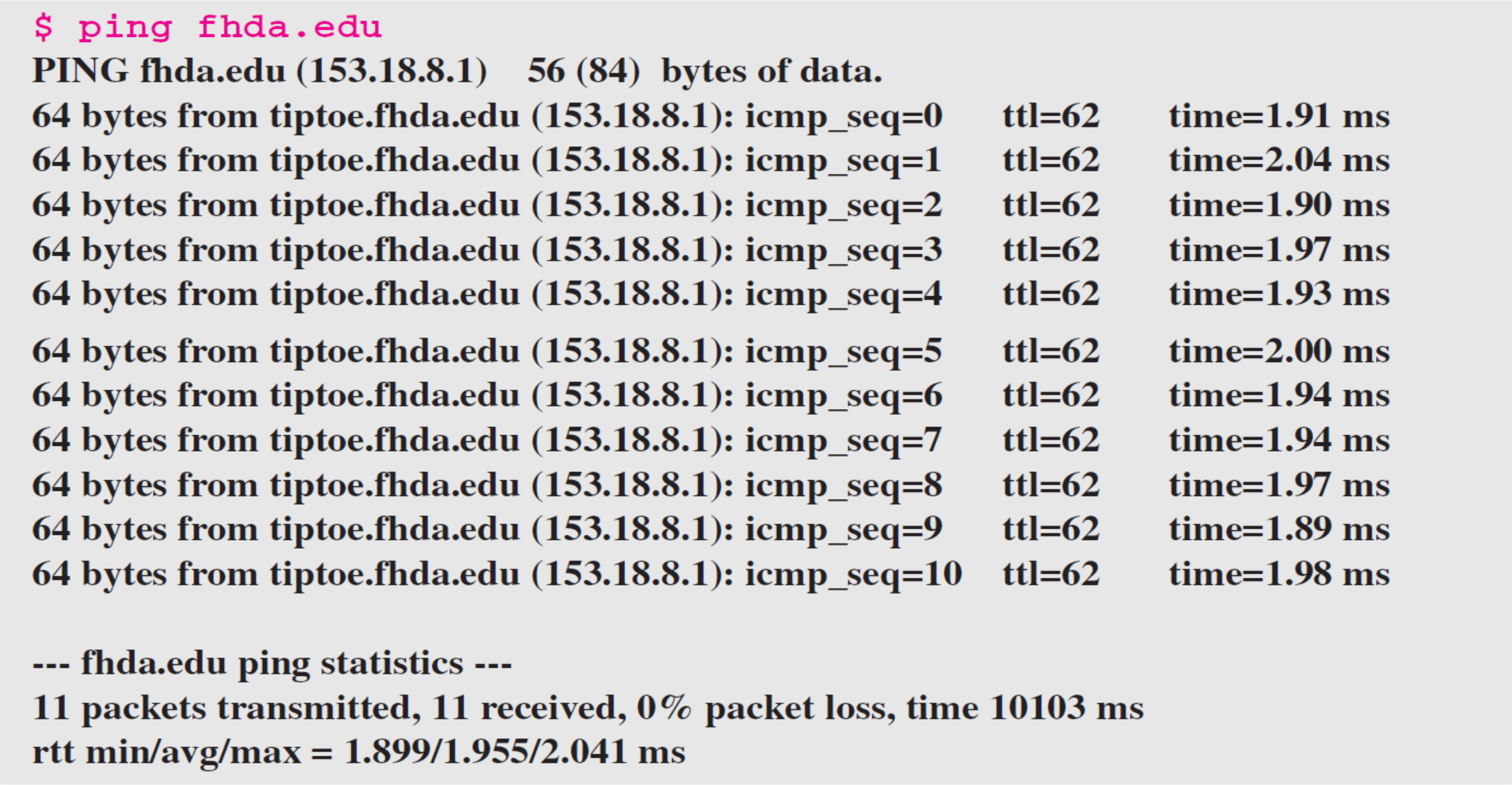
```
 8 and 0 ⟶   00001000  00000000
       0 ⟶   00000000  00000000
       1 ⟶   00000000  00000001
       9 ⟶   00000000  00001001
   T & E ⟶   01010100  01000101
   S & T ⟶   01010011  01010100
             ------------------
     Sum ⟶   10101111  10100011
Checksum ⟶   01010000  01011100
```

9.3 Debugging Tool

- Tools for debugging in Internet
  - Ping
    - Check the host or router is alive or not
  - Traceroute
    - Trace the route of the packet

Ping

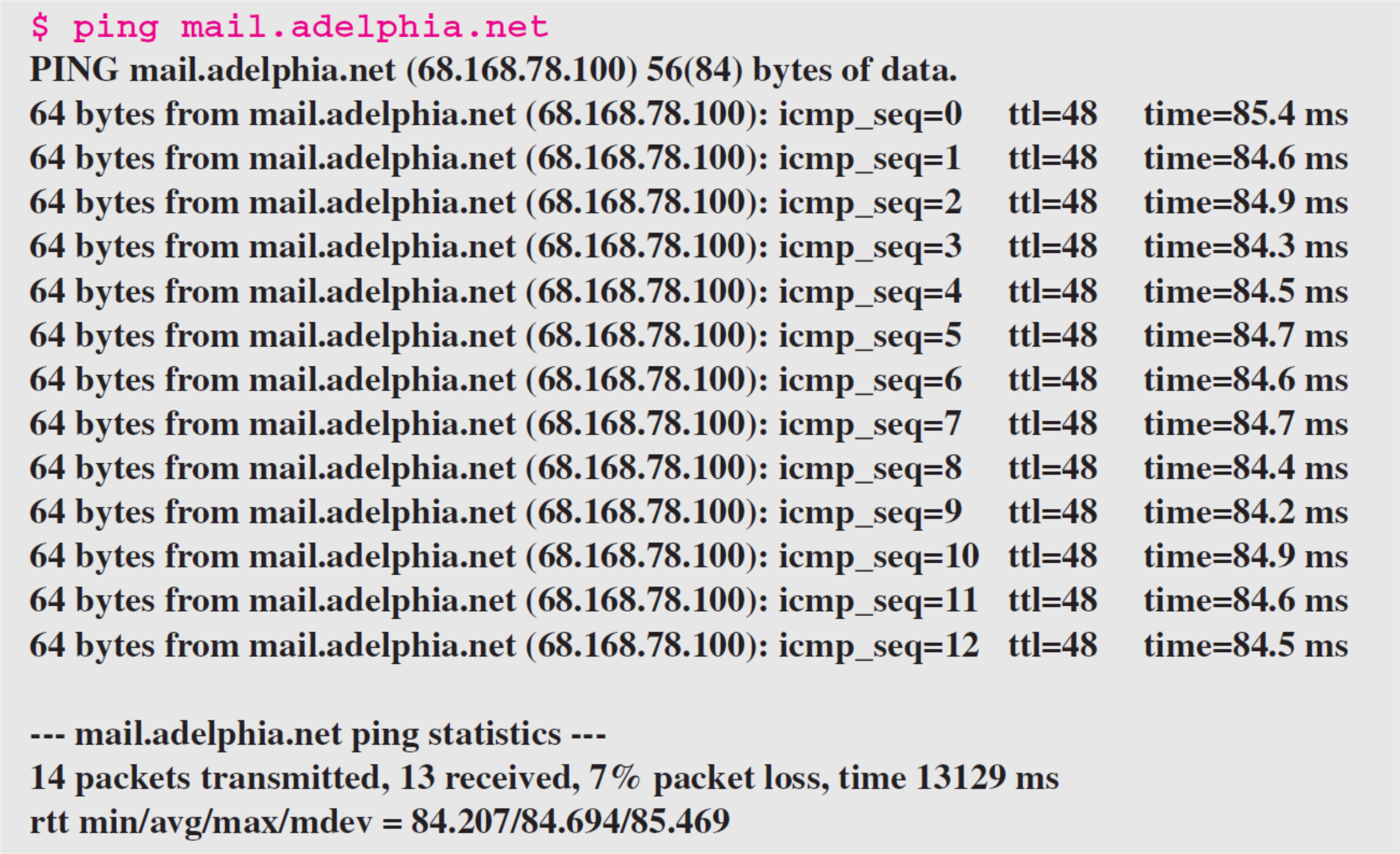
- Result of ping to test the server fhda.edu



```
$ 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

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 10103 ms
rtt min/avg/max = 1.899/1.955/2.041 ms
```

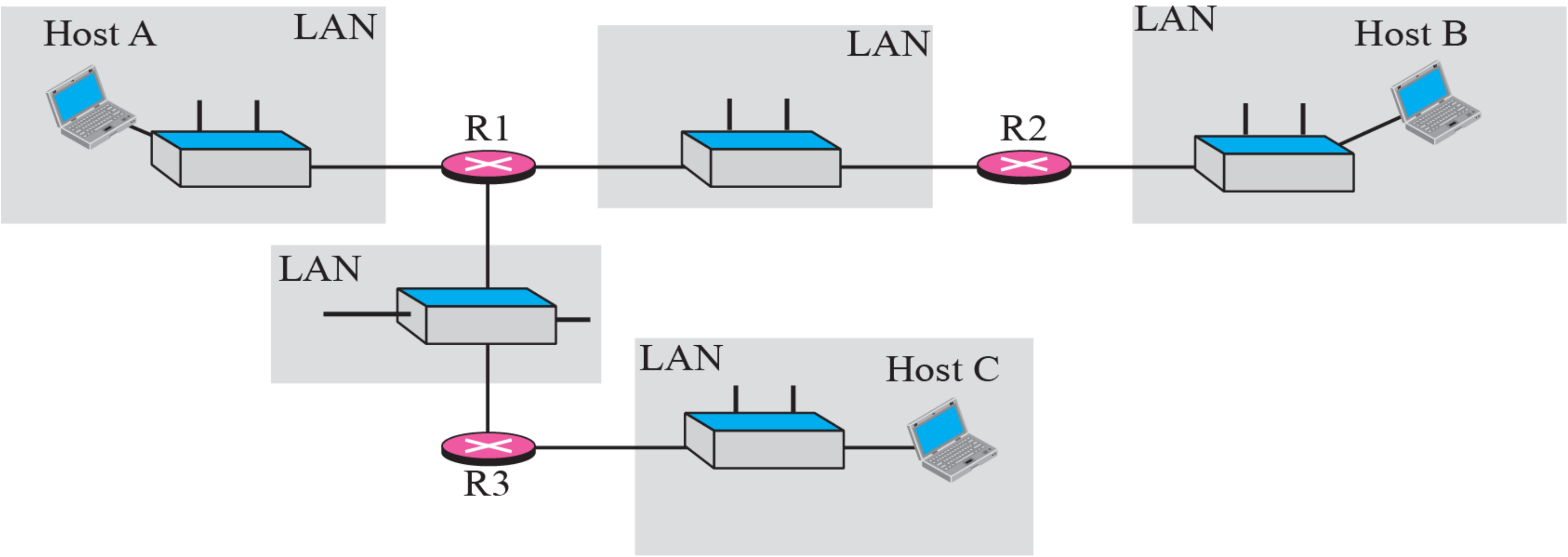
Ping

- Result of ping to test the server adelphia.net. Note that we sent 14 packets, but only 13 have been returned



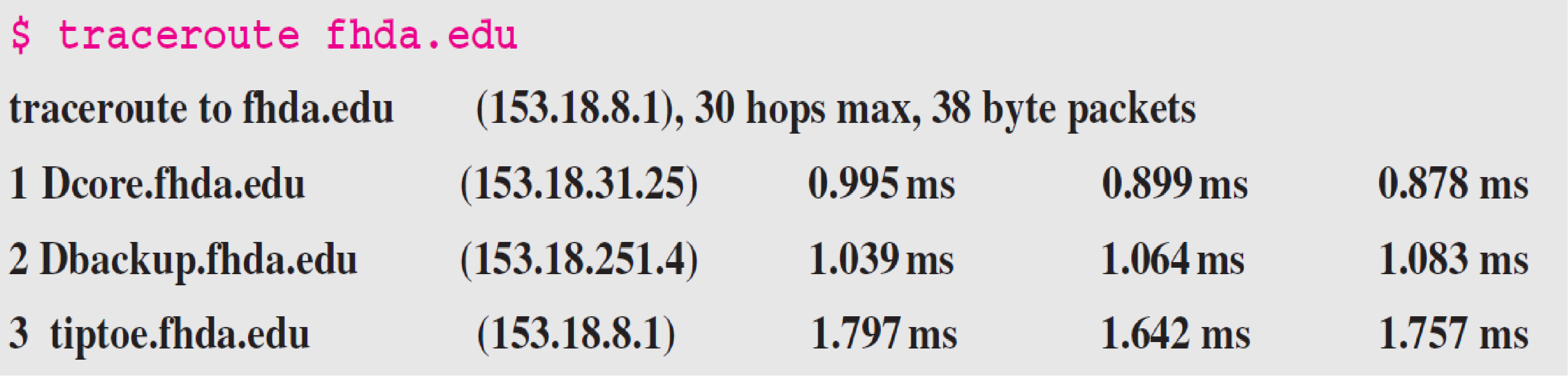
```
$ ping mail.adelphia.net
PING mail.adelphia.net (68.168.78.100) 56(84) bytes of data.
64 bytes from mail.adelphia.net (68.168.78.100): icmp_seq=0    ttl=48    time=85.4 ms
64 bytes from mail.adelphia.net (68.168.78.100): icmp_seq=1    ttl=48    time=84.6 ms
64 bytes from mail.adelphia.net (68.168.78.100): 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
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 13129 ms
rtt min/avg/max/mdev = 84.207/84.694/85.469
```

The traceroute program operation



Traceroute

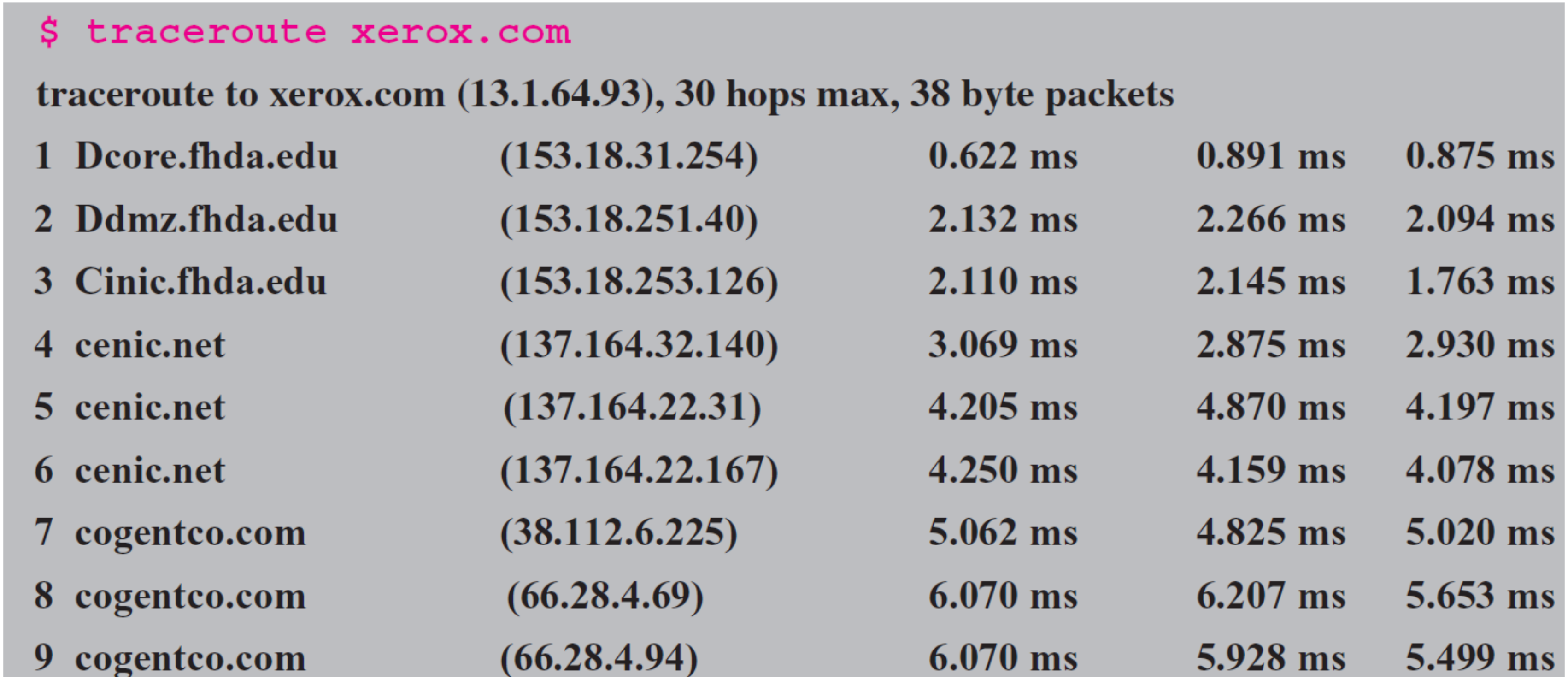
- ❑ Result of traceroute from the computer voyager.deanza.edu to the server fhda.edu



```
$ traceroute fhda.edu
traceroute to fhda.edu      (153.18.8.1), 30 hops max, 38 byte packets
1 Dcore.fhda.edu           (153.18.31.25)     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
```

Traceroute

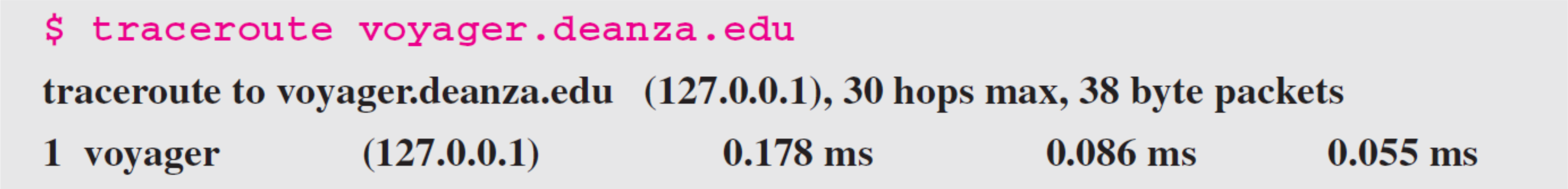
- Result of traceroute from the computer voyager.deanza.edu to the xerox.com



```
$ 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
3  Cinic.fhda.edu     (153.18.253.126)   2.110 ms   2.145 ms   1.763 ms
4  cenic.net          (137.164.32.140)   3.069 ms   2.875 ms   2.930 ms
5  cenic.net          (137.164.22.31)    4.205 ms   4.870 ms   4.197 ms
6  cenic.net          (137.164.22.167)   4.250 ms   4.159 ms   4.078 ms
7  cogentco.com       (38.112.6.225)     5.062 ms   4.825 ms   5.020 ms
8  cogentco.com        (66.28.4.69)      6.070 ms   6.207 ms   5.653 ms
9  cogentco.com       (66.28.4.94)       6.070 ms   5.928 ms   5.499 ms
```

Traceroute

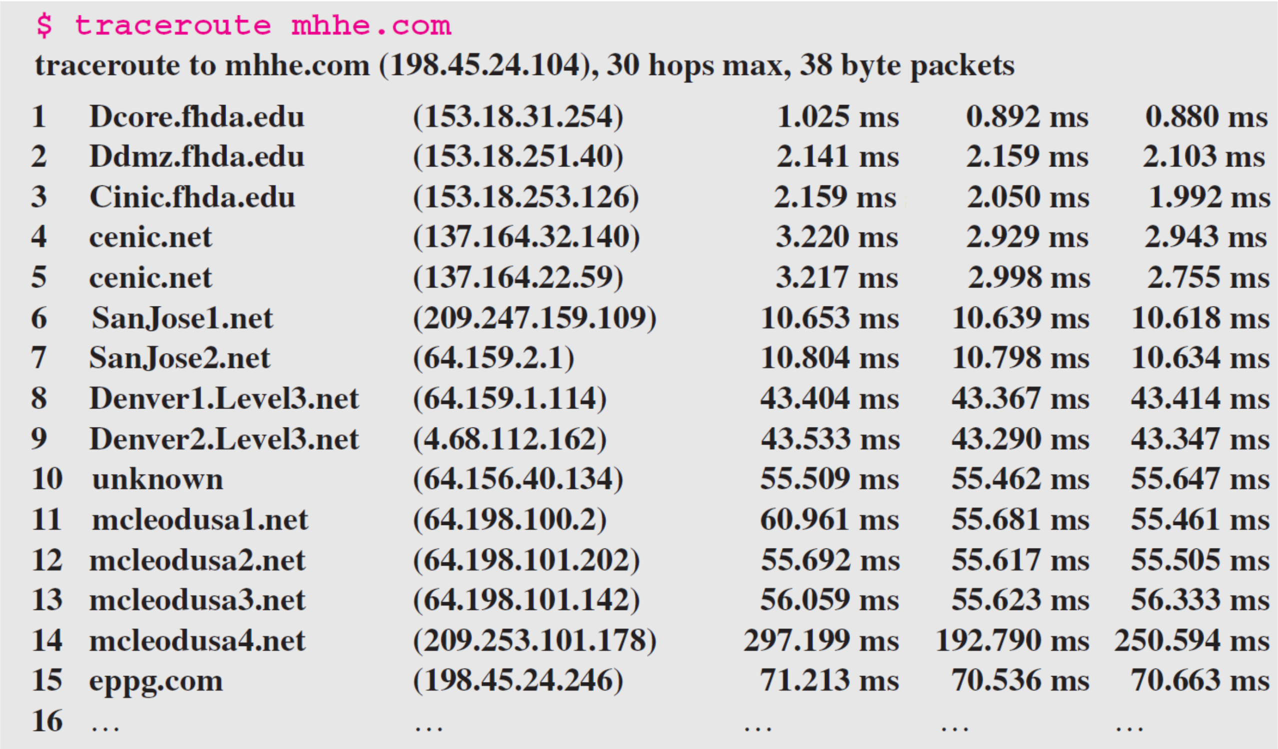
- Traceroute to localhost (loopback)



```
$ 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
```

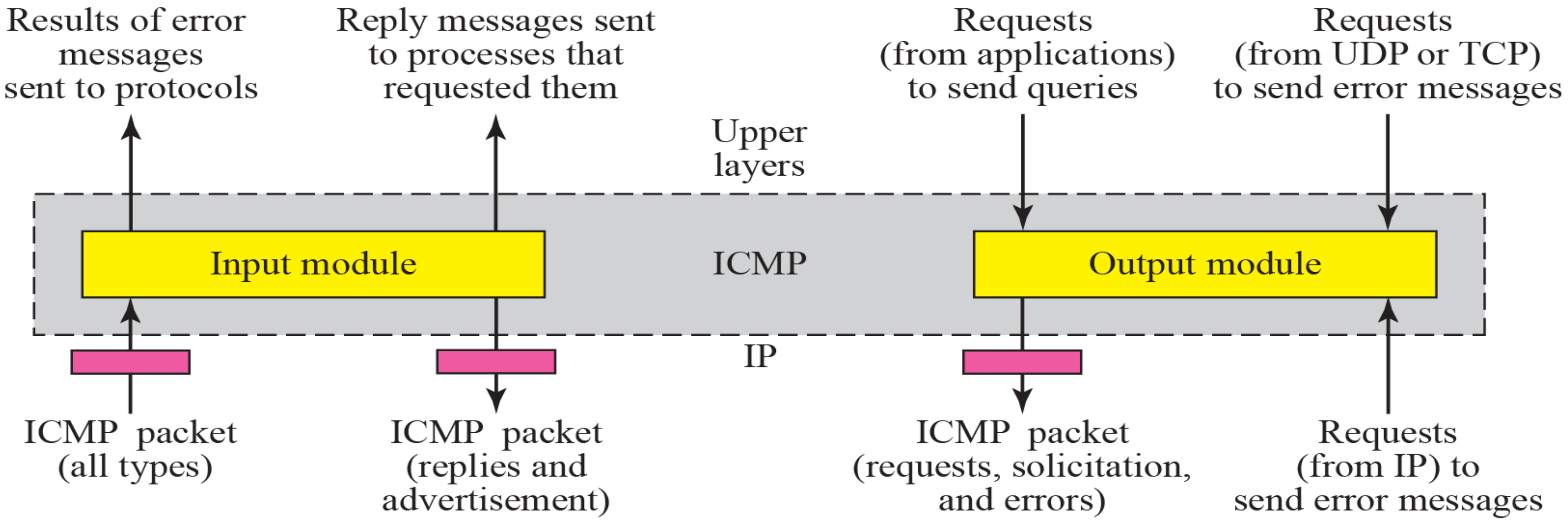
Traceroute

- Result of traceroute between fhda.edu and mhhe.com. Note that we cannot find whole route. When traceroute doesn't receive a response within 5 seconds, it prints asterisk to signify a problem



```
$ 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
4    cenic.net             (137.164.32.140)      3.220 ms     2.929 ms     2.943 ms
5    cenic.net             (137.164.22.59)       3.217 ms     2.998 ms     2.755 ms
6    SanJose1.net          (209.247.159.109)    10.653 ms    10.639 ms    10.618 ms
7    SanJose2.net          (64.159.2.1)         10.804 ms    10.798 ms    10.634 ms
8    Denver1.Level3.net    (64.159.1.114)       43.404 ms    43.367 ms    43.414 ms
9    Denver2.Level3.net    (4.68.112.162)       43.533 ms    43.290 ms    43.347 ms
10   unknown               (64.156.40.134)      55.509 ms    55.462 ms    55.647 ms
11   mcleodusa1.net        (64.198.100.2)       60.961 ms    55.681 ms    55.461 ms
12   mcleodusa2.net        (64.198.101.202)     55.692 ms    55.617 ms    55.505 ms
13   mcleodusa3.net        (64.198.101.142)     56.059 ms    55.623 ms    56.333 ms
14   mcleodusa4.net        (209.253.101.178)   297.199 ms   192.790 ms   250.594 ms
15   eppg.com              (198.45.24.246)      71.213 ms    70.536 ms    70.663 ms
16   ...                   ...                  ...          ...          ...
```

9.4 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

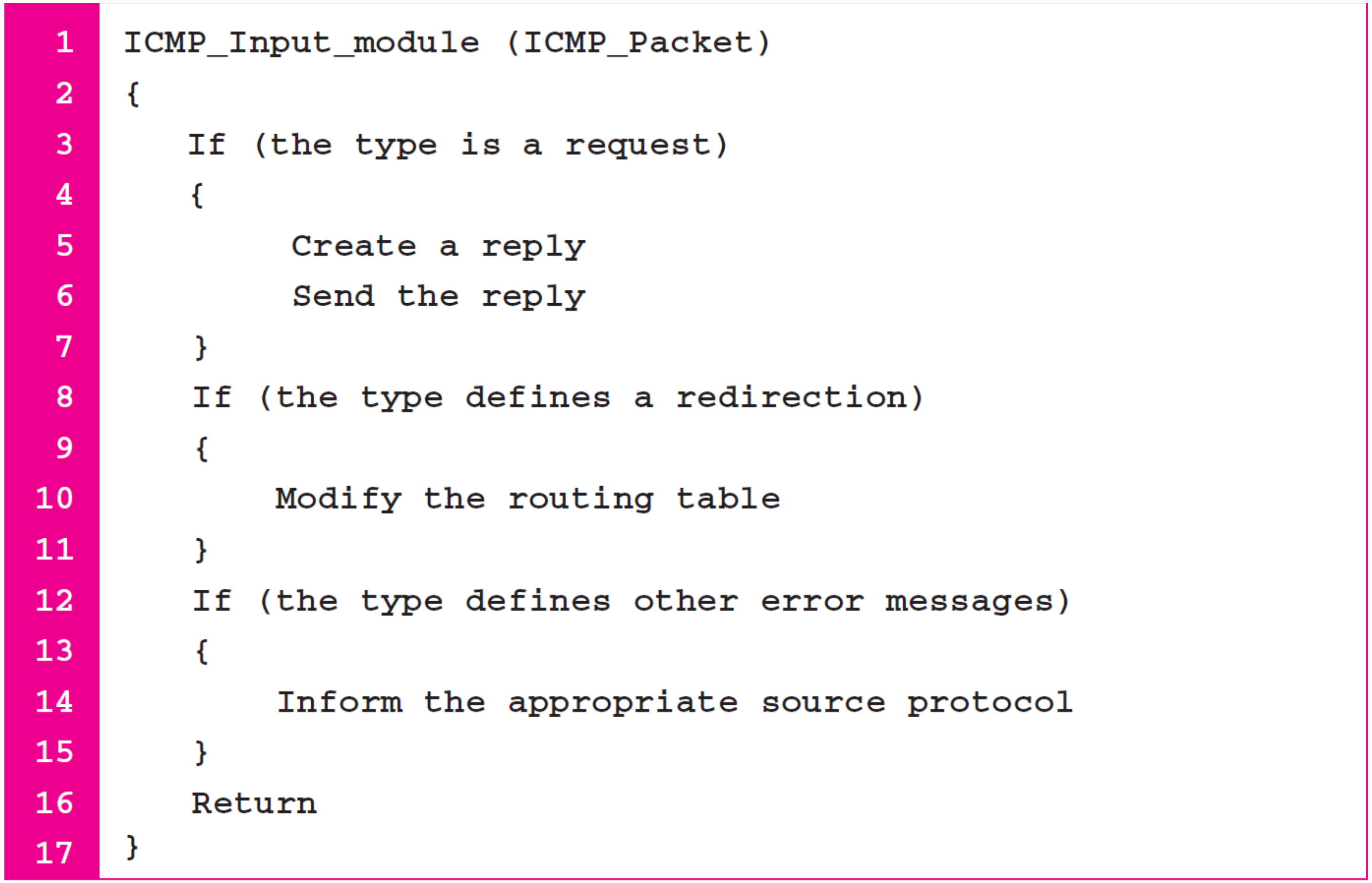
ICMP Package (cont'd)

Input module

- handling all received ICMP message
- invoked when an ICMP packet is delivered to it from the IP layer
- if the received packet is a request or solicitation, the module creates a reply or an advertisement and sends it out
- if the received packet is a redirection message, the module uses the information to update the routing table
- if the received packet is an error message, the module informs the protocol about the situation that caused the error

ICMP Package (cont'd)

❑ Pseudocode for Input Module



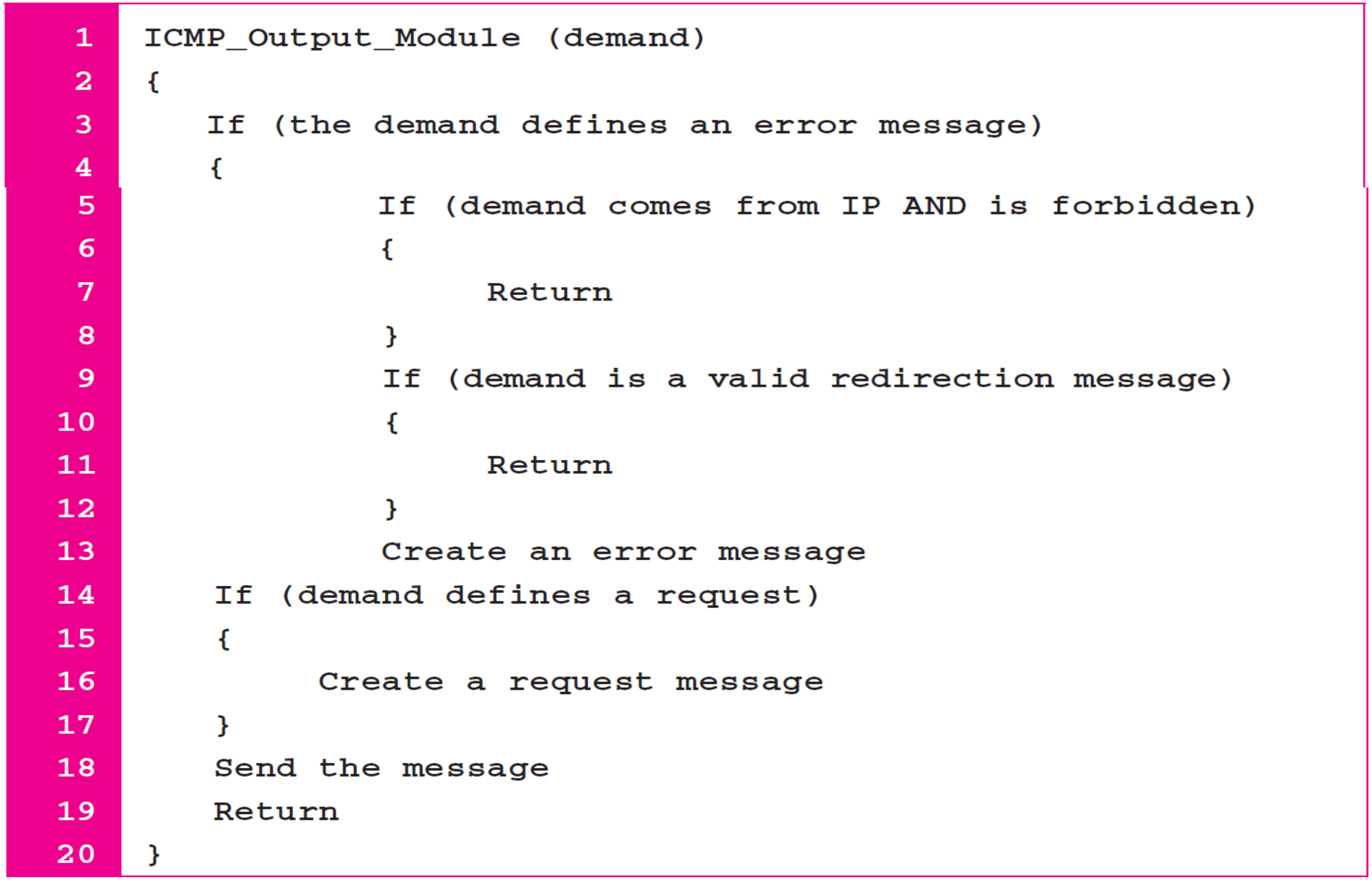
```
ICMP_Input_module (ICMP_Packet)
{
    If (the type is a request)
    {
          Create a reply
          Send the reply
    }
    If (the type defines a redirection)
    {
         Modify the routing table
    }
    If (the type defines other error messages)
    {
         Inform the appropriate source protocol
    }
    Return
}
```

ICMP Package (cont'd)

- **Output Module**
  - responsible for creating request, solicitation, or error messages requested by a higher level or the IP protocol.
  - the module receives a demand from IP, UDP or TCP to send one of the ICMP error messages
    - if the demand is from IP
      - check first that request is allowed
      - ICMP message cannot be created for four situations;
        1. ICMP error message
        2. Fragmented IP packet
        3. Multicast IP packet
        4. IP packet having IP address 0.0.0.0 or 127.X.Y.Z
  - May also receive a demand from an application program to send one of the ICMP request or solicitation messages

ICMP Package (cont’d)

❑ Pseudocode for Output Module



```
ICMP_Output_Module (demand)
{
   If (the demand defines an error message)
   {
           If (demand comes from IP AND is forbidden)
           {
                Return
           }
           If (demand is a valid redirection message)
           {
                Return
           }
           Create an error message
   If (demand defines a request)
   {
        Create a request message
   }
   Send the message
   Return
}
```

Summary

❑ The Internet Control Message Protocol (ICMP) supports the unreliable and connectionless Internet Protocol.

❑ ICMP messages are encapsulated in IP datagrams. There are two categories of ICMP messages: error-reporting and query messages. The error-reporting messages report problems that a router or a host may encounter when it processes an IP packet. The query messages, which occur in pairs, help a host or a network manager get specific information from a router or another host.

❑ The checksum for ICMP is calculated using both the header and the data fields of the ICMP message.

❑ There are several tools that can be used in the Internet for debugging. We can find if a host or router is alive and running. Two of these tools are *ping* and *teacerouter*.

❑ A simple ICMP design can consist of an input module that handle incoming ICMP packets and an output module that handles demands for ICMP services.