ELECTIVE –II - TCP/IP(18MIT35E)

UNIT-III: Group Management – IGMP Message: IGMP operation – Process to Process Communication – UDP Operation – TCP services – Flow control – Multicast Routing: Multicast routing protocols. Bootp& DHCP – Booth – UDP Ports – using TFTP – Dynamic host Configuration Protocols (DHCP) – Domain Name system (DNS) – Name Space – Domain Name Space – distribution of Name space – DNS in the Internet – Resolution – DNS Message – Types of records.

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

3.1 Group Management

- **IGMP(Internet Group Management Protocol)** is not a multicasting routing protocol; it is a protocol that manages *group membership.*

- In any network, there are one or more multicast routers that distribute multicast packets to hosts or other routers.

- The IGMP protocol gives the *multicast routers information* about the membership status of hosts (routers) connected to the network.

- A multicast router may receive thousands of multicast packets every day for different groups.

Cont…

- If a router has no knowledge about the membership status of the hosts, it must forward all of these packets.
- This creates a lot of traffic and consumes bandwidth.
- A better solution is to keep a list of groups in the network for which there is at least one loyal member.
- IGMP helps the multicast router create and update this list.
- IGMP has gone through three versions. Versions 1 and 2 provide what is called **Any Source Multicast (ASM).**

Cont

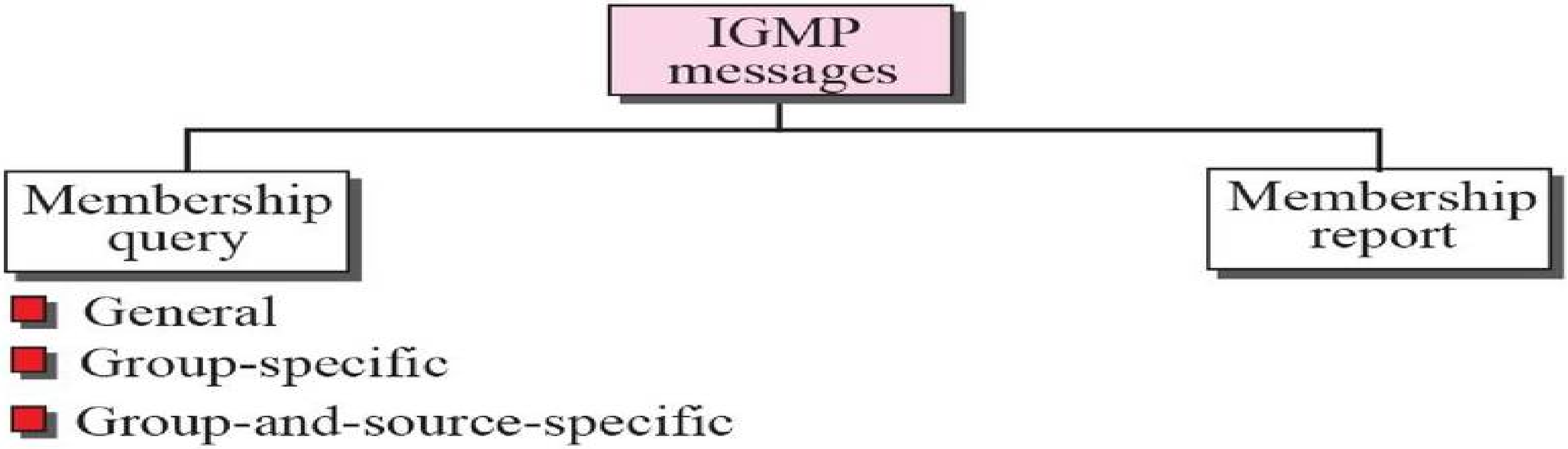
- which means that the group members receive a multicast message no matter where it comes from.

- The IGMP version 3 provides what is called **Source Specific Multicast (SSM).**
- which means that the recipient can choose to receive multicast messages coming from a list of predefined sources.

3.2 IGMP Messages

- IGMPv3 has two types of messages:
  - a) membership query message and
  - b) membership report message.

- The query message can be used in three different formats:
  - i. general,
  - ii. group specific,
  - iii. group-and-source-specific, as shown in Figure 1.



Three Formats of Query Messages

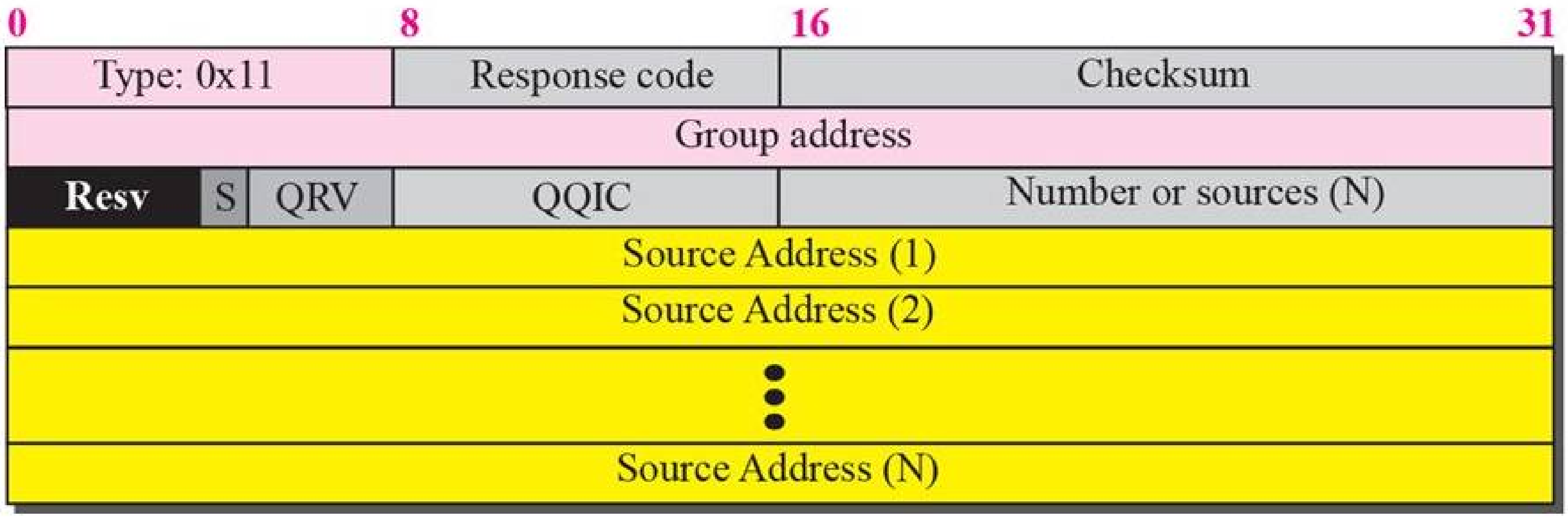
- a. In a **general query message**, the querier router probes each neighbor to report the whole list of its group membership (interest in any multicast group).

- b. In a **group-specific query message**, the querier router probes each neighbor to report if it is still interested in a specific multicast group. The multicast group address is defined as x.y.z.t in the group address field of the query.

- c. In a **group-and-source-specific query message**, the querier router probes each neighbor to report if it is still in a specific multicast group, x.y.z.t, coming from any of the N sources whose unicast addresses are defined in this packet.

a) Membership Query Message Format

- A membership query message is sent by a router to find active group members in the network.
- Figure shows the format of this message.



- ❑ **Type.** This 8-bit field defines the type of the message. The value is 0X11 for a membership query message.

Cont…

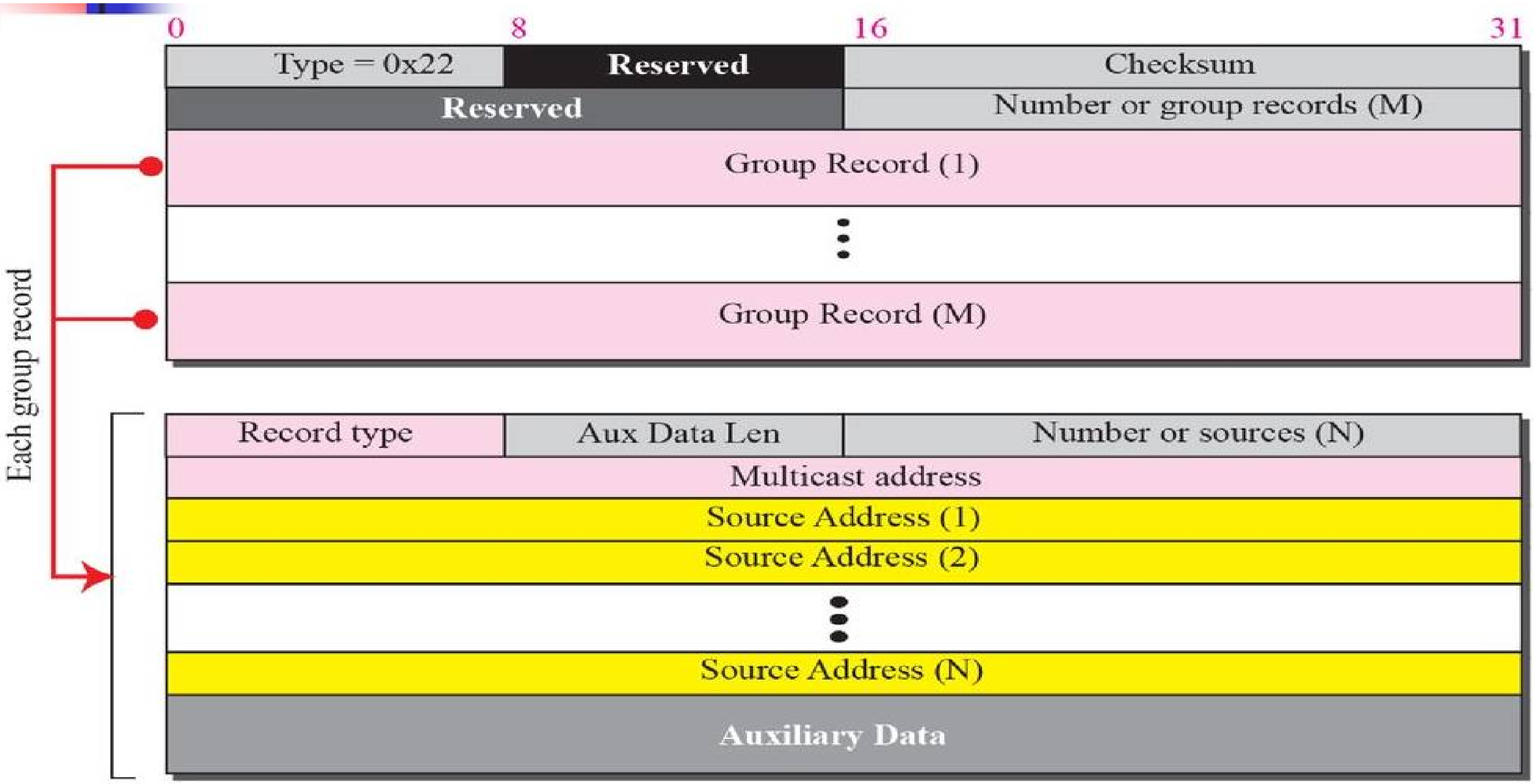
- ❑ **Maximum Response Code.** This 8-bit field is used to define the response time of a recipient of the query.
- ❑ **Checksum.** This is a 16-bit field holding the checksum. The checksum is calculated over the whole IGMP message.
- **Group Address.** This 32-bit field is set to 0 in a general query message; it is set to IP multicast being queried when sending a group-specific or group-and-source specific query message.
- ❑ **Resv.** This 4-bit field is reserved for the future and it is not used.
- ❑ **S.** This is a 1-bit suppress flag. When this field is set to 1, it means that the receivers of the query message should suppress the normal timer updates.

Cont…

- ❑ **QRV.** This 3-bit field is called *querier's robustness variable. It is used to monitor* the robustness in the network.
- ❑ **QQIC.** This 8-bit field is called *querier's query interval code. This is used to calculate* the querier's query interval (QQI).
- ❑ **Number of sources (N).** This 16-bit field defines the number of 32-bit unicast source addresses attached to the query. The value of this field is zero for the general query and the group-specific query, and nonzero in the group-and-source-specific query.
- ❑ **Source Addresses.** These multiple 32-bit fields list the N source addresses, the origin of multicast messages. The value of N is defined in the previous field.

b) Membership Report Message Format

- Figure shows the format of an IGMP membership report message format.



0 8 16 31

Type = 0x22 | Reserved | Checksum

Reserved | Number or group records (M)

Group Record (1)

Group Record (M)

Each group record

Record type | Aux Data Len | Number or sources (N)

Multicast address

Source Address (1)

Source Address (2)

Source Address (N)

Auxiliary Data

- ❑ **Type.** This 8-bit field with the value 0x22 defines the type of the message.

3.3 Process-to-Process Communication

- **Process-to-Process Communication :**
- The IP address is responsible for communication at the computer level (host – to –host communication).
- As a network layer protocol , IP can deliver the message only to the destination computer.
- The message still needs to be handed to the correct process.
- TCP provides process-to-process communication using port numbers.

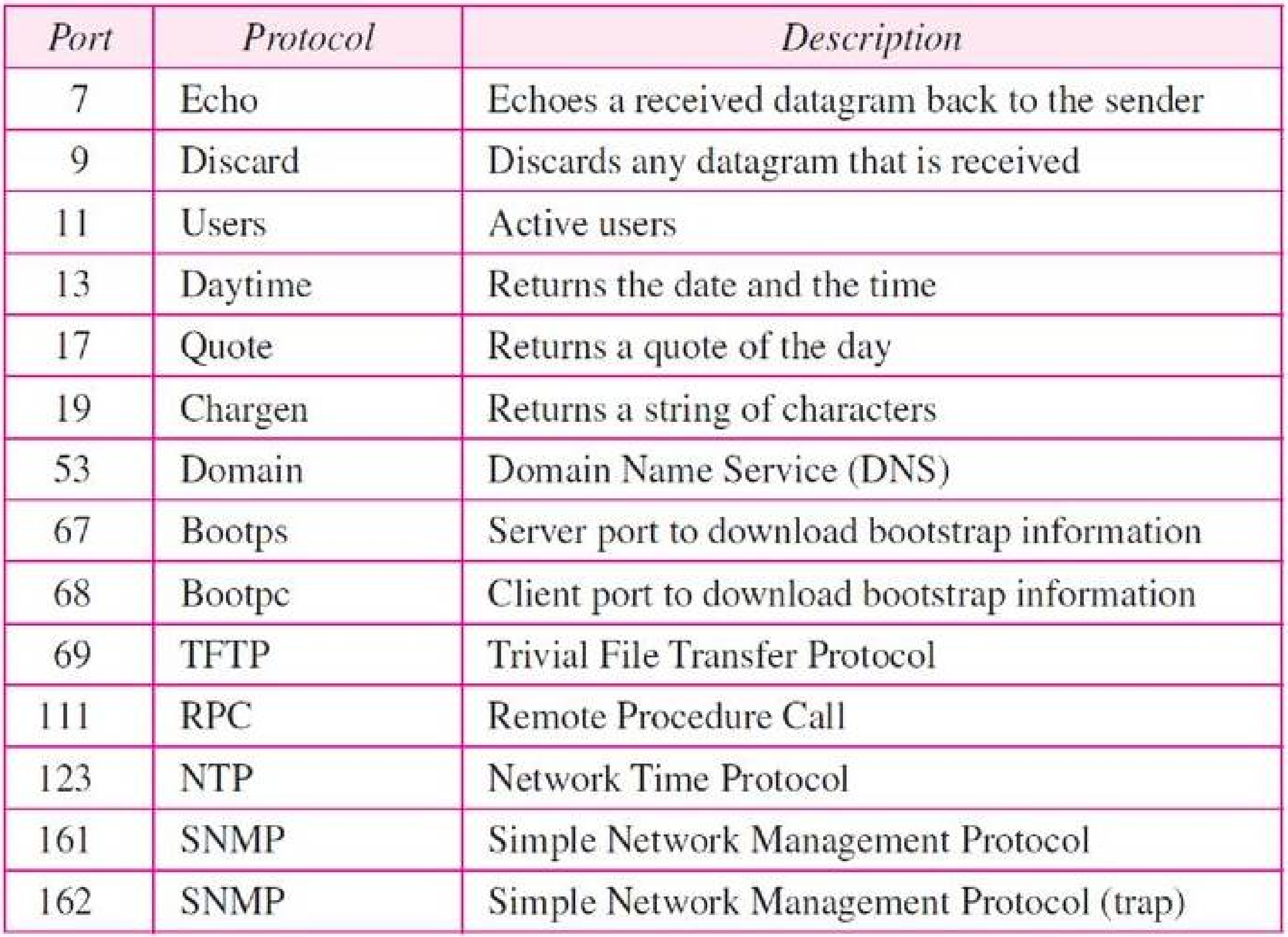
Port numbers

- There are several ways to achieve process to process communication the most common is through the client server paradigm.
- A process on the local host called a **client**.
- A process on the remote host called a **server.**
- Both client and server have the same name.
- A remote computer can run several server programs at the same time , just as several local computer can run or more client program at the same time.
- The local host and the remote host are defined using IP addresses, we need second identifiers **called port numbers.**
- The client program defines itself with a port number, called **ephemeral port number.**

Well –known port numbers

- TCP/IP has decided to use universal port number for server, these are called **well known port numbers.**
- It should be clear IP addresses and port number play different roles in selecting the final destination of data.
- The destination IP address defines the host among the different hosts in the world.
- The following table lists some well-known port numbers used by TCP.

Table : Well-known Ports used by UDP



Port	*Protocol*	*Description*
7	Echo	Echoes a received datagram back to the sender
9	Discard	Discards any datagram that is received
11	Users	Active users
13	Daytime	Returns the date and the time
17	Quote	Returns a quote of the day
19	Chargen	Returns a string of characters
53	Domain	Domain Name Service (DNS)
67	Bootps	Server port to download bootstrap information
68	Bootpc	Client port to download bootstrap information
69	TFTP	Trivial File Transfer Protocol
111	RPC	Remote Procedure Call
123	NTP	Network Time Protocol
161	SNMP	Simple Network Management Protocol
162	SNMP	Simple Network Management Protocol (trap)

ICANN(Internet Corporation for Assigned Names and Numbers) ranges

ICANN has divided the port number into three ranges,

- Well-known ports
- Registered ports
- Dynamic ports

- **Well –known ports(Standard ports) :** The ports ranging from 0 to 1,023 are assigned and controlled by ICANN. These are well known ports.
- **Registered ports:** The ports ranging from 1,024 to 49,151 are assigned or controlled by ICANN.
- **Dynamic ports :** The ports ranging from 49,152 to 65,535 are neither controller nor registered. They can be used as temporary or private port number.

Figure : ICANN ranges

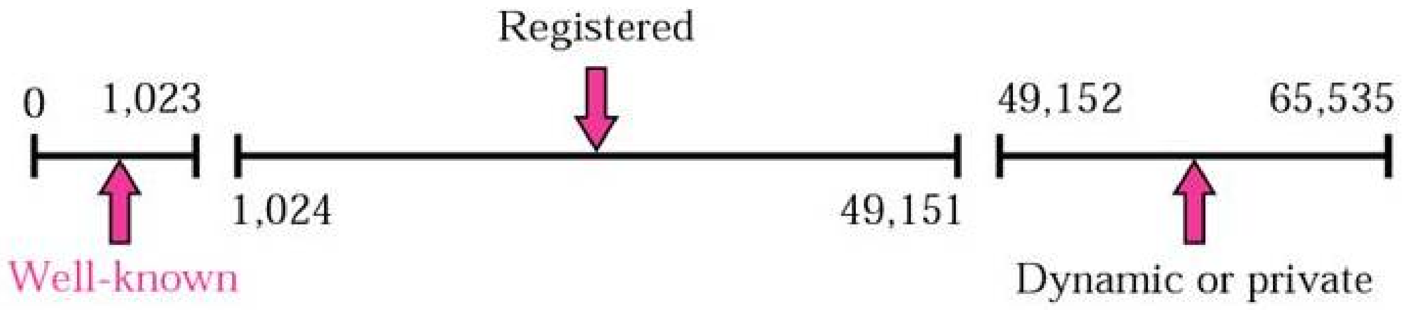
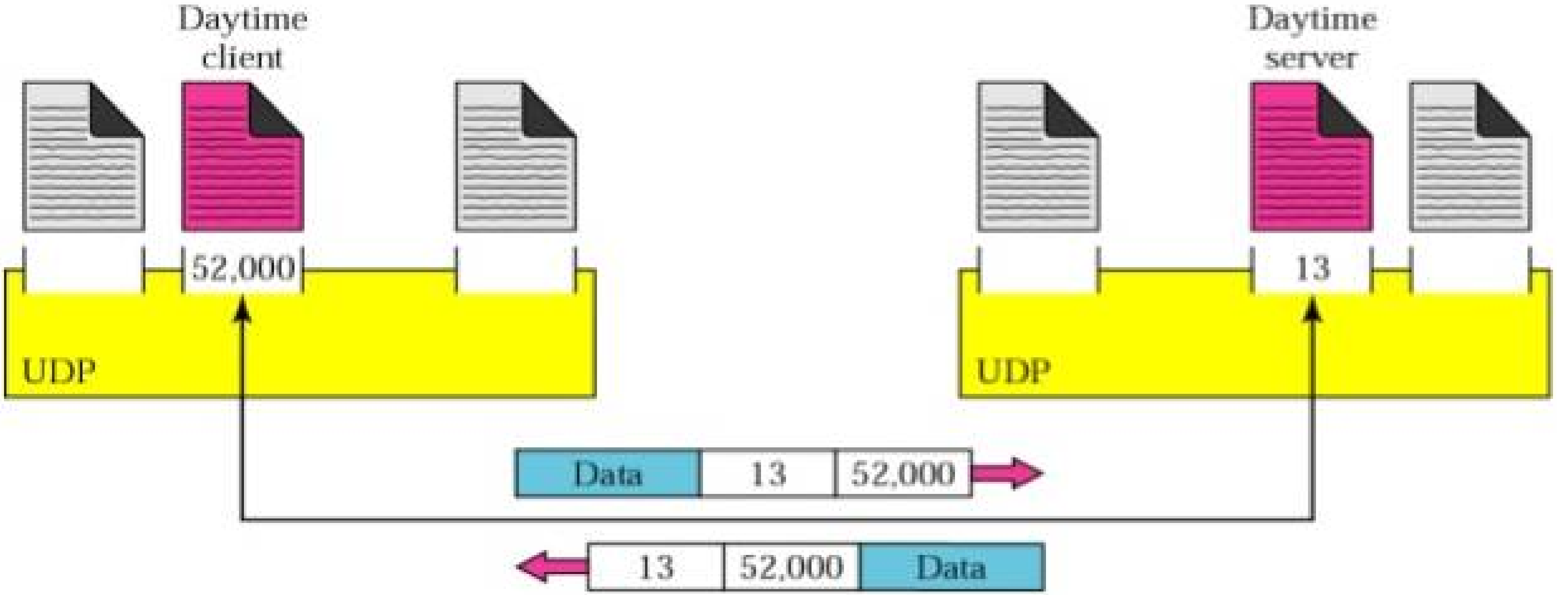


Figure : Port number



3.4 UDP Operation

- UDP users concepts common to the transport layer.
- Below link provide the difference between TCP and UDP protocol

Types of UDP operations are

- Connectionless services
- Flow and Error control
- Encapsulation and Decapsulation
- Queuing

Connectionless services

- UDP provides a connectionless service. This means that each user datagram sent by UDP is an independent datagram.
- There is no relationship between the different user datagrams even if they are coming from the same source process and going to the same destination program.
- The user datagram are not numbered.
- There is no connection establishment and no connection termination as is the case for TCP/IP.

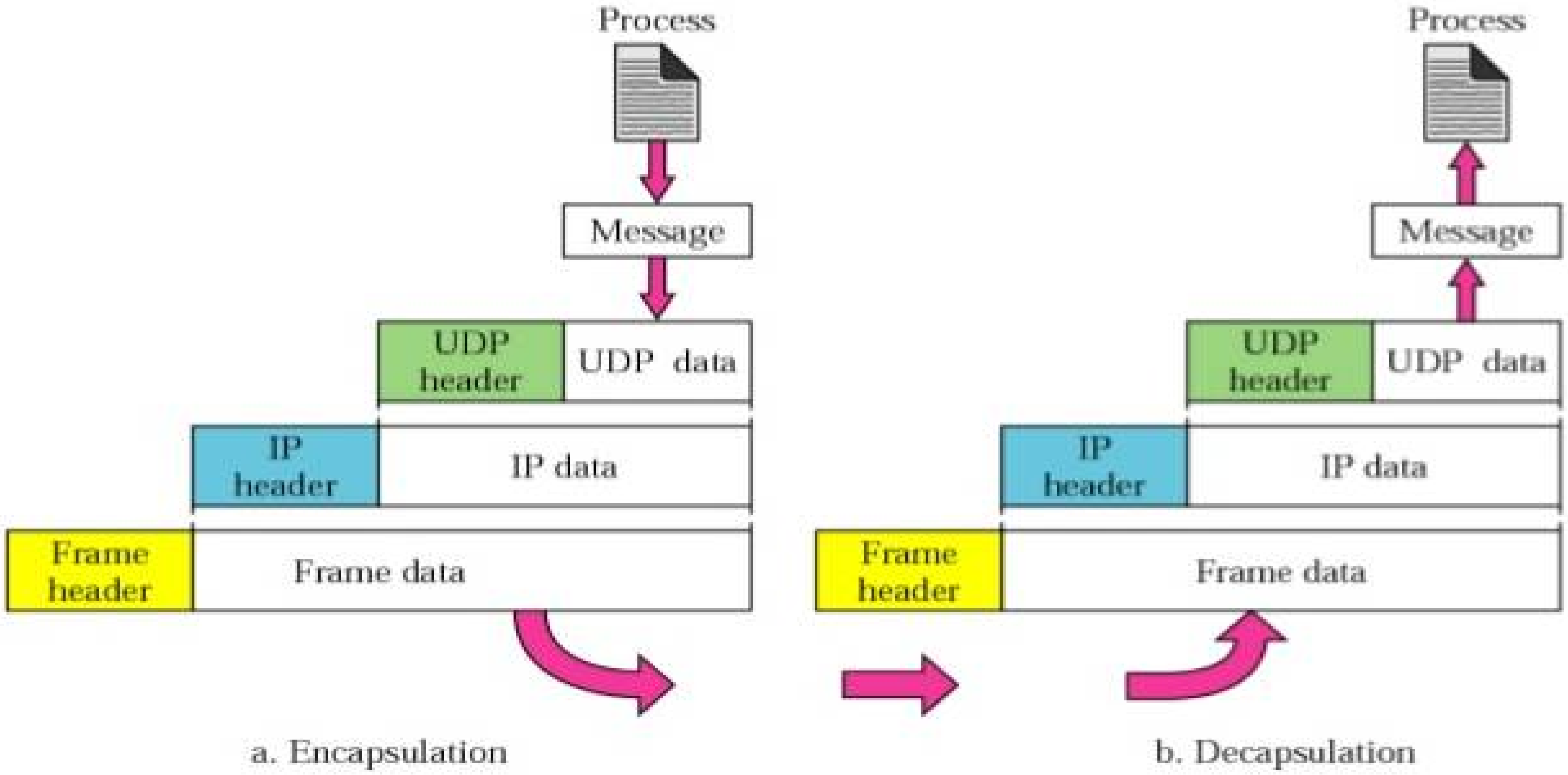
Flow and Error control

- UDP is a very simple , unreliable transport protocol.
- There is no flow control and hence no window mechanism.
- The receiver may overflow with incoming messages.
- This means that the sender does not know if a message has been lost or duplicated.
- UDP protocol is lack in flow control and error control mechanism.

Encapsulation and Decapsulation

- **https://www.youtube.com/watch?v=blV7WUZpkCE&t=240s**
- **Encapsulation :** When a process has a message to send through UDP, it process the message to UDP along with a pair of socket addresses and the length of data.
- **Decapsulation :** When a message arrives at the destination host , the physical layer decodes the signals into bits and passes it to the data link layer.
- The data link layer uses the header to check the data.
- The header and trailer are dropped and the datagram is passed to IP.

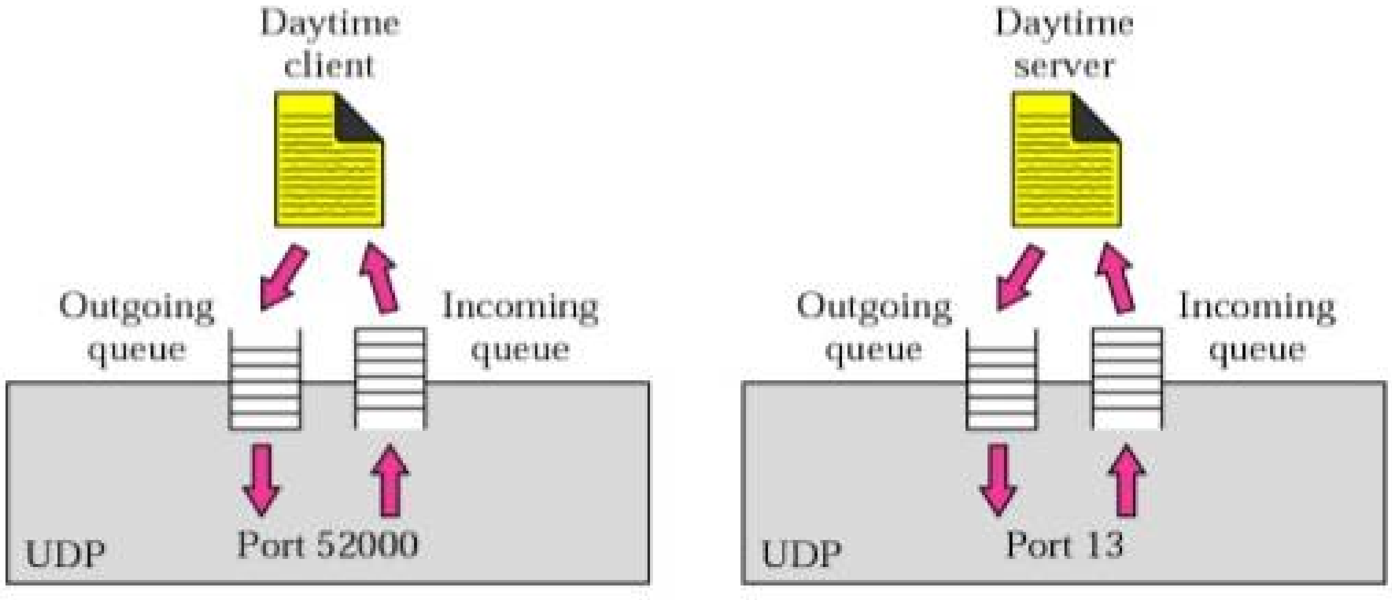
Figure: Encapsulation and Decapsulation



Queuing

- In UDP Queues are associated with ports(see figure).
- At the client site, when a process starts, it requests a port number from the operating system.
- Some implementations create both an incoming and an outgoing queue associated with each process.
- Other implementations create only an incoming queue associated with each process.

Figure: Queues in UDP



3.5 TCP SERVICES

- TCP is a connection oriented protocol.
- TCP lies between the application layer and the network layer, and serves as the intermediary between the application programs and the network operations.
- Its a reliable protocol. It carry data and control information.
- **Features of TCP protocol is as follows:**

A) Process-to-Process Communication
B) Stream Delivery Service
C) Full-Duplex Communication
D) Multiplexing and Demultiplexing
E) Connection-Oriented Service
F) Reliable Service

A) Process-to-Process Communication

- Process-to-Process Communication :

- TCP provides process-to-process communication using port numbers.

- Table 2 lists some well-known port numbers used by TCP.

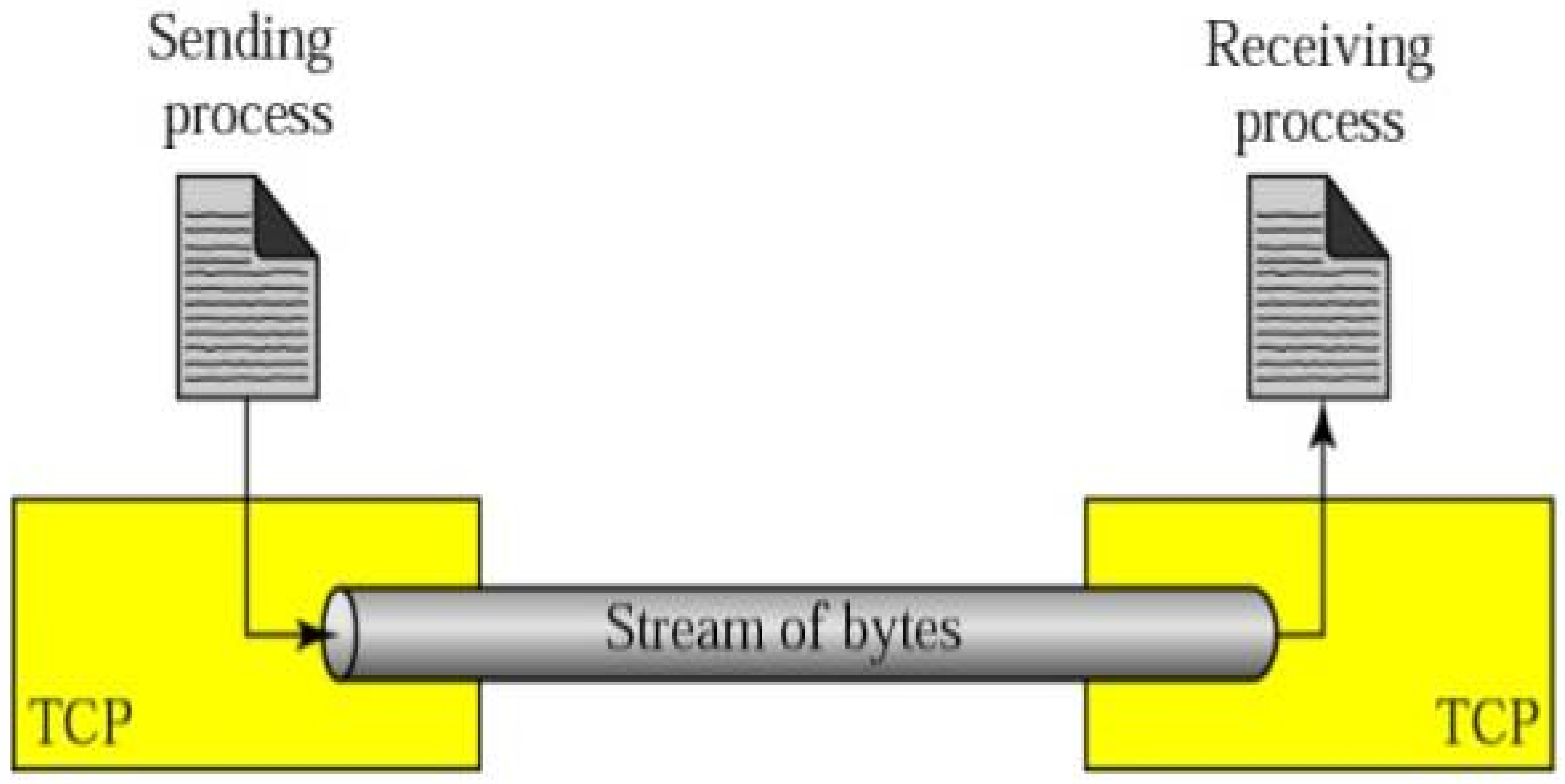
B) Stream Delivery Service

- TCP, unlike UDP, is a stream-oriented protocol.
- In UDP, a process sends messages with predefined boundaries to UDP for delivery.
- UDP adds its own header to each of these messages and delivers it to IP for transmission.
- Each message from the process is called a **user datagram**, and becomes a IP datagram.
- Neither IP nor UDP recognizes any relationship between the datagrams.

Cont…

- TCP, on the other hand, allows the sending /receiving process to deliver /obtain data as a stream of bytes.
- TCP create an environment in which the two processes seem to be connected by an imaginary “tube” that carries their bytes across the Internet.
- This imaginary environment is shown in Figure .
- The sending process produces (writes to) the stream of bytes and the receiving process consumes (reads from) them.

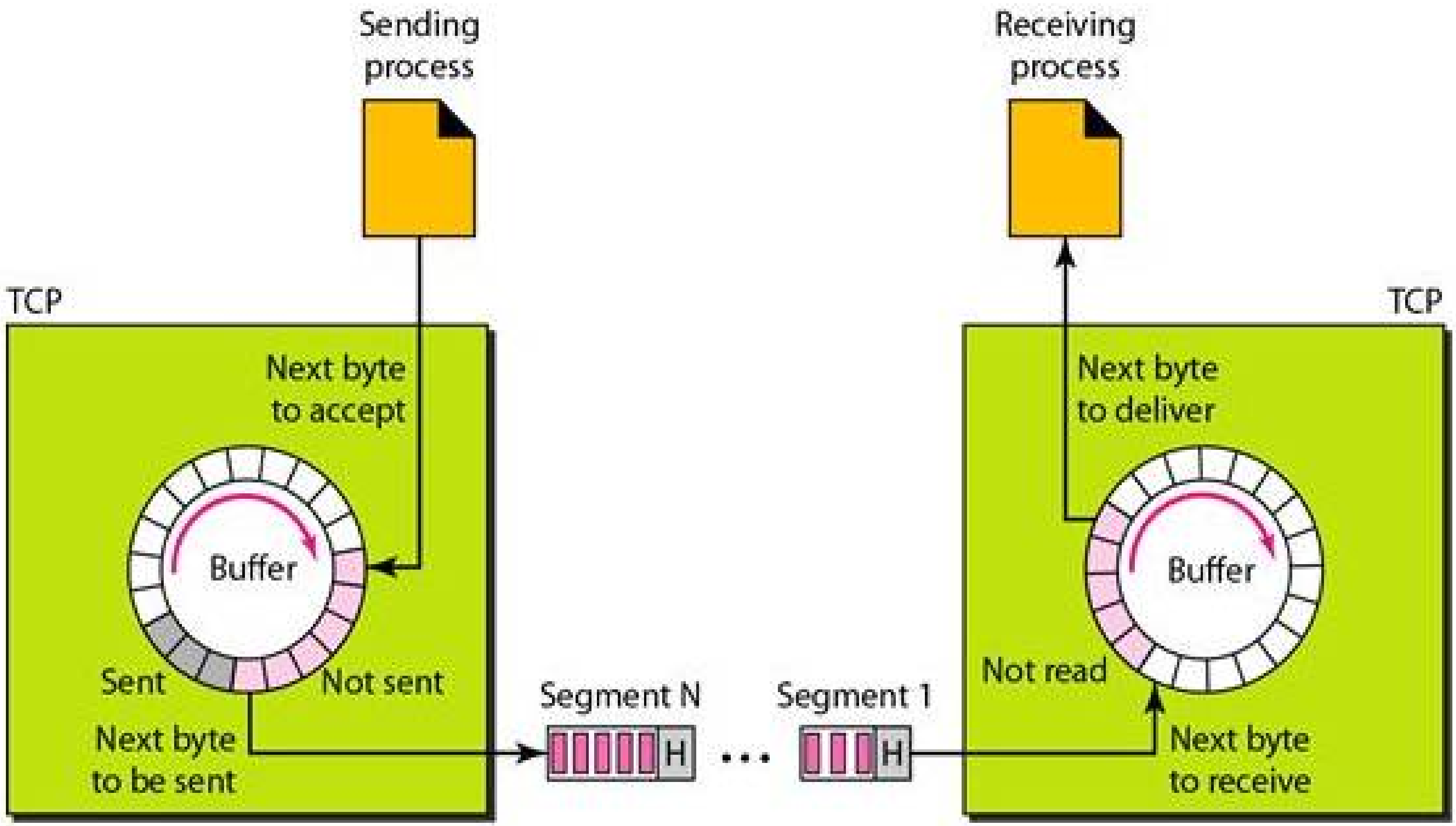
Figure : *Stream delivery*



Segments

- The IP layer, as a service provider for TCP, needs to send data in packets, not as a stream of bytes.
- At the transport layer, TCP groups a number of bytes together into a packet called a **segment.**
- TCP adds a header to each segment (for control purposes) and delivers the segment to the IP layer for transmission.

Figure : *TCP segments*



C) Full-Duplex Communication

- TCP offers full-duplex service, where data can flow in both directions at the same time.

- Each TCP endpoint then has its own sending and receiving buffer, and segments move in both directions.

D) Multiplexing and Demultiplexing

- Like UDP, TCP performs multiplexing at the sender and demultiplexing at the receiver.

- However, since TCP is a connection-oriented protocol, a connection needs to be established for each pair of processes.

E) Connection-Oriented Service

- TCP, unlike UDP, is a connection-oriented protocol.

- when a process at site A wants to send to and receive data from another process at site B, the following **Three phases occur:**

- **1. The two TCPs establish a virtual connection between them.**
- **2. Data are exchanged in both directions.**
- **3. The connection is terminated.**

Cont…

- TCP have virtual connection, not a physical connection.
- The TCP segment is encapsulated in an IP datagram and can be sent out of order, or lost, or corrupted, and then resent.
- Each may be routed over a different path to reach the destination.
- There is no physical connection.
- TCP creates a stream-oriented environment in which it accepts

the responsibility of delivering the bytes in order to the other site.

F) Reliable Service

- TCP is a reliable transport protocol.
- It uses an acknowledgment mechanism to check the safe and sound arrival of data.
- It has three way handshaking mechanism.

3.6 Flow Control

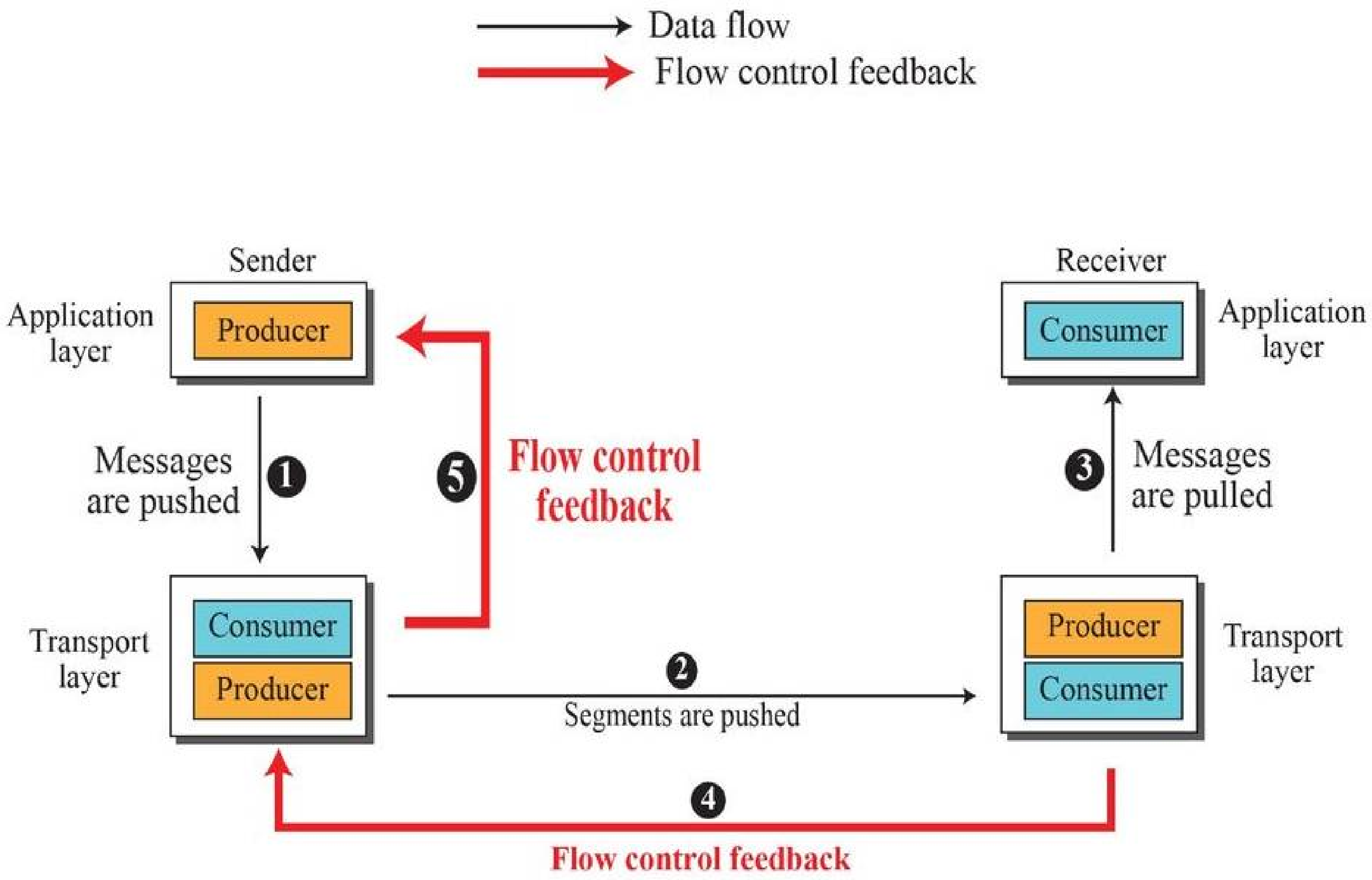
- *Flow control balances the rate a producer creates data with* the rate a consumer can use the data.

- TCP separates flow control from error control. In this section we discuss flow control, ignoring error control.

- We temporarily assume that the logical channel between the sending and receiving TCP is error-free.

- Figure shows unidirectional data transfer between a sender and a receiver; bidirectional data transfer can be deduced from unidirectional.

Figure : *Data flow and flow control feedbacks in TCP*



Cont…

- The figure shows that data travel from the sending process down to the sending TCP, from the sending TCP to the receiving TCP, and from receiving TCP up to the receiving process (paths 1, 2, and 3).

- Flow control feedbacks, however, are traveling from the receiving TCP to the sending TCP and from the sending TCP up to the sending process (paths 4 and 5).

- Most implementations of TCP do not provide flow control feedback from the receiving process to the receiving TCP, they let the receiving process pull data from the receiving TCP whenever it is ready to do so.

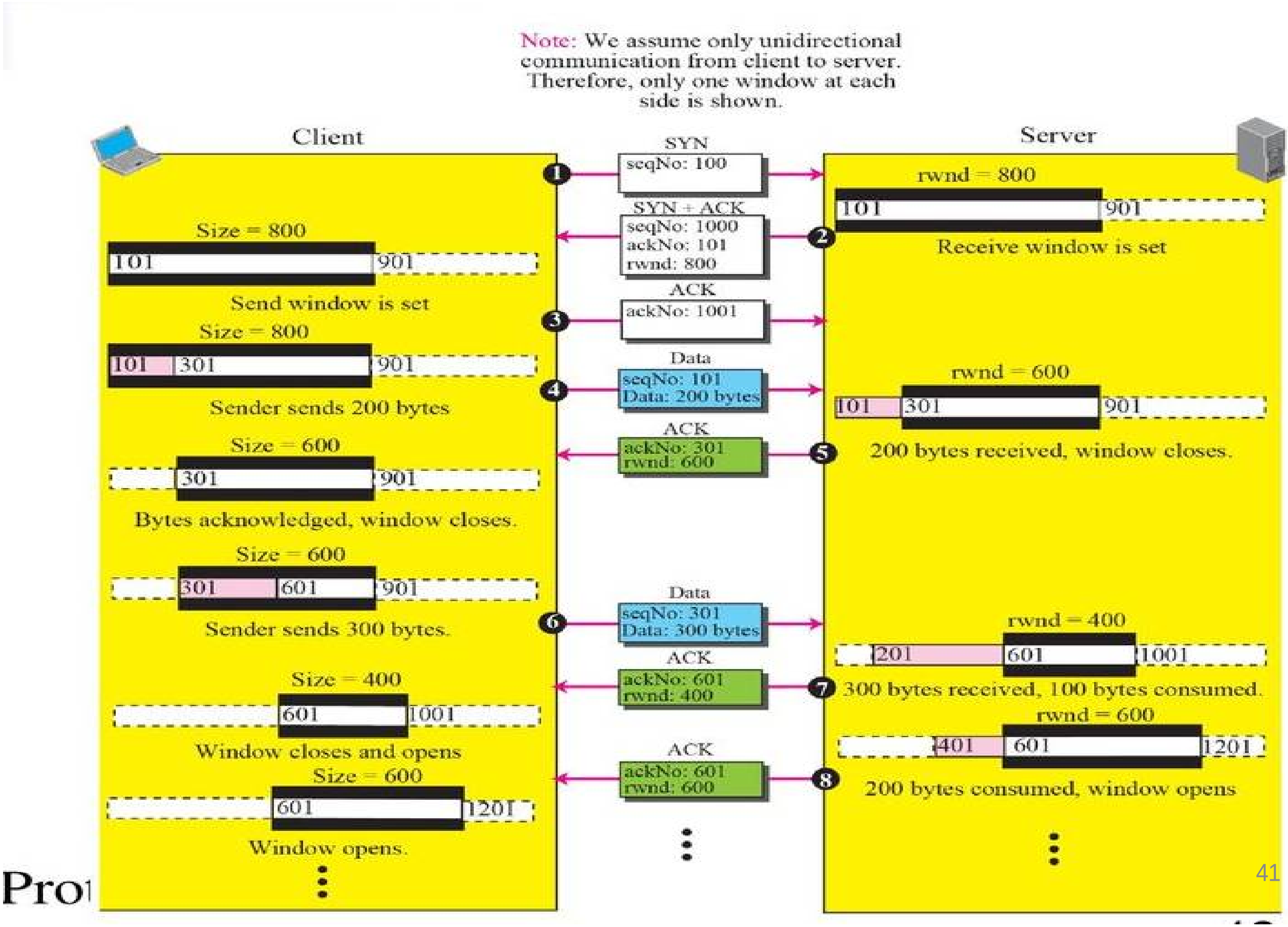
Cont…

- In other words, the receiving TCP controls the sending TCP; the sending TCP controls the sending process.

- Flow control feedback from the sending TCP to the sending process (path 5) is achieved through simple rejection of data by sending TCP when its window is full.

- This means that our discussion of flow control concentrates on the feedback sent from the receiving TCP to the sending TCP (path 4).

Figure : *An example of flow control*



Note: We assume only unidirectional communication from client to server. Therefore, only one window at each side is shown.

Cont…

- **5. The fifth segment is the feedback from the server to the client.**
- The server acknowledges bytes up to and including 300 (expecting to receive byte 301). The segment also carries the size of the receive window after decrease (600). The client, after receiving this segment, purges the acknowledged bytes from its window and closes its window to show that the next byte to send is byte 301.
- **6. Segment 6 is sent by the client after its process pushes 300 more bytes.**

Cont…

- **7. In segment 7,** the server acknowledges the receipt of data, and announces that its window size is 400. When this segment arrives at the client, the client has no choice but to reduce its window again and set the window size to the value of rwnd = 400

- **8. Segment 8 is also from the server after its process has pulled another 200 bytes.** Its window size increases. The new rwnd value is now 600. The segment informs the client that the server still expects byte 601, but the server window size has expanded to 600

3. 8 Multicasting and Multicast Routing Protocols

Introduction:

- Multicast applications are in more and more demand everyday, but as we will see multicast routing is more difficult than unicast routing .

- A multicast router is response to send a copy of a multicast packet to all members of the corresponding group.

OBJECTIVES

- *The chapter has several objectives:*

- ❑ To compare and contrast unicasting, multicasting, and broadcasting communication.

- ❑ To define multicast addressing space in IPv4 and show the division of the space into several blocks.

- ❑ To discuss the IGMP protocol, which is responsible for collecting group membership information in a network.

Cont…

- ❑ To discuss the general idea behind multicast routing protocols and their division into two categories based on the creation of the shortest path trees.

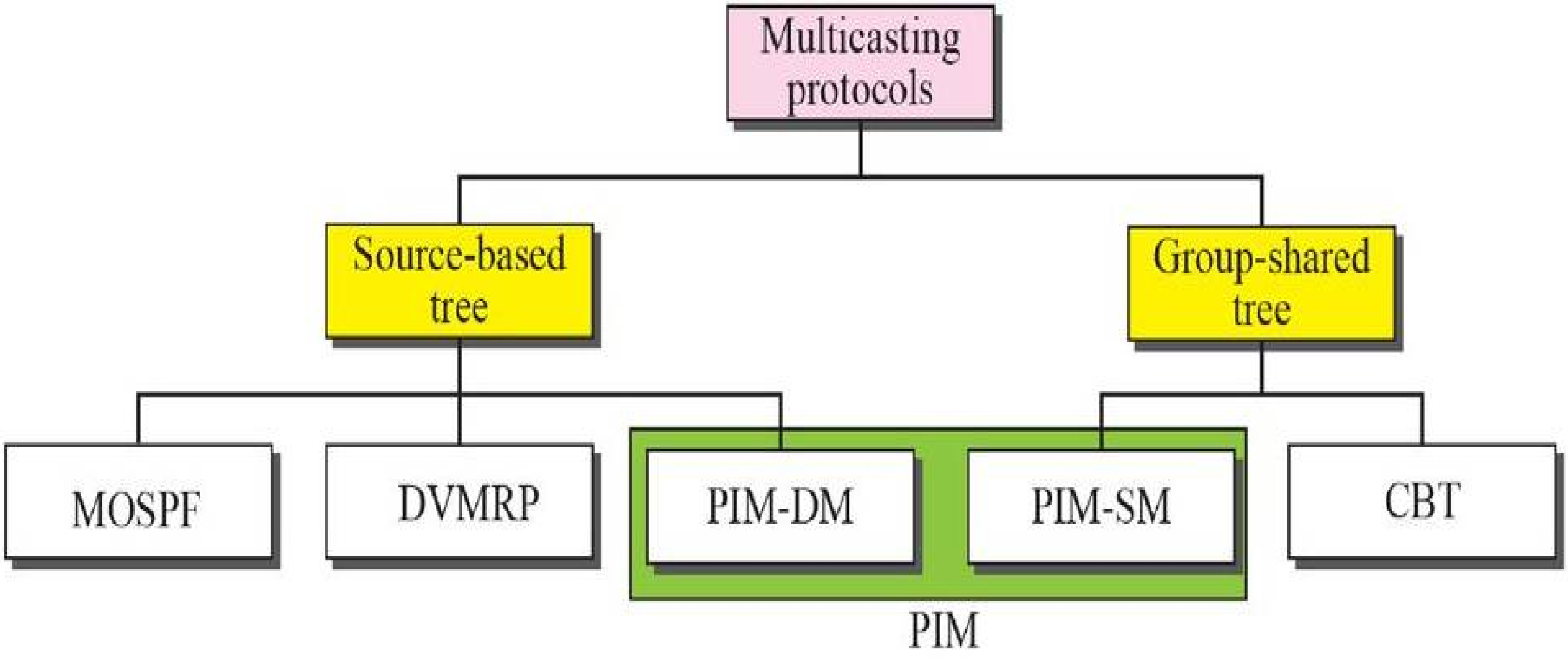
- ❑ To discuss *multicast link state routing in general and its implementation* in the Internet: a protocol named MOSPF.

- ❑ To discuss *multicast distance vector routing in general and its implementation* in the Internet: a protocol named DVMRP.

3.8 ROUTING PROTOCOLS

- During the last few decades, several multicast routing protocols have emerged.
- Some of these protocols are extensions of unicast routing protocols, some are totally new.
- Figure shows the taxonomy of these protocols.

Figure : *Taxonomy of common multicast protocols*



Multicast Link State Routing: MOSPF

- In this section, we briefly discuss multicast link state routing and its implementation in the Internet, MOSPF.
- **Multicast Link State Routing**
- We said that each router creates a shortest path tree using Dijkstra's algorithm.
- The routing table is a translation of the shortest path tree.
- Multicast link state routing is a direct extension of unicast routing and uses a source-based tree approach.
- Although unicast routing is quite involved, the extension to multicast routing is very simple and straightforward.

Cont…

- Recall that in unicast routing, each node needs to advertise the state of its links.
- For multicast routing, a node needs to revise the interpretation of *state.*
- *A node advertises* every group that has any loyal member on the link.
- Here the meaning of state is “what groups are active on this link.”
- The information about the group comes from IGMP
- Each router running IGMP solicits the hosts on the link to find out the membership status.

Cont…

- When a router receives all these LSPs, it creates *n (n is the number of groups)* topologies, from which *n shortest path trees are made using Dijkstra's algorithm.*

- *So* each router has a routing table that represents as many shortest path trees as there are groups.

MOSPF

- **Multicast Open Shortest Path First (MOSPF)** protocol is an extension of the OSPF protocol that uses multicast link state routing to create source-based trees.
- The protocol requires a new link state update packet to associate the unicast address of a host with the group address or addresses the host is sponsoring.
- This packet is called the group membership LSA.
- In this way, we can include in the tree only the hosts (using their unicast addresses) that belong to a particular group.

Cont..

- In other words, we make a tree that contains all the hosts belonging to a group, but we use the unicast address of the host in the calculation.

- For efficiency, the router calculates the shortest path trees on demand (when it receives the first multicast packet).

- In addition, the tree can be saved in cache memory for future use by the same source/group pair.

- MOSPF is a **data-driven protocol,** the first time an MOSPF router sees a datagram with a given source and group address, the router constructs the Dijkstra shortest path tree.

Multicast Distance Vector

- In this section, we briefly discuss multicast distance vector routing and its implementation in the Internet, DVMRP.

Multicast Distance Vector Routing

- Unicast distance vector routing is very simple; extending it to support multicast routing is complicated.

- Multicast routing does not allow a router to send its routing table to its neighbors.

- The idea is to create a table from scratch using the information from the unicast distance vector tables.

Flooding

- Flooding is the first strategy that comes to mind. A router receives a packet and without even looking at the destination group address, sends it out from every interface except the one from which it was received.
- Flooding accomplishes the first goal of multicasting: every network with active members receives the packet.
- However, so will networks without active members. This is a broadcast, not a multicast.
- There is another problem it creates loops.

Reverse Path Forwarding (RPF)

- **Reverse path forwarding (RPF)** is a modified flooding strategy.

- To prevent loops, only one copy is forwarded; the other copies are dropped.

- In RPF, a router forwards only the copy that has traveled the shortest path from the source to the router. To find this copy, RPF uses the unicast routing table.

- The router receives a packet and extracts the source address (a unicast address).

Cont…

- It consults its unicast routing table as though it wants to send a packet to the source address.
- The routing table tells the router the next hop.
- There is duplication because a tree has not been made, instead of a tree we have a graph.
- **Net3 has two parents: routers R2 and R4.**

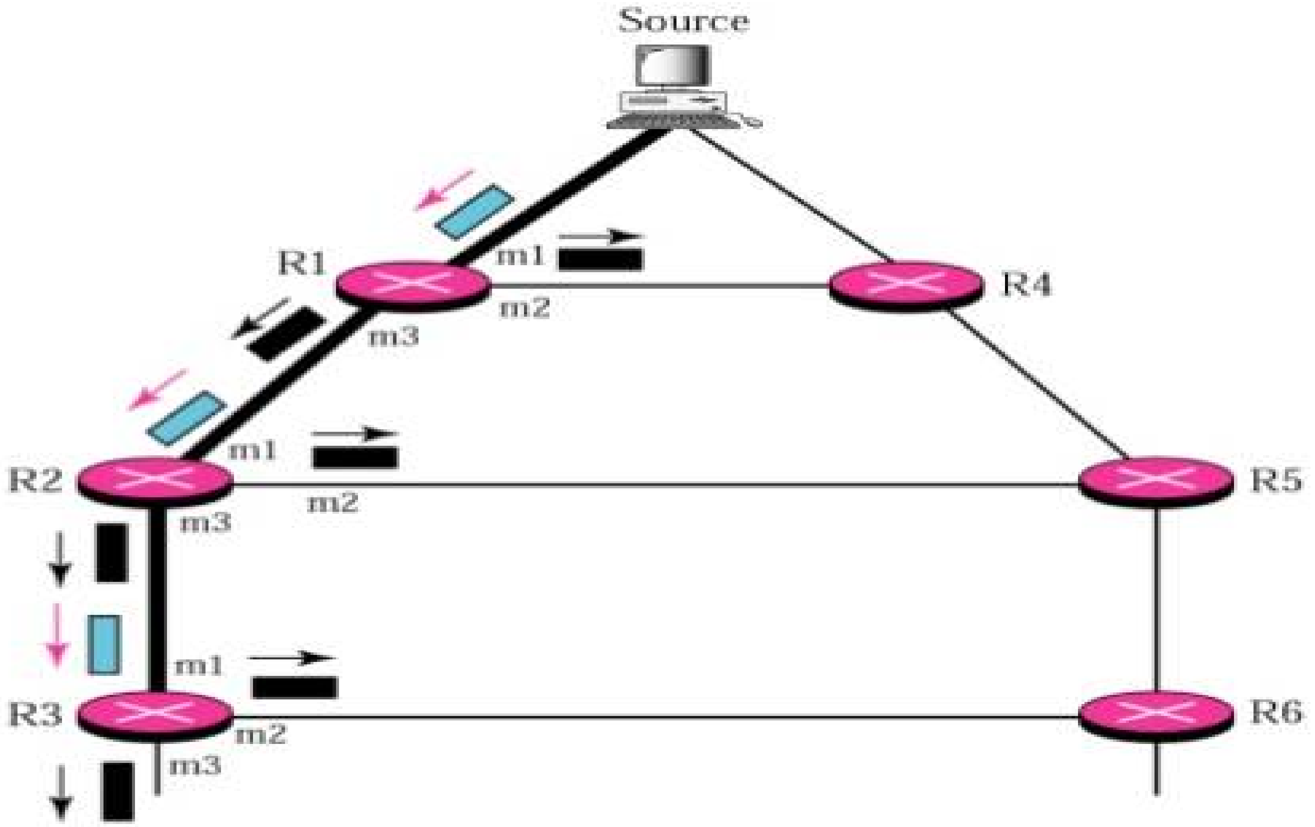
Cont…

- To eliminate duplication, we must define only one parent router for each network.

- We must have this restriction, A network can receive a multicast packet from a particular source only through a designated parent router.

- Figure shows the RPF.

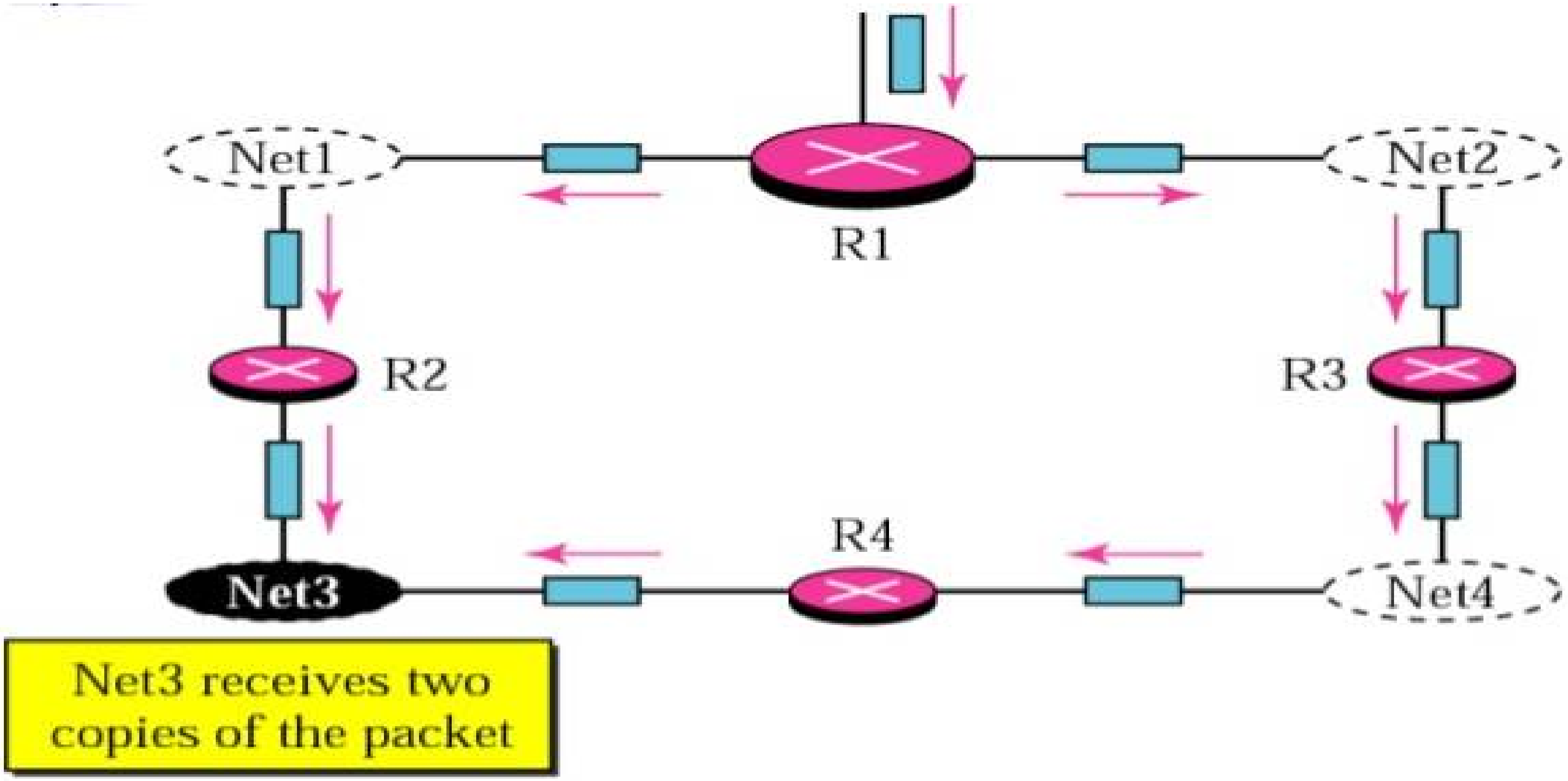
Figure : *RPF*



Reverse Path Broadcasting (RPB)

- RPF guarantees that each network receives a copy of the multicast packet without formation of loops.
- However, RPF does not guarantee that each network receives only one copy.
- A network may receive two or more copies.
- The reason is that RPF is not based on the destination address (a group address); forwarding is based on the source address.
- To visualize the problem, let us look at Figure .

Figure : Problem with RPF



Cont…

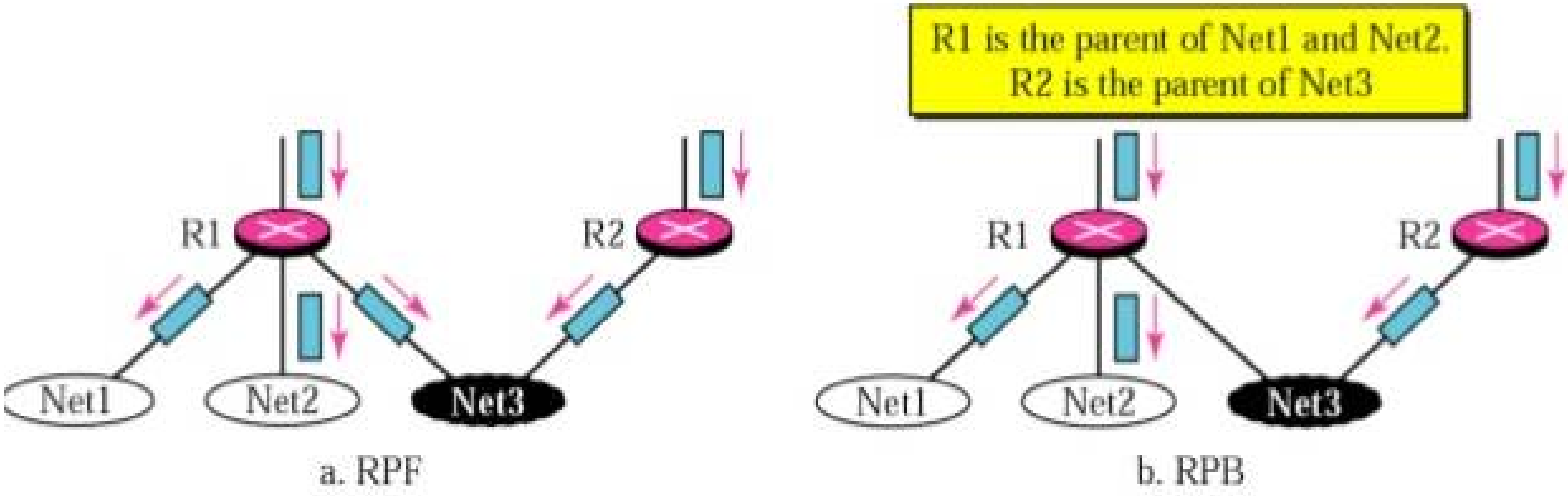
- Net3 in this figure receives two copies of the packet even though each router just sends out one copy from each interface.

- There is duplication because a tree has not been made; instead of a tree we have a graph.

- Net3 has two parents: routers R2 and R4.
- To eliminate duplication, we must define only one parent router for each network.

- We must have this restriction: A network can receive a multicast packet from a particular
- source only through a designated parent router.

Figure : *RPF versus RPB*



R1 is the parent of Net1 and Net2.
R2 is the parent of Net3

a. RPF

b. RPB

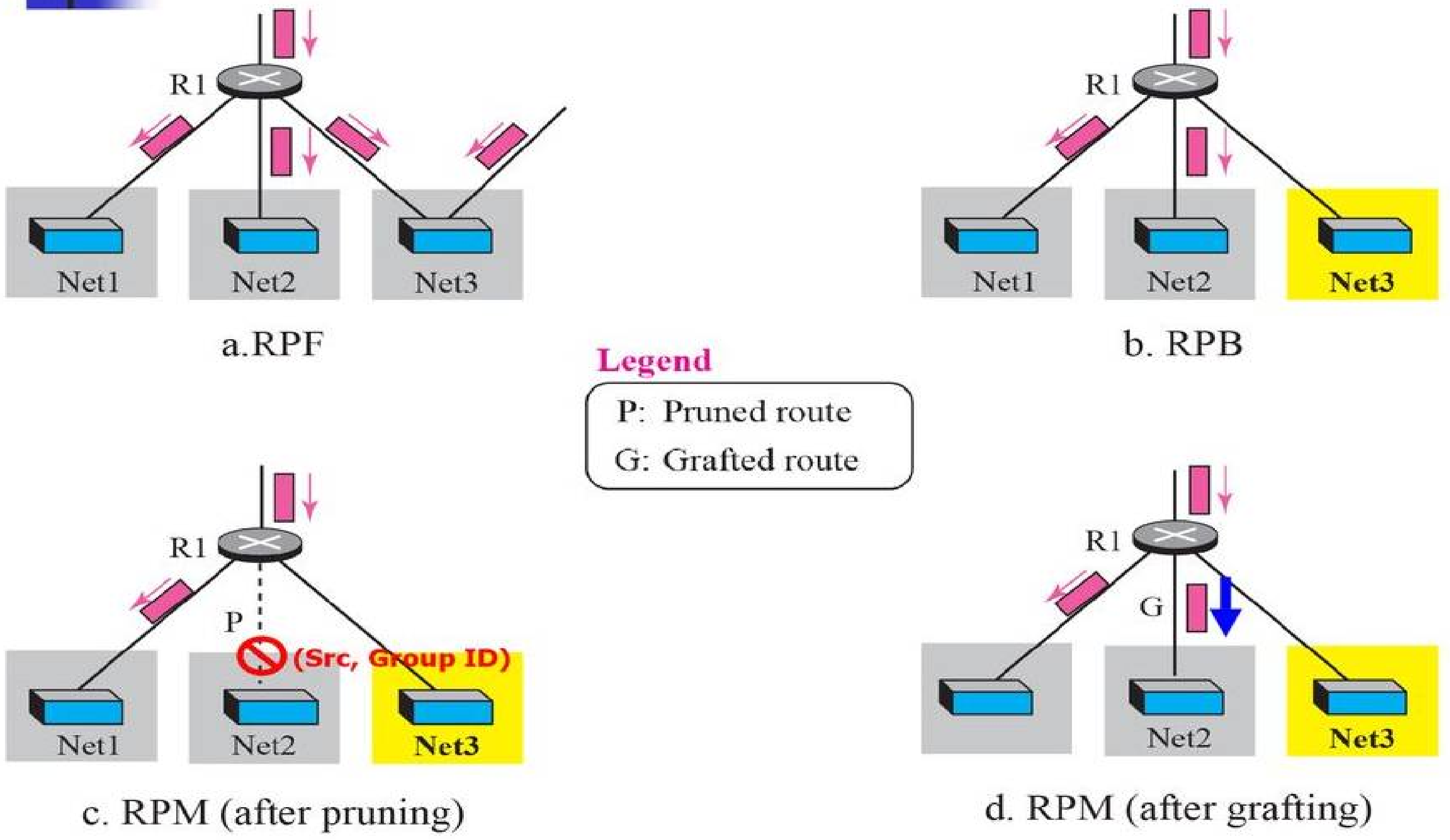
Cont…

- Now the policy is clear.
- For each source, the router sends the packet only out of those interfaces for which it is the designated parent.
- This policy is called reverse path broadcasting (RPB).
- RPB guarantees that the packet reaches every network and that every network receives only one copy.
- Figure shows the difference between RPF and RPB.

Reverse Path Multicasting (RPM)

- As you may have noticed, RPB does not multicast the packet, it broadcasts it.
- This is not efficient.
- To increase efficiency, the multicast packet must reach only those networks that have active members for that particular group.
- This is called reverse path multicasting (RPM). To convert broadcasting to multicasting, the protocol uses two procedures, pruning and grafting.
- Figure shows the idea of pruning and grafting.

Figure *RPF, RPB, and RPM*



a.RPF

b. RPB

c. RPM (after pruning)

d. RPM (after grafting)

Pruning

- The designated parent router of each network is responsible for holding the membership information.

- The process starts when a router connected to a network finds that there is no interest in a multicast packet.

- The router sends a prune message to the upstream router so that it can prune the corresponding interface.

- That is, the upstream router can stop sending multicast messages for this group through that interface.

Grafting

- What if a leaf router (a router at the bottom of the tree) has sent a prune message but suddenly realizes, through IGMP, that one of its networks is again interested.

- In receiving the multicast packet? It can send a graft message.

- The graft message forces the upstream router to resume sending the multicast messages.

PIM-SM

- PIM-SM is used when there is a slight possibility that each router is involved in multicasting (sparse mode).

- In this environment, the use of a protocol that broadcasts the packet is not justified; a protocol such as CBT that uses a group-shared tree is more appropriate.

- PIM-SM is a group-shared tree routing protocol that has a rendezvous point (RP) as the source of the tree.

- Its operation is like CBT; however, it is simpler because it does not require acknowledgment from a join message.

3.10 BOOTP

- The Bootstrap Protocol (BOOTP) is the pre runner of DHCP.
- It is a client/server protocol designed to overcome the two deficiencies of the RARP protocol.
- First, since it is a client/server program, the BOOTP server can be anywhere in the Internet.
- Second, it can provide all pieces of information we mentioned above, including the IP address.
- To provide the four pieces of information described above.

Cont…

- It removes all restriction about the RARP protocol. BOOTP, however, is a *static configuration protocol*.

- *When a client* requests its IP address, the BOOTP server consults a table that matches the physical address of the client with its IP address.

- This implies that the binding between the physical address and the IP address of the client already exists.

- The binding is predetermined address and the IP address of the client already exists.

Cont…

- There are some situations in which we need a *dynamic configuration protocol.*

- *For* example, when a host moves from one physical network to another, its physical address changes.

- As another example, there are occasions when a host wants a temporary IP address to be used for a period of time.

- BOOTP cannot handle these situations because the binding between the physical and IP addresses is static and fixed in a table until changed by the administrator.

3.11 UDP PORTS

- SNMP uses the services of UDP on two well-known ports, 161 and 162.

- The well known port 161 is used by the server (agent), and the well-known port 162 is used by the client (manager).

- The agent (server) issues a passive open on port 161. It then waits for a connection from a manager (client).

- A manager (client) issues an active open using an ephemeral port.

Cont…

- The request messages are sent from the client to the server using the ephemeral port as the source port and the well-known port 161 as the destination port.

- The response messages are sent from the server to the client using the well-known port 161 as the source port and the ephemeral port as the destination port.

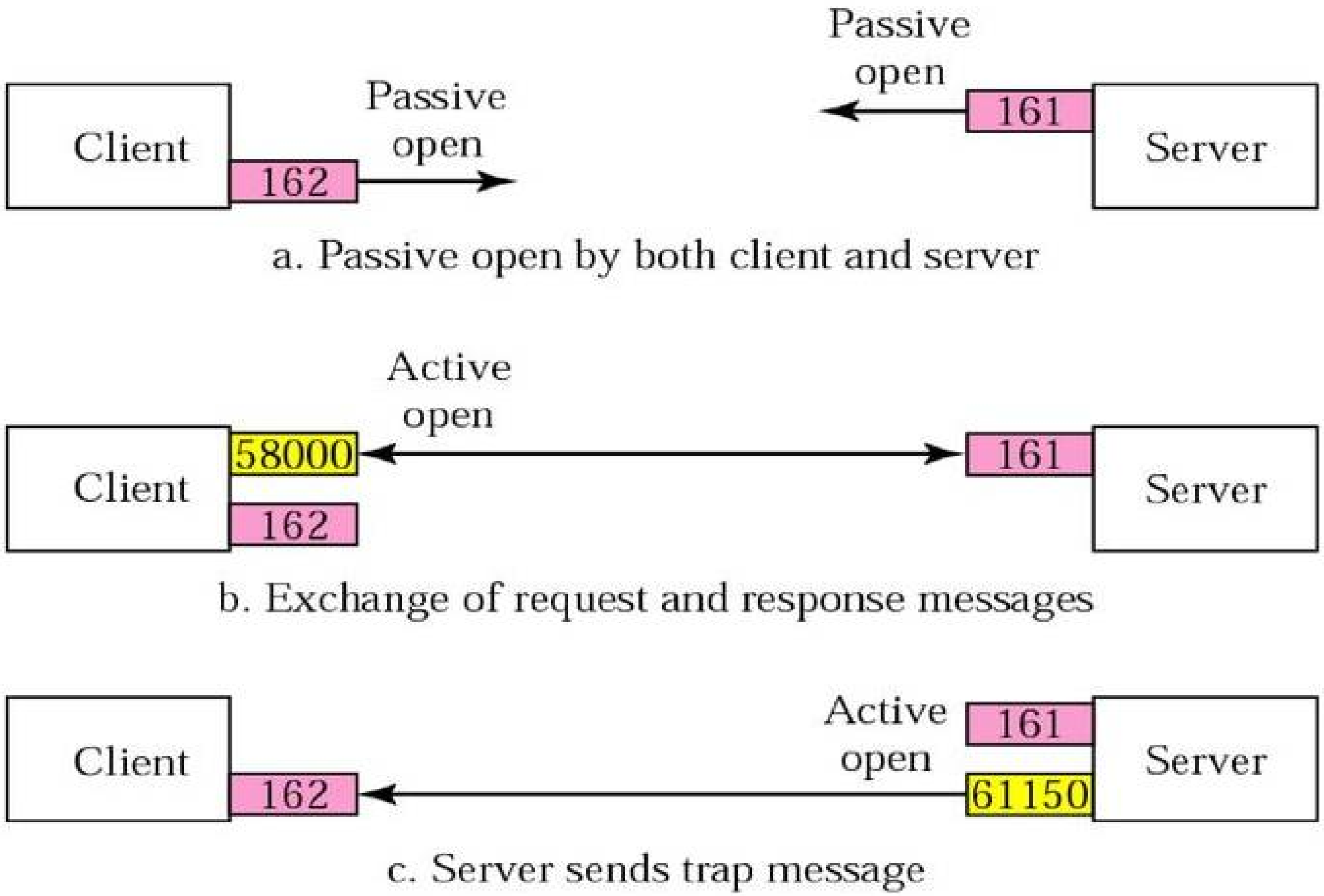
- The manager (client) issues a passive open on port 162. It then waits for a connection from an agent (server).

- Whenever it has a Trap message to send, an agent (server) issues an active open, using an ephemeral port.

Cont…

- This connection is only one-way, from the server to the client (see Figure).
- The client-server mechanism in SNMP is different from other protocols.
- Here both the client and the server use well-known ports.

Figure : *Port numbers for SNMP*



a. Passive open by both client and server

b. Exchange of request and response messages

c. Server sends trap message

3.12 Using TFTP(Trivial File Transfer Protocol)

- The server does not send all of the information that a client may need for booting.
- In the reply message, the server defines the pathname of a file in which the client can find complete booting information.
- The client can then use a TFTP message .
- which is encapsulated in a UDP user datagram, to obtain the rest of the needed information.

3.13 Dynamic host Configuration Protocols (DHCP)

- Dynamic Host Configuration Protocol (DHCP). This application is discussed first because it is the first client/server application program that is used after a host is booted.

- In other words, it serves as a bootstrap when a host is booted and supposed to be connected to the Internet, but the host does not know its IP address.

OBJECTIVES

- ***The chapter has several objectives:***

- ❑ To give the reasons why we need host configuration.

- ❑ To give a historical background of two protocols used for host configuration in the past.

- ❑ To define DHCP as the current Dynamic Host Configuration Protocol.

- ❑ To discuss DHCP operation when the client and server are on the same network or on different networks.

Cont…

- ❑ To show how DHCP uses two well-known ports of UDP to achieve configuration.

- ❑ To discuss the states the clients go through to lease an IP address from a DHCP server.

INTRODUCTION:

Four pieces of information are normally needed:

- **1. The IP address of the computer**
- **2. The subnet mask of the computer**
- **3. The IP address of a router**
- **4. The IP address of a name server**

A) RARP

- Before DHCP became the formal protocol for host configuration, some other protocols were used for this propose.

- At the beginning of the Internet era, a protocol called Reverse Address Resolution Protocol (RARP) was designed to provide the IP address for a booted computer.

- RARP maps a physical address to an IP address.

- However, RARP is deprecated today for two reasons.

- First, RARP used the broadcast service of the data link layer, which means that a RARP server must be present in each network.

- Second, RARP can provide only the IP address of the computer, but a computer today needs all four pieces of information mentioned above.

B) BOOTP

- The Bootstrap Protocol (BOOTP) is the pre runner of DHCP.
- It is a client/server protocol designed to overcome the two deficiencies of the RARP protocol.
- First, since it is a client/server program, the BOOTP server can be anywhere in the Internet.
- Second, it can provide all pieces of information we mentioned above, including the IP address.
- To provide the four pieces of information described above.

Cont…

- It removes all restriction about the RARP protocol. BOOTP, however, is a *static configuration protocol*.
- *When a client* requests its IP address, the BOOTP server consults a table that matches the physical address of the client with its IP address.
- This implies that the binding between the physical address and the IP address of the client already exists.
- The binding is predetermined address and the IP address of the client already exists.

Cont…

- There are some situations in which we need a *dynamic configuration protocol*.

- *For* example, when a host moves from one physical network to another, its physical address changes.

- As another example, there are occasions when a host wants a temporary IP address to be used for a period of time.

- BOOTP cannot handle these situations because the binding between the physical and IP addresses is static and fixed in a table until changed by the administrator.

C) DHCP

- The **Dynamic Host Configuration Protocol (DHCP)** is a client/server protocol .
- Designed to provide the four pieces of information for a diskless computer or a computer that is booted for the first time.
- DHCP is a successor to BOOTP and is backward compatible with it.
- Although BOOTP is considered deprecated, there may be some systems that may still use BOOTP for host configuration.

3.14 Domain Name system (DNS)

- DNS is a client/server application program used to help other application programs.

- DNS is used to map a host name in the application layer to an IP address in the network layer.

A) NEED FOR DNS

- To identify an entity, TCP/IP protocols use the IP address, which uniquely identifies the connection of a host to the Internet.

- However, people prefer to use names instead of numeric addresses.

- Therefore, we need a system that can map a name to an address or an address to a name.

Cont…

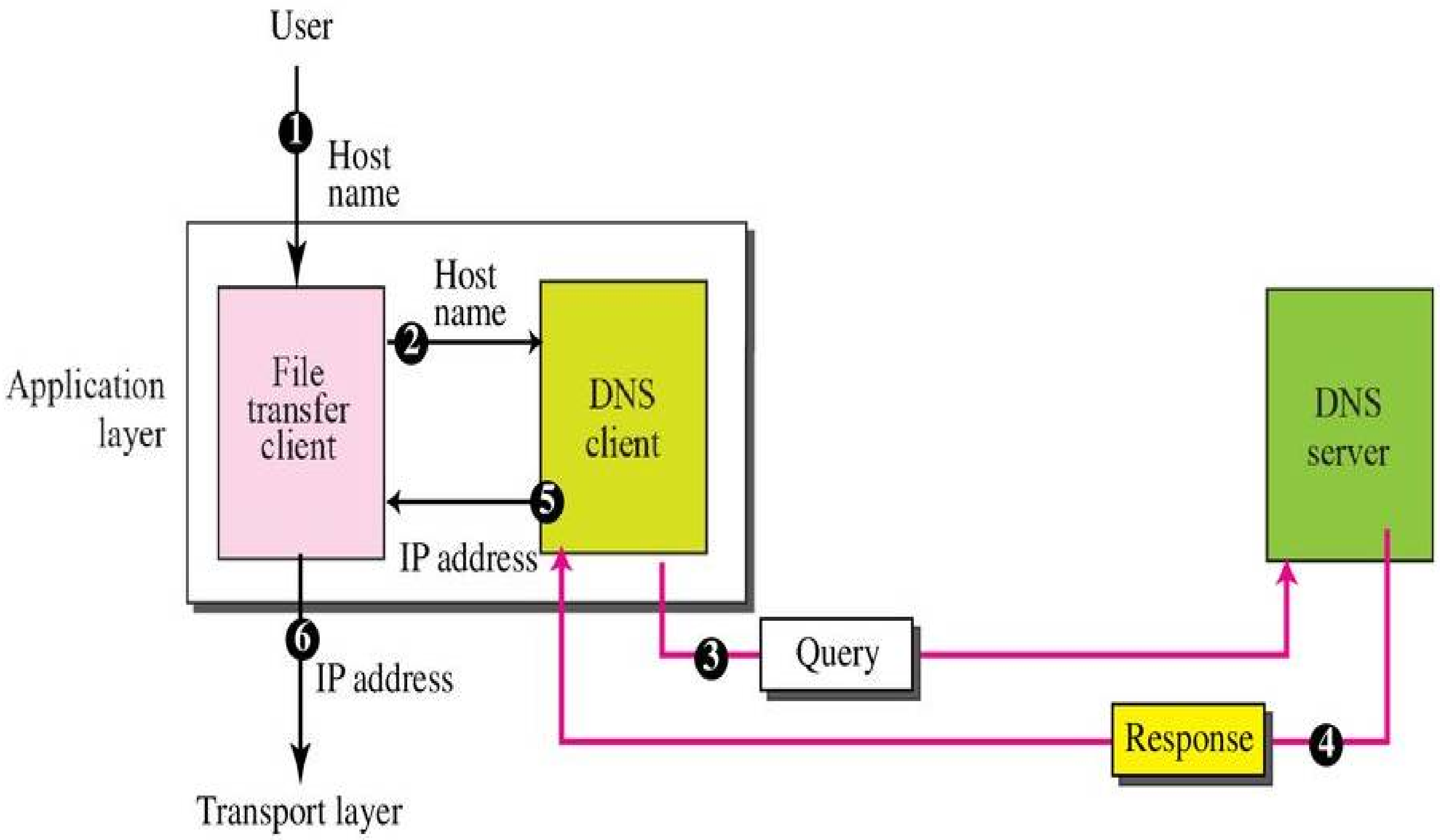
- When the Internet was small, mapping was done using a *host file.*

- ***The host file had* only two columns: name and address.**

- Every host could store the host file on its disk and update it periodically from a master host file.

- When a program or a user wanted to map a name to an address, the host consulted the host file and found the mapping.

Figure : *Purpose of DNS*



Cont…

- In Figure , a user wants to use a file transfer client to access the corresponding file transfer server running on a remote host.
- The user knows only the file transfer server name, such as *forouzan.com.*
- *However, the TCP/IP suite needs the IP address of* the file transfer server to make the connection.

Cont…

- The following six steps map the host name to an IP address:

- 1. The user passes the host name to the file transfer client.
- 2. The file transfer client passes the host name to the DNS client.
- 3. We know that each computer, after being booted, knows the address of one DNS server.
- The DNS client sends a message to a DNS server with a query that gives the file transfer server name using the known IP address of the DNS server.

Cont…

- 4. The DNS server responds with the IP address of the desired file transfer server.
- 5. The DNS client passes the IP address to the file transfer server.
- 6. The file transfer client now uses the received IP address to access the file transfer server.

3.15 NAME SPACE

- To be unambiguous, the names assigned to machines must be carefully selected from a name space with complete control over the binding between the names and IP addresses.

- In other words, the names must be unique because the addresses are unique.

- A name space that maps each address to a unique name can be organized in **two ways:**

  flat or hierarchical.

Flat Name Space

- In a flat name space, a name is assigned to an address. A name in this space is a sequence of characters without structure.

- The names may or may not have a common section.
- If they do, it has no meaning.

- The main disadvantage of a flat name space is that it cannot be used in a large system such as the Internet because it must be centrally controlled to avoid ambiguity and duplication.

Hierarchical Name Space

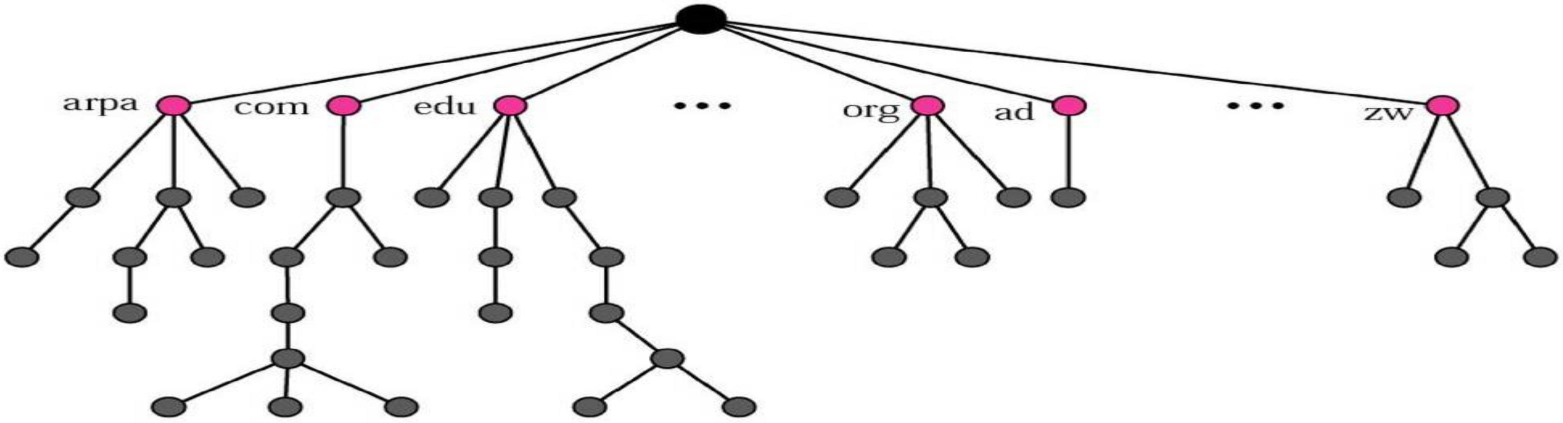
- In a hierarchical name space, each name is made of several parts.

- The first part can define the nature of the organization, the second part can define the name of an organization, the third part can define departments in the organization, and so on.

- In this case, the authority to assign and control the name spaces can be decentralized.

3.16 Domain Name Space

- To have a hierarchical name space, a domain name space was designed.
- In this design the names are defined in an inverted-tree structure with the root at the top.
- The tree can have only 128 levels: level 0 (root) to level 127



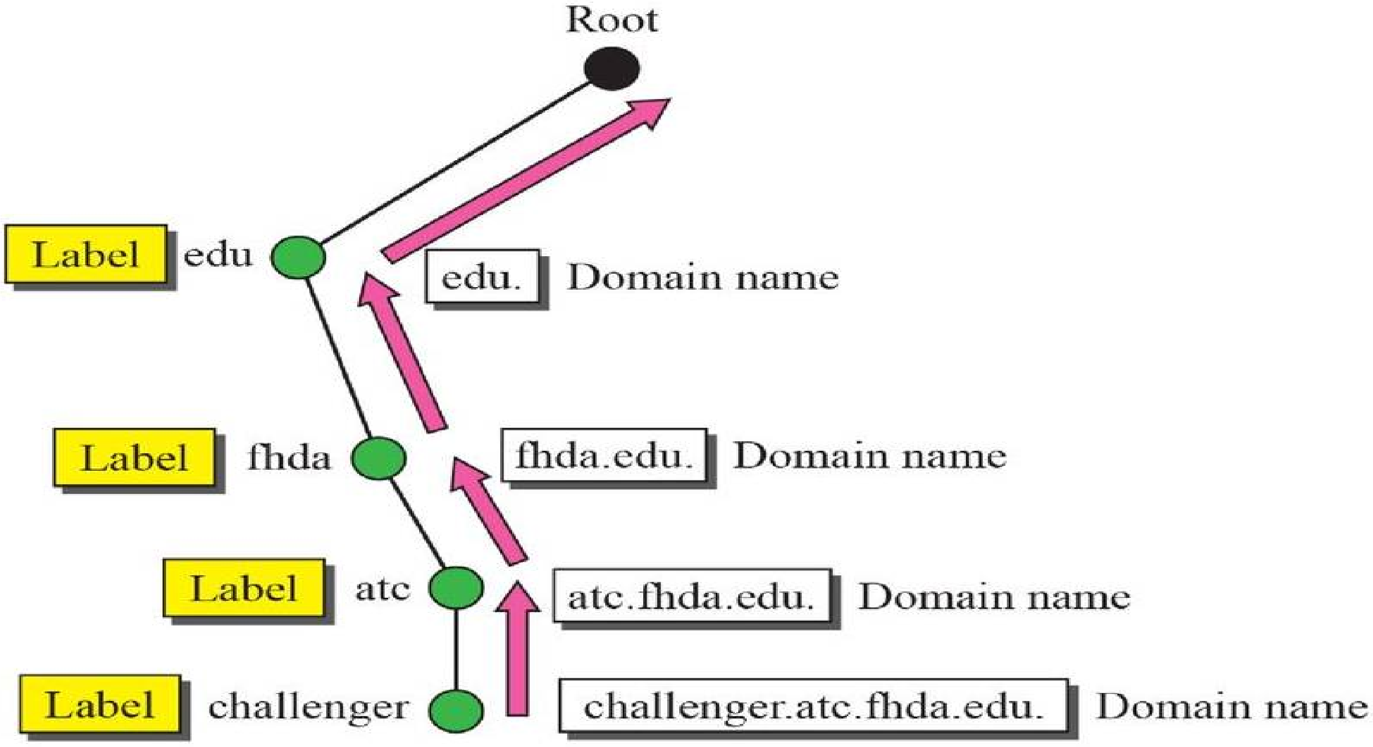
Label

- Each node in the tree has a label, which is a string with a maximum of 63 characters.
- The root label is a null string (empty string). DNS requires that children of a node (nodes that branch from the same node) have different labels.
- which guarantees the uniqueness of the domain names.

Domain Name

- Each node in the tree has a domain name.
- A full **domain name is a sequence of labels** separated by dots (.).
- The domain names are always read from the node up to the root.
- The last label is the label of the root (null).
- This means that a full domain name always ends in a null label, which means the last character is a dot because the null string is nothing.
- Figure shows some domain names.

Figure : *Domain names and labels*



3.17 Distribution of Name Space

- The information contained in the domain name space must be stored.

- However, it is very inefficient and also not reliable to have just one computer store such a huge amount of information.

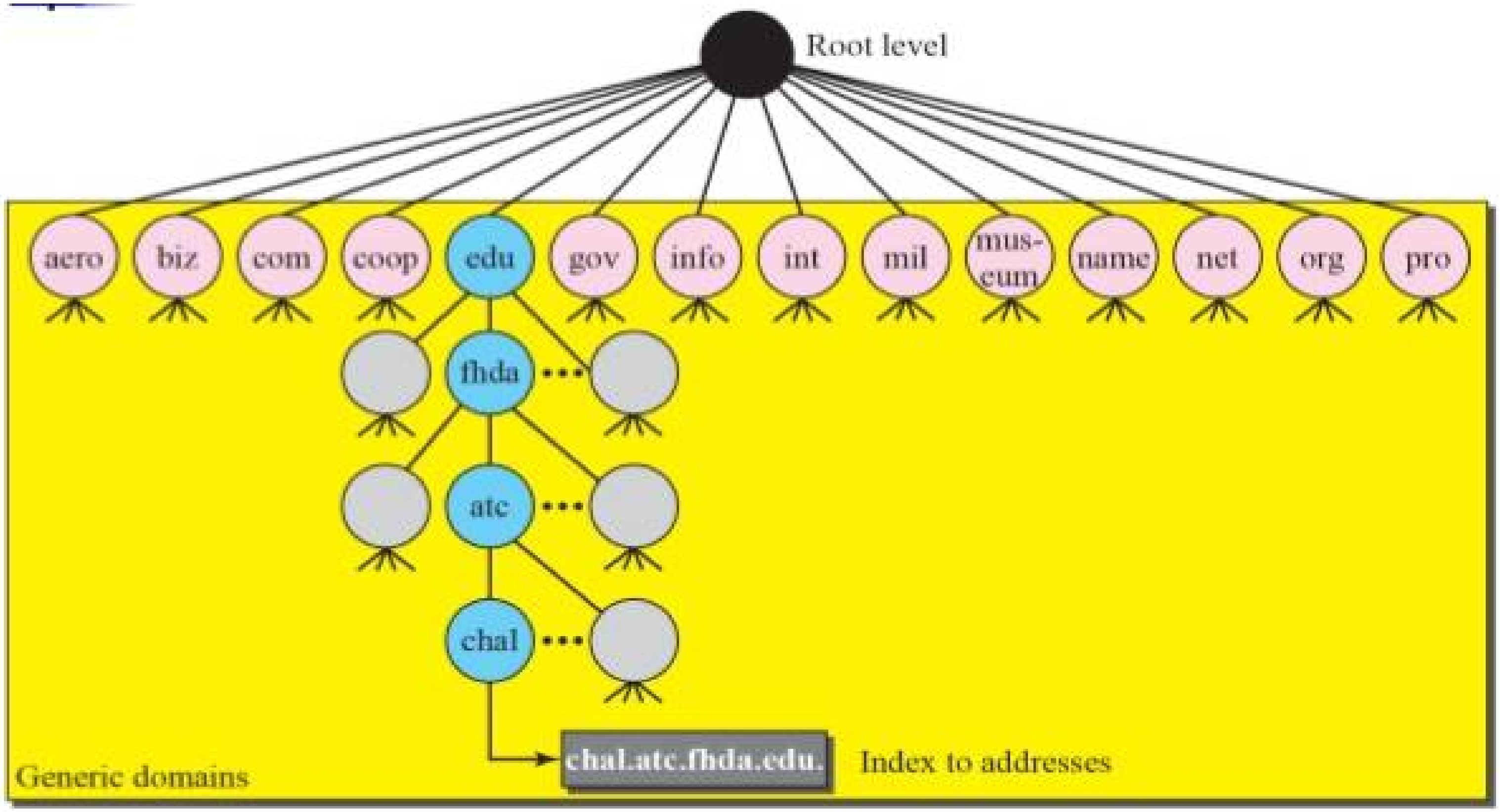
- It is inefficient because responding to requests from all over the world places a heavy load on the system.

- It is not reliable because any failure makes the data inaccessible.

3.18 DNS IN THE INTERNET

- DNS is a protocol that can be used in different platforms.
- In the Internet, the domain name space (tree) is divided into three different sections: generic domains, country domains, and the inverse domain (Figure).
- **Generic Domains**
- The generic domains define registered hosts according to their generic behavior.
- Each node in the tree defines a domain, which is an index to the domain name space database (Figure).

Figure : *Generic domains*



Country Domains

- The country domains section uses two-character country abbreviations (e.g., us for United States).

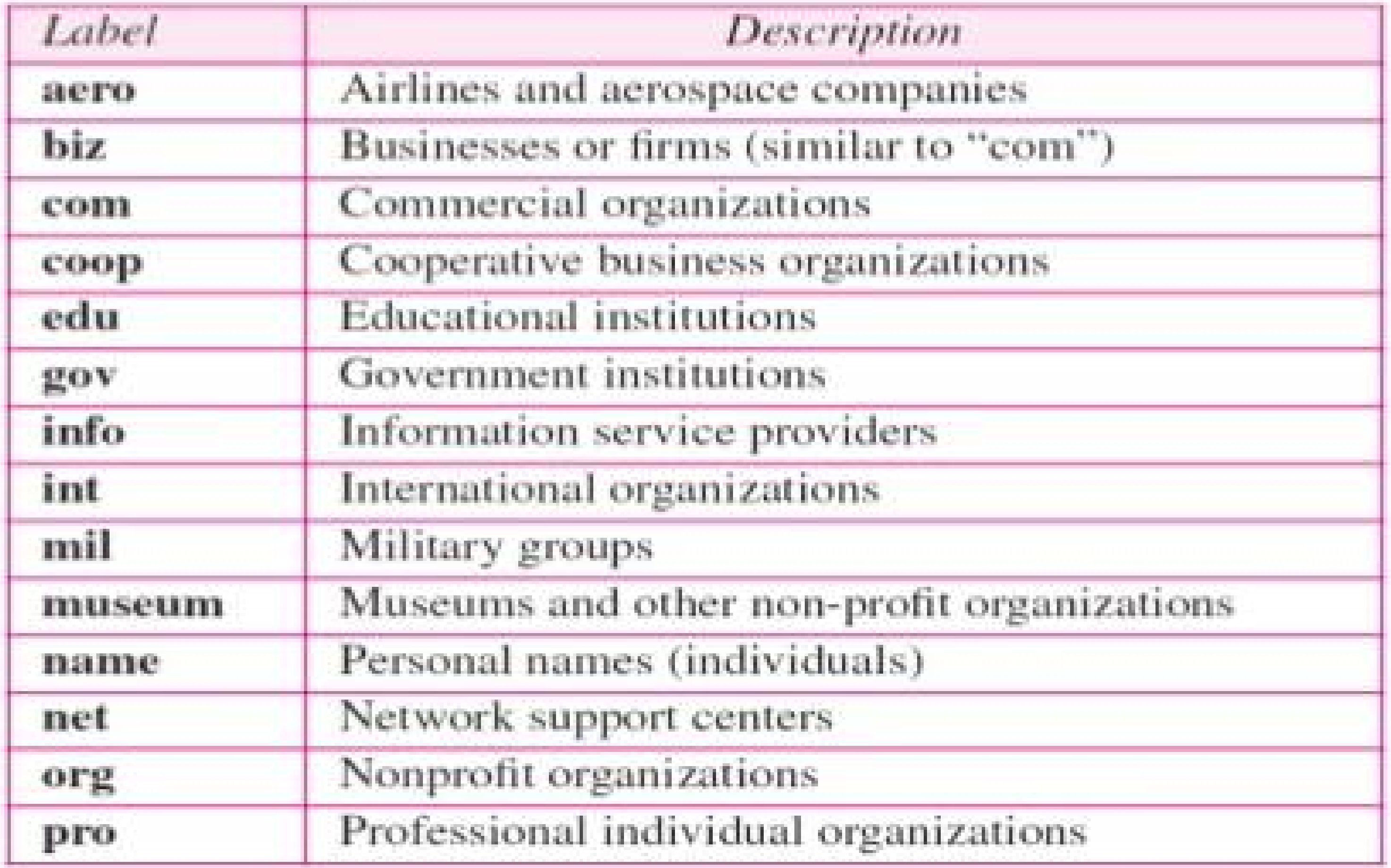
- Second labels can be organizational, or they can be more specific, national designations.

- The United States, for example, uses state abbreviations as a subdivision of us (e.g., ca.us.).

- Figure shows the country domains section.

- The address *anza.cup.ca.us can* be translated to De Anza College in Cupertino in California in the United States.

Table : *Generic domain labels*



Label	*Description*
aero	Airlines and aerospace companies
biz	Businesses or firms (similar to "com")
com	Commercial organizations
coop	Cooperative business organizations
edu	Educational institutions
gov	Government institutions
info	Information service providers
int	International organizations
mil	Military groups
museum	Museums and other non-profit organizations
name	Personal names (individuals)
net	Network support centers
org	Nonprofit organizations
pro	Professional individual organizations

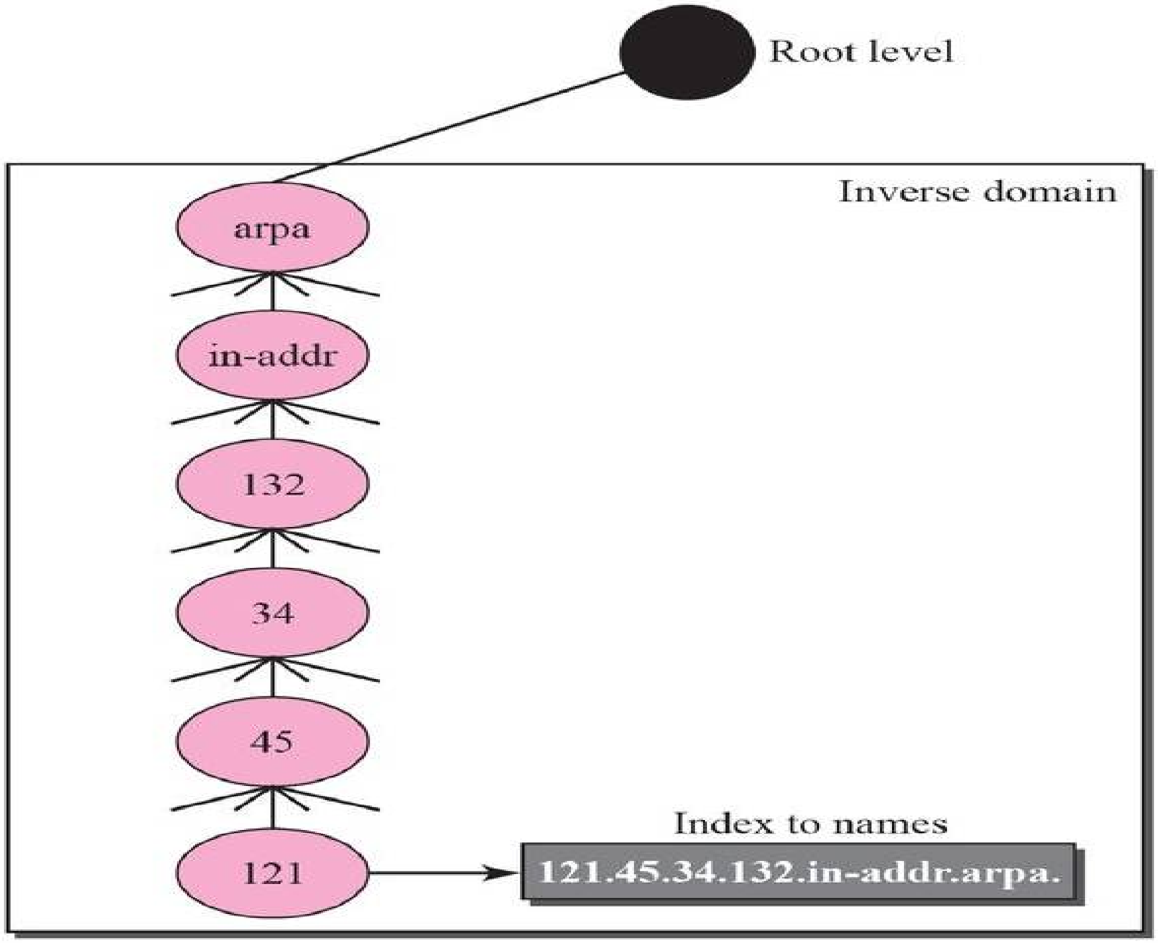
Inverse Domain

- The inverse domain is used to map an address to a name.

- when a server has received a request from a client to do a task.
-
- Although the server has a file that contains a list of authorized clients, only the IP address of the client (extracted from the received IP packet) is listed.

- The server asks its resolver to send a query to the DNS server to map an address to a name to determine if the client is on the
- authorized list.

Figure : *Inverse domain*



Cont…

- This type of query is called an inverse or pointer (PTR) query.

- To handle a pointer query, the inverse domain is added to the domain name space with the first-level node called *arpa (for* historical reasons).

- The second level is also one single node named in-addr (for inverse address).

- The rest of the domain defines IP addresses.

- The servers that handle the inverse domain are also hierarchical.

Cont…

- This means the netid part of the address should be at a higher level than the subnetid part, and the subnetid part higher than the hostid part.
- In this way, a server serving the whole site is at a higher level than the servers serving each subnet.
- This configuration makes the domain look inverted when compared to a generic or country domain.
- To follow the convention of reading the domain labels from the bottom to the top, an IP address such as 132.34.45.121 (a class B address with netid 132.34) is read as 121.45.34.132.in-addr. arpa.
- See Figure for an illustration of the inverse domain configuration.

3.19 Resolution

- Mapping a name to an address or an address to a name is called **name-address resolution.**

Resolver

- DNS is designed as a client-server application. A host that needs to map an address to a name or a name to an address calls a DNS client called a **resolver.**
- **The resolver** accesses the closest DNS server with a mapping request.
- If the server has the information, it satisfies the resolver; otherwise, it either refers the resolver to other servers or
- asks other servers to provide the information.

Mapping Names to Addresses

- Most of the time, the resolver gives a domain name to the server and asks for the corresponding address.

- In this case, the server checks the generic domains or the country domains to find the mapping.

- If the domain name is from the generic domains section, the resolver receives a domain name such as "*chal.atc.fhda.edu.*".

- The query is sent by the resolver to the local DNS server for resolution.

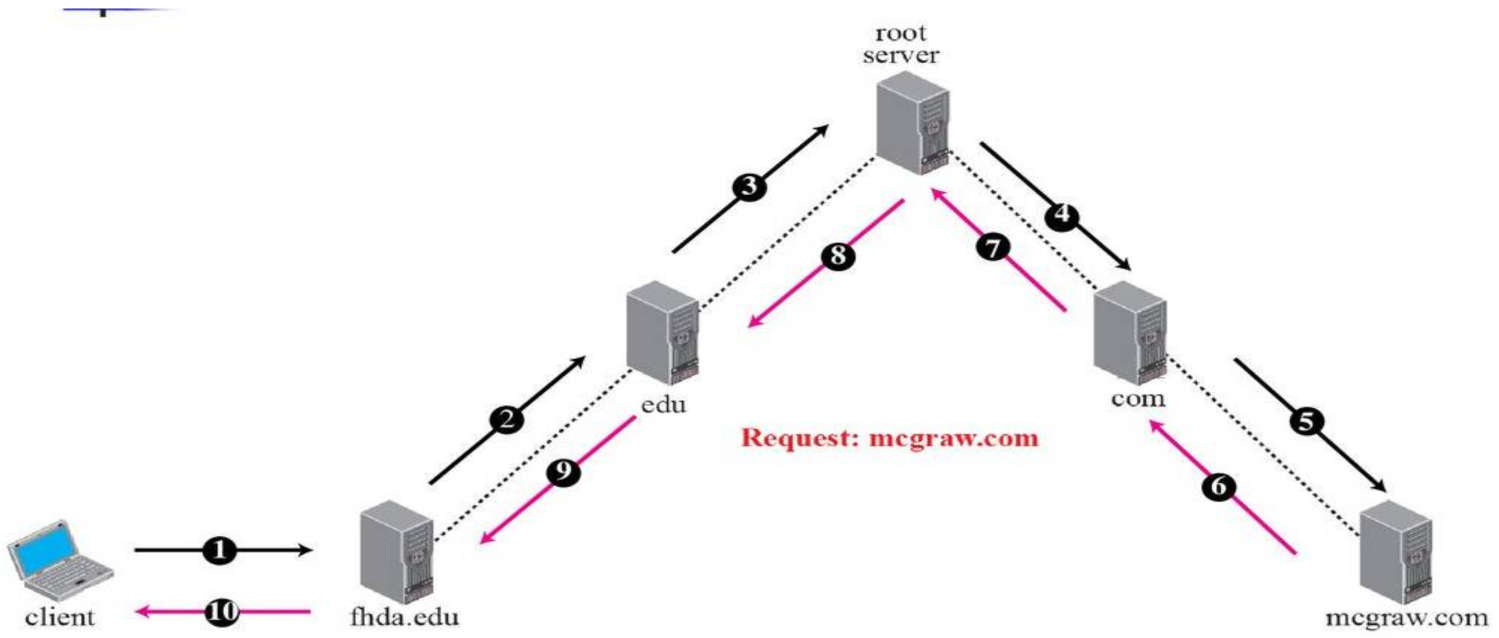
- If the local server cannot resolve the query, it either refers the resolver to other servers or asks other servers directly.

Mapping Addresses to Names

- A client can send an IP address to a server to be mapped to a domain name.
- As mentioned before, this is called a PTR query.
- To answer queries of this kind, DNS uses the inverse domain.
- However, in the request, the IP address is reversed and two labels, *in-addr and arpa, are appended to create a domain acceptable by*
- the inverse domain section.

Recursive Resolution

- Figure shows the recursive resolution.
- The client (resolver) can ask for a recursive answer from a name server.
- This means that the resolver expects the server to supply the final answer.
- If the server is the authority for the domain name, it checks its database and responds.



Cont…

- If the server is not the authority, it sends the request to another server (the parent usually) and waits for the response.

- If the parent is the authority, it responds; otherwise, it sends the query to yet another server.

- When the query is finally resolved, the response travels back until it finally reaches the requesting client.

Iterative Resolution

- If the client does not ask for a recursive answer, the mapping can be done iteratively.
- If the server is an authority for the name, it sends the answer.
- If it is not, it returns (to the client) the IP address of the server that it thinks can resolve the query.
- The client is responsible for repeating the query to this second server.
- If the newly addressed server can resolve the problem, it answers the query with the IP address; otherwise, it returns
- the IP address of a new server to the client.
- Now the client must repeat the query to the third server. This process is **called *iterative***

Caching

- Each time a server receives a query for a name that is not in its domain, it needs to search its database for a server IP address.
- Reduction of this search time would increase efficiency.
- DNS handles this with a mechanism called caching.
- When a server asks for a mapping from another server and receives the response, it stores this information in its cache memory before sending it to the client.

Cont…

- If the same or another client asks for the same mapping, it can check its cache memory and resolve the problem.

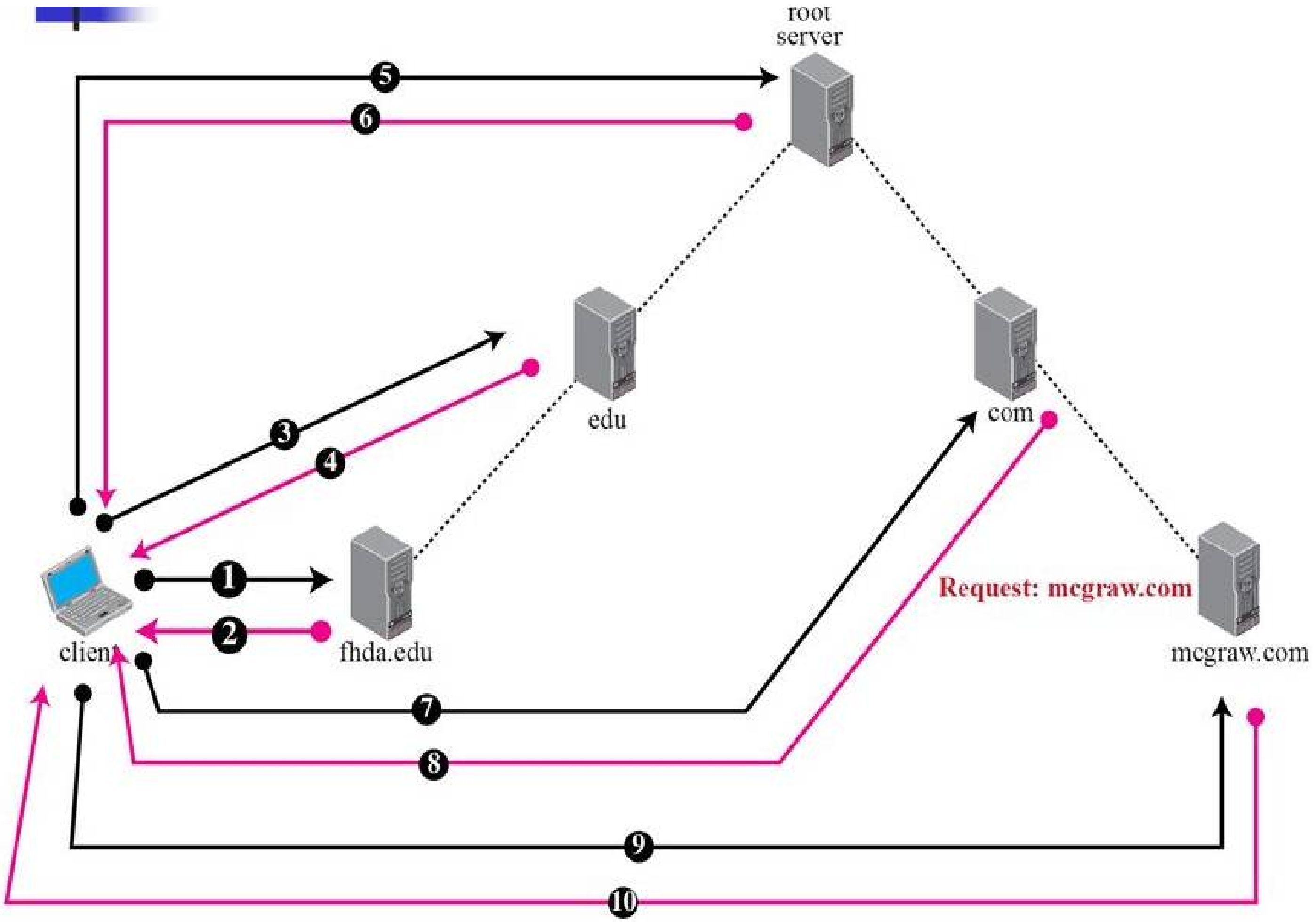
- However, to inform the client that the response is coming from the cache memory and not from an authoritative source, the server marks the response as *unauthoritative*.

- Caching speeds up resolution, but it can also be problematic.

- If a server caches a mapping for a long time, it may send an outdated mapping to the client.

- First, the authoritative server always adds information to the mapping **called *time-to-live (TTL)*.**

Figure : *Iterative resolution*



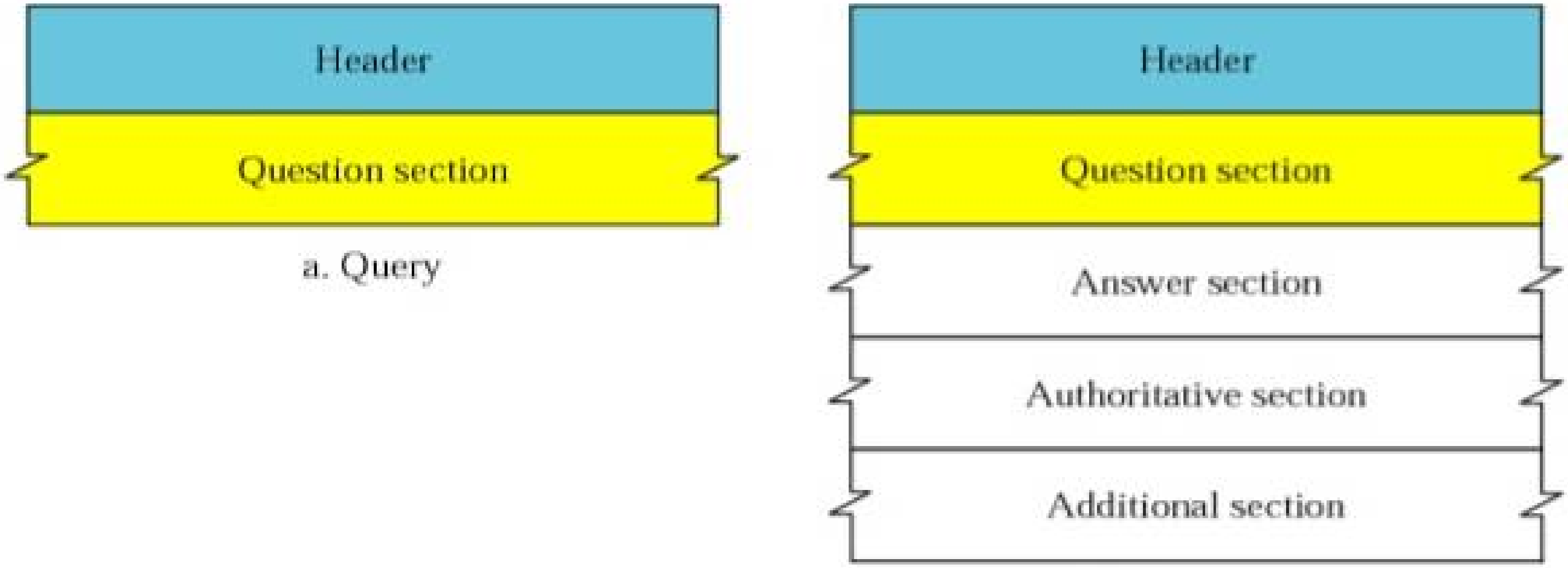
3.20 DNS MESSAGES

- DNS has two types of messages: query and response. Both types have the same format.

- The query message consists of a header and question records.

- the response message consists of a header, question records, answer records, authoritative records, and additional records (see Figure).

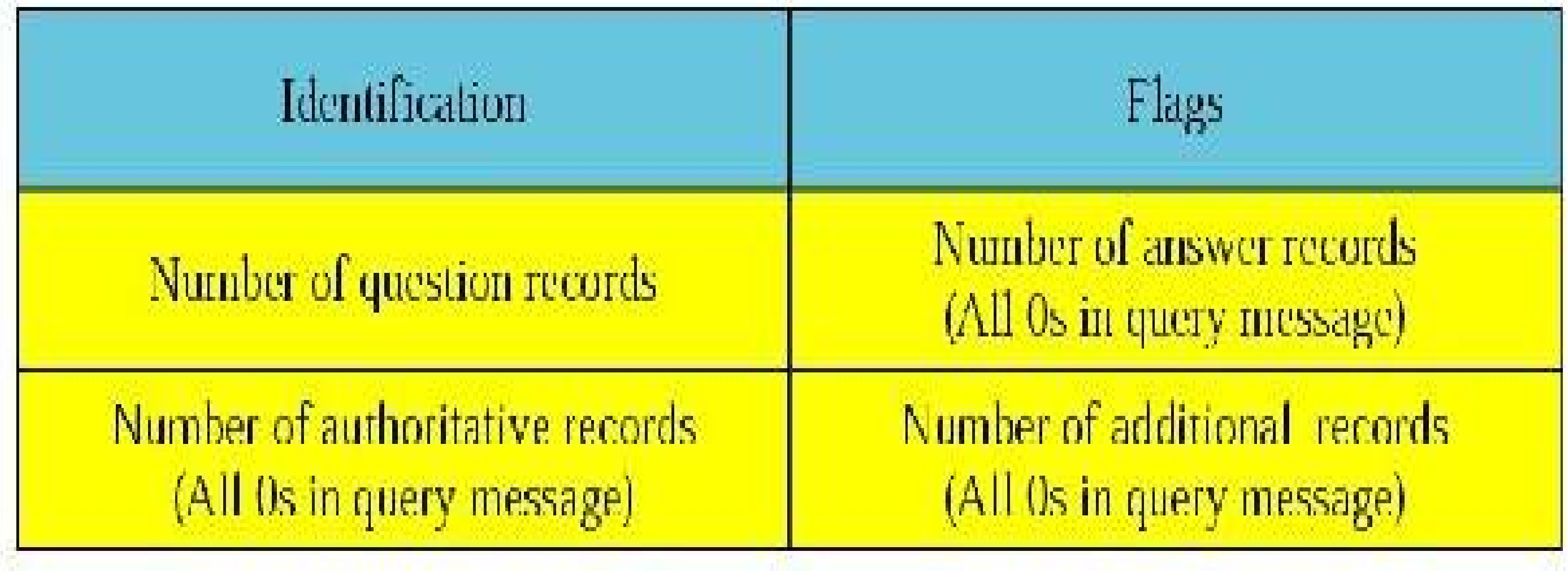
Figure : *Query and response messages*



a. Query

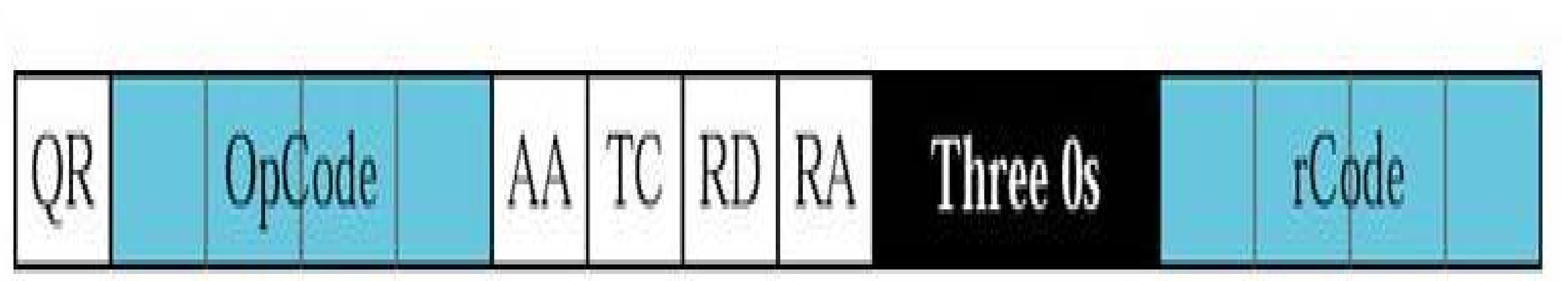
b. Response

Figure : *Header format*



Identification	Flags
Number of question records	Number of answer records (All 0s in query message)
Number of authoritative records (All 0s in query message)	Number of additional records (All 0s in query message)

Figure : *Flags field*



QR | OpCode | AA | TC | RD | RA | Three 0s | rCode

3.21 TYPES OF RECORDS

- As we saw in the previous section, two types of records are used in DNS.

- The question records are used in the question section of the query and response messages.

- The resource records are used in the answer, authoritative, and additional information sections of the response message.

Cont…

Question Record

- A question record is used by the client to get information from a server.
- This contains the domain name. Figure shows the format of a question record.
- The list below describes question record fields.

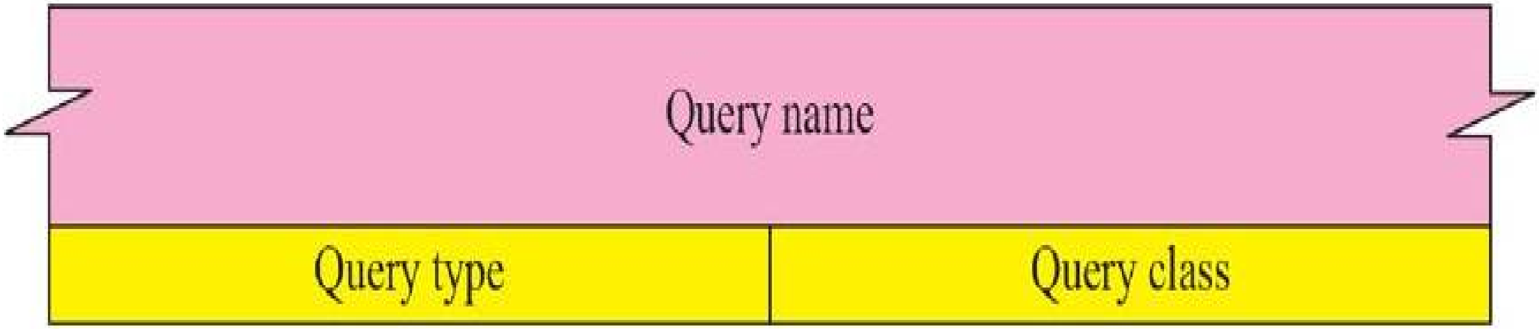
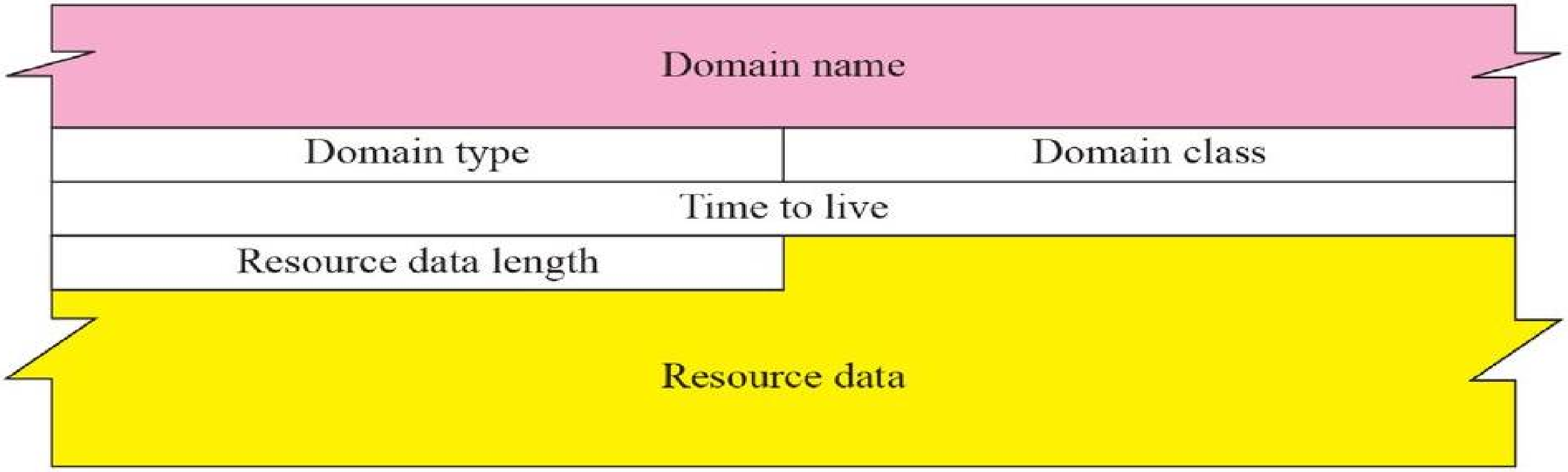


Figure : *Resource record format*



Domain name

Domain type | Domain class

Time to live

Resource data length

Resource data