ELECTIVE –II - TCP/IP(18MIT35E)

UNIT-II: ARP & RARP – ARP over ATM – Proxy ARP. ARP Package – RARP – Internet Protocol (IP) – Datagram – Fragmentation – options – Checksum: IP Package. Internet Control Messang Protocol (ICMP) – Types of Message – Message format – error Reporting – Query – Checksum – ICMP Package.

Text Book :

1.Behrouz A. Forouzan, “TCP/IP Protocol Suite”, Tata Mcgraw-Hill Publishing Company, Second edition.

Reference Books:

1.W. Richard Stevens, “TCP/IP Illustrated: The Protocols”, Vol.1, Pearson Education.

2. Comer , ” Inter networking with TCP/IP : Principles ,protocols & Architecture”,Vol.1,fourth Edition, Pearson Education.

Prepared by

Dr.M.Soranamageswari

Introduction

→An internet is made of a combination of physical n/w's connected together by internetworking devices such as routers.

→A packet starting from a source host and it pass through several different physical N/W's & reach the designation host.

→By logical address (IP address) the host & routers are recognized .

→At the physical level the hosts & routers are recognized by physical address. Its a local address(48 bits)

→Both the address are must. The logical address is map to its corresponding physical address & vice versa.

→It can be done by **static/dynamic mapping**

→**Static mapping**: Create a table that associates with a logical address with a physical address. This table is stored in each m/c on the n/w.

→**Dynamic mapping**: The table is updated periodically. Each time the m/c knows one of the two address (physical /logical).

→Two protocols are designed to perform dynamic mapping.

i.**ARP**-Address Resolution Protocol.
(It maps logical address to physical address)

ii.**RARP**-Reverse Address Resolution Protocol
(It maps physical address to logical address)

2.1 Address Resolution Protocol (ARP) over ATM

→When IP packet(IP DATAGRAM) are moving through an ATM WAN, a mechanism protocol is needed to find (map) the physical address of the exiting-point router in the ATM WAN.

→This is the same task performed by ARP on a LAN. However, there is a difference between a LAN and an ATM network. A LAN is a broadcast network (at the data link layer).

→ ARP uses the broadcasting capability of a LAN to send (broadcast) an ARP request.

→ An ATM network is not a broadcast network; another solution is needed to handle the task.

Packet Format :

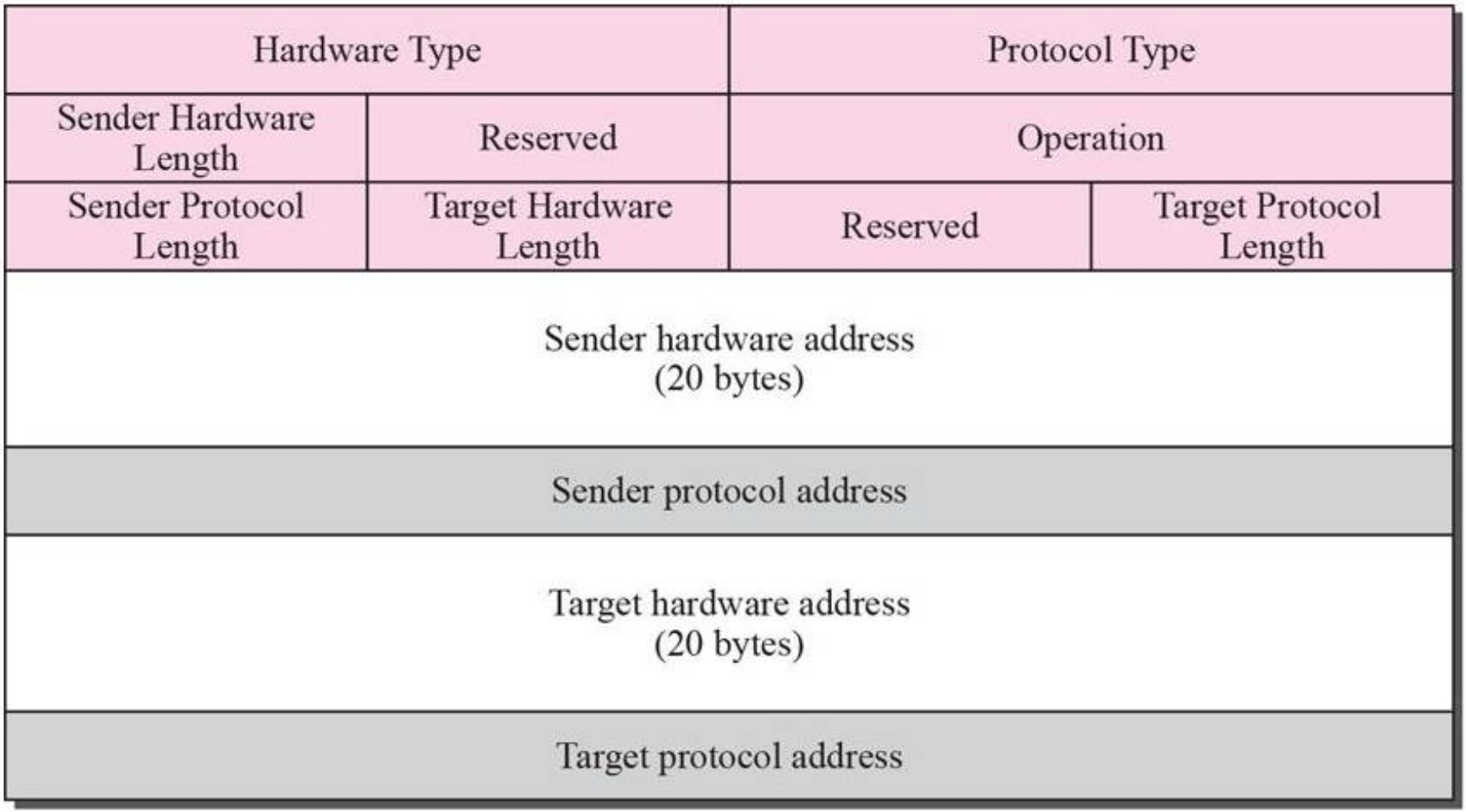
The format of an ATMARP packet, which is similar to the ARP packet, is shown in

❑ **Hardware type (HTYPE) :**
The 16-bit HTYPE field defines the type of the physical network. Its value is 001316 for an ATM network.

❑ **Protocol type (PTYPE) :**
The 16-bit PTYPE field defines the type of the protocol.
For IPv4 protocol the value is 080016.

❑ **Sender hardware length (SHLEN):**
The 8-bit SHLEN field defines the length of the sender's physical address in bytes. For an ATM network the value is 20.
Note

Figure : *ATMARP packet*



❑ **Operation (OPER):**

The 16-bit OPER field defines the type of the packet.

Types of packets are defined as,

❑ **Sender protocol length (SPLEN):** The 8-bit SPLEN field defines the length of the address in bytes. For IPv4 the value is 4 bytes.

❑ **Target hardware length (TLEN) :** The 8-bit TLEN field defines the length of the receiver's physical address in bytes. For an ATM network the value is 20. Note that if the binding is done across an ATM network and two levels of hardware addressing are necessary, the neighboring 8-bit reserved field is used to define the length of the second address.

❑ **Target protocol length (TPLEN):** The 8-bit TPLEN field defines the length of the address in bytes. For IPv4 the value is 4 bytes.

❑ **Sender hardware address (SHA):** The variable-length SHA field defines the physical address of the sender. For ATM networks defined by the ATM Forum, the length is 20 bytes.

❑ **Sender protocol address (SPA):** The variable-length SPA field defines the address of the sender. For IPv4 the length is 4 bytes.

ss

❑ **Target hardware address (THA):** The variable-length THA field defines the physical address of the receiver. For ATM networks defined by the ATM Forum, the length is 20 bytes. This field is left empty for request messages and filled in for reply and NACK messages.

❑ **Target protocol address (TPA):** The variable-length TPA field defines the address of the receiver. For IPv4 the length is 4 bytes.

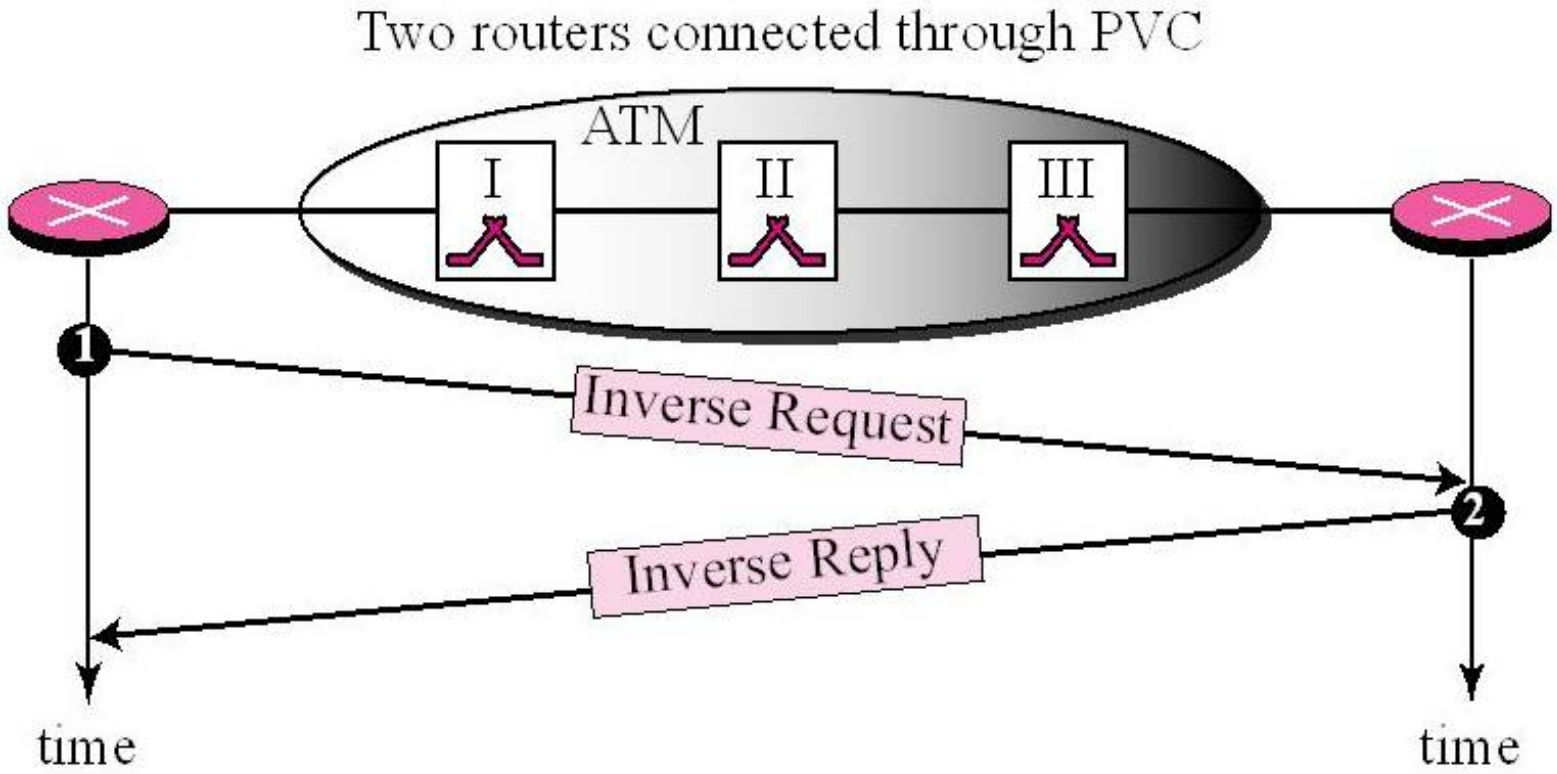
ATMARP Operation :

There are two methods to connect two routers on an ATM network: through a **permanent virtual circuit** (PVC) or through a **switched virtual circuit (SVC)**. The operation of ATMARP depends on the connection method.

PVC Connection :

A permanent virtual circuit (PVC) connection is established between two end points by the network provider. The VPIs and VCIs are defined for the permanent connections and the values are entered in a table for each switch.

Figure : *Binding with PVC*



SVC Connection :

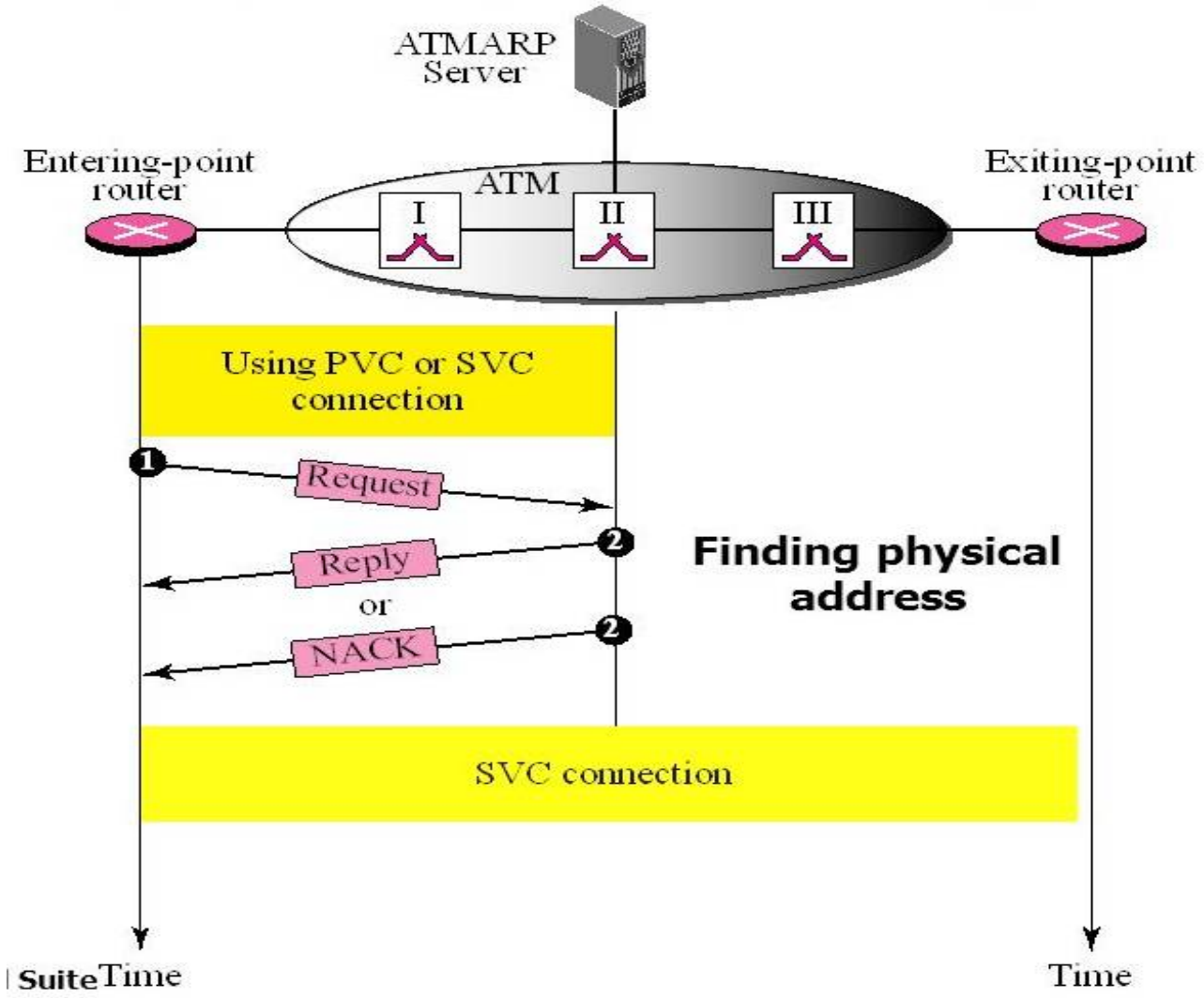
→In a switched virtual circuit (SVC) connection, each time a router wants to make a connection with another router (or any computer), a new virtual circuit must be established.

→However, the virtual circuit can be created only if the entering-point router knows the **physical address** of the exiting-point router (ATM does not recognize IP addresses).

→To map the IP addresses to physical addresses, each router runs a client ATMARP program, but only one computer runs an ATMARP server program.

→ An ARP client can broadcast an ARP request message and each router on the network will receive it; only the target router will respond.

Figure : *Binding with ATMARP*



Cont…

→The process of establishing a virtual connection requires three steps:

1) connecting to the server,
2) receiving the physical address, and
3) establishing the connection

Connecting to the Server :

→ Normally, there is a permanent virtual circuit established between each router and the server.

→ If there is no PVC connection between the router and the server, the server must at least know the physical address of the router to create an SVC connection just for exchanging ATMARP request and reply messages.

Receiving the Physical Address :

→ When there is a connection between the entering point router and the server, the router sends an ATMARP request to the server.
→ The server sends back an ATMARP reply if the physical address can be found or an ATMARP NACK otherwise. If the entering-point router receives a NACK, the datagram is dropped.

Establishing the connection :

→ Virtual Circuits After the entering-point router receives the physical address of the exiting-point router, it can request an SVC between itself and the exiting point router.
→ The ATM network uses the two physical addresses to set up a virtual circuit which lasts until the entering-point router asks for disconnection.

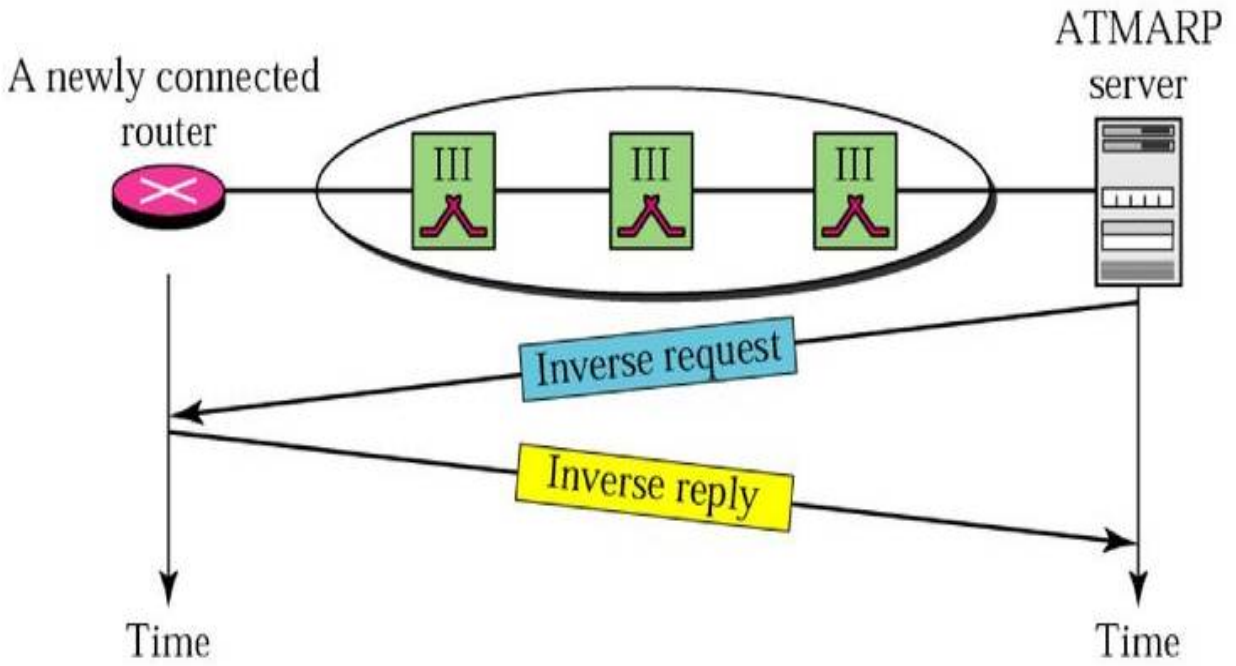
Building the Table :

→How does the ATM server build its mapping table? This is also done through the use of ATMARP and the two inverse messages (inverse request and inverse reply).

→ When a router is connected to an ATM network for the first time and a permanent virtual connection is established between the router and the server, the server sends an inverse request message to the router.

→ The router sends back an inverse reply message, which includes its IP address and physical address. Using these two addresses, the server creates an entry in its routing table to be used if the router becomes an exiting-point router in the future.

Figure: Building a table

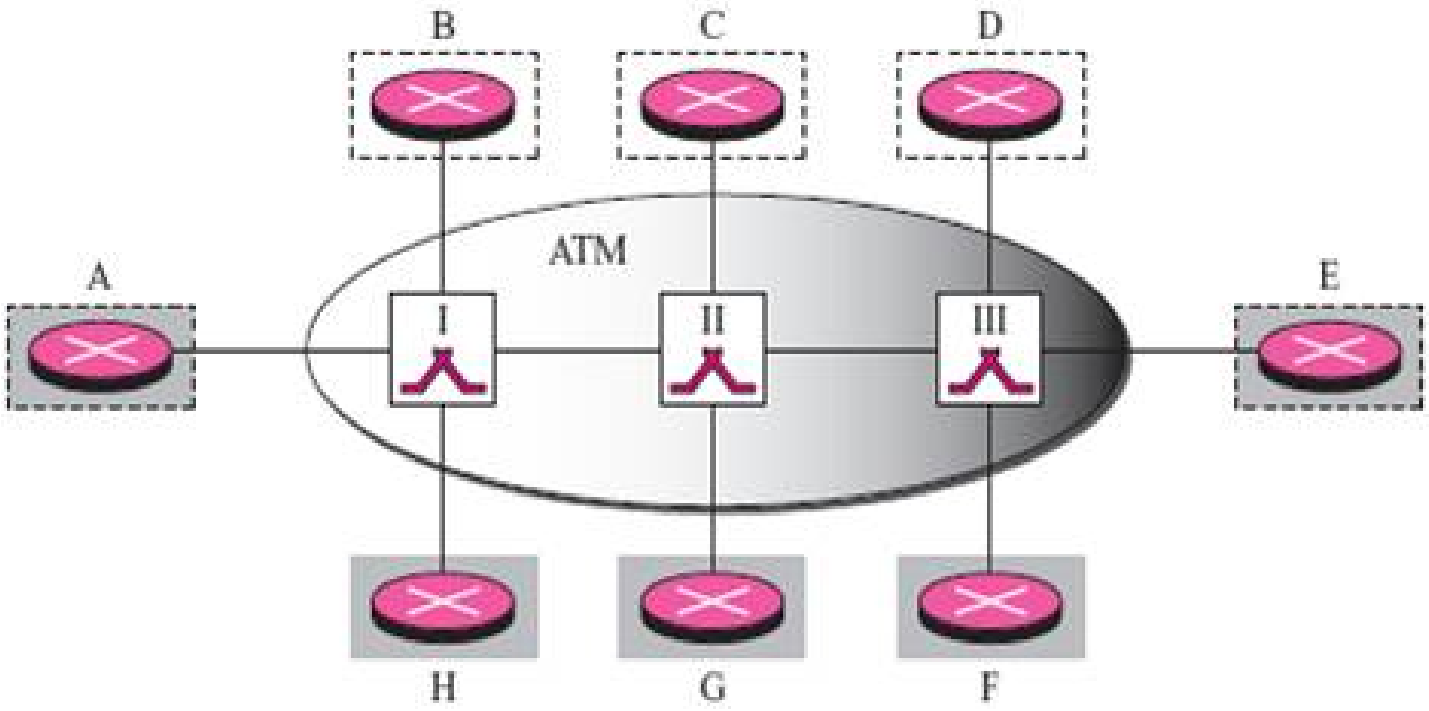


Logical IP Subnet (LIS) :

→ Before we leave the subject of IP over ATM, we need to discuss a concept called **logical IP subnet (LIS).**

→ For the same reason that a large LAN can be divided into several subnets, an ATM network can be divided into logical (not physical) subnetworks.

Figure : *LIS*



Cont…

→Routers connected to an ATM network can belong to one or more logical subnets, routers B, C, and D belong to one logical subnet , routers F, G, and H belong to another logical subnet .

→ Routers A and E belong to both logical subnets. A router can communicate and send IP packets directly to a router in the same subnet.

→However, if it needs to send a packet to a router that belongs to another subnet, the packet must first go to a router that belongs to both subnets.

→ For example, router B can send a packet directly to routers C and D. But a packet from B to F must first pass through A or E.

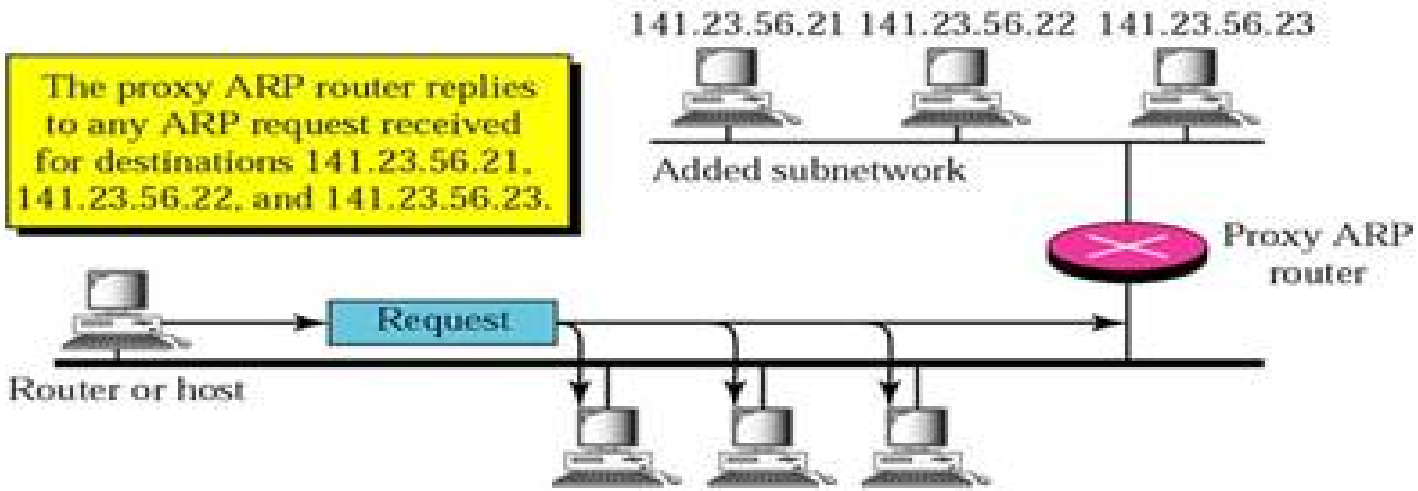
2.2 Proxy ARP

→ A technique called proxy ARP is used to create a subnetting effect.

→ A proxy ARP is an ARP that acts on behalf of a set of hosts.

→Whenever a router running a proxy ARP receives an ARP request looking for the IP address of one of these hosts, the router sends an ARP reply announcing its own hardware (physical) address.

→ After the router receives the actual IP packet, it sends the packet to the appropriate host or router.


The proxy ARP router replies to any ARP request received for destinations 141.23.56.21, 141.23.56.22, and 141.23.56.23.
141.23.56.21
141.23.56.22
141.23.56.23
Added subnetwork
Proxy ARP router
Request
Router or host

Cont…

→However, the administrator may need to create a subnet without changing the whole system to recognize subnetted addresses. One solution is to add a router running a proxy ARP.

→ In this case, the router acts on behalf of all of the hosts installed on the subnet.

→ When it receives an ARP request with a target IP address that matches the address of one of its protégés (141.23.56.21, 141.23.56.22, and 141.23.56.23), it sends an ARP reply and announces its hardware address as the target hardware address.

→ When the router receives the IP packet, it sends the packet to the appropriate host.

2.3 ARP PACKAGE

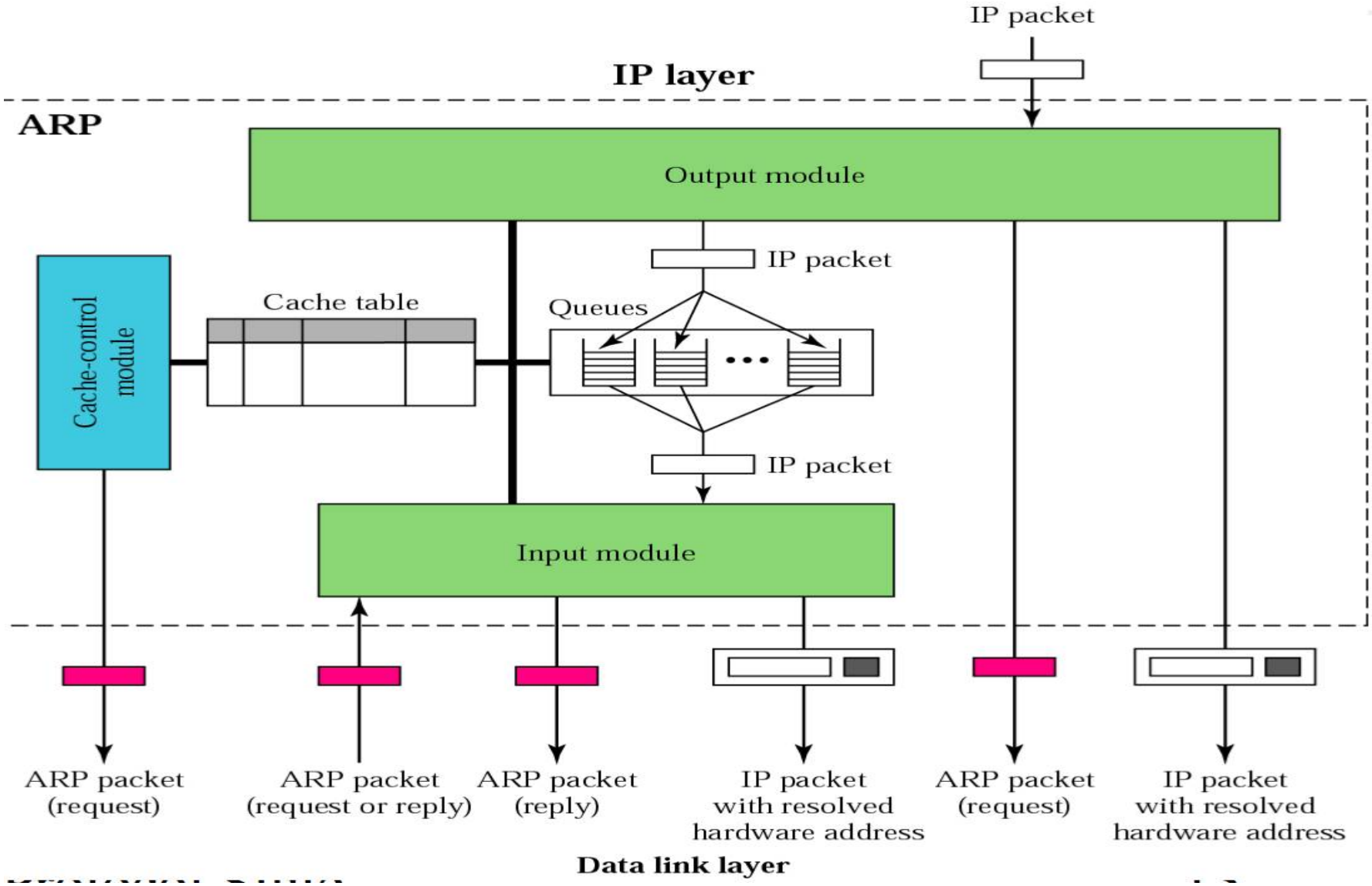
→The purpose is to show the components of a ARP package and the relationships between the components.

ARP package involves five components:

1) cache table,
2) queues,
3) an output module,
4) an input module, and
5) a cache-control module.

→ The package receives an IP datagram that needs to be encapsulated in a frame that needs the destination physical (hardware) address.

Figure: ARP components



→ If the ARP package finds this address, it delivers the IP packet and the physical address to the data link layer for transmission.

a) Cache Table :

→A sender usually has more than one IP datagram to send to the same destination.

→ It is inefficient to use the ARP protocol for each datagram destined for the same host or router.

→ The do this in efficient manner cache table is used. When a host or router receives the corresponding physical address for an IP datagram, the address can be saved in the cache table.

→ This address can be used for the data grams destined for the same receiver within the next few minutes.

Cont…

The cache table is implemented as an array of entries. The fields are:

- **State:** This column shows the state of the entry. It can have one of three values: *FREE, PENDING, or RESOLVED.*

→*The **FREE** state means that the time-to-live(TTL) for* this entry has expired.The space can be used for a new entry.

→ The **PENDING** state means a request for this entry has been sent, but the reply has not yet been received.

→ The **RESOLVED** state means that the entry is complete. The entry now has the physical (hardware) address of the destination.

→ The packets waiting to be sent to this destination can use the information in this entry.

- **Hardware type:** This is the 16 bit field define the type of N/W on which ARP is running.
- **Protocol type:** This is the 16 bit field define the protocol.
  Ex. IPv4 protocol is 0800_{16}
- **Hardware length:** This is the 8 bit field define the length of the physical address. Ex. Ethernet value is 6
- **Protocol length:** This is the 8 bit field define the length of the logical address. Ex. IPv4 protocol value is 4.
- **Interface number:** A router can be connected to different networks, each with a different interface number. Each network can have different hardware and protocol types.
- **Queue number.** ARP uses numbered queues to enqueue the packets waiting for address resolution. Packets for the same destination are usually enqueued in the same queue.

❑ **Attempts:** This column shows the number of times an ARP request is sent out for this entry.
❑ **Time-out:** This column shows the lifetime of an entry in seconds.
❑ **Hardware address:** This column shows the destination hardware address. It remains empty until resolved by an ARP reply.
❑ **Protocol address :** This column shows the destination IP address.

b) Queues:

→ ARP package maintains a set of queues, one for each destination, to hold the IP packets while ARP tries to resolve the hardware address.

→The output module sends unresolved packets into the corresponding queue.

→The input module removes a packet from a queue and sends it, with the resolved physical address, to the data link layer for transmission.

c) Output Module

→The output module waits for an IP packet.

→The output module checks the cache table to find an entry corresponding to the destination IP address of this packet.

→ The destination IP address of the IP packet must match the protocol address of the entry.

→If the entry is Resolved→ the packet waits until the designation H/W address is passed to the data link layer for transmission.

→If the entry is PENDING→ the packet waits until the designation H/W address is found. so queue will be created for this designation.

→If no entry is found the module creates a queue and enqueues the packet.A new entry with the state of pending is created for this designation and the value of the **ATTEMPTS** fields is set to 1.

→An ARP request packet is then broadcast.

d) Input Module :

→The input module waits until an ARP packet (request or reply) arrives.
→The input module checks the cache table to find an entry corresponding to this ARP packet.
→The target protocol address should match the protocol address of the entry.
→State of entry== PENDING, the module updates the entry and changed to RESOLVED. The module also sets the value of the TIME OUT for this entry.
→RESOLVED →The module still updates the entry. Because target H/W address could have been change. Time out field is also reset.
→If no entry is found the module create a new entry and add it into the table. The protocol requires that any information received is added to the table for future use.

e) Cache-Control Module :

→The cache-control module is responsible for maintaining the cache table. It periodically every 5 secs. checks the cache table, entry by entry.

→ If the state of the entry is FREE, it continues to the next entry.
→If the state is PENDING, the module increments the value of the attempts field by 1.It then checks the value of the attempts field. If this value is greater than the maximum number of attempts allowed, the state is changed to FREE and the corresponding queue is destroyed.
→If the state is RESOLVED →Decrement the value of time out by the value of elapsed time.

2.4 INTRODUCTION : Internet Protocol (IP)

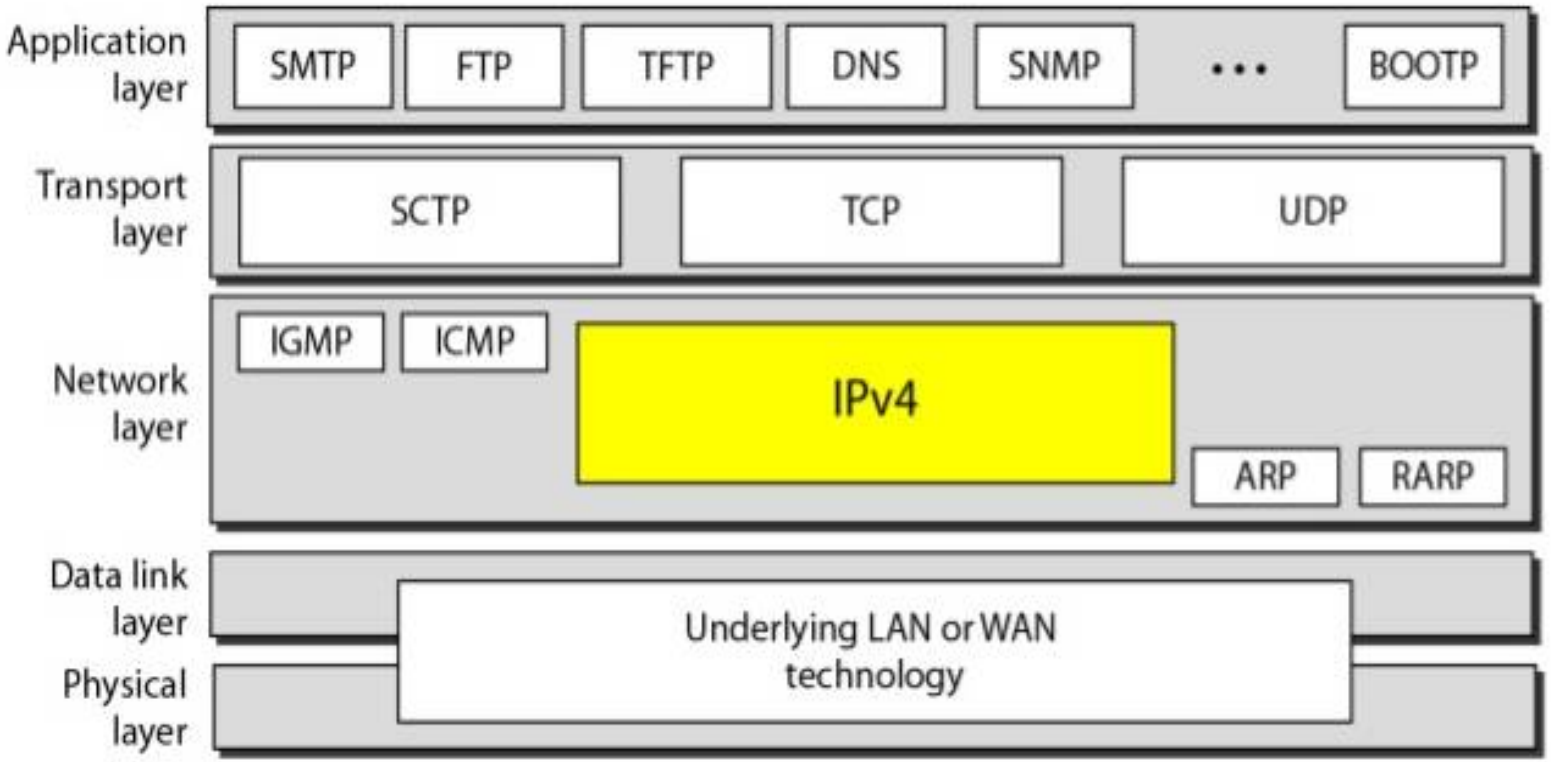
→The **Internet Protocol (IP)** is the transmission mechanism used by the TCP/IP protocol at the network layer .

→IP is an unreliable and connectionless datagram protocol—a best-effort delivery service.

→ The term best-effort means that IP packets can be corrupted, lost, arrive out of order, or delayed and may create congestion for the network.

→ If reliability is important, IP must be paired with a reliable protocol such as TCP.

Figure : Position of IP in TCP/IP protocol suite



Cont…

→The post office does its best to deliver the mail but does not always succeed.

→If an unregistered letter is lost, it is up to the sender or recipient to discover the loss and rectify the problem.

→ The post office itself does not keep track of every letter
and cannot notify a sender of loss or damage.

→IP is also a connectionless protocol for a packet switching network that uses the datagram approach .
→ This means that each datagram is handled independently,
and each datagram can follow a different route to the destination.

→This implies that datagram's sent by the same source to the same destination could arrive out of order.

→Also, some could be lost or corrupted during transmission.

→ Again, IP relies on a higher-level protocol to take care of all these problems.

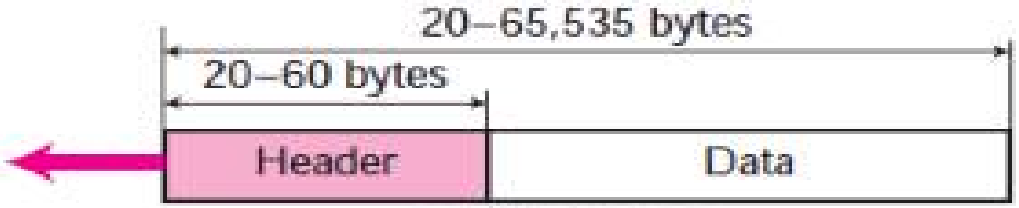
2.5 DATAGRAMS

→Packets in the network (internet) layer are called **datagrams.**

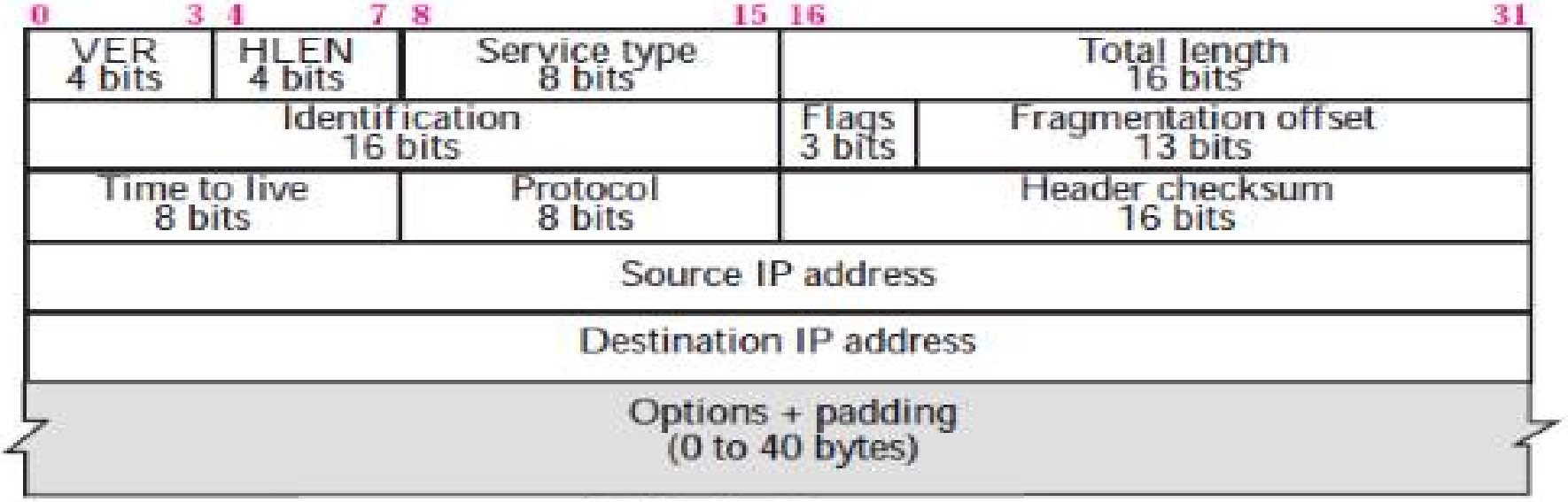
→A datagram is a variable-length packet consisting of two parts: **header and data.**

→ The header is 20 to 60 bytes in length and contains information essential to routing and delivery.

Fig : IP DATAGRAM



a. IP datagram



b. Header format

A brief description of each field is as follows.

❑ **Version (VER):**

→ This 4-bit field defines the version of the IP protocol. Currently the version is 4.

→However, version 6 (or IPv6) may totally replace version 4 in the future.

→ If the machine is using some other version of IP, the datagram is discarded rather than interpreted incorrectly.

❑ **Header length (HLEN) :**

→ This 4-bit field defines the total length of the datagram header in 4-byte words.

→ This field is needed because the length of the header is variable (between 20 and 60 bytes).

→When there are no options, the header length is 20 bytes, and the value of this field is 5 (5 × 4= 20).

→ When the option field is at its maximum size, the value of this field is 15 (15 × 4 = 60).

❑ **Service type :**

→In the original design of IP header, this field was referred to as **type of service (TOS),** which defined how the datagram should be handled.

→ Part of the field was used to define the precedence of the datagram; the rest defined the type of service (low delay, high throughput, and so on).

❑ **Total length :**

→This is a 16-bit field that defines the total length (header plus data) of the IP datagram in bytes.

→To find the length of the data coming from the upper layer, subtract the header length from the total length.

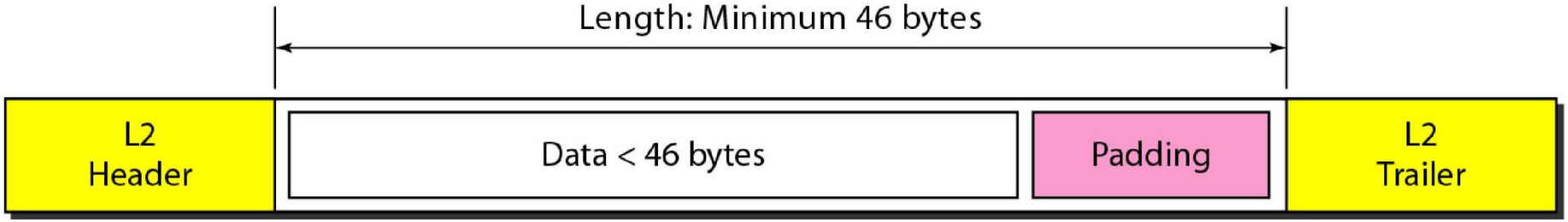
→The header length can be found by multiplying the value in the HLEN field by four.

❑ **Identification :** This field is used in fragmentation

❑ **Flags:** This field is used in fragmentation

❑ **Fragmentation offset :**This field is used in fragmentation

Figure : Encapsulation of a small datagram in an Ethernet frame



❑ **Time to live**:

→A datagram has a limited lifetime in its travel through an internet.

→ This field was originally designed to hold a timestamp, which was decremented by each visited router.

→ The datagram was discarded when the value became zero. However, for this scheme, all the machines must have synchronized clocks and must know how long it takes for a datagram to go from one machine to another.

→Today, this field is mostly used to control the maximum number of hops (routers) visited by the datagram. When a source host sends the datagram,

❑ **Protocol:**

→This 8-bit field defines the higher-level protocol that uses the services of the IP layer.

→ An IP datagram can encapsulate data from several higher level protocols such as TCP, UDP, ICMP, and IGMP. This field specifies the final destination protocol to which the IP datagram should be delivered.

❑ **Source address :**

→This 32-bit field defines the IP address of the source.

→This field must remain unchanged during the time the IP datagram travels from the source host to the destination host.

❑ **Destination address :**

→This 32-bit field defines the IP address of the destination.

→ This field must remain unchanged during the time the IP datagram travels from the source host to the destination host.

2.6 FRAGMENTATION

→A datagram can travel through different networks. Each router decapsulates the IP datagram from the frame it receives, processes it, and then encapsulates it in another frame.

→The format and size of the received frame depend on the protocol used by the physical network through which the frame has just traveled.

→The format and size of the sent frame depend on the protocol used by the physical network through which the frame is going to travel.

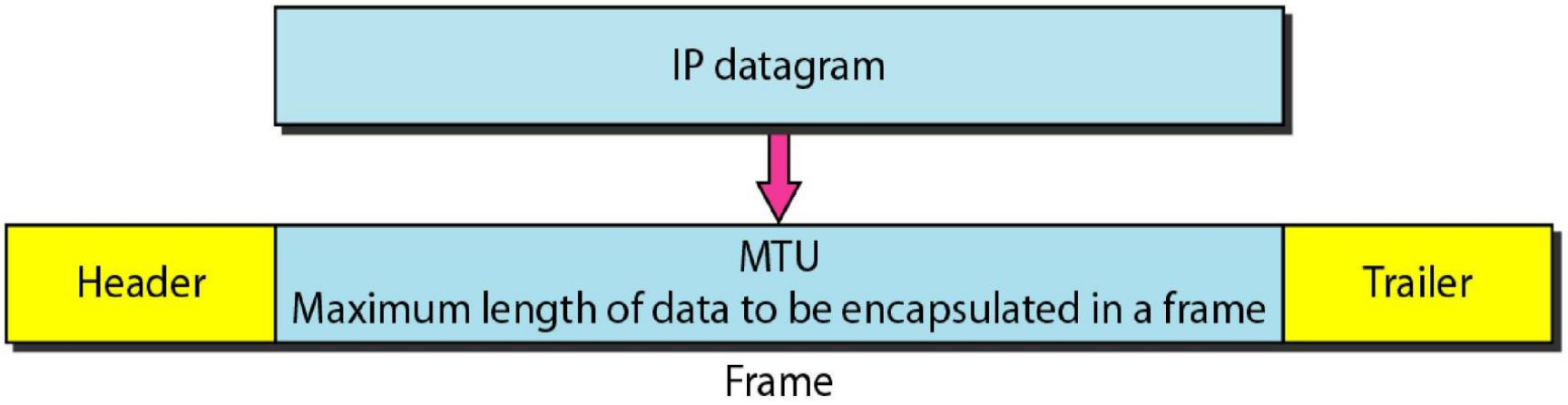
→ For example, if a router connects a LAN to a WAN, it receives a frame in the LAN format and sends a frame in the WAN format.

Maximum Transfer Unit (MTU) :

→Each data link layer protocol has its own frame format in most protocols.

→when a datagram is encapsulated in a frame, the total size of the datagram must be less than this maximum size, which is defined by the hardware and software used in the network .

Figure : MTU



Cont…

→The value of the MTU differs from one physical network protocol to another.

→For other physical networks, we must divide the datagram to make it possible to pass through these networks. This is called **fragmentation.**

→The source usually does not fragment the IP packet.

→ The **transport layer** will instead segment the data into a size that can be accommodated by IP and the data link layer in use.

→When a datagram is fragmented, each fragment has its own header with most of the fields repeated, but some changed.

→ A fragmented datagram may itself be fragmented
if it encounters a network with an even smaller MTU.

→ In other words, a datagram can be fragmented several times before it reaches the final destination.

Cont…

→A datagram can be fragmented by the source host or any router in the path.

→ But the reassembly of the datagram, however, is done only by the destination host because each fragment becomes an independent datagram.

→ Whereas the fragmented datagram can travel through different routes, and we can never control or guarantee which route a fragmented datagram may take, all of the fragments belonging to the same datagram should finally arrive at the destination host.

→ So it is logical to do the reassembly at the final destination.

Fields Related to Fragmentation

❑ **Identification :**

→This 16-bit field identifies a datagram originating from the source host.
→ The combination of the identification and source IP address must uniquely define a datagram as it leaves the source host.
→ To guarantee uniqueness, the IP protocol uses a counter to label the datagram's.
→ The counter is initialized to a positive number. When the IP protocol sends a datagram, it copies the current value of the counter to the identification field and increments the counter by one.
→ As long as the counter is kept in the main memory, uniqueness is guaranteed.

Cont…

→When a datagram is fragmented, the value in the identification field is copied into all fragments.

→ In other words, all fragments have the same identification number, which is also the same as the original datagram.

→ The identification number helps the destination in reassembling the datagram.

→ It knows that all fragments having the same identification value should be assembled into one datagram.

❑ Flags :

→This is a three-bit field.
→The first bit is **reserved** (not used).
→The second bit is called the **do not fragment** bit. If its **value is 1**, the machine must not fragment the datagram.

→ If it cannot pass the datagram through any available physical network, it discards the datagram and sends an ICMP error message to the source host .

→ If its **value is 0**, the datagram **can be fragmented** if necessary.

→ The third bit is called the **more fragment** bit.
→If its value is 1, it means the datagram is not the last fragment; there are more fragments after this one.

❑ Fragmentation offset

→ This 13-bit field shows the relative position of this fragment with respect to the whole datagram.
→It is the offset of the data in the original datagram measured in units of 8 bytes.
→ This Figure, shows a datagram with a data size of 4000 bytes fragmented into three fragments.
→ The bytes in the original datagram are numbered 0 to 3999.
→ The first fragment carries bytes 0 to 1399. The offset for this datagram is 0/8 = 0.
→The second fragment carries bytes 1400 to 2799; the offset value for this fragment is 1400/8 = 175.
→Finally, the third fragment carries bytes 2800 to 3999. The offset value for this fragment is 2800/8 = 350.

Figure : Fragmentation example

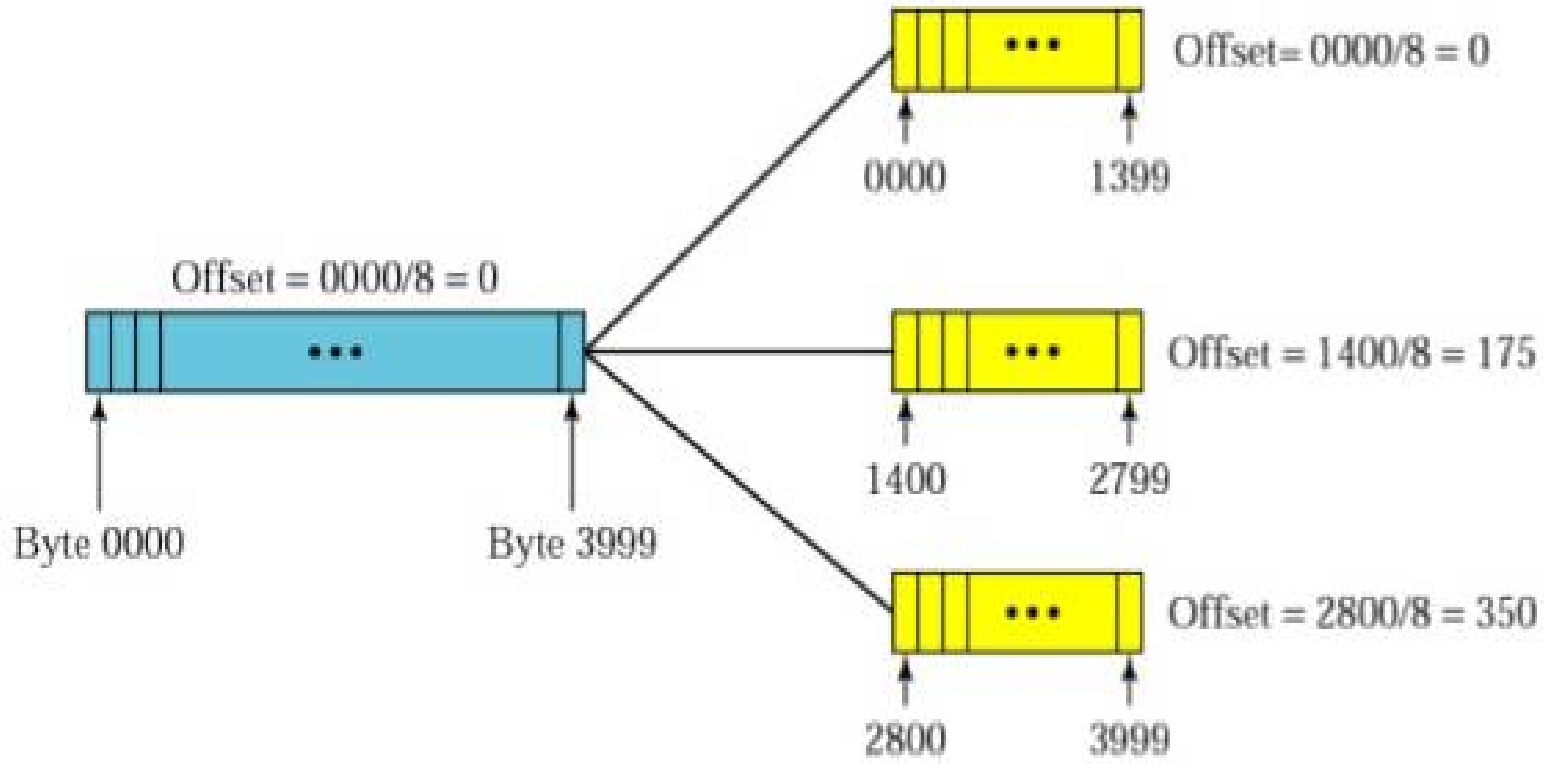
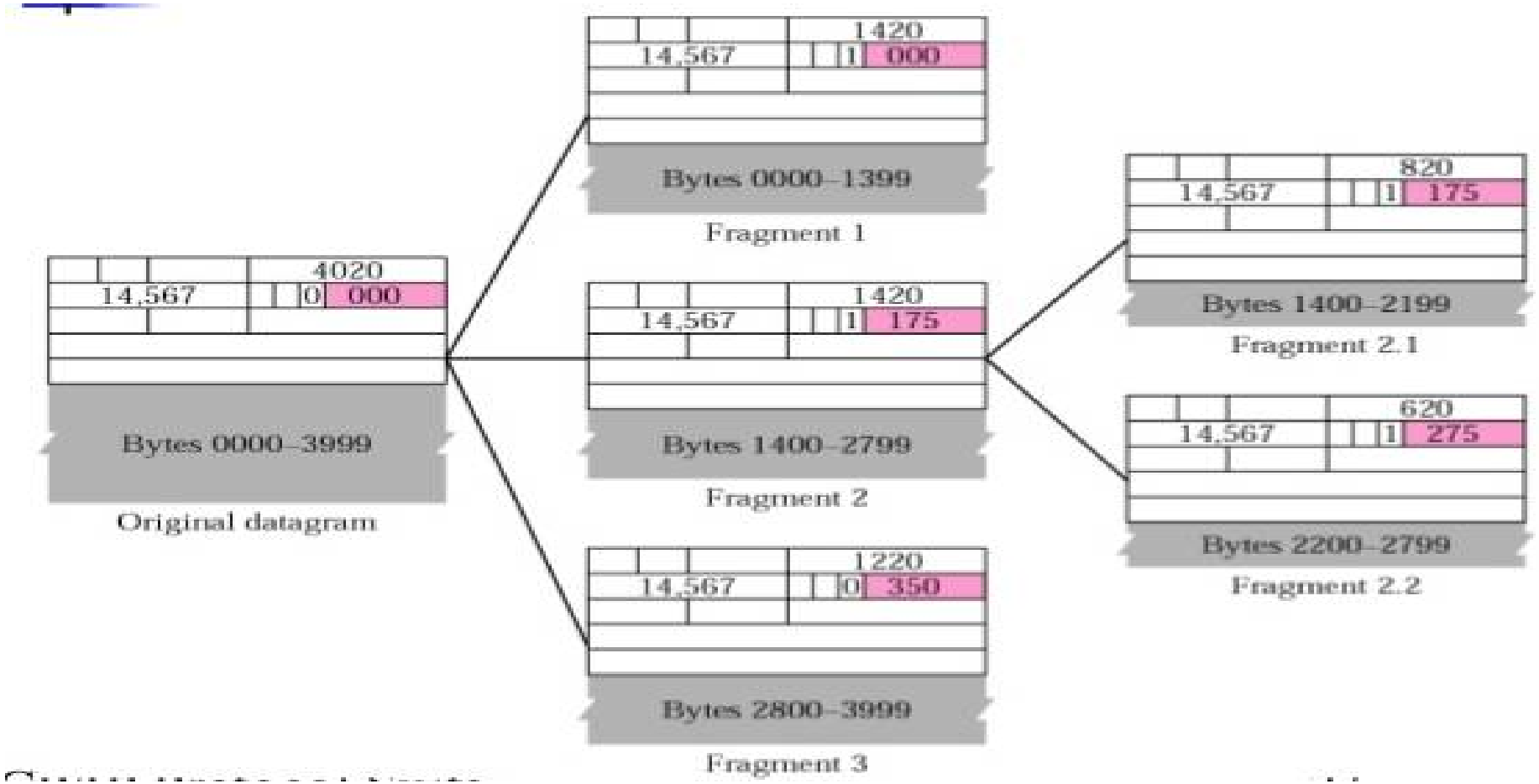


Figure : Detailed fragmentation example



Fragment using the following strategy,

Fig. shows even each fragment follows a different path and arrives out of order, the final destination host will reassemble the original datagram from the fragments received

a. The first fragment has an offset field value of zero.

b. Divide the length of the first fragment by 8. The second fragment has an offset value equal to that result.

c. Divide the total length of the first and second fragment by 8. The third fragment has an offset value equal to that result.

d. Continue the process. The last fragment has a more bit value of 0.

2.7 OPTIONS

→The **header** of the IP datagram is made of two parts: a **fixed part and a variable part.**

→ The fixed part is 20 bytes long and the variable part comprises the **options**, which can be a maximum of 40 bytes.

→ Options, as the name implies, are not required for a datagram.

→ They can be used for network **testing and debugging**.

→ Although options are not a required part of the IP header, option processing is required of the IP software.

Cont…

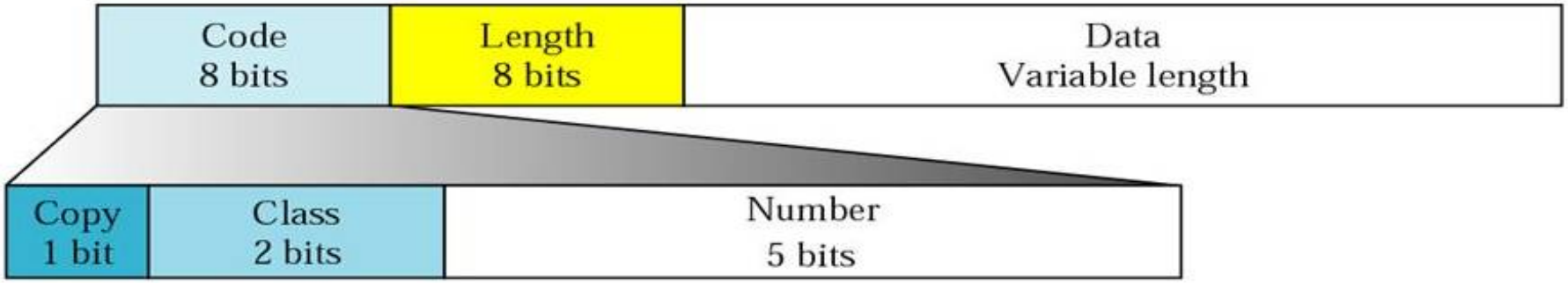
→ This means that all implementations must be able to handle options if they are present in the header.

a) Format :

→Figure shows the format of an option. It is composed of a 1 byte type field, a 1-byte length field, and a variable-sized value field.

→ The three fields are often referred to as type-length-value or TLV.

Figure : Option format



Copy
0 Copy only in first fragment
1 Copy into all fragments

Class
00 Datagram control
01 Reserved
10 Debugging and management
11 Reserved

Number
00000 End of option
00001 No operation
00011 Loose source route
00100 Timestamp
00111 Record route
01001 Strict source route

Cont…

b) Type

→The type field is 8 bits long and contains three subfields:

-- Copy,

-- Class, and

-- Number.

❑ **Copy:** This 1-bit subfield controls the presence of the option in fragmentation . When its value is 0, it means that the option must be copied only to the first fragment. If its value is 1, it means the option must be copied to all fragments.

❑ **Class:** This 2-bit subfield defines the general purpose of the option. When its value is 00, it means that the option is used for datagram control. When its value is 10, it means that the option is used for debugging and management.

Cont…

→ The other two possible values (01 and 11) have not yet been defined.

❑ **Number :** This 5-bit subfield defines the type of option. Although 5 bits can define up to 32 different types, currently only 6 types are in use. These will be discussed in a later section.

c) Length

The length field defines the total length of the option including the type field and the length field itself. This field is not present in all of the option types.

d) Value

The value field contains the data that specific options require. Like the length field, this field is also not present in all option types.

Cont…

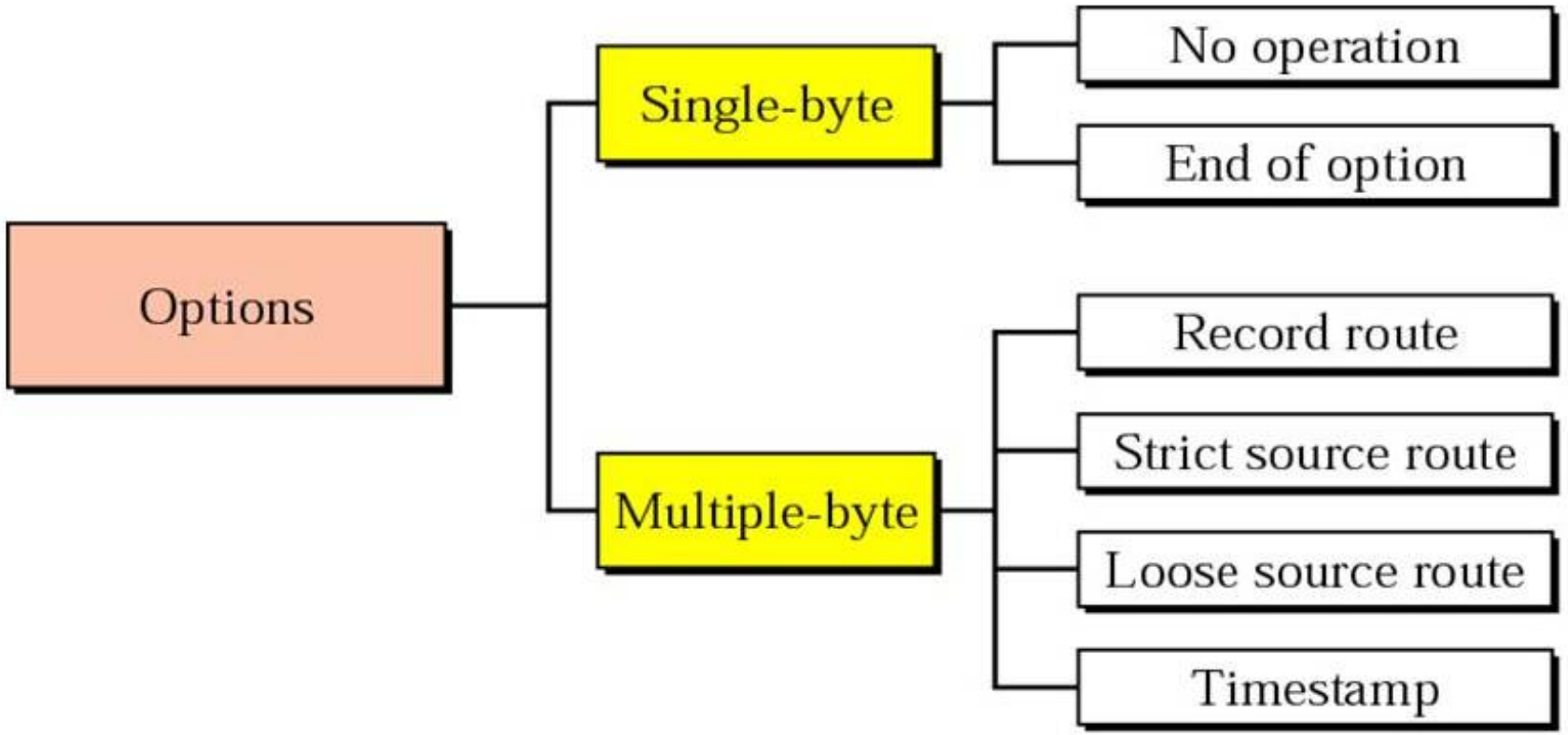
Option Types

→only six options are currently being used.

→ Two of these are 1-byte options, and they do not require the length or the data fields.

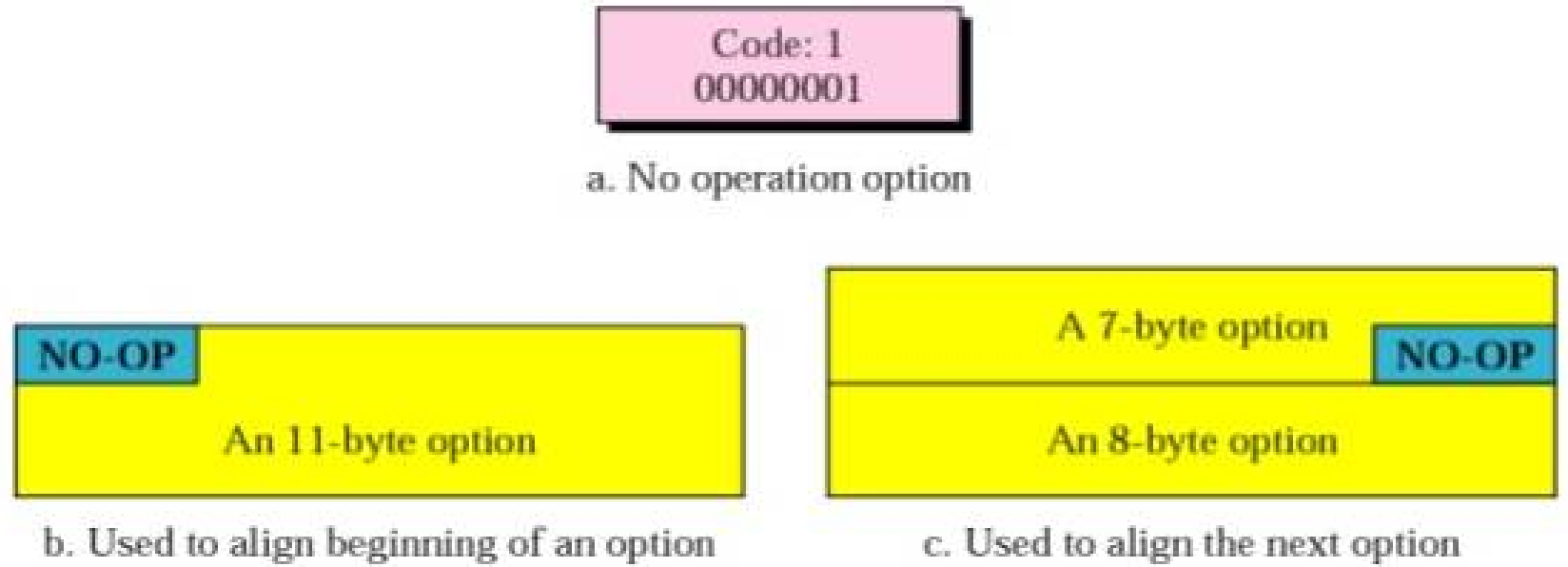
→ Four of them are multiple-byte options; they require the length and the data fields (see Figure).

Figure : ***Categories of options***



No-Operation Option :

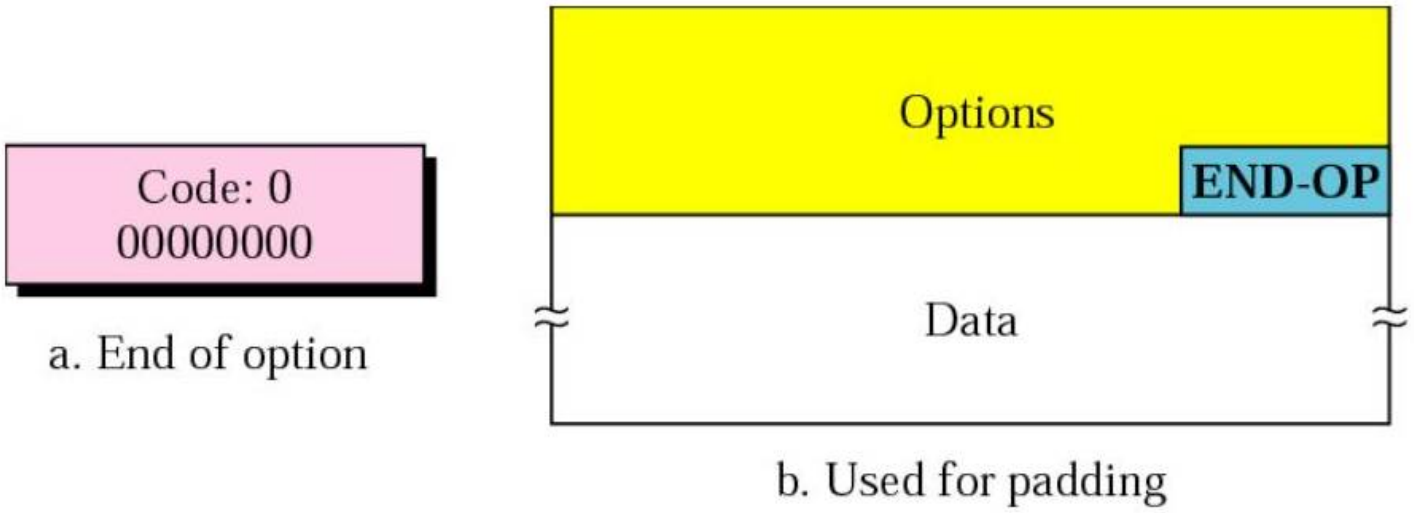
A no-operation option is a 1-byte option used as a filler between options. For example, it can be used to align the next option on a 16-bit or 32-bit boundary (see Figure).



Cont…

End-of-Option Option :

→An end-of-option option is also a 1-byte option used for padding at the end of the option field. It can only be used as the last option. Only one end-of-option can be used. After this option, the receiver looks for the payload data.



Cont…

Record-Route Option :

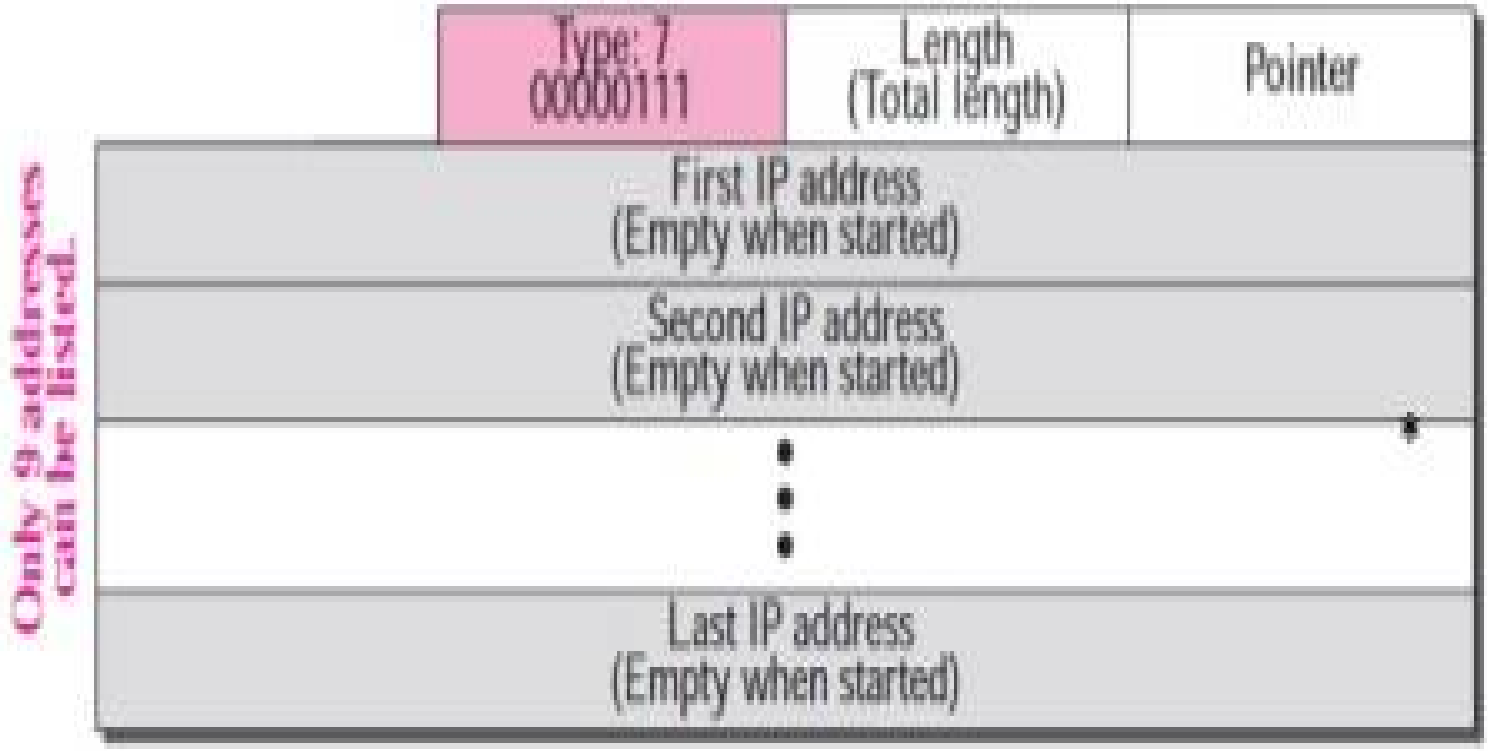
→A record-route option is used to record the Internet routers that handle the datagram.

→ It can list up to nine router IP addresses since the maximum size of the header is 60 bytes, which include 20 bytes for the base header and only 40 bytes for the option part.

→ The source creates placeholder fields in the option to be filled by the visited routers.

→ Figure 7.14 shows the format of the record route option.

Figure : *Record-route option*



Cont…

→Both the code and length fields have been described above.

→ Pointer fields points to the first available entry. The source creates empty fields for the IP addresses in the data field of the option.

→When the datagram leaves the source, all of the fields are empty. The pointer field has a value of 4, pointing to the first empty field.

→When the datagram is traveling, each router that processes the datagram compares the value of the pointer with the value of the length.

Cont…

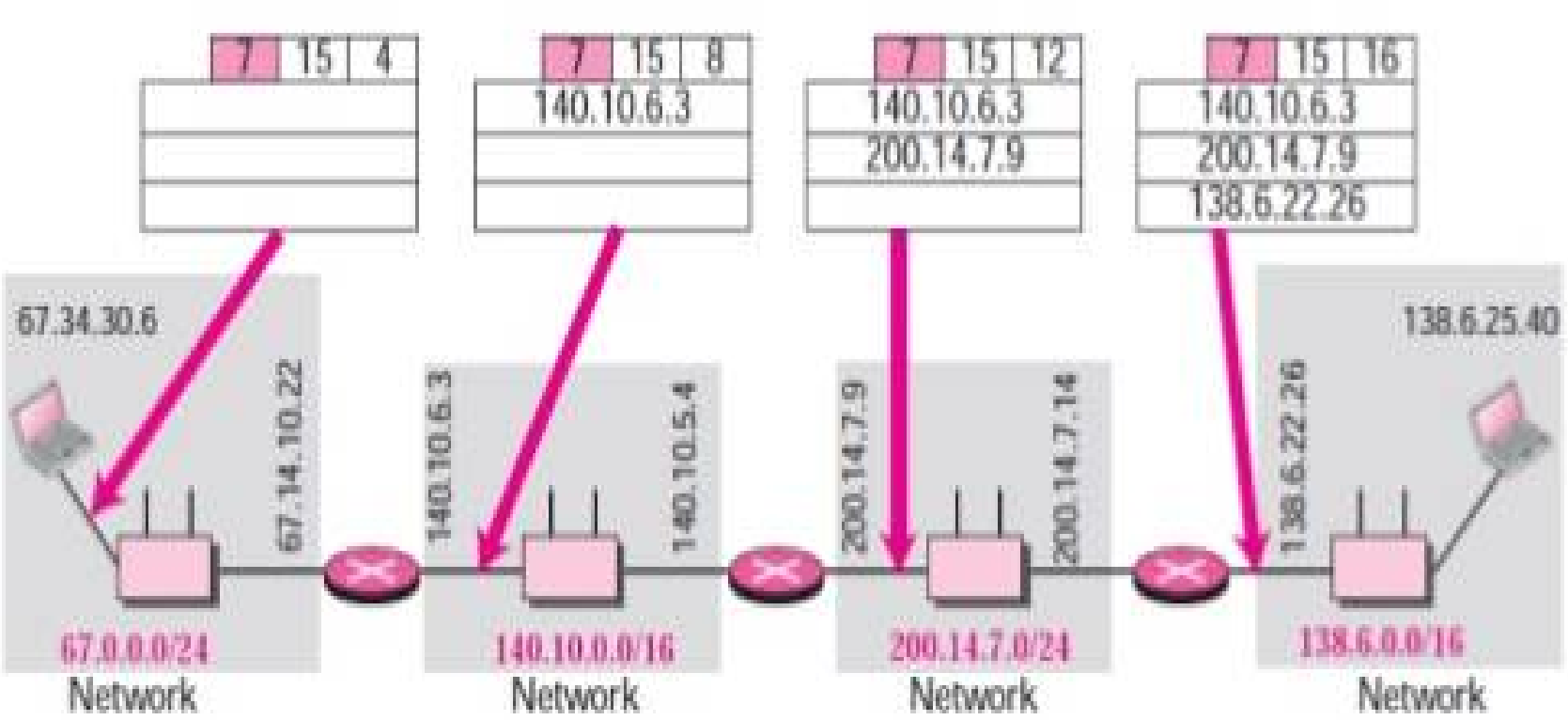
→ If the value of the pointer is **greater** than the value of the length, the **option is full** and no changes are made.

→ However, if the value of the pointer is **not greater** than the value of the length, the **router inserts its outgoing IP address** in the next empty field (remember that a router has more than one IP address).

→In this case, the router adds the IP address of its interface from which the datagram is leaving.

→The router then increments the value of the pointer by 4. Figure shows the entries as the datagram travels left to right from router to router.

Figure : *Record-route concept*



Strict-Source-Route Option

→A strict-source-route option is used by the source to **predetermine a route** for the datagram as it travels through the Internet.

→ Dictation of a route by the source can be useful for several purposes.

→The sender can choose a route with a specific type of service, such as minimum delay or maximum throughput.

→ Alternatively, it may choose a route that is safer or more reliable for the sender's purpose.

→ For example, a sender can choose a route so that its datagram does not travel through a competitor's network.

Cont…

→If a datagram specifies a strict source route, all of the routers defined in the option must be visited by the datagram.

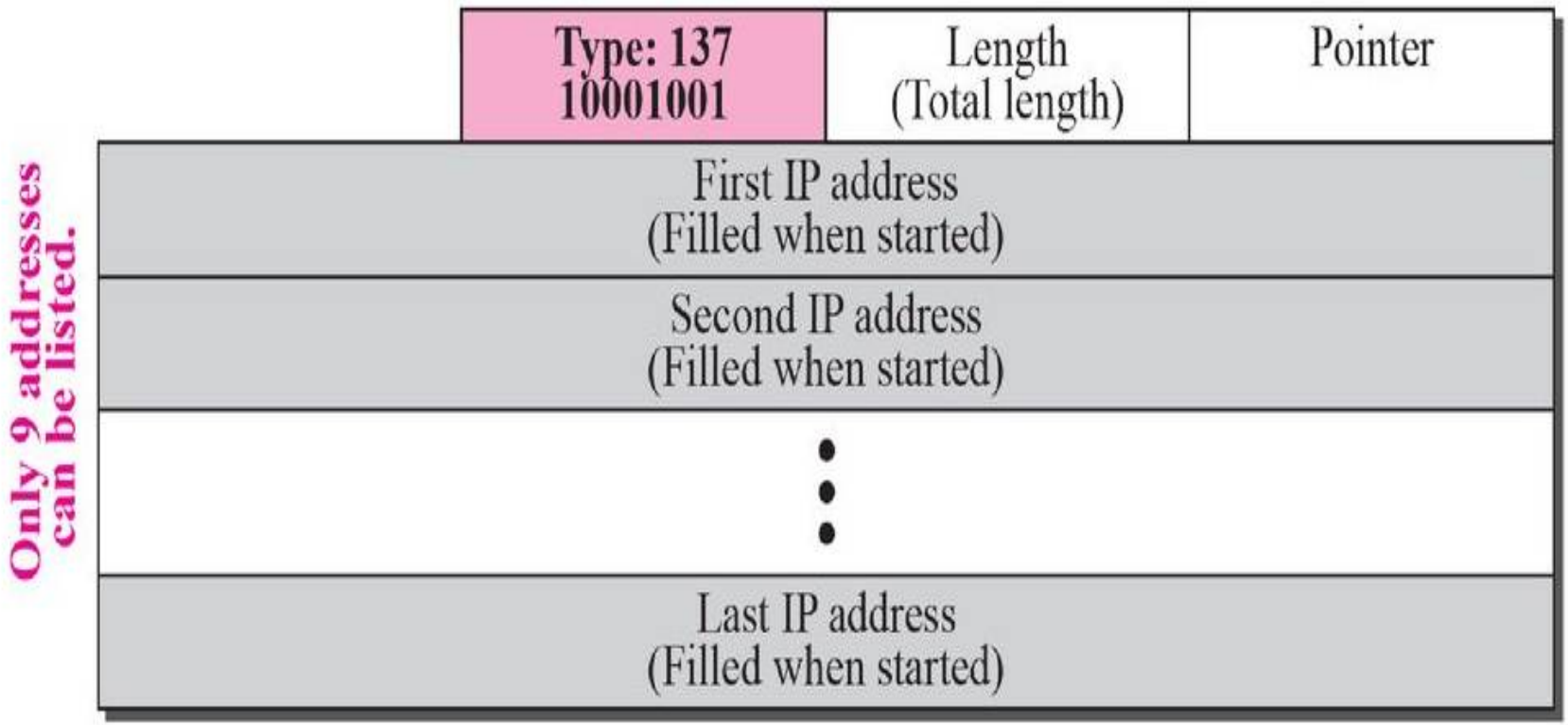
→ A router must not be visited if its IP address is not listed in the datagram.

→If the datagram visits a router that is not on the list, the datagram is discarded and an error message is issued.

→ If the datagram arrives at the destination and some of the entries were not visited, it will also be discarded and an error message issued.

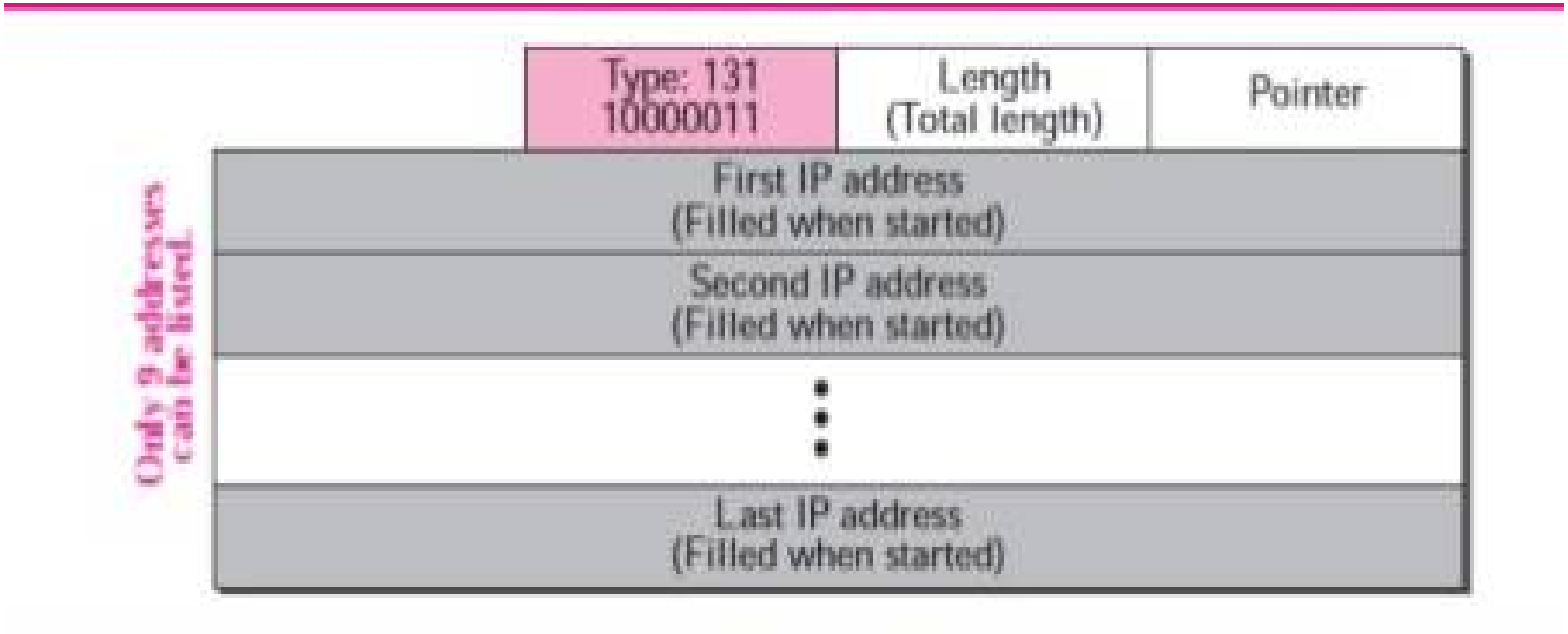
→Regular users of the Internet, however, are not usually aware of the physical topology of the Internet.

Figure : *Strict-source-route option*



Loose-Source-Route Option

→ A loose-source-route option is similar to the strict source route, but it is more relaxed. Each router in the list must be visited, but the datagram can visit other routers as well. Figure shows the format of the loose source route option.



Timestamp

→ A timestamp option is used to record the time of datagram processing by a router.

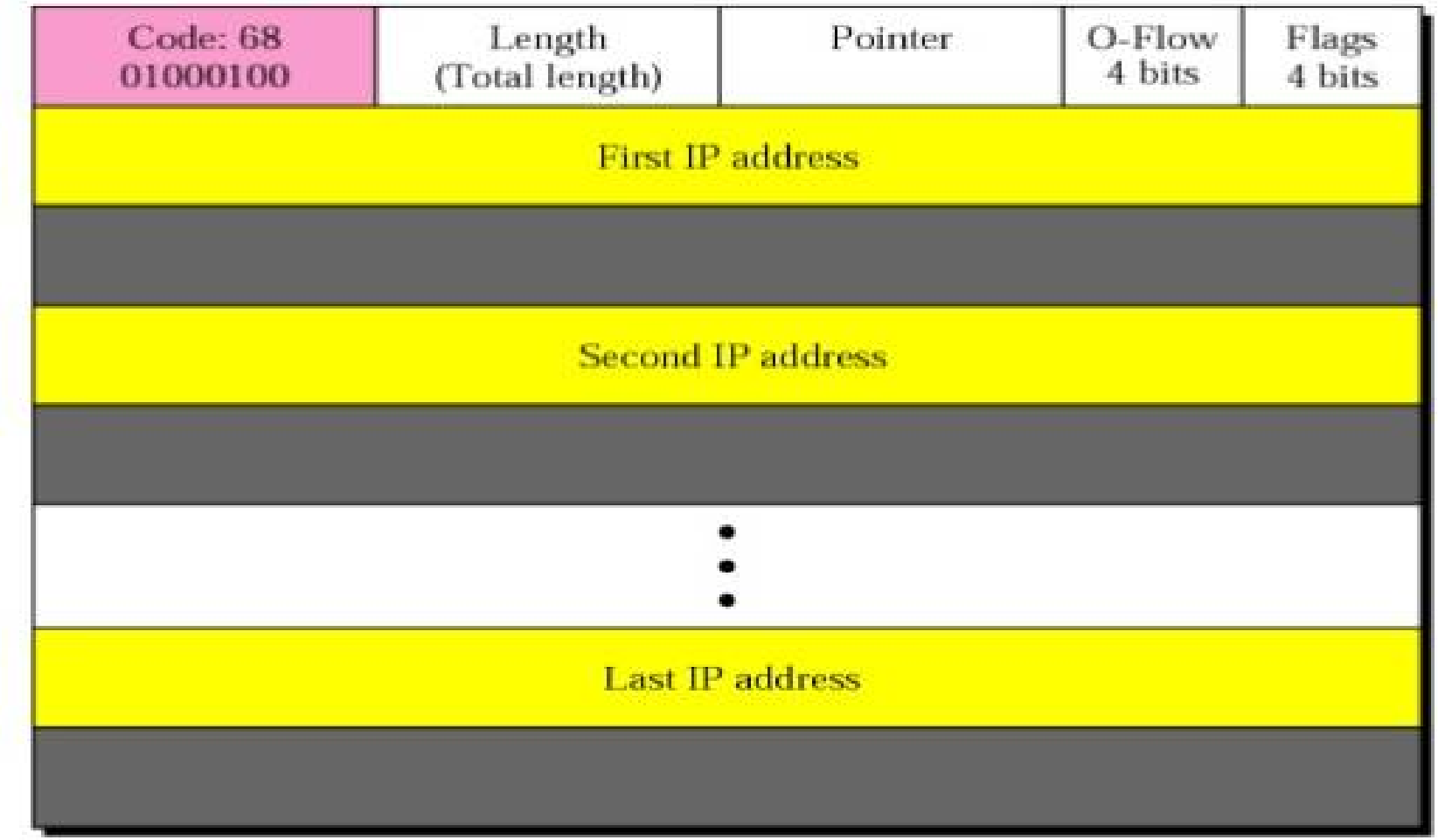
→ The time is expressed in milliseconds from midnight, Universal Time.

→ Knowing the time a datagram is processed can help users and managers track the behavior of the routers in the Internet.

→We can estimate the time it takes for a datagram to go from one router to another.

→ We say estimate because, although all routers may use Universal Time, their local clocks may not be synchronized.

Figure : *Timestamp option*



Cont…

→The overflow field records the number of routers that could not add their timestamp because no more fields were available.

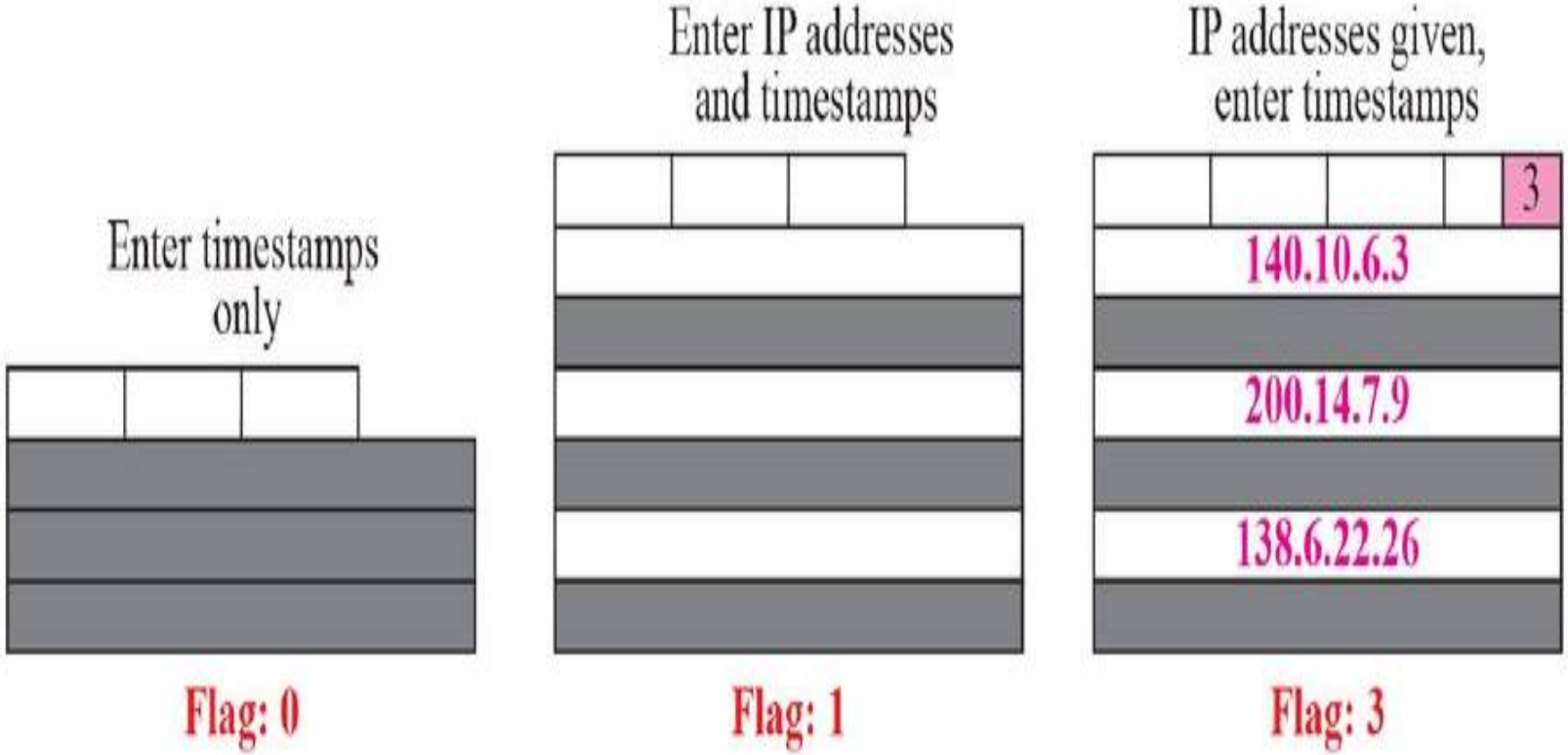
→The flags field specifies the visited router responsibilities.

→ If the flag value is 0, each router adds only the timestamp in the provided field. If the flag value is 1, each router must add its outgoing IP address and the timestamp.

→ If the value is 3, the IP addresses are given, and each router must check the given IP address with its own incoming IP address.

→ If there is a match, the router overwrites the IP address with its outgoing IP address and adds the timestamp.

Figure : *Use of flag in timestamp*



Example :

Which of the six options must be copied to each fragment?

Solution

We look at the first (left-most) bit of the type for each option.
a. No operation: type is 00000001; not copied.
b. End of option: type is 00000000; not copied.
c. Record route: type is 00000111; not copied.
d. Strict source route: type is 10001001; copied.
e. Loose source route: type is 10000011; copied.
f. Timestamp: type is 01000100; not copied.

2.8 CHECKSUM

→The error detection method used by most TCP/IP protocols is called the **checksum.**

→The checksum protects against the corruption that may occur during the transmission of a packet.

→ It is redundant information added to the packet.

→The checksum is calculated at the sender and the value obtained is sent with the packet.

→ The receiver repeats the same calculation on the whole packet including the checksum.

→ If the result is satisfactory (see below), the packet is accepted; otherwise, it is rejected.

Checksum Calculation at the Sender

→At the sender, the packet header is divided into n-bit sections (n is usually 16).

→ These sections are added together using one's complement arithmetic resulting in a sum that is also n bits long.

→The sum is then complemented (all 0s changed to 1s and all 1s to 0s) to produce the checksum.

Cont…

To create the checksum the sender does the following:

❑ The packet is divided into *k sections, each of n bits.*
❑ All sections are added together using one's complement arithmetic.
❑ The final result is complemented to make the checksum.

Checksum Calculation at the Receiver

→ The receiver divides the received packet into *k sections and adds all sections. It then* complements the result. If the final result is 0, the packet is accepted; otherwise, it is rejected

Cont…

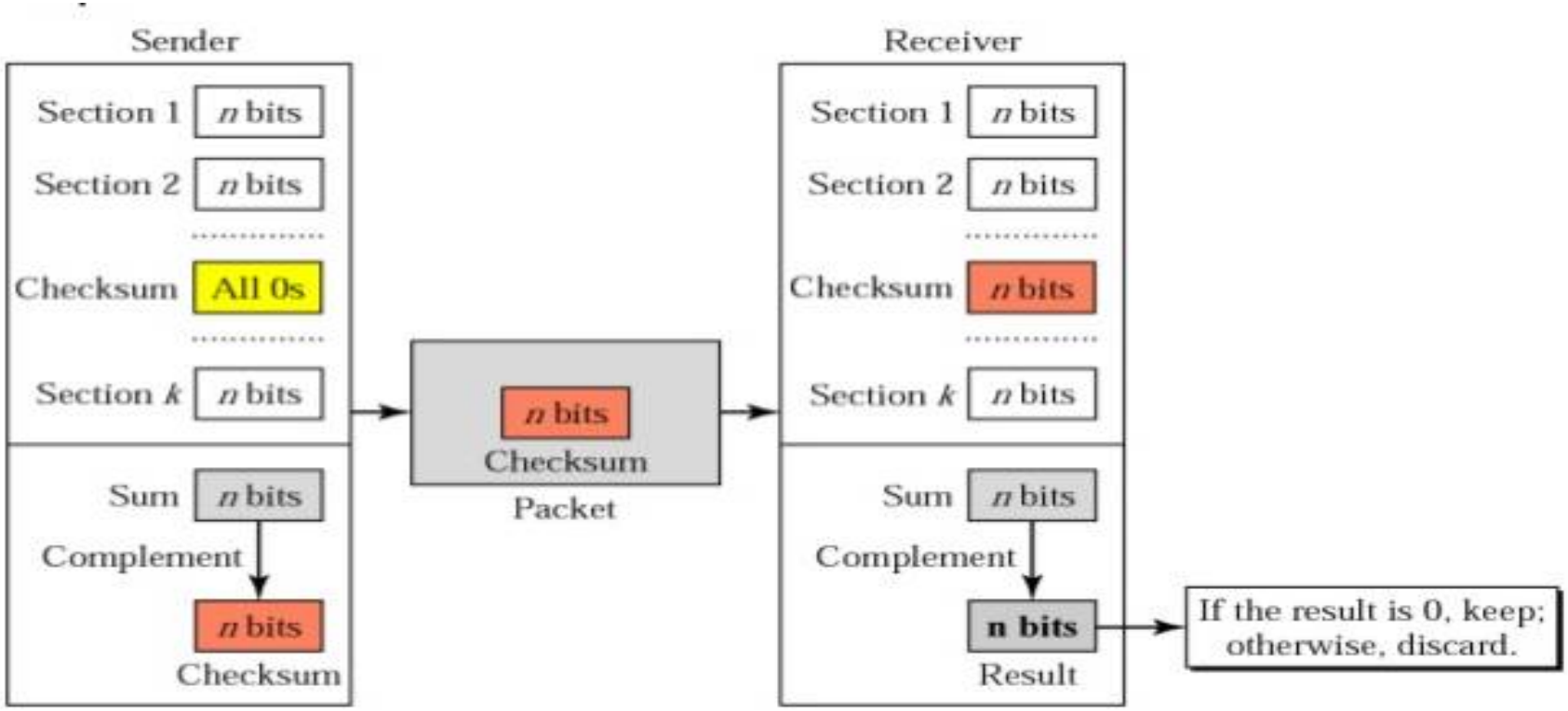
→We said when the receiver adds all of the sections and complements the result, it should get zero if there is no error in the data during transmission or processing.

→Assume that we get a number called T when we add all the sections in the sender.

→ When we complement the number in one's complement arithmetic, we get the negative of the number.

→ This means that if the sum of all sections is T, the checksum is −T.

Figure : *Checksum concept*



→The implementation of the checksum in the IP packet follows the same principles discussed above.

2.9 IP PACKAGE

→This class presents the example of a hypothetical IP package.

→ IP package involves eight components:

(ie) header-adding module, a processing module, a forwarding module, a fragmentation module, a reassembly module, a routing table, an MTU table, and a reassembly table.

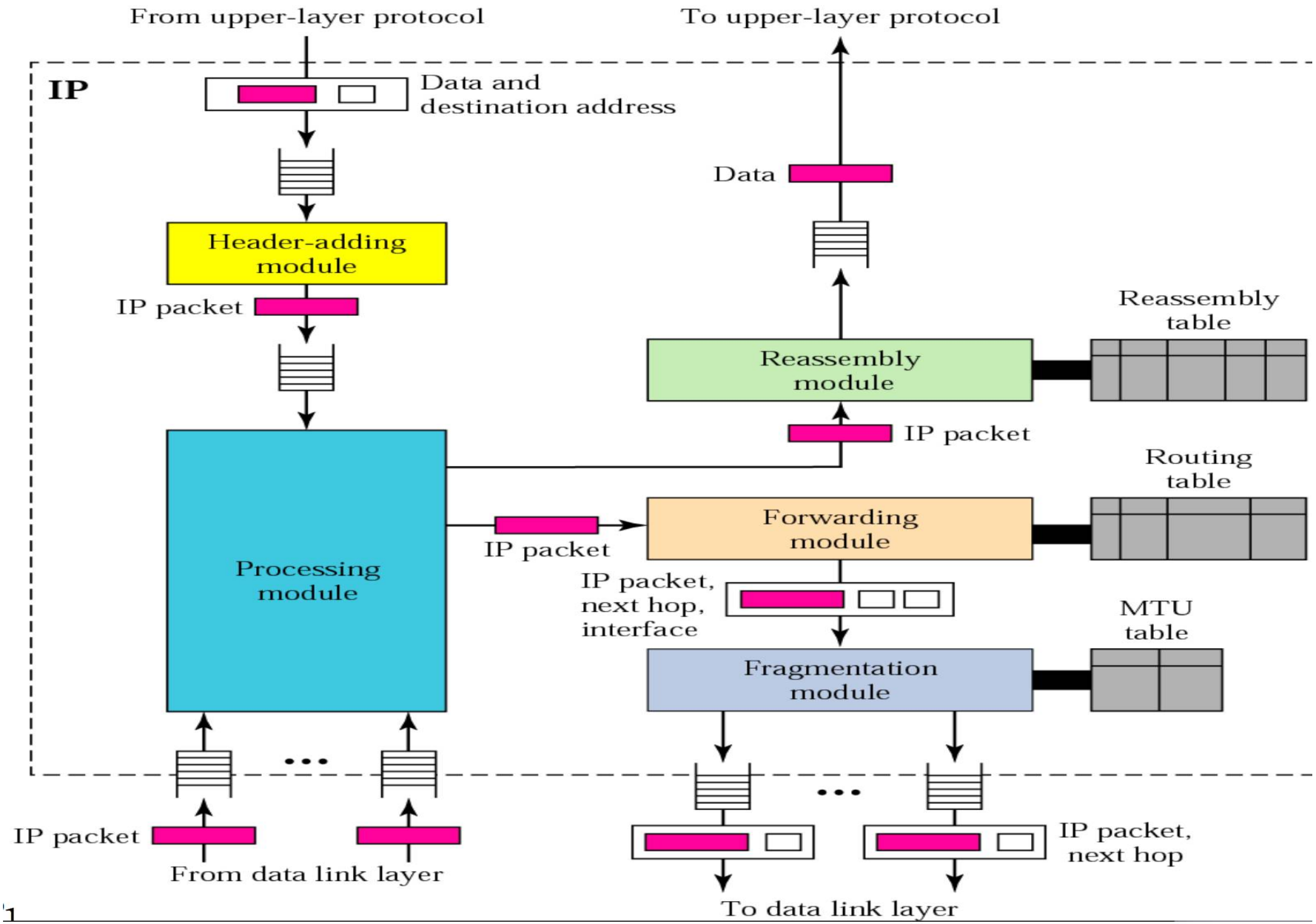
→In addition, the package includes input and output queues.

→The package receives a packet, either from the data link layer or from a higher level protocol.

→ If the packet comes from an upper-layer protocol, it is delivered to the data link layer for transmission.

→ If the packet comes from the data link layer, it is either delivered to the data link layer for forwarding (in a router) or it is delivered to a higher-layer protocol.

Figure : *IP components*



Header-Adding Module

→The header-adding module receives data from an upper-layer protocol along with the destination IP address.

→ It encapsulates the data in an IP datagram by adding the IP header.

```
IP_Adding_Module (data, destination_address)
 {
Encapsulate data in an IP datagram
 Calculate checksum and insert it in the checksum field
 Send data to the corresponding queue
 Return
 }
```

Processing Module

→The processing module (Table) is the heart of the IP package.

→ The processing module receives a datagram from an interface or from the header-adding module. It treats both cases the same.

→A datagram must be processed and routed regardless of where it comes from.

→The processing module first checks to see if the datagram has reached its final destination.

→ In this case, the packet is sent to the reassembly module.If the node is a router, it decrements the time-to-live (TTL) field by one.

→If this value is less than or equal to zero, the datagram is discarded and an ICMP message is sent to the original sender.

→ If the value of TTL is greater than zero after decrement,the processing module sends the datagram to the forwarding module.

Queues

→IP package uses two types of queues: (ie) input queues and output queues.

→The input queues store the datagrams coming from the data link layer or the upper-layer protocols.

→The output queues store the datagrams going to the data link layer or the upper layer protocols.

→ The processing module dequeues (removes) the datagrams from the input queues.

→ The fragmentation and reassembly modules enqueue (add) the datagrams into the output queues.

Routing Table :

→The routing table is used by the forwarding module to determine the next-hop address of the packet.

Forwarding Module :

→The forwarding module receives an IP packet from the processing module.

→ If the packet is to be forwarded, it is passed to this module.

→ The module finds the IP address of the next station along with the interface number to which the packet should be sent.

→ It then sends the packet with this information to the fragmentation module.

MTU Table :

→ The MTU table is used by the fragmentation module to find the **maximum transfer unit (MTU)** of a particular interface.

→ It can have only **two columns: interface and MTU.**

Fragmentation Module :

→In IP package, the fragmentation module receives an IP datagram from the forwarding module.

→The forwarding module gives the IP datagram and the IP address of the next station (either the final destination in a direct delivery or the next router in an indirect delivery), and the interface number through which the datagram is sent out.

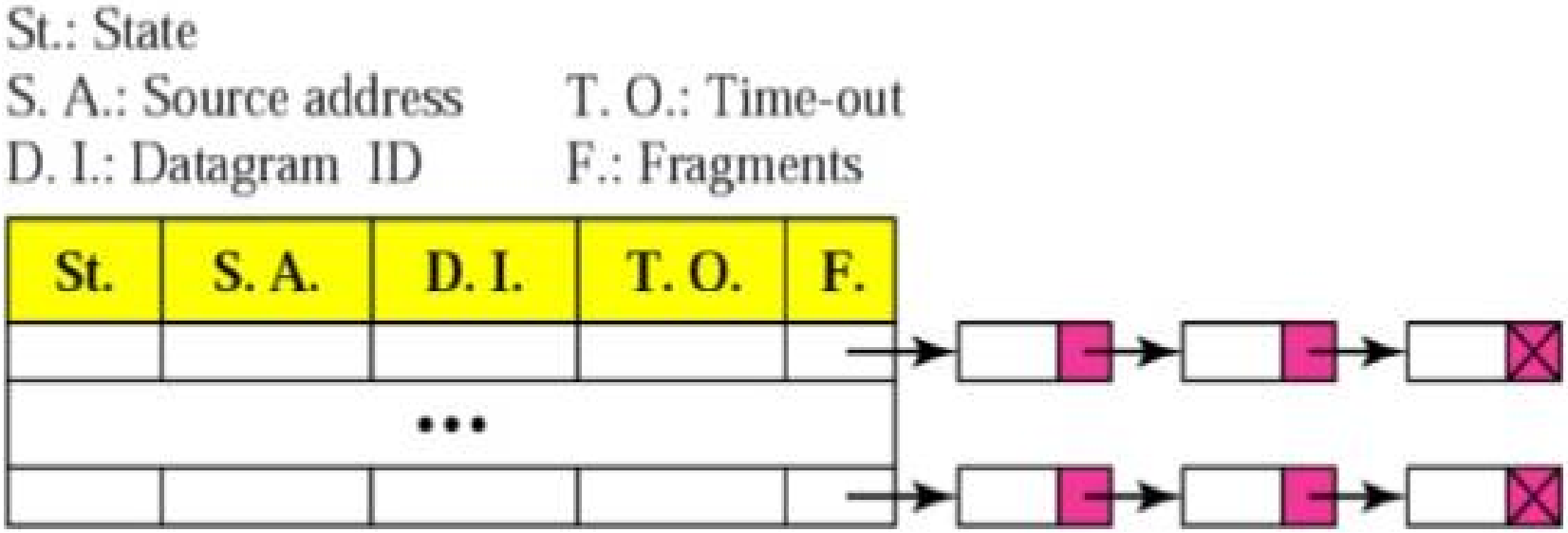
→The fragmentation module consults the MTU table to find the MTU for the specific interface number.

→ If the length of the datagram is larger than the MTU, the fragmentation module fragments the datagram, adds a header to each fragment, and sends them to the ARP package for address resolution and delivery.

Reassembly Table

→The reassembly table is used by the reassembly module. In our package, the reassembly table has **five fields:**

(ie) **state, source IP address, datagram ID, time-out, and fragments**



Cont…

→The value of the state field can be either FREE or IN-USE.

→ The IP address field defines the source IP address of the datagram.

→ The datagram ID is a number that uniquely defines a datagram and all of the fragments belonging to that datagram.

→ The time-out is a predetermined amount of time in which all fragments must arrive.

→ Finally, the fragments field is a pointer to a linked list of fragments.

Reassembly Module

→The reassembly module (Table 7.6) receives, from the processing module, those datagram fragments that have arrived at their final destinations.

→ In our package, the reassembly module treats an unfragmented datagram as a fragment belonging to a datagram with only one fragment.

→because the IP protocol is a connectionless protocol, there is no guarantee that the fragments arrive in order.

→ Besides, the fragments from one datagram can be intermixed with fragments from another datagram.

Cont…

→To keep track of these situations, the module uses a reassembly table with associated linked lists, as we described earlier.

→ The job of the reassembly module is to find the datagram to which a fragment belongs, to order the fragments belonging to the same datagram, and reassemble all fragments of a datagram when all have arrived.

→ If the established time-out has expired and any fragment is missing, the module discards the fragments.

2.10 Internet Control Message Protocol (ICMP)

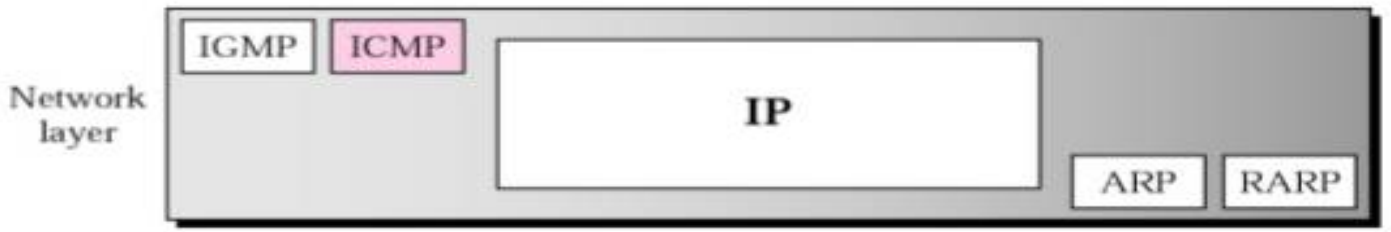
OBJECTIVES :

- ❑ **ICMP messages are divided into two categories:**

  i. Error reporting ii. Query messages.

- ❑ To discuss the purpose and format of error-reporting messages and format of query messages.
- ❑ To show how the checksum is calculated for an ICMP message.
- ❑ To show how debugging tools using the ICMP protocol.

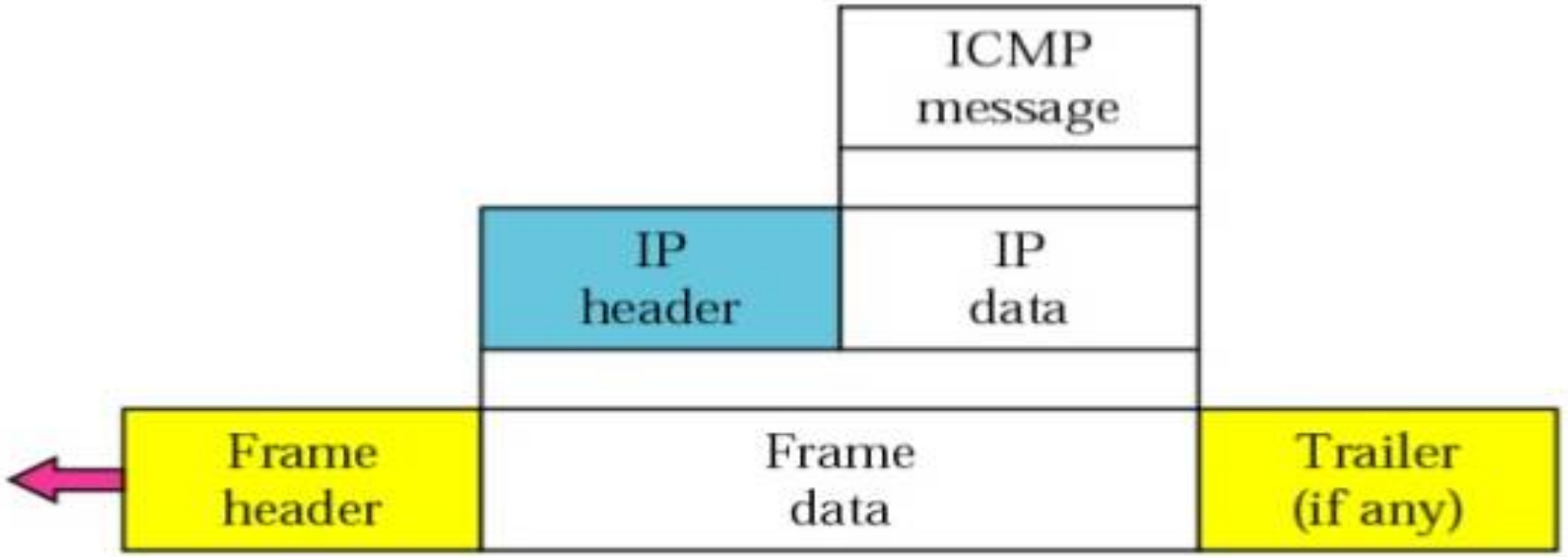
Figure : *Position of ICMP in the network layer*



→ICMP is a network layer protocol. However, its messages are not passed directly to the data link layer.

→ Instead, the messages are first encapsulated inside IP datagrams before going to the lower layer.

Figure : *ICMP encapsulation*



- The value of the protocol field in the IP datagram is 1 to indicate that the IP data is an ICMP message.

2.11 Messages

- **ICMP messages are divided into two broad categories: error-reporting messages and query messages.**

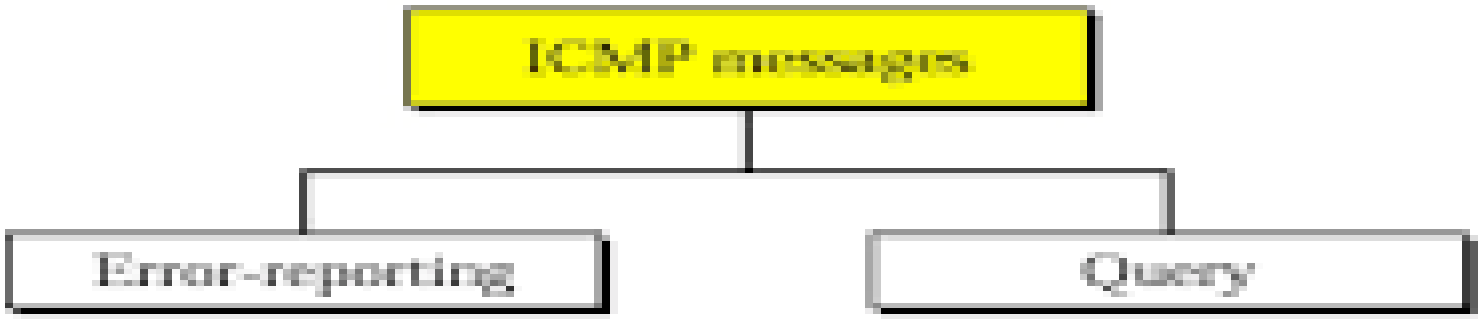
- The **error-reporting messages** report problems that a router or a host (destination) may encounter when it processes an IP packet.

- The **query messages**, help a host or a network manager get specific information from a router or another host.

- For example, nodes can discover their neighbors. Also, hosts can discover and learn about routers on their network and routers can help a node redirect its messages.

Table : *ICMP messages*

- Table lists the ICMP messages in each category.



2.12 Message Format

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

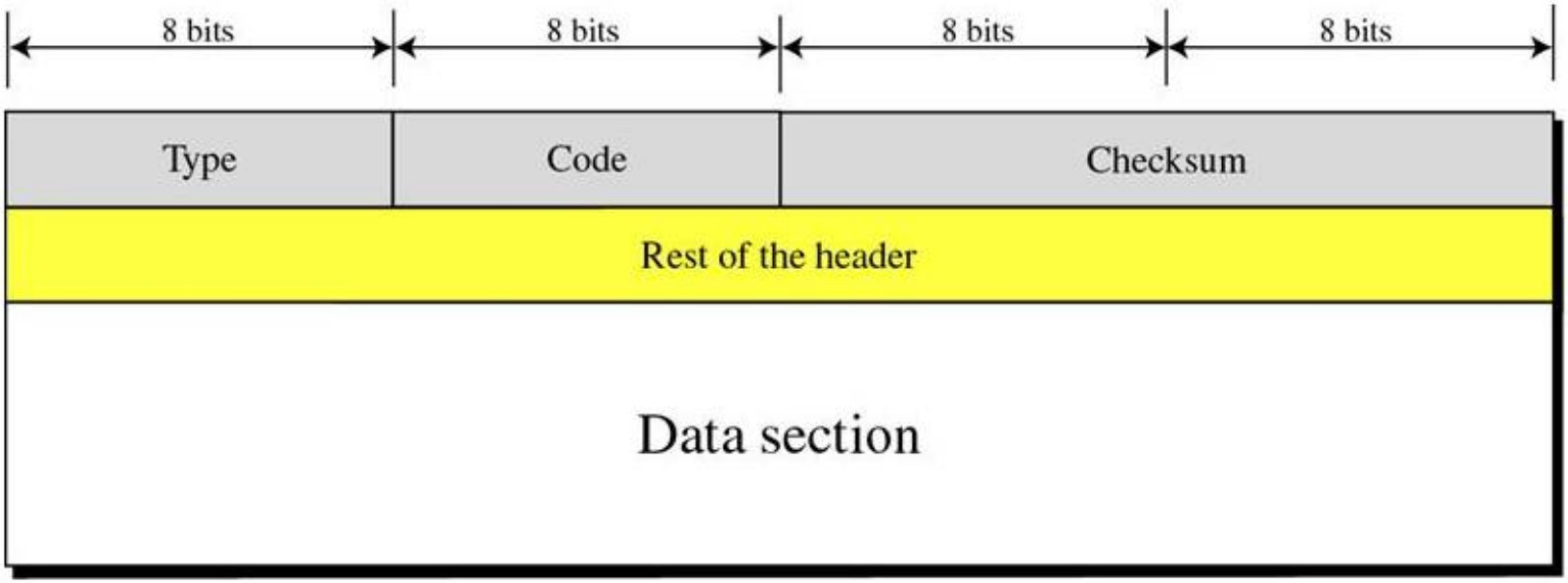
→ Although the general format of the header is different for each message type, the first 4 bytes are common to all.

→As Figure shows, the first field, ICMP type, defines the type of the message.

→ The code field specifies the reason for the particular message type.

→ The last common field is the checksum field , the rest of the header is specific for each message type.

Figure : ***General format of ICMP messages***



2.13 Error Reporting Messages

→One of the main responsibilities of ICMP is to report errors.

→ ICMP does not correct errors, it simply reports them.

→Error correction is left to the higher-level protocols.

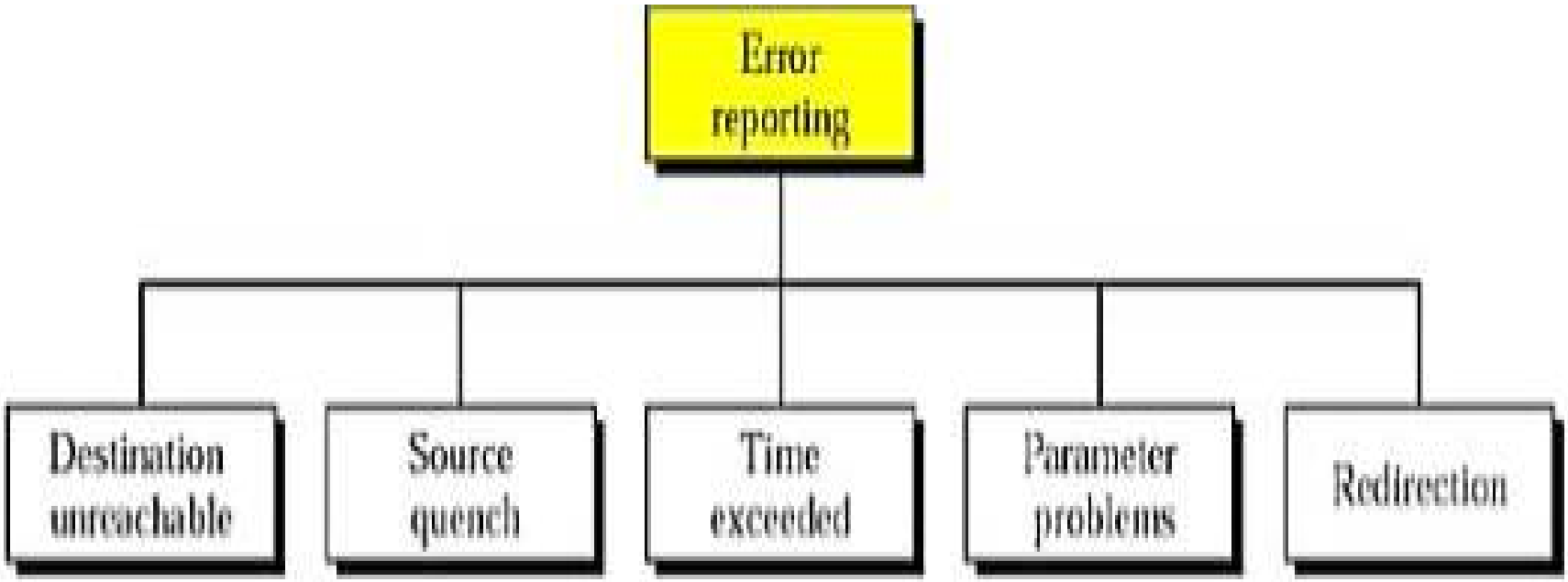
→Error messages are always sent to the original source because the only information available in the datagram about the route is the source and destination IP addresses.

→ ICMP uses the source IP address to send the error message to the source (originator) of the datagram.

Five types of errors are handled:

→destination unreachable,
→source quench,
→Time exceeded,
→parameter problems, and
→ redirection

Figure : ***Error-reporting messages***



Cont…

The following are important points about ICMP error messages:

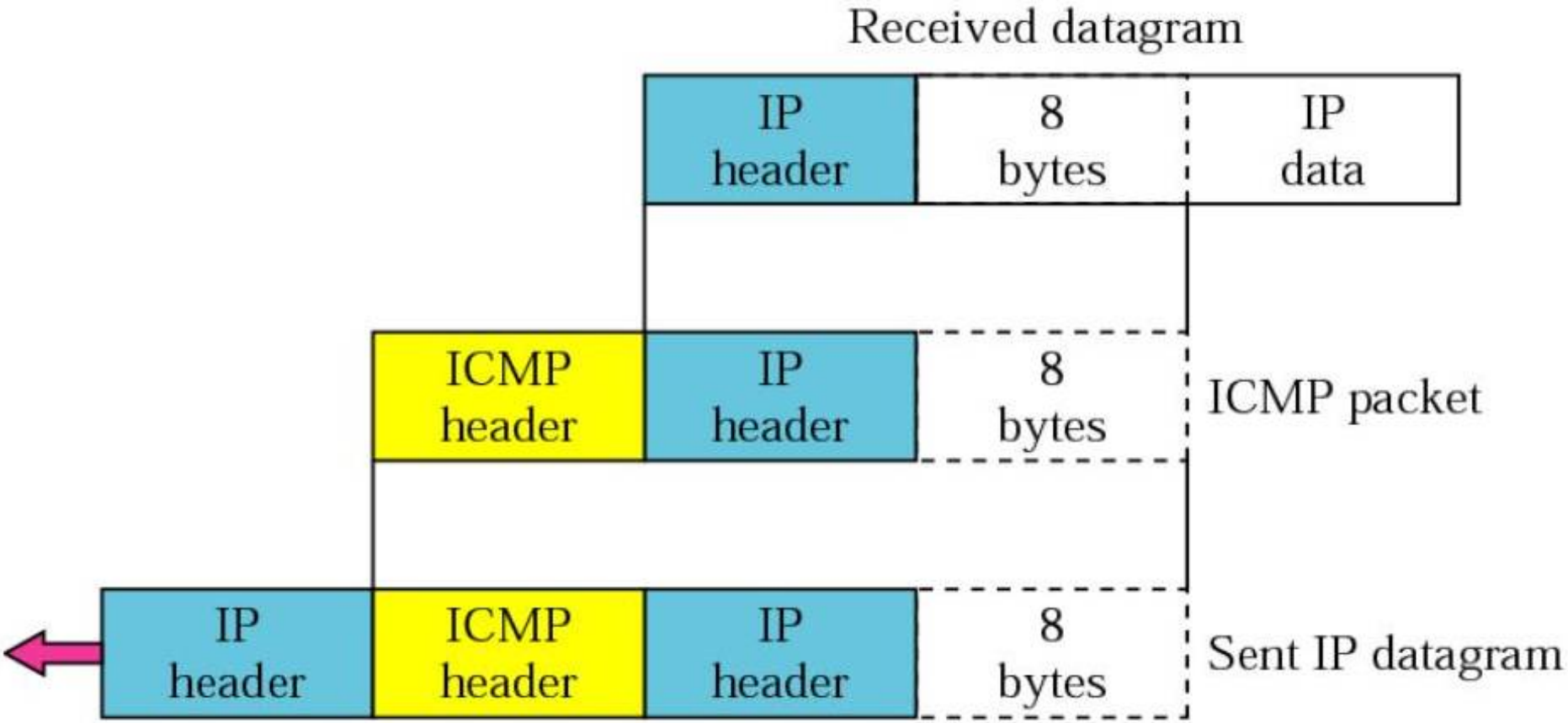
❑ No ICMP error message will be generated in response to a datagram carrying an ICMP error message.

❑ 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.

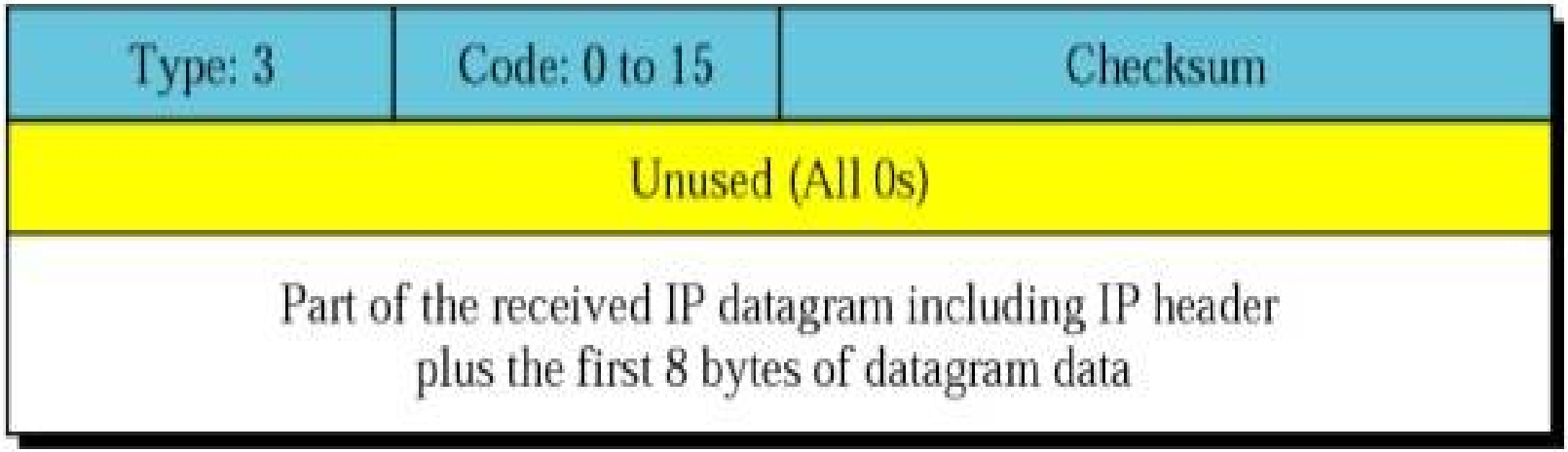
Figure : ***Contents of data field for the error messages***



Destination Unreachable

→When a router cannot route a datagram or a host cannot deliver a datagram. The datagram is discarded and the router or the host sends a **destination-unreachable message** back to the source host that initiated the datagram.

→Figure shows the format of the destination-unreachable message.



The code field for this type specifies the reason for discarding the datagram:

- **Code 0.** The network is unreachable, possibly due to hardware failure. It can be Generated by router.
- **Code 1**. The host is unreachable. This can also be due to hardware failure.
- **Code 2.** The protocol is unreachable. An IP datagram can carry data belonging to higher-level protocols such as UDP, TCP, and OSPF.

→ If the destination host receives a datagram that must be delivered, for example, to the TCP protocol, but the TCP protocol is not running at the moment, a code 2 message is sent.

Cont…

- **Code 3.** The port is unreachable. 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 of the datagram has been set. In other words, the sender of the datagram has specified that the datagram not be fragmented, but routing is impossible without fragmentation.

- **Code 5.** Source routing cannot be accomplished. In other words, one or more routers defined in the source routing option cannot be visited.

Cont…

- **Code 6.** The destination network is unknown. This is different from code 0. In code 0, the router knows that the destination network exists, but it is unreachable at the moment. For code 6, the router has no information about the destination network.

- **Code 7.** The destination host is unknown. This is different from code 1. In code 1, the router knows that the destination host exists, but it is unreachable at the moment. For code 7, the router is unaware of the existence of the destination host.

- **Code 8.** The source host is isolated.

- **Code 9.** Communication with the destination network is administratively prohibited

Cont…

❑ **Code 10.** Communication with the destination host is administratively prohibited.

❑ **Code 11.** The network is unreachable for the specified type of service. This is different from code 0. Here the router can route the datagram if the source had requested an available type of service.

❑ **Code 12.** The host is unreachable for the specified type of service. This is different from code 1. Here the router can route the datagram if the source had requested an available type of service.

❑ **Code 13.** The host is unreachable because the administrator has put a filter on it.

Cont…

❑ **Code 14.** The host is unreachable because the host precedence is violated.
--The message is sent by a router to indicate that 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, but the datagram was sent with a precedence
below this level.

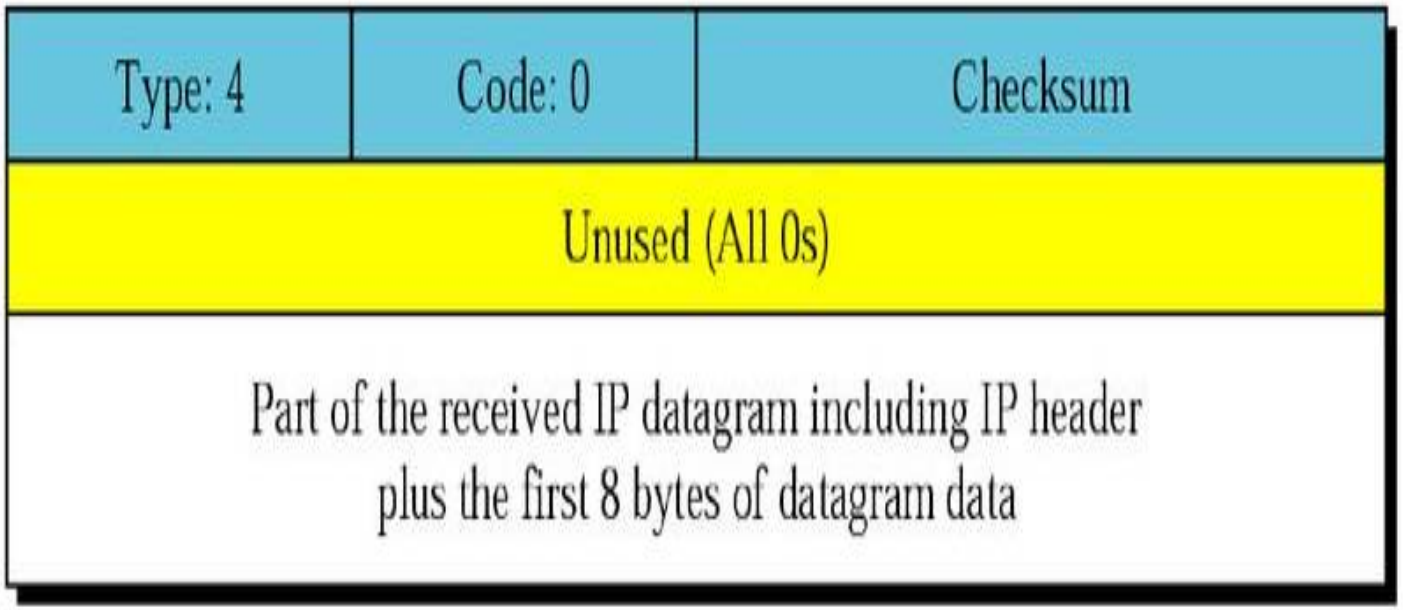
Source Quench

→The IP protocol is a connectionless protocol.

→ There is no communication between the source host, which produces the datagram, the routers, which forward it, and the destination host, which processes it.

→ One of the ramifications of this absence of communication is the lack of *flow control and congestion control.*

Figure : ***Source-quench format***



Time Exceeded

The time-exceeded message is generated in two cases:

→First, the routers use routing tables to find the next hop (next router) that must receive the packet.

→ If there are errors in one or more routing tables, a packet can travel in a loop or a cycle, going from one router to the next or visiting a series of routers endlessly.

→When a datagram visits a router, the value of this field is decremented by 1.

→ When the time-to-live value reaches 0, after decrementing, the router discards the datagram.
→ However, when the datagram is discarded, a time-exceeded message must be sent by the router to the original source.

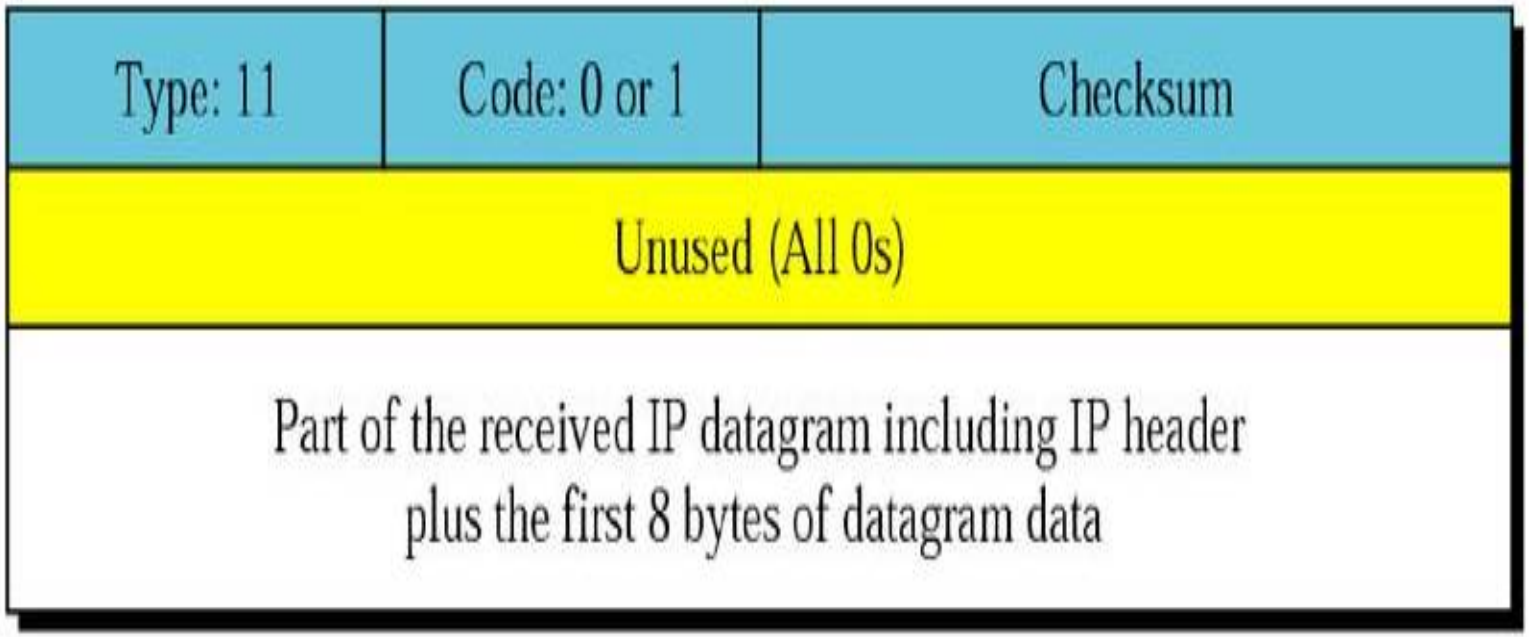
Cont…

→Second, a time-exceeded message is also generated when all fragments that make up a message do not arrive at the destination host within a certain time limit.

→ When the first fragment arrives, the destination host starts a timer.

→If all the fragments have not arrived when the time expires, the destination discards all the fragments and sends a time-exceeded message to the original sender.

Figure : ***Time-exceeded message format***



Parameter Problem

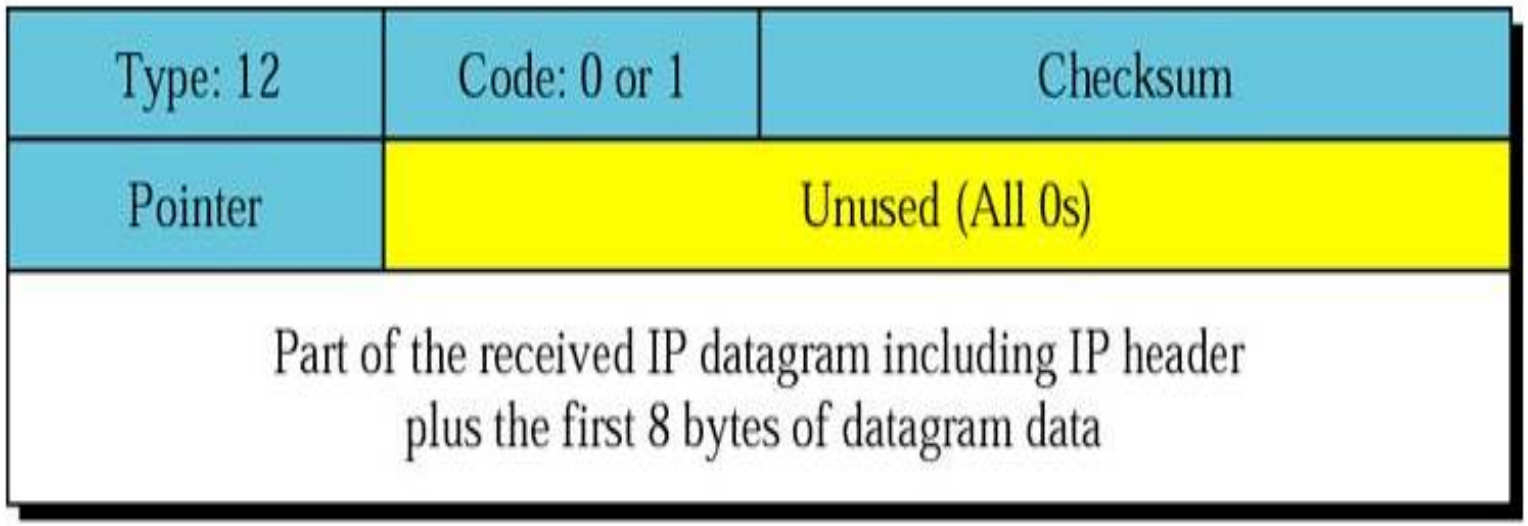
→Any ambiguity in the header part of a datagram can create serious problems as the datagram travels through the Internet.

→ If a router or the destination host discovers an ambiguous or missing value in any field of the datagram, it discards the datagram and sends a parameter-problem message back to the source.

❑ **Code 0.** There is an error or ambiguity in one of the header fields. In this case, the value in the pointer field points to the byte with the problem. For example, if the value is zero, then the first byte is not a valid field.

❑ **Code 1.** The required part of an option is missing. In this case, the pointer is not used.

Figure : ***Parameter-problem message format***



Redirection

→When a router needs to send a packet destined for another network, it must know the IP address of the next appropriate router.

→ The same is true if the sender is a host.

→ Both routers and hosts then must have a routing table to find the address of the router or the next router.

→ Routers take part in the routing update process and are supposed to be updated constantly.

→Routing is dynamic.

Figure : *Redirection concept*

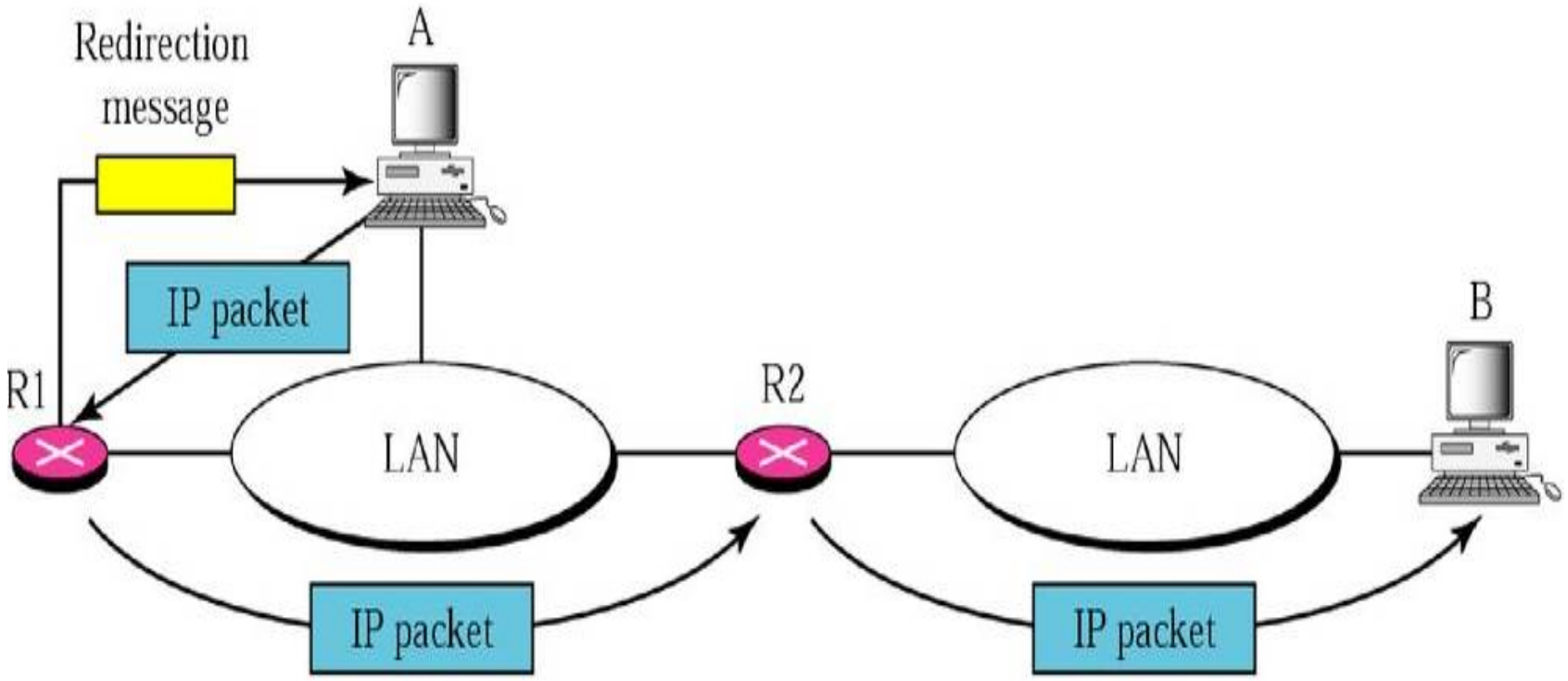
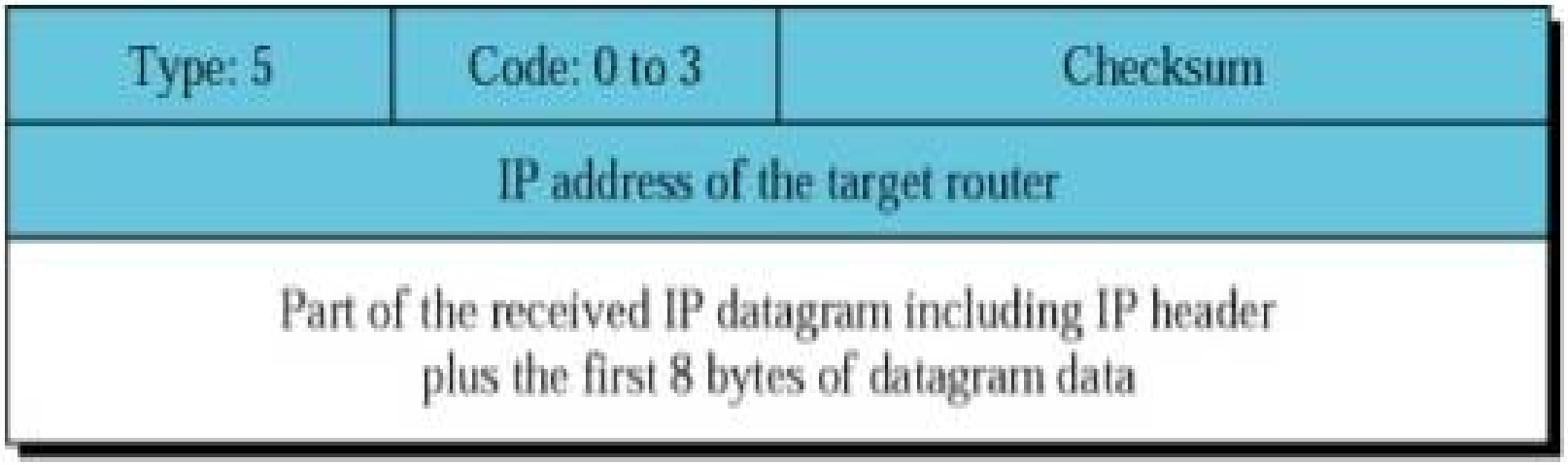


Figure : Redirection message format



Cont…

→ Although the redirection message is considered an error-reporting message, it is different from other error messages.

→ The router does not discard the datagram in this case, it is sent to the appropriate router.

→ The code field for the redirection message narrows down the redirection:

- **Code 0.** Redirection for a network-specific route.
- **Code 1.** Redirection for a host-specific route.
- **Code 2.** Redirection for a network-specific route based on a specified type of service.
- **Code 3.** Redirection for a host-specific route based on a specified type of service.

2.14 Query Messages

→In addition to error reporting, ICMP can also diagnose some network problems.

→ This is accomplished through the query messages.

→ A group of five different pairs of messages have been designed for this purpose.

→ Only two pairs are used today: echo request and replay and timestamp request and replay.

→ In this type of ICMP message, a node sends a message that is answered in a specific format by the destination node.

Echo Request and Reply

→The echo-request and echo-reply messages are designed for diagnostic purposes.

→ Network managers and users utilize this pair of messages to identify network problems.

→The combination of echo-request and echo-reply messages determines whether two systems (hosts or routers) can communicate with each other.

→A host or router can send an echo-request message to another host or router.

→ The host or router that receives an echo-request message creates an echo-reply message and returns it to the original sender

Timestamp Request and Reply

→Two machines (hosts or routers) can use the **t**imestamp-request and timestamp-reply messages to determine the round-trip time needed for an IP datagram to travel between him.

→It can also be used to synchronize the clocks in two machines.

→The three timestamp fields are each 32 bits long.

→ Each field can hold a number representing time measured in milliseconds from midnight in Universal Time (formerly called Greenwich Mean Time).

The formulas are

sending time = receive timestamp − original timestamp
receiving time = returned time − transmit timestamp
round-trip time = sending time + receiving time

→ The sending and receiving time calculations are accurate only if the two clocks in the source and destination machines are synchronized.

→ However, the round-trip calculation is correct even if the two clocks are not synchronized because each clock contributes twice to the round-trip calculation, thus canceling any difference in synchronization.

Cont…

For example, given the following information:

original timestamp: 46 receive timestamp: 59
transmit timestamp: 60 return time: 67

We can calculate the round-trip time to be 20 milliseconds:

sending time= 59− 46 = 13 milliseconds
receiving time= 67− 60= 7 milliseconds
round-trip time = 13 + 7= 20 milliseconds

Deprecated Messages

Three pairs of messages are declared obsolete by IETF:

→ Information request and replay messages are not used today because their duties are done by Address Resolution Protocol (ARP).

→ Address mask request and reply messages are not used today because their duties are done by Dynamic Host Configuration Protocol (DHCP).

→ Router solicitation and advertisment messages are not used today because their duties are done by Dynamic Host Configuration Protocol (DHCP).

2.15 Checksum

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

Checksum Calculation

The sender follows these steps using one’s complement arithmetic:

1. The checksum field is set to zero.
2. The sum of all the 16-bit words (header and data) is calculated.
3. The sum is complemented to get the checksum.
4. The checksum is stored in the checksum field.

Checksum Testing

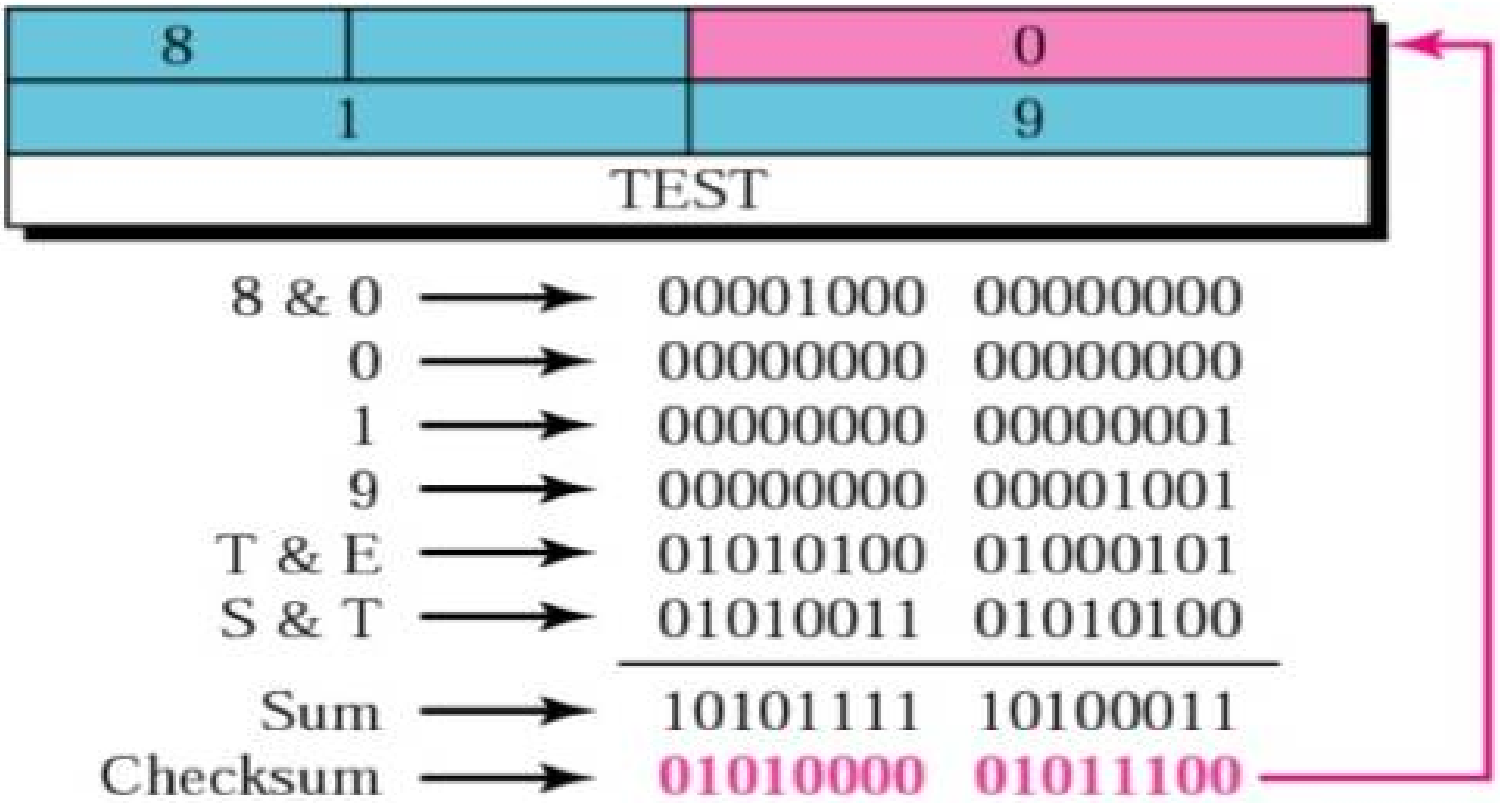
The receiver follows these steps using one's complement arithmetic:

1. The sum of all words (header and data) is calculated.

2. The sum is complemented.

3. If the result obtained in step 2 is 16 0s, the message is accepted; otherwise, it is rejected.

Figure : ***Example of checksum calculation***



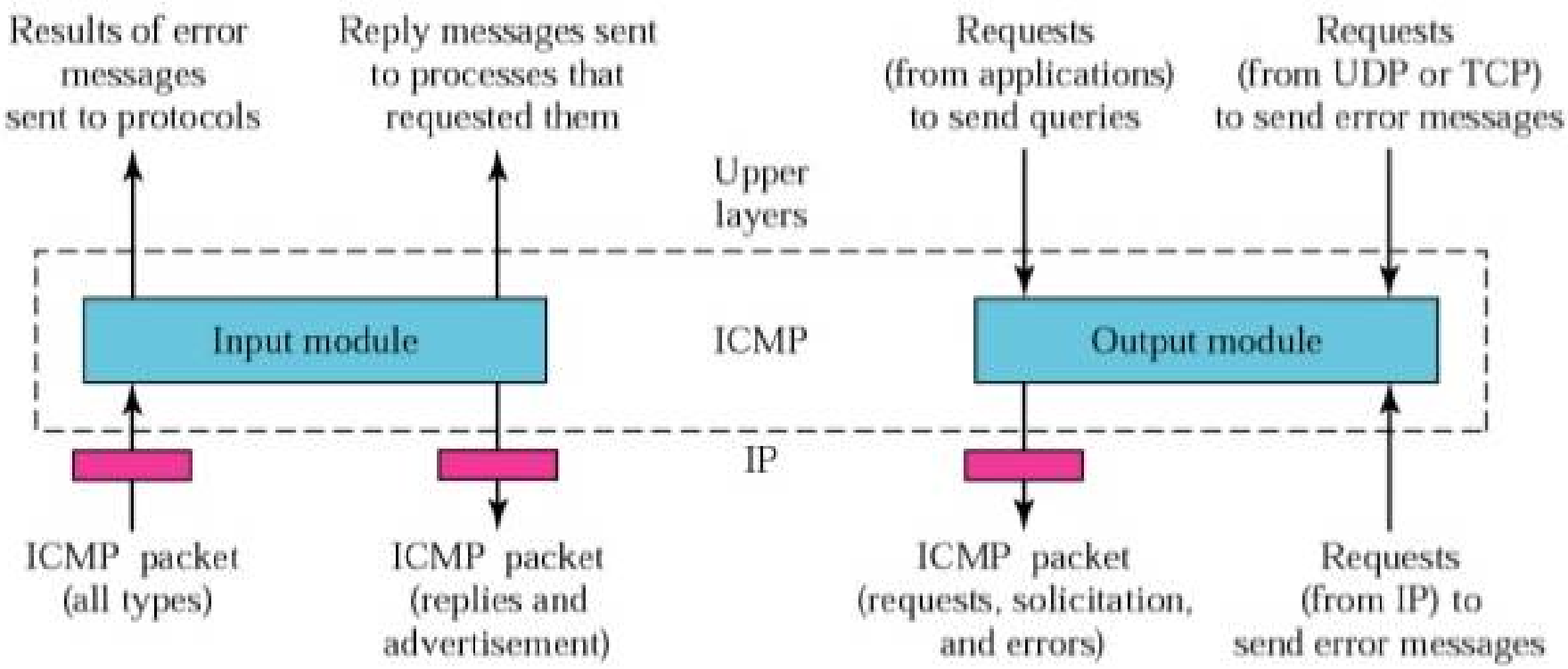
2.16 ICMP PACKAGE

→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.

→Figure shows these two modules.

Figure : *ICMP package*



Input Module

→The input module handles all received ICMP messages. It is invoked when an ICMP packet is delivered to it from the IP layer.

→ If the received packet is a request, the module creates a reply 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.

The pseudocode is shown below:

Output Module

→The output module is 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, the output module must first check that the request is allowed.

→ Remember, an ICMP message cannot be created for four situations,
→ An IP packet carrying an ICMP error message,
→ A fragmented IP packet,
→ A multicast IP packet, or
→ An IP packet having IP