Unit 4

Transport Layer Protocols

Transport Layer

Segments

Transport
End-to-End Connections
and Reliability

- Transport layer (called end-to-end layer) manages end to end (source to destination) (process to process) message delivery in a network
- Ensures that all packets of a message arrive intact and in order.
- Provides acknowledgement of successful data transmission and retransmits data if found error.
- Ensures messages are delivered error-free, in sequence, and with no losses or duplications.
- Size and complexity of a transport protocol depends on type of service it can get from network layer.
- Transport layer is at core of OSI model.
- Provides services to application layer and takes services from network layer.
- Divides message received from upper layer into packets at source and reassembles these packets again into message at destination.

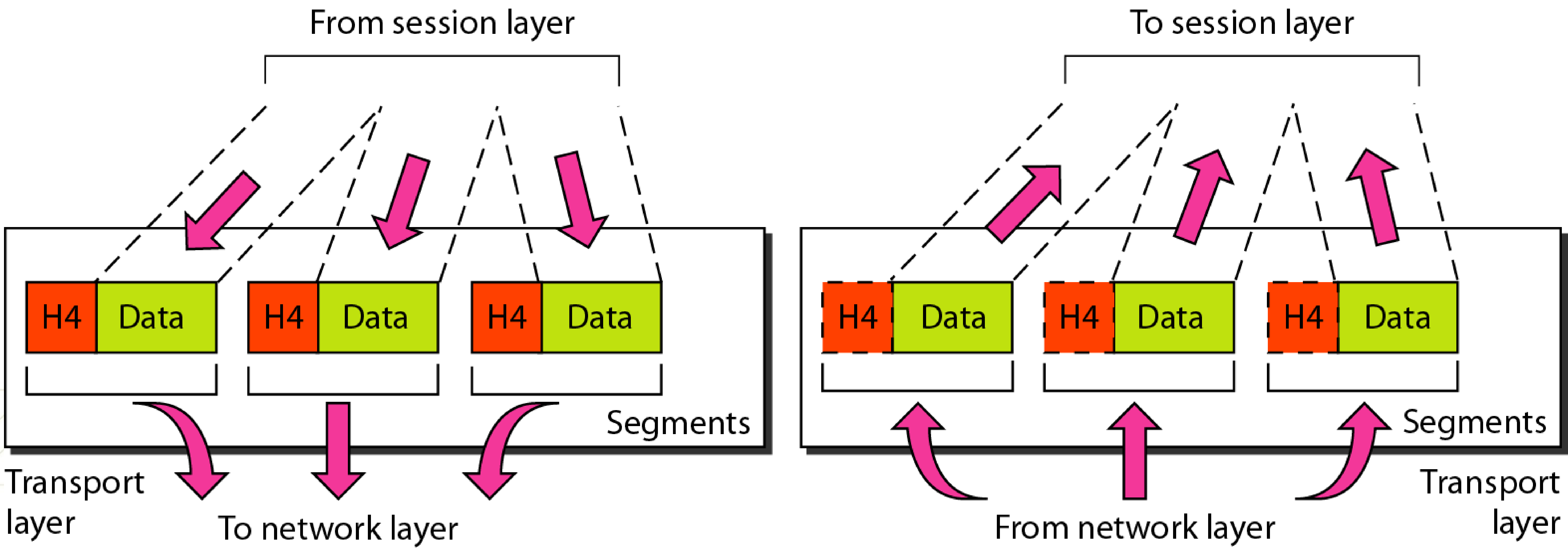
Transport Layer Services

1. **Connection Oriented Transmission**
   a. Receiving device sends an acknowledgment to source after one or group of packet is received.
   b. Also known as reliable transport method.
   c. Slower transmission method.
   d. If data sent has problems, destination requests source for retransmission by acknowledging only packets that have been received and are recognizable.
   e. Once destination computer receives all data necessary to reassemble packet, transport layer assembles data in correct sequence and then passes it up, to session layer.
2. **Connectionless Transmission**
   a. Receiver does not acknowledge receipt of a packet.
   b. Sending device assumes that packet arrive just fine.
   c. Allows for much faster communication between devices.
   d. Less reliable than connection oriented.

Transport Layer

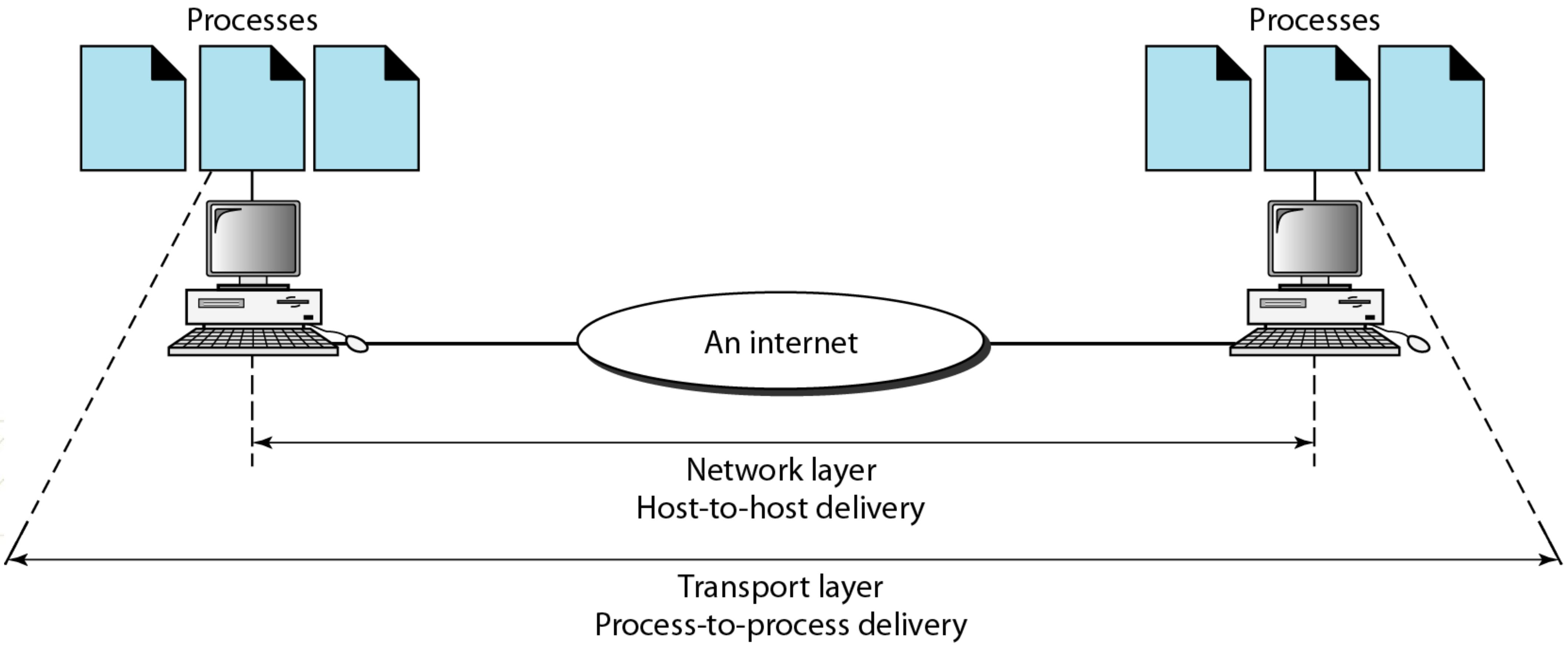
Segments

Transport
End-to-End Connections
and Reliability



Transport Layer

Reliable process-to-process delivery of a message



Functions of Transport Layer:

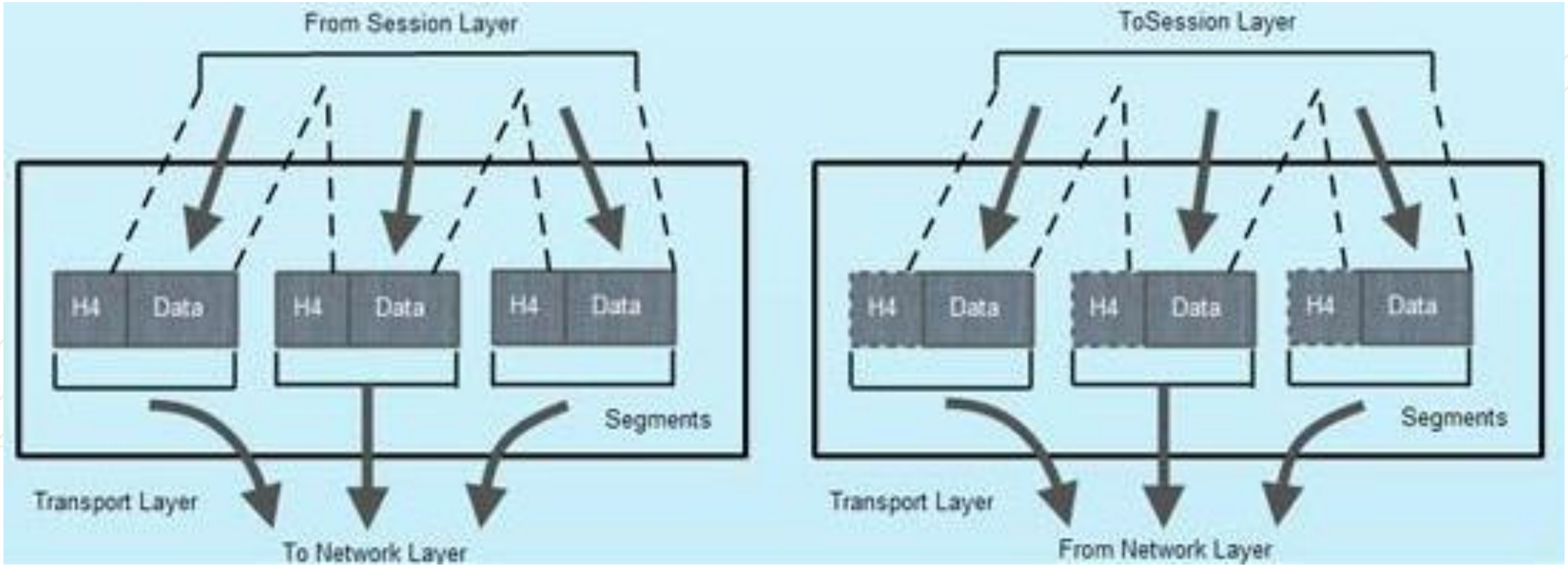
1. **Segmentation of message into packet and reassembly of packets into message**: accepts a message from (session) layer above it, splits into smaller units (if not already small enough), and passes smaller units down to network layer. Transport layer at destination station reassembles message.
2. **Message acknowledgment**: provides reliable end-to-end message delivery with acknowledgments.
3. **Message traffic control** : tells transmitting station to "back-off" when no message buffers are available.
4. **Session multiplexing** : multiplexes several message streams, or sessions onto one logical link and keeps track of which messages belong to which sessions.

Functions of Transport Layer:

5. **Service point addressing**: Purpose is to deliver message from one process running on source machine to another process running on destination machine simulataneously. To deliver message to correct process, transport layer header includes a type of address called **service point address or port address**. Thus by specifying this address, transport layer makes sure that message is delivered to correct process on destination machine.
6. **Protocols:** Protocols of transport layer are TCP, SPX, NETBIOS, ATP and NWLINK.
7. **Flow control**: Ensures that sender and receiver communicate at a rate they both can handle which prevents source from sending data packets faster than destination can handle. Here, flow control is performed end-to-end rather than across a link.

Functions of Transport Layer:

8. **Error control**: Error control is performed end-to-end rather than across a single link. Sending transport layer ensures that entire message arrives at receiving transport layer without loss, error (damage, or duplication). Error correction is achieved through retransmission.



TRANSPORT LAYER PROTOCOLS – Design Issues

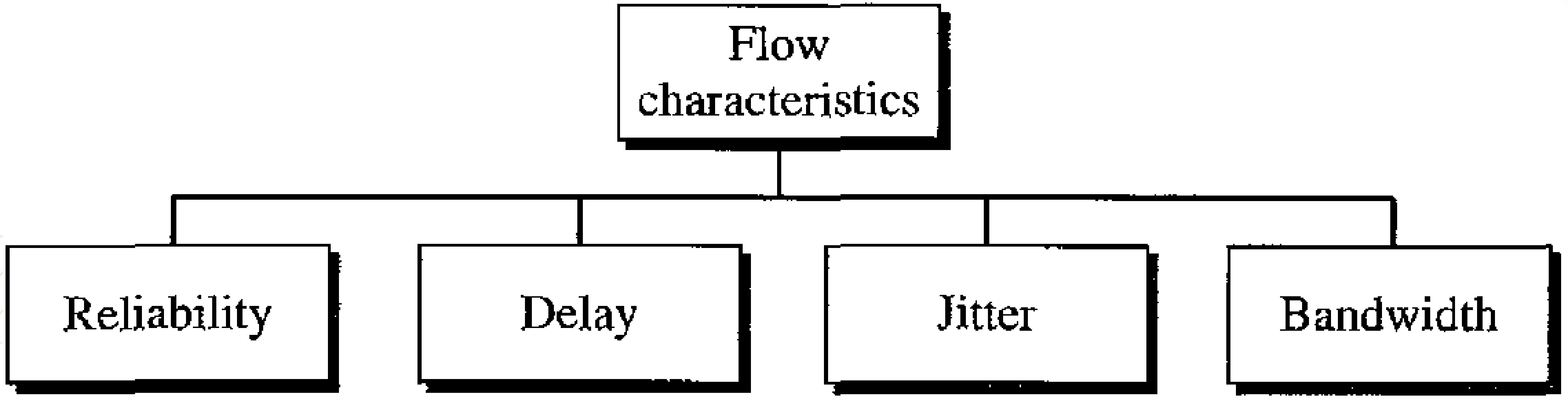
1. **Reliability:** To ensure that data transmitted between hosts is delivered reliably. TCP provides reliable data delivery by using acknowledgment and retransmission mechanisms. In contrast, UDP does not provide reliable delivery and is used in applications where occasional loss of data is acceptable.
2. **Flow Control:** Process of regulating the flow of data between two network nodes to prevent the receiver from being overwhelmed with data. Flow control results in a better network utilization by avoiding packet loss. TCP uses flow control mechanisms to ensure that the sender does not send data faster than the receiver can process it.
3. **Congestion Control:** Process of preventing network from becoming congested. Achieved by controlling rate at which data is sent by sender. TCP uses effective congestion control to prevent packet loss, improve network performance, and avoid network collapse.
4. **Multiplexing and Demultiplexing:** May have multiple connections running on a single physical network. Multiplexing combines multiple connections into a single stream while demultiplexing splits single stream into multiple connections.

TRANSPORT LAYER PROTOCOLS – Design Issues

5. **Connection Establishment and Termination:** Hosts exchange information for connection establishment. Before data transfer, a connection is established which gets terminated after completion of data transfer between sending and receiving hosts. TCP uses a three-way handshake mechanism to establish a connection and a four-way handshake to terminate a connection.
6. **Quality of Service:** Ensures to provide an acceptable level of QoS for traffic. Different types of traffic have different QoS requirements, like guarantee of minimum bandwidth or maximum delay.
7. **Security:** Critical issue – as any vulnerability can compromise confidentiality or integrity of transmitted data. Encryption and authentication mechanisms can be used to secure data.
8. **Performance:** High performance and low latency ensures efficient communication between network nodes. Achieved through optimized algorithms and efficient use of network resources. Measured by various metrics such as throughput, delay, and error rate.

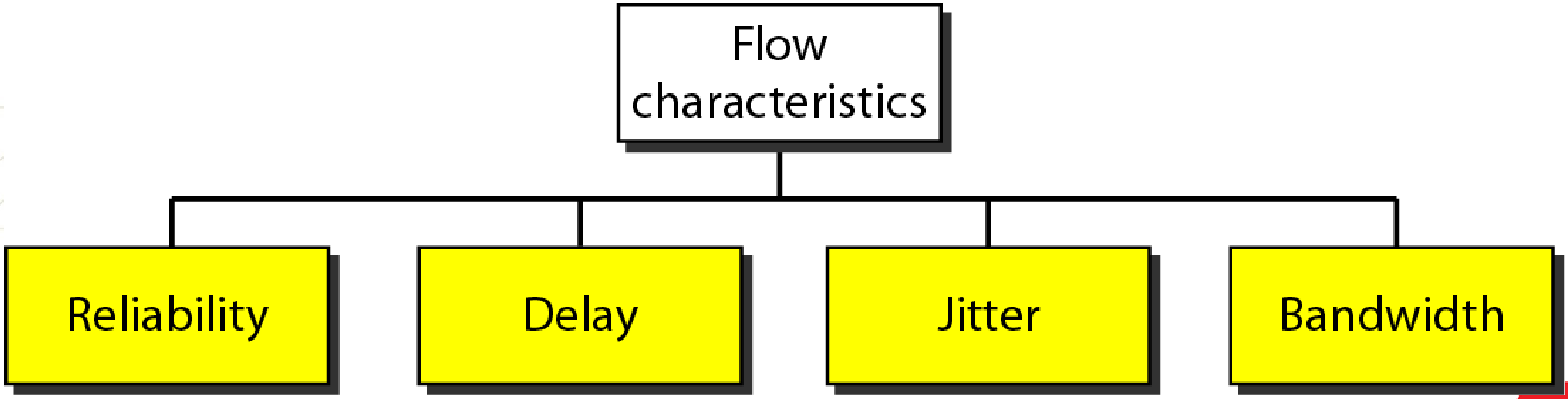
Quality of Service

- QoS is critical for applications that require reliable and timely data delivery.
- Transport Layer Protocols that do not provide QoS may cause degraded performance, user frustration, increased latency, and dropped packets.
- Quality of service is something a flow seeks to attain



Factors affecting QoS

1. **Reliability:** Lack of reliability means losing a packet or acknowledgment, which entails retransmission.
2. **Delay:** Applications can tolerate delay in different degrees.
3. **Jitter:** Jitter is variation in delay for packets belonging to same flow. High jitter means difference between delays is large; low jitter means variation is small.
4. **Bandwidth:** Different applications need different bandwidths



Techniques to improve QoS

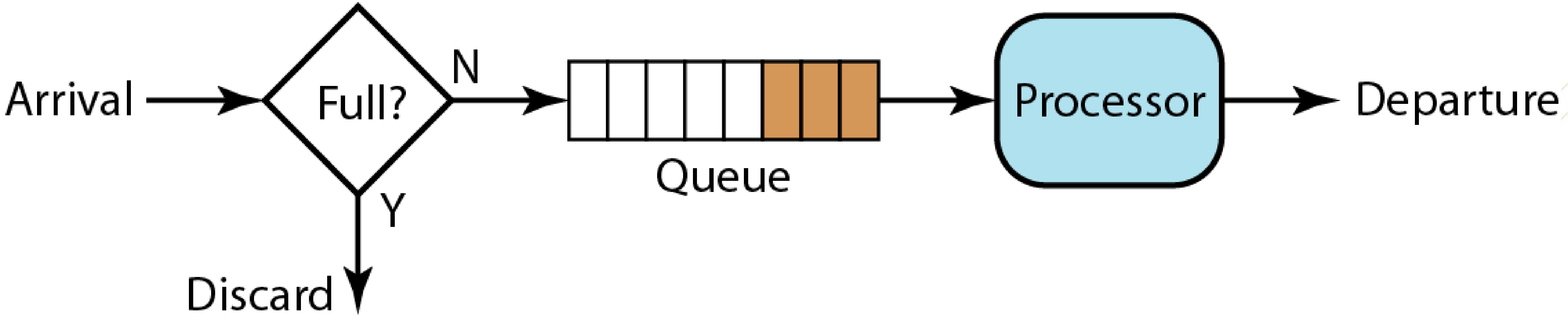
1. **Scheduling**
2. **Traffic Shaping**
3. **Resource Reservation**
4. **Admission Control**

Techniques to improve QoS

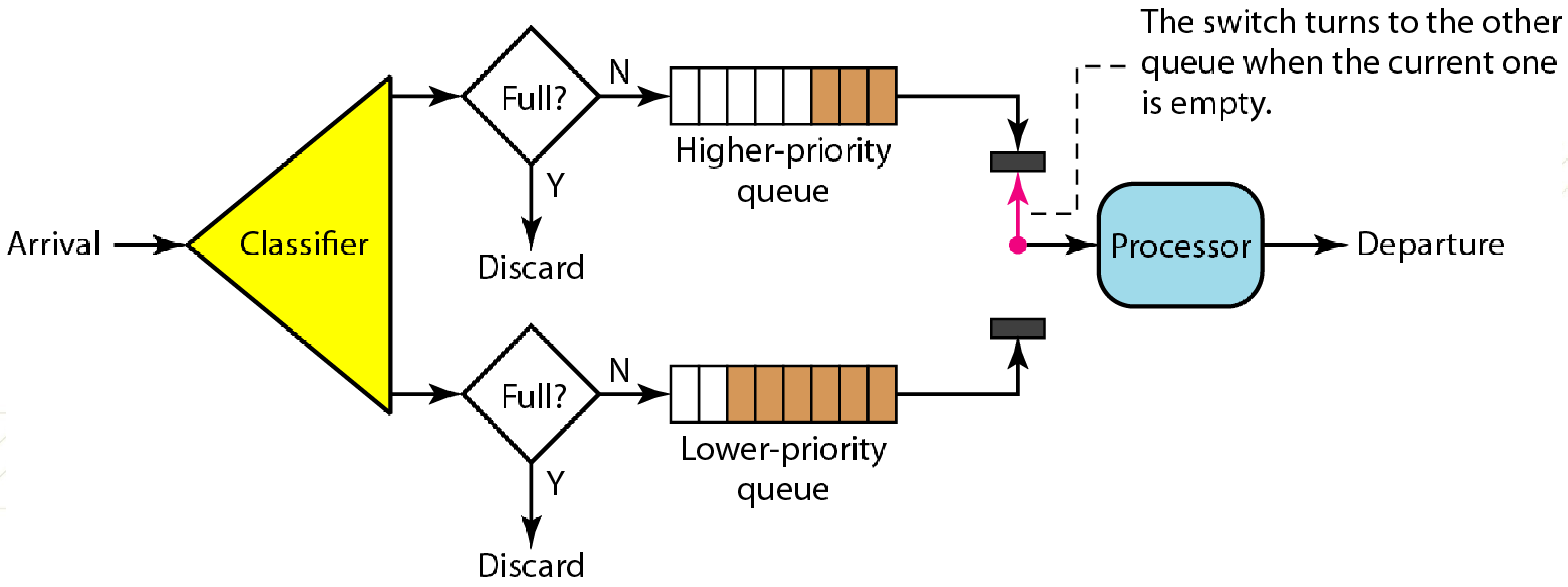
Scheduling:

- Packets from different flows arrive at a switch or router for processing.
- Good scheduling technique treats different flows in a fair and appropriate manner.
- Multiple techniques designed to improve QoS are
    1. FIFO queuing,
    2. priority queuing, and
    3. weighted fair queuing.

FIFO queue

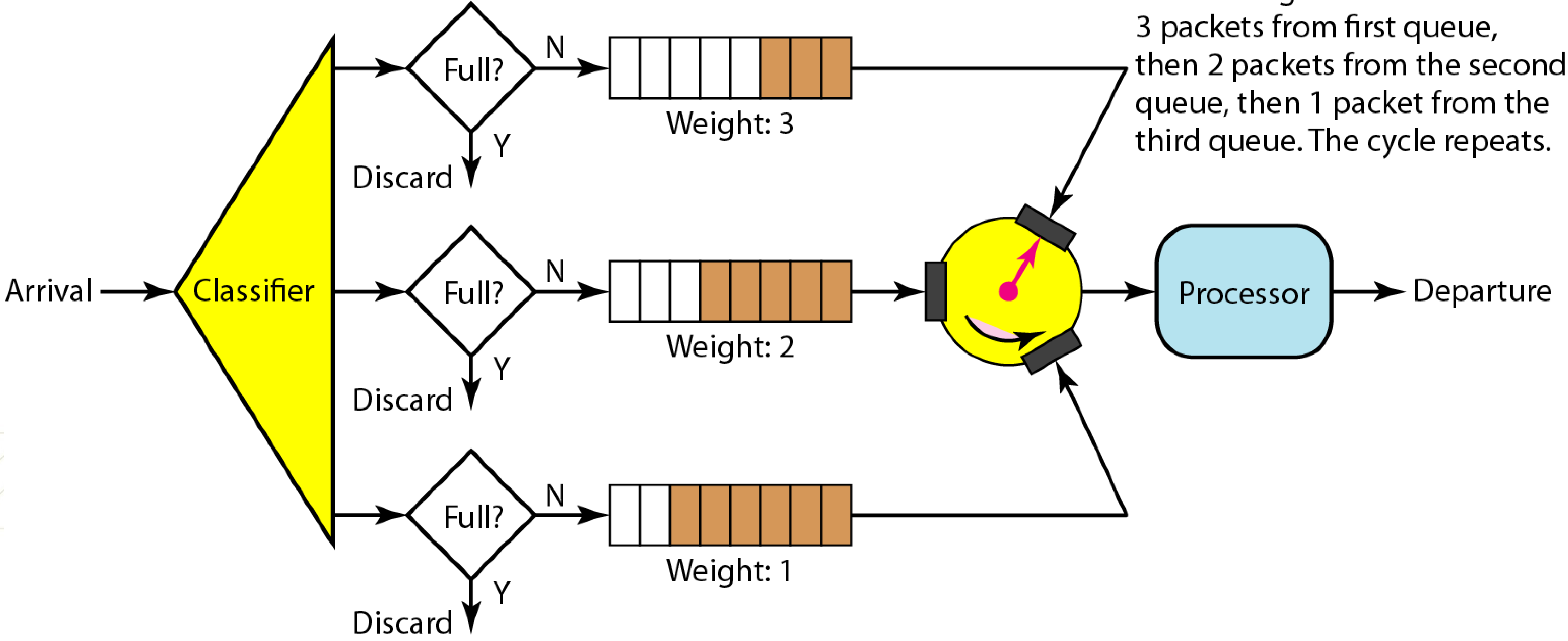


Priority queuing



Weighted fair queuing

The turning switch selects 3 packets from first queue, then 2 packets from the second queue, then 1 packet from the third queue. The cycle repeats.

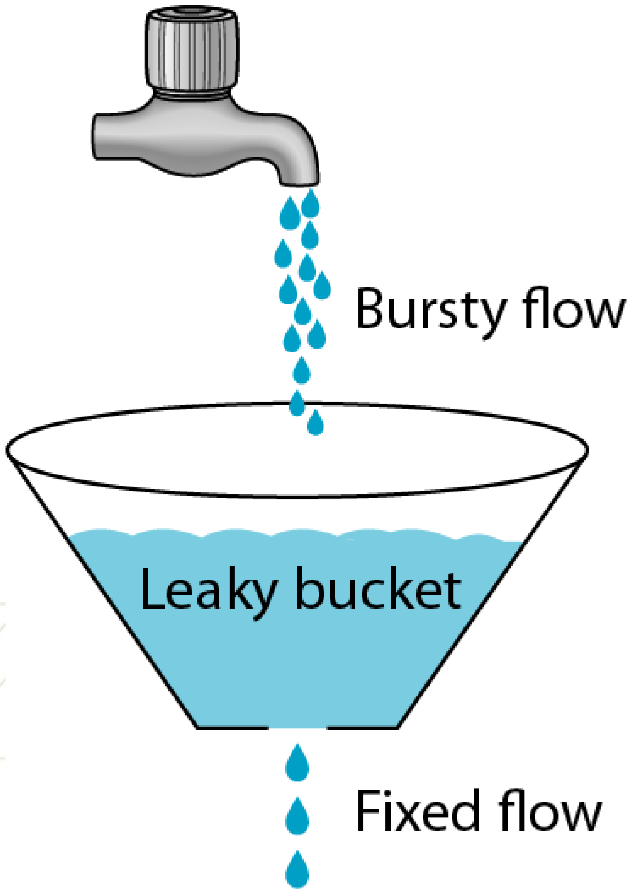


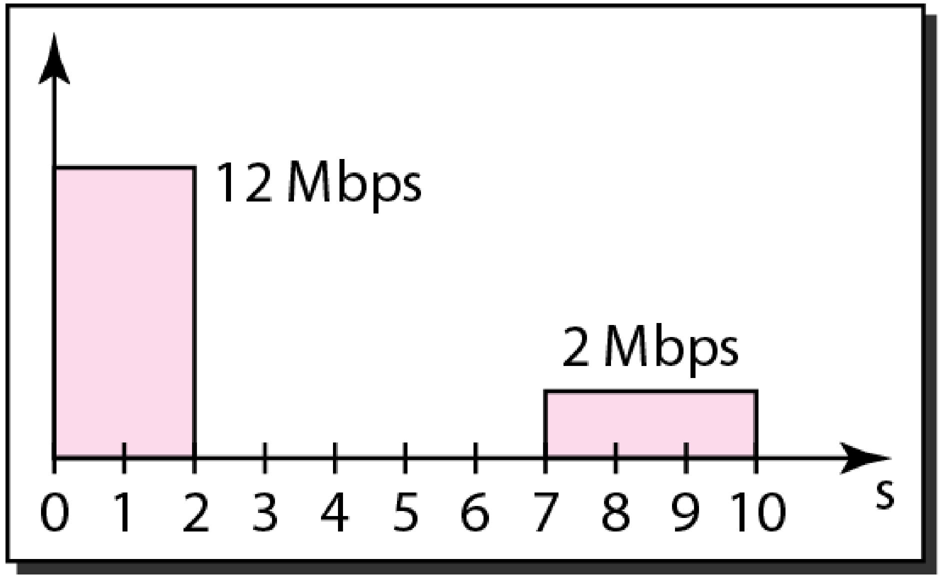
Techniques to improve QoS

Traffic Shaping:

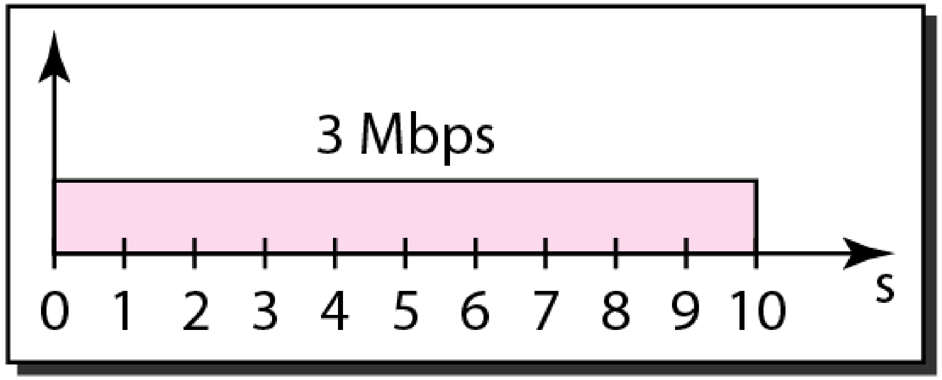
- Traffic is shaped before it enters network.
- Controls rate at which packets are sent (not just how many).
- Used in ATM and Integrated Services networks.
- At connection set-up time, sender and carrier negotiate a traffic pattern (shape).
- Two traffic shaping algorithms are
  1. Leaky Bucket
  2. Token Bucket

Leaky bucket





Bursty data



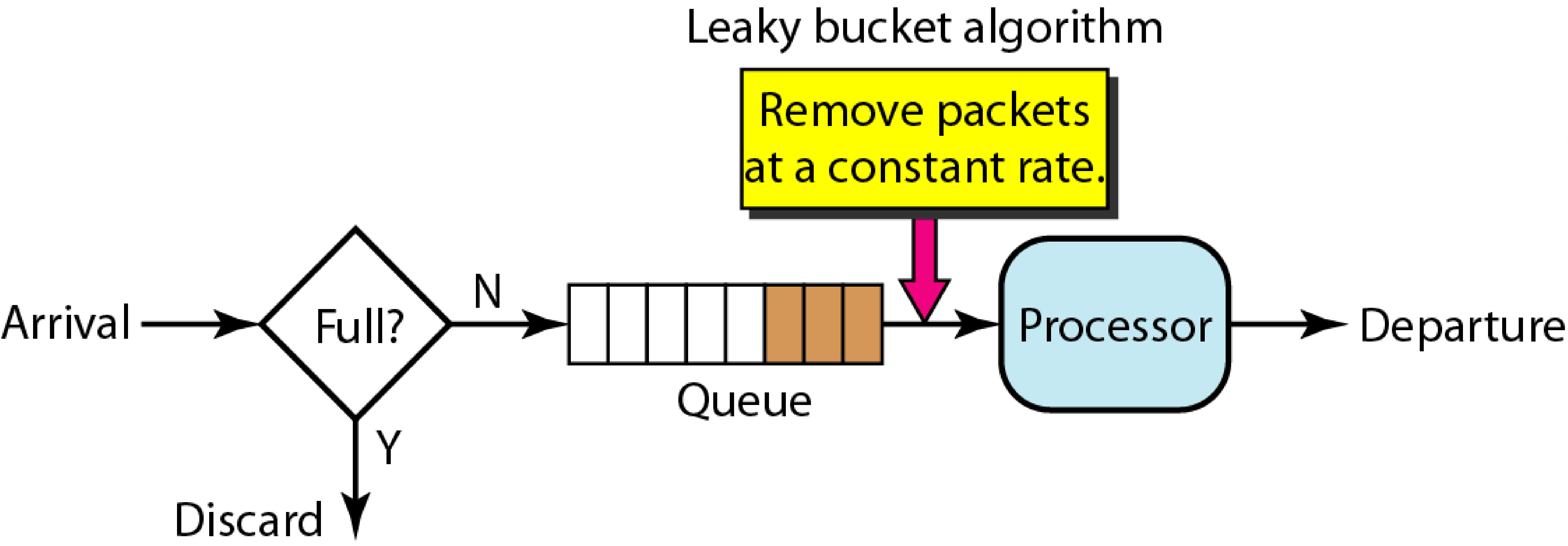
Fixed-rate data

- Leaky bucket algorithm shapes bursty traffic into fixed-rate traffic by averaging data rate.
- It may drop packets if bucket is full.

Leaky bucket

- Enforces a constant output rate (average rate) regardless of burstiness of input.
- Implemented as a singleserver queue with constant service time.
- If bucket (buffer) overflows then packets are discarded.
- Does nothing when input is idle.
- Host injects one packet per clock tick onto network.
- This results in a uniform flow of packets, smoothing out bursts and reducing congestion.
- When packets are same size one packet per tick is okay.
- For variable length packets, allows fixed number of bytes per tick.
- E.g. 1024 bytes per tick will allow one 1024-byte packet or two 512-byte packets or four 256-byte packets on 1 tick

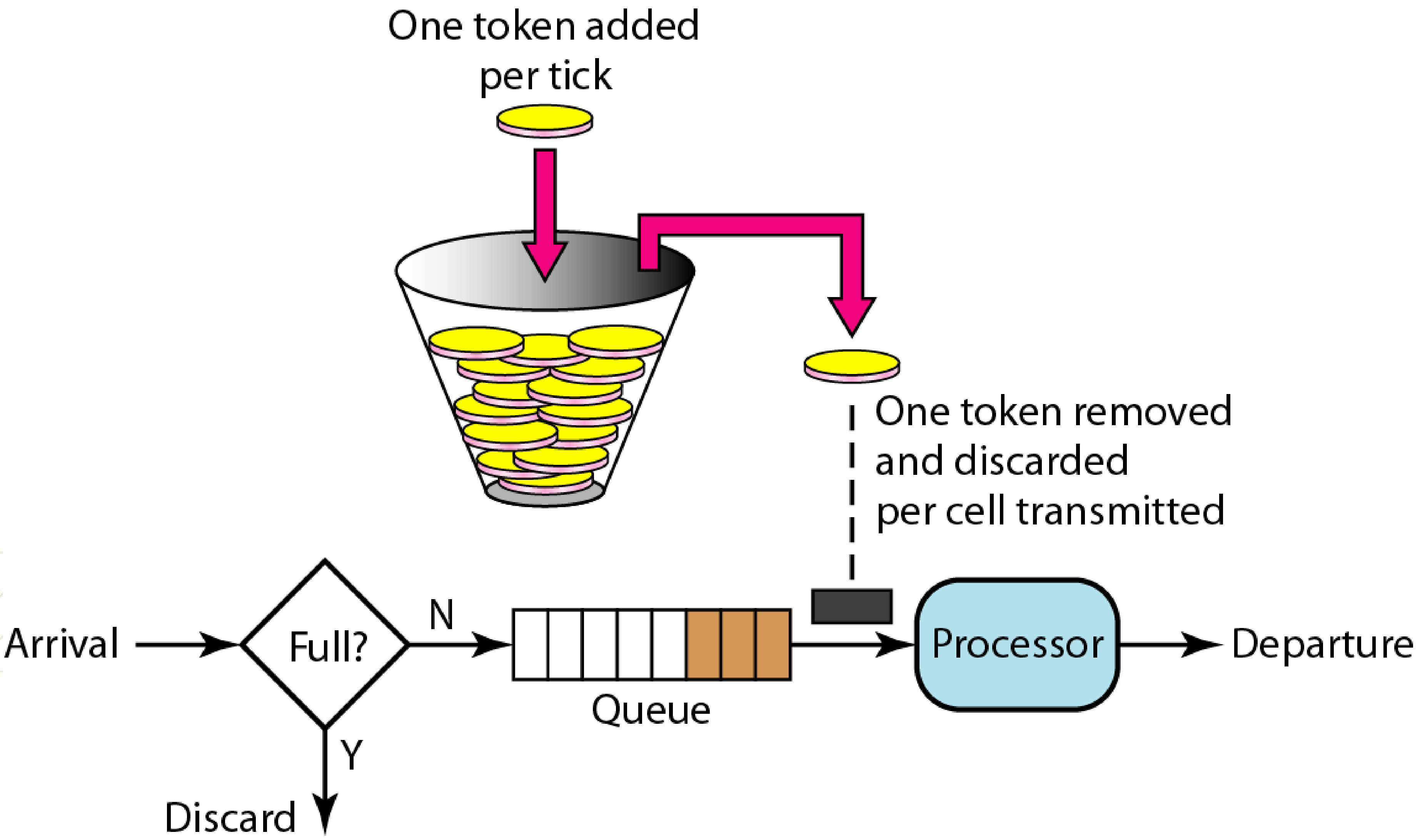
Leaky bucket implementation



Token Bucket Algorithm

- TBA allows output rate to vary, depending on size of burst.
- Here, bucket holds tokens.
- To transmit a packet, host must capture and destroy one token.
- Tokens are generated by a clock at rate of one token every Δt sec.
- Idle hosts can capture and save up tokens (up to max. size of bucket) in order to send larger bursts later.

Token Bucket Algorithm



Token bucket allows bursty traffic at a regulated maximum rate.

Techniques to improve QoS

Resource Reservation

- Flow of data needs resources such as a buffer, bandwidth, CPU time, and so on.
- QoS is improved if these resources are reserved beforehand.

Admission Control

- Refers to mechanism used by a router, or a switch, to accept or reject a flow based on predefined parameters called flow specifications.
- Before a router accepts a flow for processing, it checks flow specifications to see if its capacity and previous commitments to other flows can handle new flow.

IPv4 ADDRESSES

- IPv4 address is a 32-bit address that uniquely and universally defines connection of device (for example, a computer or a router) to Internet.
- IPv4 addresses are unique that each address defines only one, connection to Internet.
- Two devices on Internet can never have same address at same time.
- An address may be assigned to a device for a time period and then taken away and assigned to another device.
- If device operating at network layer has m connections to Internet, it needs to have m addresses.

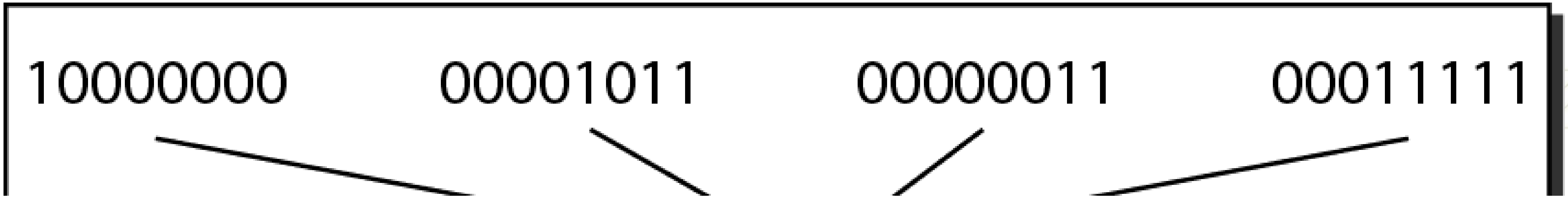
IPv4 Address Space

- Address space is total number of addresses used by protocol.
- If a protocol uses *N* bits to define an address, address space is 2^N because each bit can have two different values (0 or 1) and *N* bits can have 2^N values.
- IPv4 uses 32-bit addresses, which means that address space is **2^{32} or 4,294,967,296**.

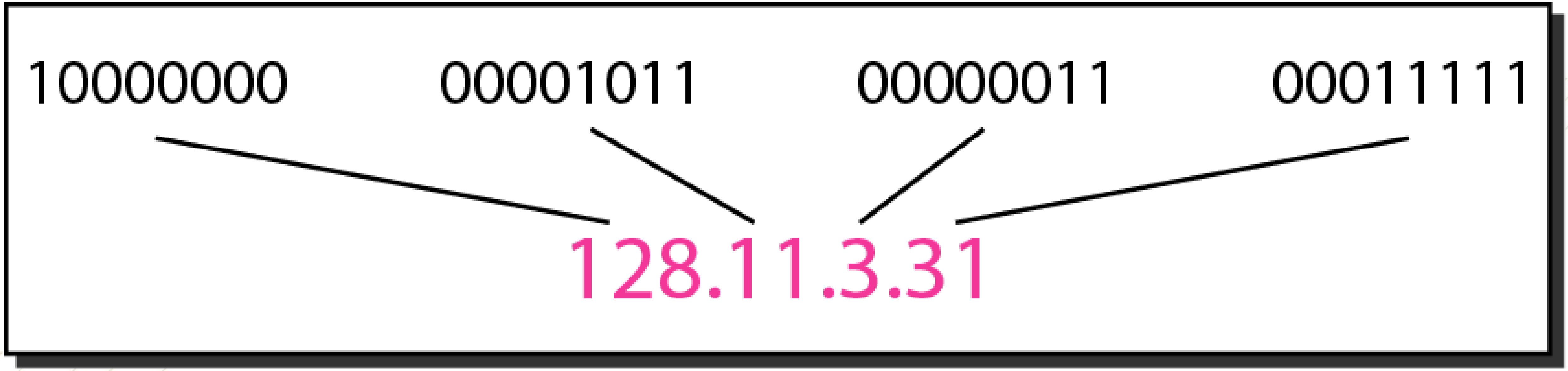
IPv4 Notations

1. **Binary Notation:** Displayed as 32 bits.
   - Each octet is referred as byte.
   - IPv4 address referred to as a 32-bit address or 4-byte address.
   - Example: 01110101   10010101   00011101   00000010
2. **Dotted-Decimal Notation:**
   - To make address more compact and easier to read, Internet addresses are written in decimal form with a decimal point (dot) separating bytes.
   - Example: 117.149.29.2

Question: Find decimal notation of binary equivalent



Solution: Find decimal notation of binary equivalent



Question: Find decimal notation of binary equivalent

- Change following IPv4 addresses from binary notation to dotted-decimal notation.

a) 10000001 00001011 00001011 11101111

b) 11000001 10000011 00011011 11111111

Solution: Find decimal notation of binary equivalent

- Change following IPv4 addresses from binary notation to dotted-decimal notation.

a) 10000001 00001011 00001011 11101111

b) 11000001 10000011 00011011 11111111

- **Solution:**

a) 129.11.11.239

b) 193.131.27.255

Question: Find binary notation of Decimal equivalent

- Change following IPv4 addresses from dotted-decimal notation to binary notation.

a) 111.56.45.78

b) 221.34.7.82

Solution: Find binary notation of decimal equivalent

- Change following IPv4 addresses from dotted-decimal notation to binary notation.

a) 111.56.45.78

b) 221.34.7.82

- **Solution:**

a) 01101111 00111000 00101101 01001110

b) 11011101 00100010 00000111 01010010

Question: Find errors if any

- Find error, if any, in the following IPv4 addresses.

1. a. 111.56.045.78
2. b. 221.34.7.8.20
3. c. 75.45.301.14
4. d. 11100010.23.14.67

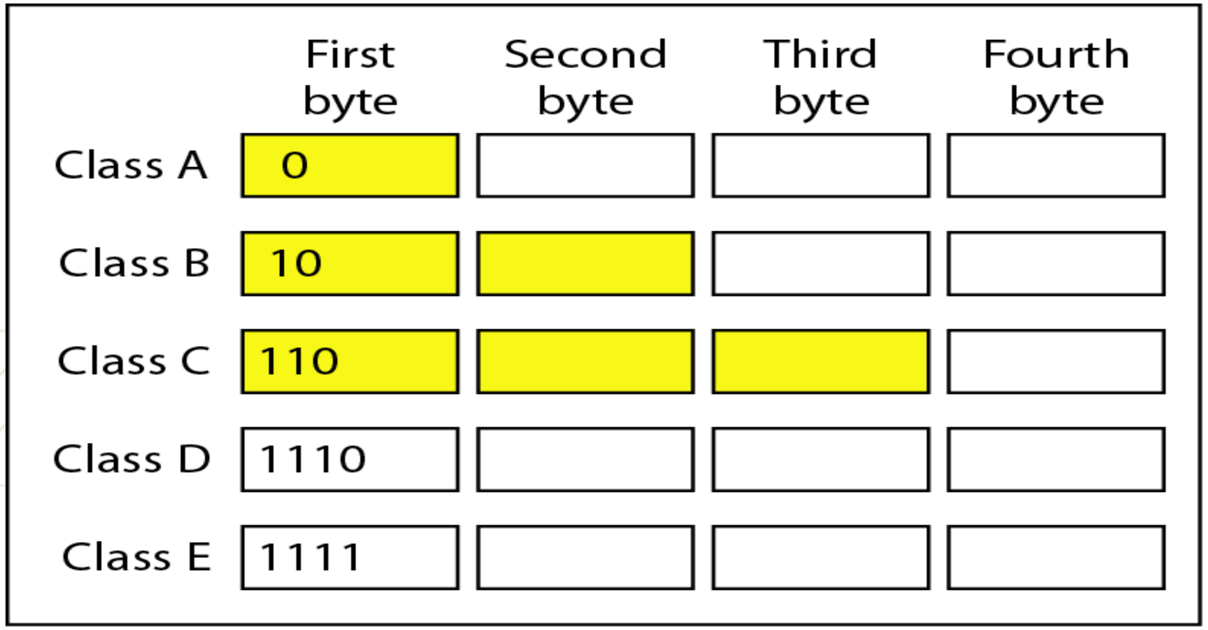
Solution: Find errors if any

- Find error, if any, in the following IPv4 addresses.

1. 111.56.045.78 : No leading zero allowed (045).
2. 221.34.7.8.20: no more than four numbers in an IPv4 address.
3. 75.45.301.14 : Each number needs to <=255 (301 is outside this range)
4. 11100010.23.14.67 : mixture of binary notation and dotted-decimal notation not allowed

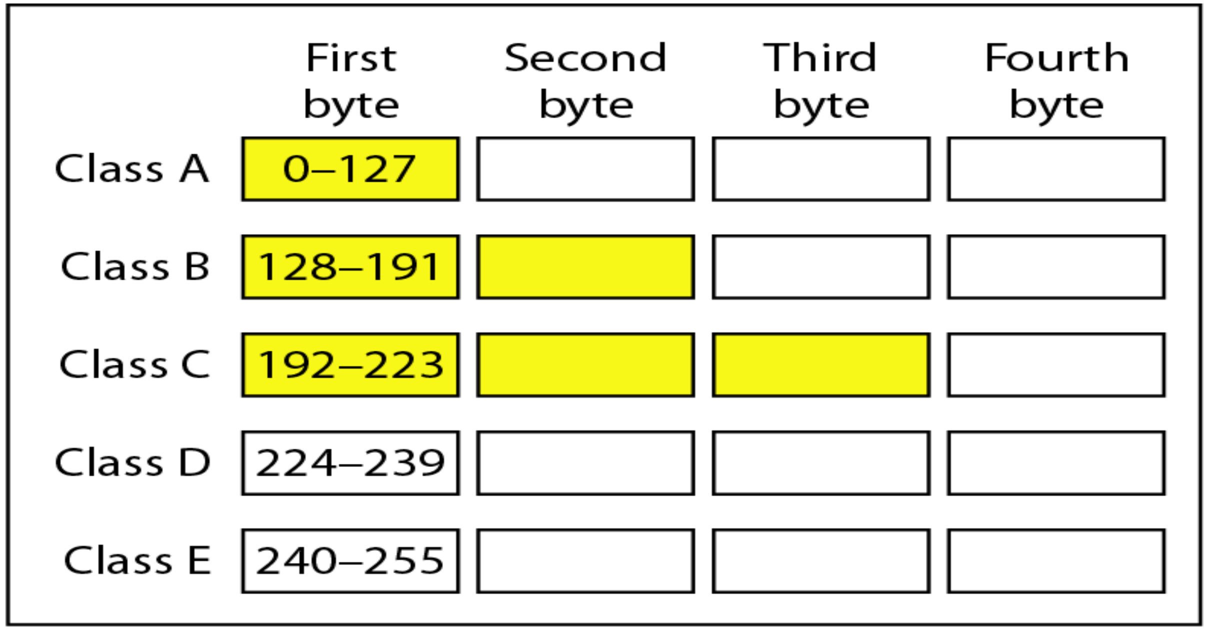
IPv4 Notations

- In classful addressing, address space is divided into five classes: **A, B, C, D, and E.**
- For binary notation address, first few bits defines class of address.
- For decimal-dotted notation address, first byte defines class



	First byte	Second byte	Third byte	Fourth byte
Class A	🟨 0			
Class B	🟨 10	🟨		
Class C	🟨 110	🟨	🟨	
Class D	1110			
Class E	1111			

a. Binary notation



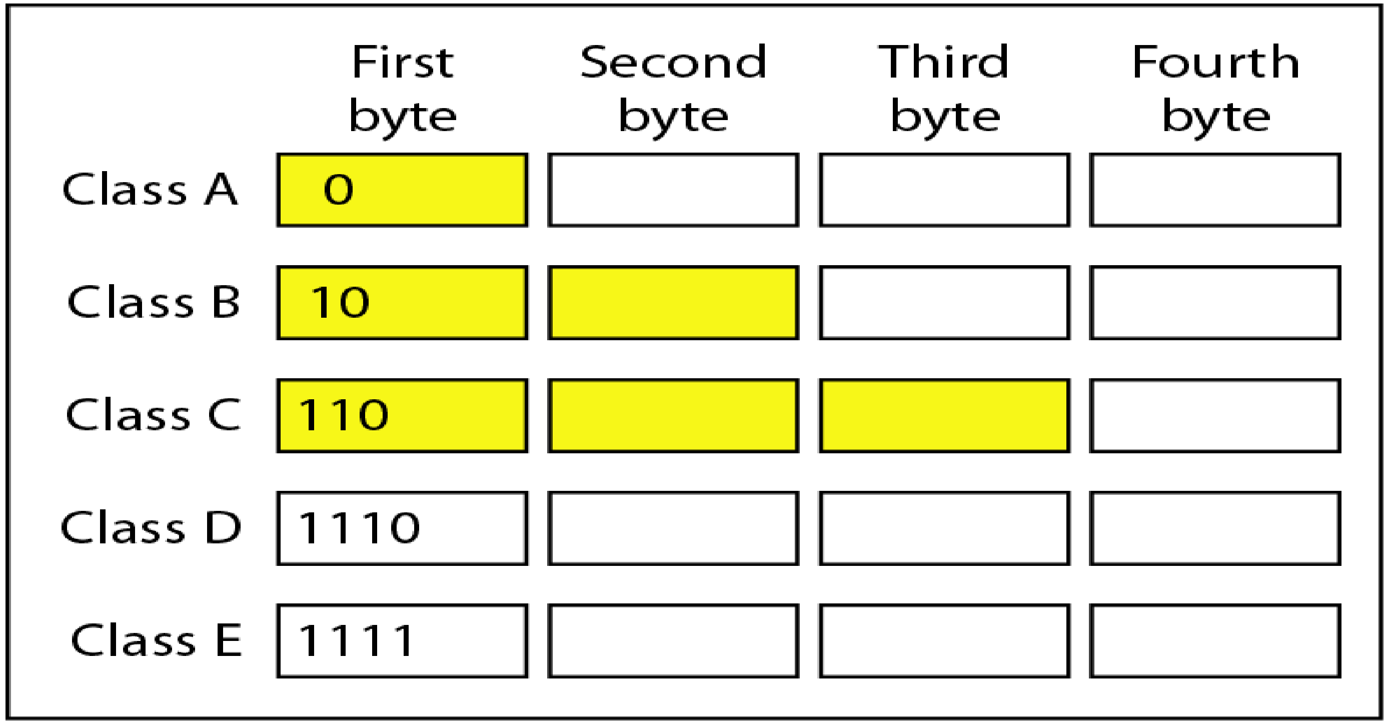
	First byte	Second byte	Third byte	Fourth byte
Class A	🟨 0–127			
Class B	🟨 128–191	🟨		
Class C	🟨 192–223	🟨	🟨	
Class D	224–239			
Class E	240–255			

b. Dotted-decimal notation

Question: Find class of each address.

- Find class of each address.

A. 00000001 00001011 00001011 11101111 :



	First byte	Second byte	Third byte	Fourth byte
Class A	0			
Class B	10	🟨		
Class C	110	🟨	🟨	
Class D	1110			
Class E	1111			

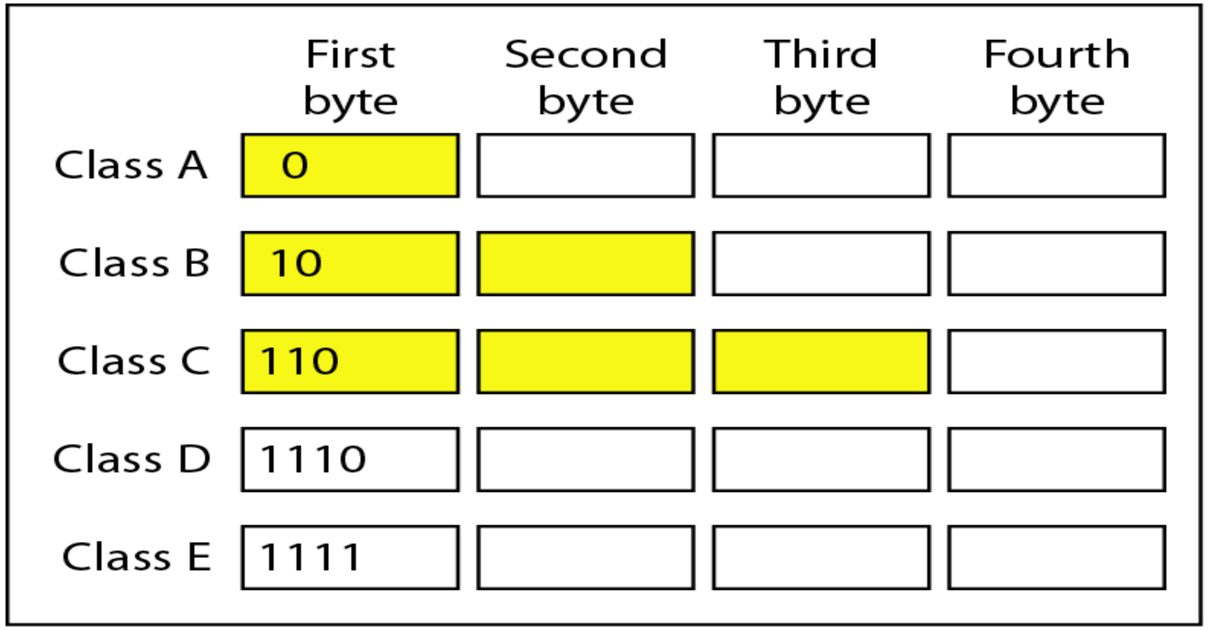
a. Binary notation

Question: Find class of each address.

- Find class of each address.

A. 00000001 00001011 00001011 11101111 : First bit is O. So, Class A address

B. 11000001 10000011 00011011 11111111



	First byte	Second byte	Third byte	Fourth byte
Class A	🟨 0			
Class B	🟨 10	🟨		
Class C	🟨 110	🟨	🟨	
Class D	1110			
Class E	1111			

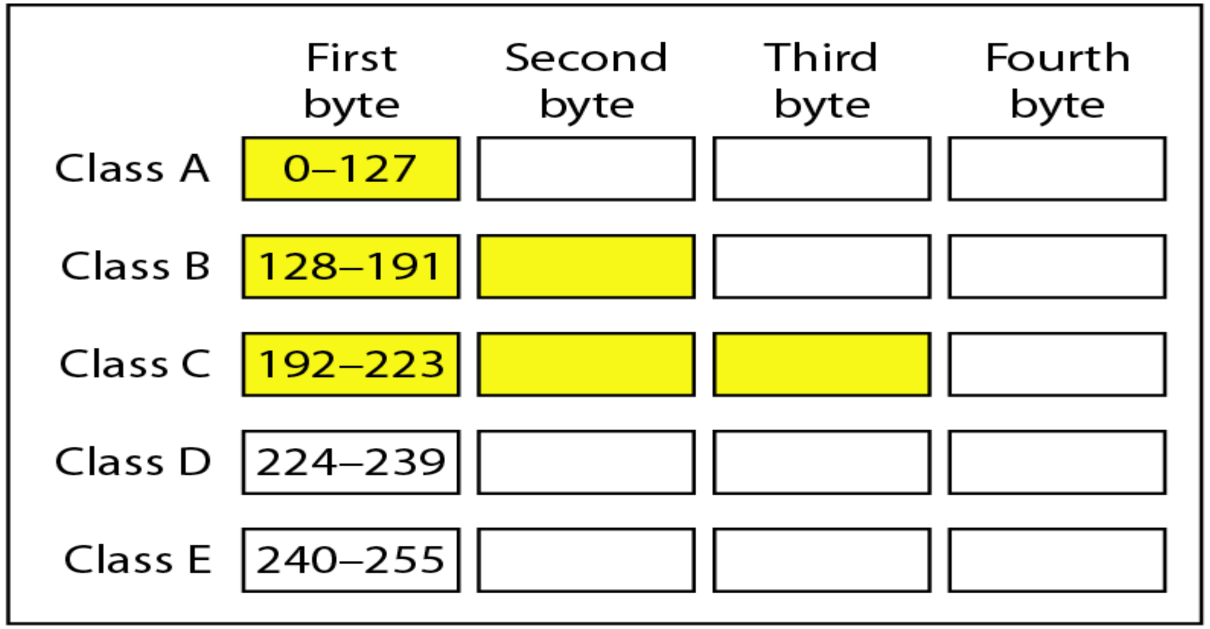
a. Binary notation

Question: Find class of each address.

- Find class of each address.

A. 00000001 00001011 00001011 11101111 : First bit is O. So, **Class A address**

B. 11000001 10000011 00011011 11111111: First 2 bits are 1; third bit is O. So, **class C address**



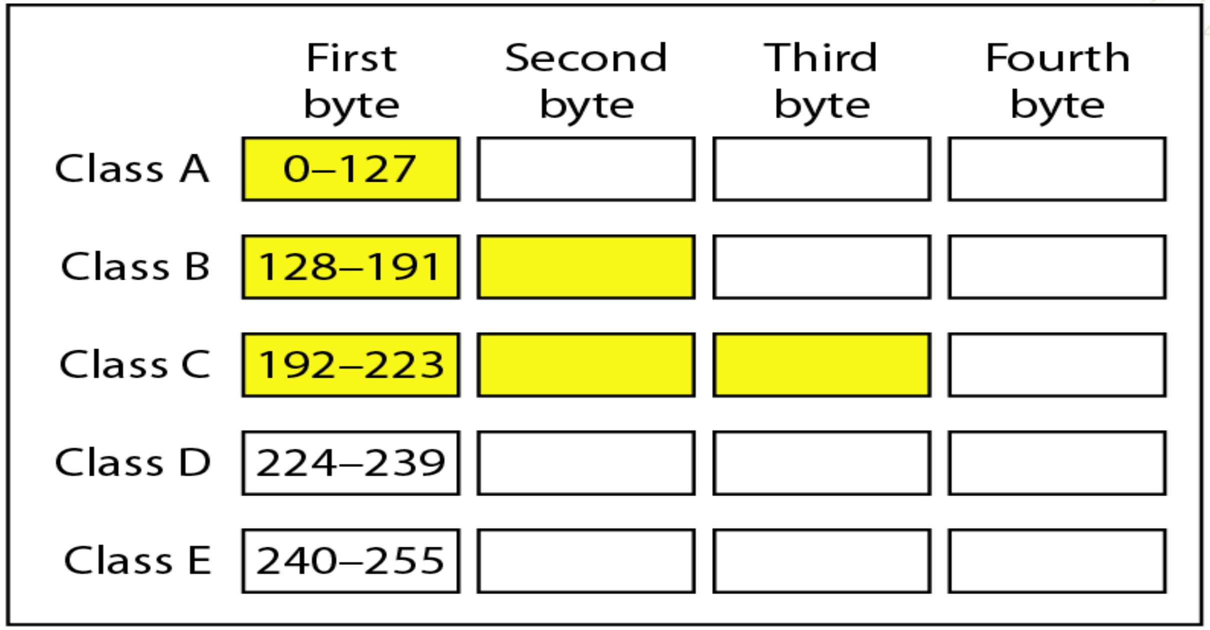
	First byte	Second byte	Third byte	Fourth byte
Class A	0–127			
Class B	128–191	🟨		
Class C	192–223	🟨	🟨	
Class D	224–239			
Class E	240–255			

b. Dotted-decimal notation

Question: Find class of each address.

- Find class of each address.

A. 14.23.120.8 :



	First byte	Second byte	Third byte	Fourth byte
Class A	0–127			
Class B	128–191			
Class C	192–223			
Class D	224–239			
Class E	240–255			

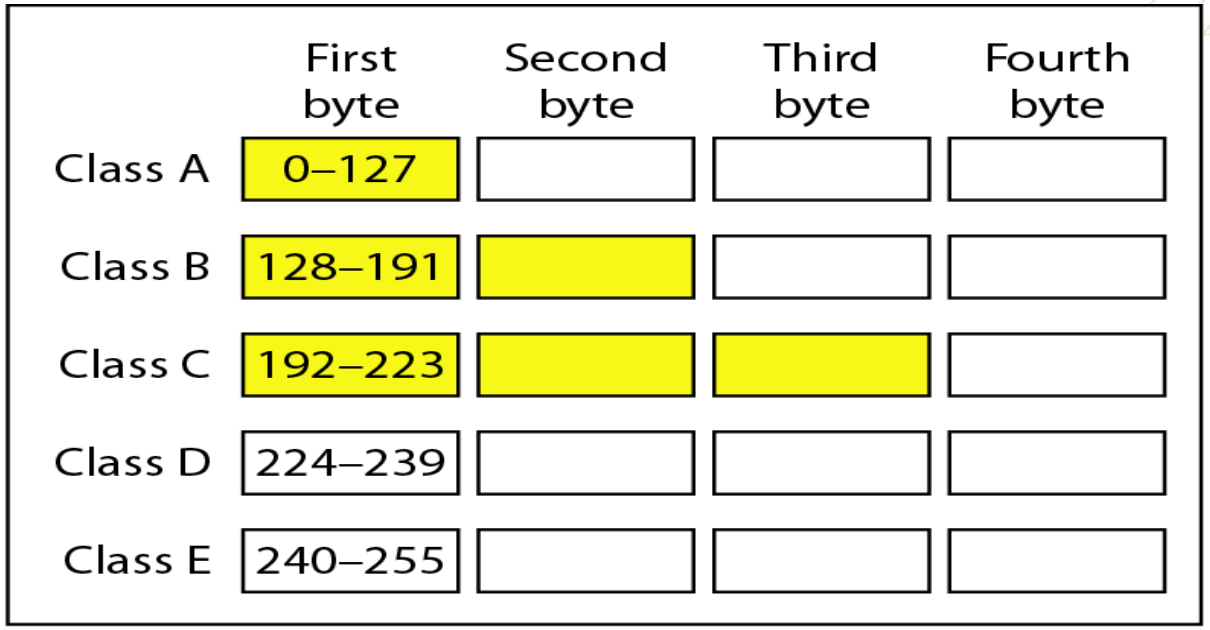
b. Dotted-decimal notation

Question: Find class of each address.

- Find class of each address.

A. 14.23.120.8 : First byte is 14 (between 0 and 127); So, **Class A address**

B. 252.5.15.111 :



	First byte	Second byte	Third byte	Fourth byte
Class A	0–127			
Class B	128–191	🟨		
Class C	192–223	🟨	🟨	
Class D	224–239			
Class E	240–255			

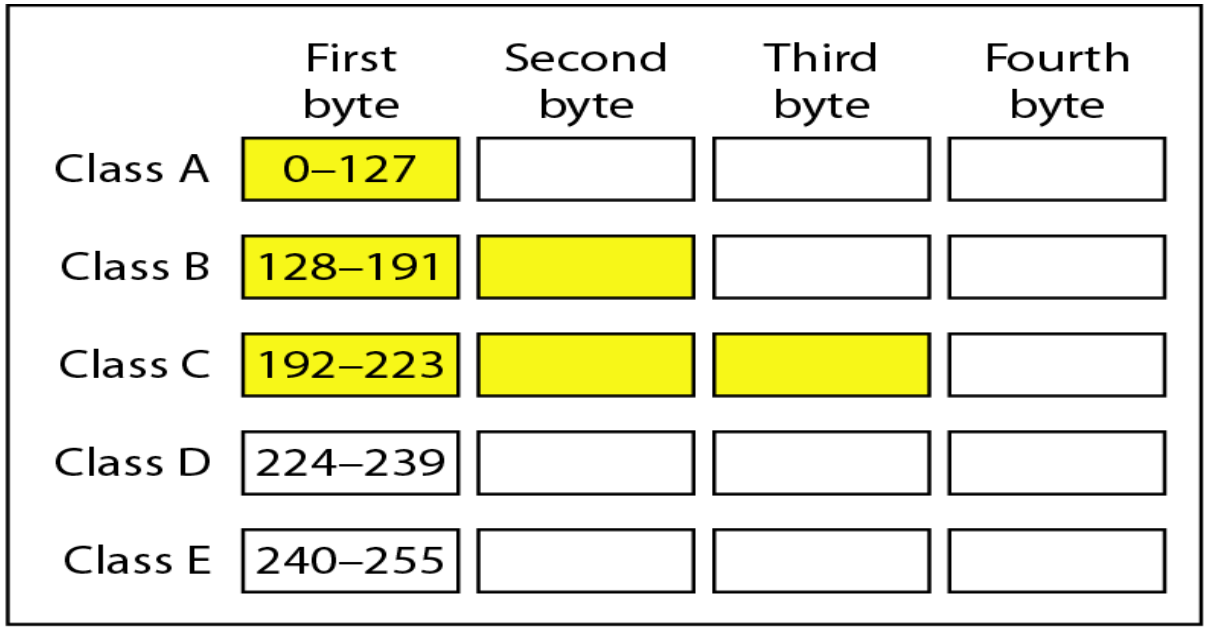
b. Dotted-decimal notation

Question: Find class of each address.

- Find class of each address.

A. 14.23.120.8 : First byte is 14 (between 0 and 127); So, **Class A address**

B. 252.5.15.111 : First byte is 252 (between 240 and 255); So, **class E address**

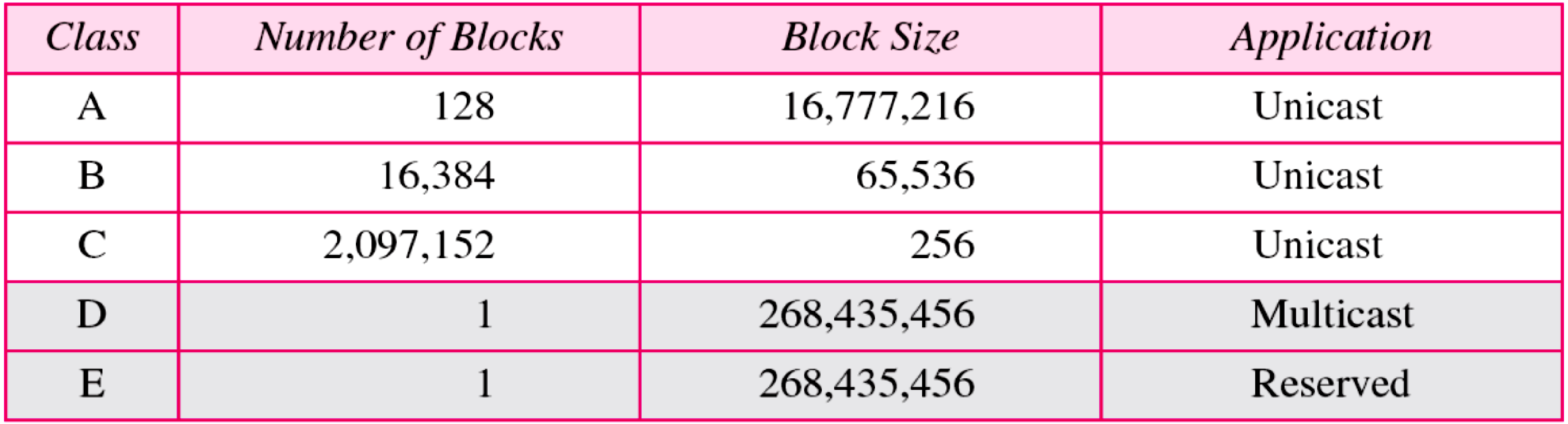


	First byte	Second byte	Third byte	Fourth byte
Class A	0–127			
Class B	128–191	🟨		
Class C	192–223	🟨	🟨	
Class D	224–239			
Class E	240–255			

b. Dotted-decimal notation

Classful Addressing : Problem of Classes and Blocks

- One problem with classful addressing is that each class is divided into a fixed number of blocks with each block having a fixed size

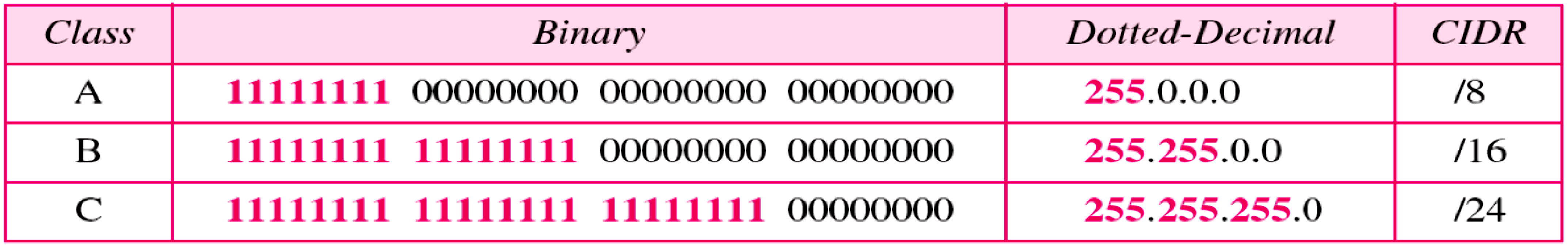


Class	*Number of Blocks*	*Block Size*	*Application*
A	128	16,777,216	Unicast
B	16,384	65,536	Unicast
C	2,097,152	256	Unicast
D	1	268,435,456	Multicast
E	1	268,435,456	Reserved

- In classful addressing, a large part of the available addresses were wasted

Classful Addressing : Netid and Hostid

- In classful addressing, IP address in class A, B, or C is divided into netid and hostid.
- Varying lengths, depending on class of address.
- Netid is in color, hostid is in white.

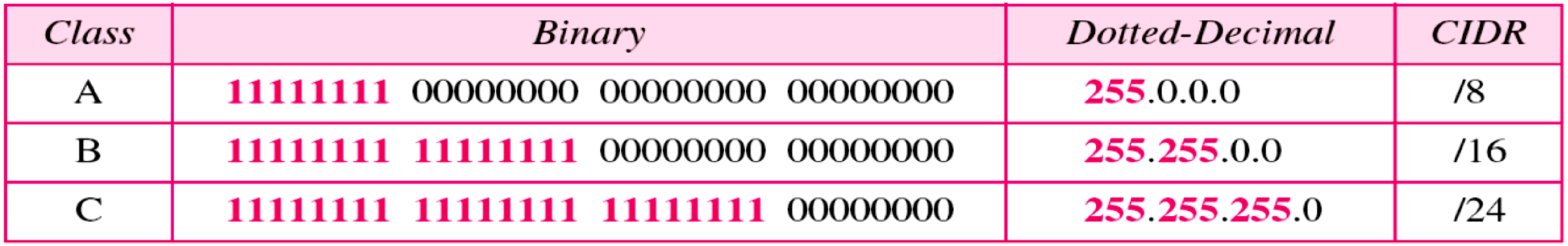


Class	*Binary*	*Dotted-Decimal*	*CIDR*
A	11111111 00000000 00000000 00000000	255.0.0.0	/8
B	11111111 11111111 00000000 00000000	255.255.0.0	/16
C	11111111 11111111 11111111 00000000	255.255.255.0	/24

- In class A, Netid = one byte, hostid = 3 bytes
- In class B, Netid = two bytes, hostid = two bytes
- In class C, Netid = three bytes, hostid = 1 byte

Classful Addressing : Mask

- Default mask, is 32-bit number made of continuous 1s followed by continuous 0s.
- Mask can help to find netid and hostid.
- Mask for class A has eight 1s, meaning netid = 1st 8 bits and hostid = next 24 bits
- Mask is represented by slash or Classless Interdomain Routing (CIDR) notation
- This notation is used in classless addressing



Class	*Binary*	*Dotted-Decimal*	*CIDR*
A	11111111 00000000 00000000 00000000	255.0.0.0	/8
B	11111111 11111111 00000000 00000000	255.255.0.0	/16
C	11111111 11111111 11111111 00000000	255.255.255.0	/24

Classful Addressing : Subnetting

- Method of dividing a single physical network into logical sub-networks (subnets).
- Allows a business to expand its network without requiring a new network number from its ISP
- Helps to reduce network traffic and also conceals network complexity

Classful Addressing : Supernetting

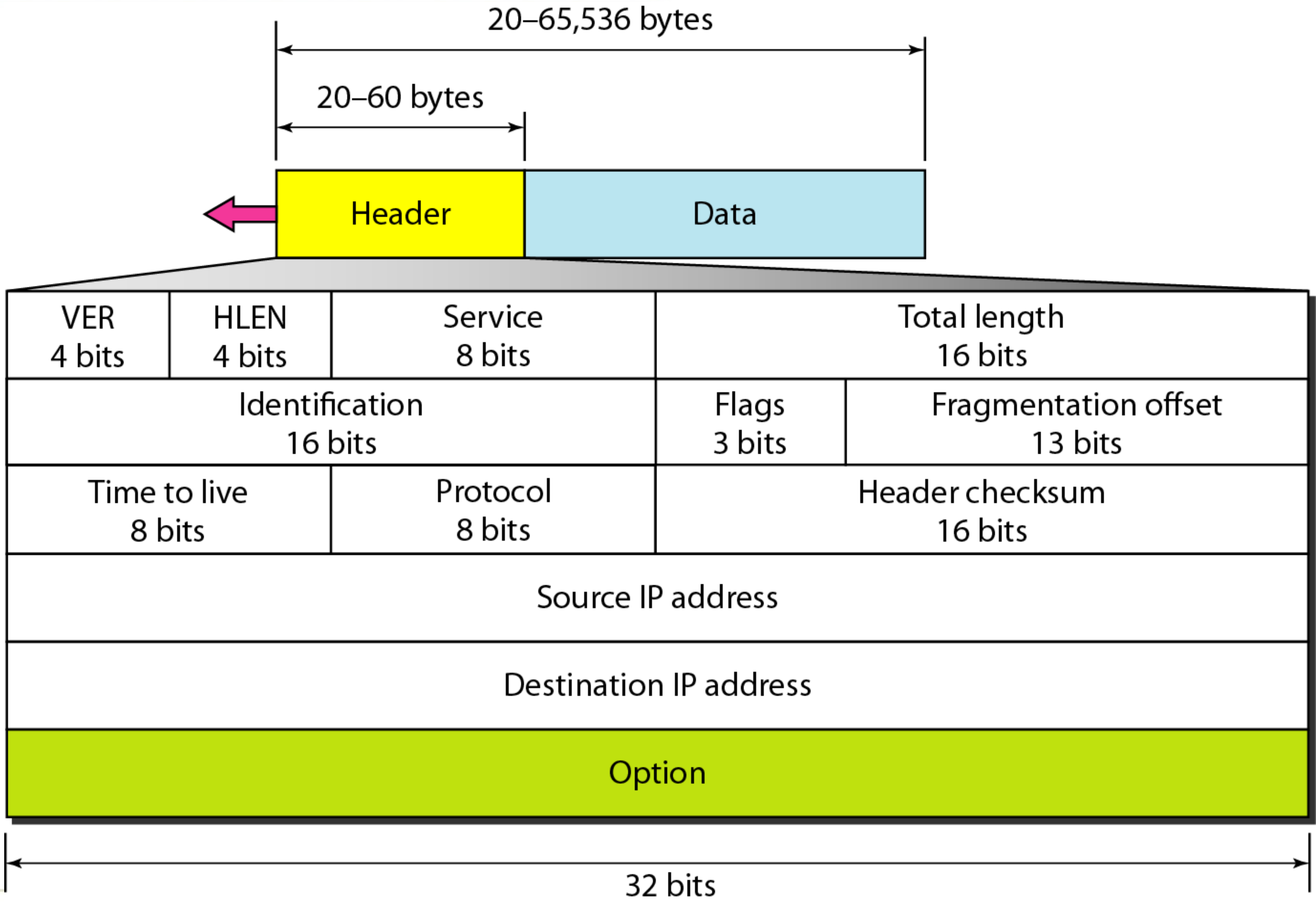
- Since, most of class A and class B addresses were depleted; there was huge demand for midsize blocks.
- Class C block with maximum 256 addresses did not satisfy needs of organizations.
- Even a midsize organization needed more addresses.
- Solution was supernetting.
- In supernetting, several networks are combined to create a supernetwork
- Organizations can apply for sets of class C blocks instead of just one.
- For example, an organization that needs 1000 addresses can be granted four continuous class C blocks which can create one supernetwork.
- Supernetting decreases number of 1s in mask.
- For example, if an organization is given four class C addresses, mask changes from /24 to /22.
- Classless addressing eliminated need for supernetting.

Classless Addressing

- Classful addressing is almost obsolete - replaced with classless addressing.
- To overcome address depletion and give more organizations access to Internet, classless addressing was designed and implemented.
- There are no classes, but addresses are granted in blocks.
- When an entity, small or large, needs to connect to Internet, it is granted a block (range) of addresses whose size (number of addresses) varies based on nature and size of entity.
- For example, a household may be given only two addresses; a large organization may be given thousands of addresses.
- An ISP may be given hundreds of thousands based on number of customers it may serve.
- **Restriction:** To simplify handling of addresses, Internet authorities impose three restrictions:
    1. Addresses in a block must be contiguous, one after another.
    2. Number of addresses in a block must be a power of 2 (I, 2, 4, 8, ...).
    3. First address must be evenly divisible by number of addresses.

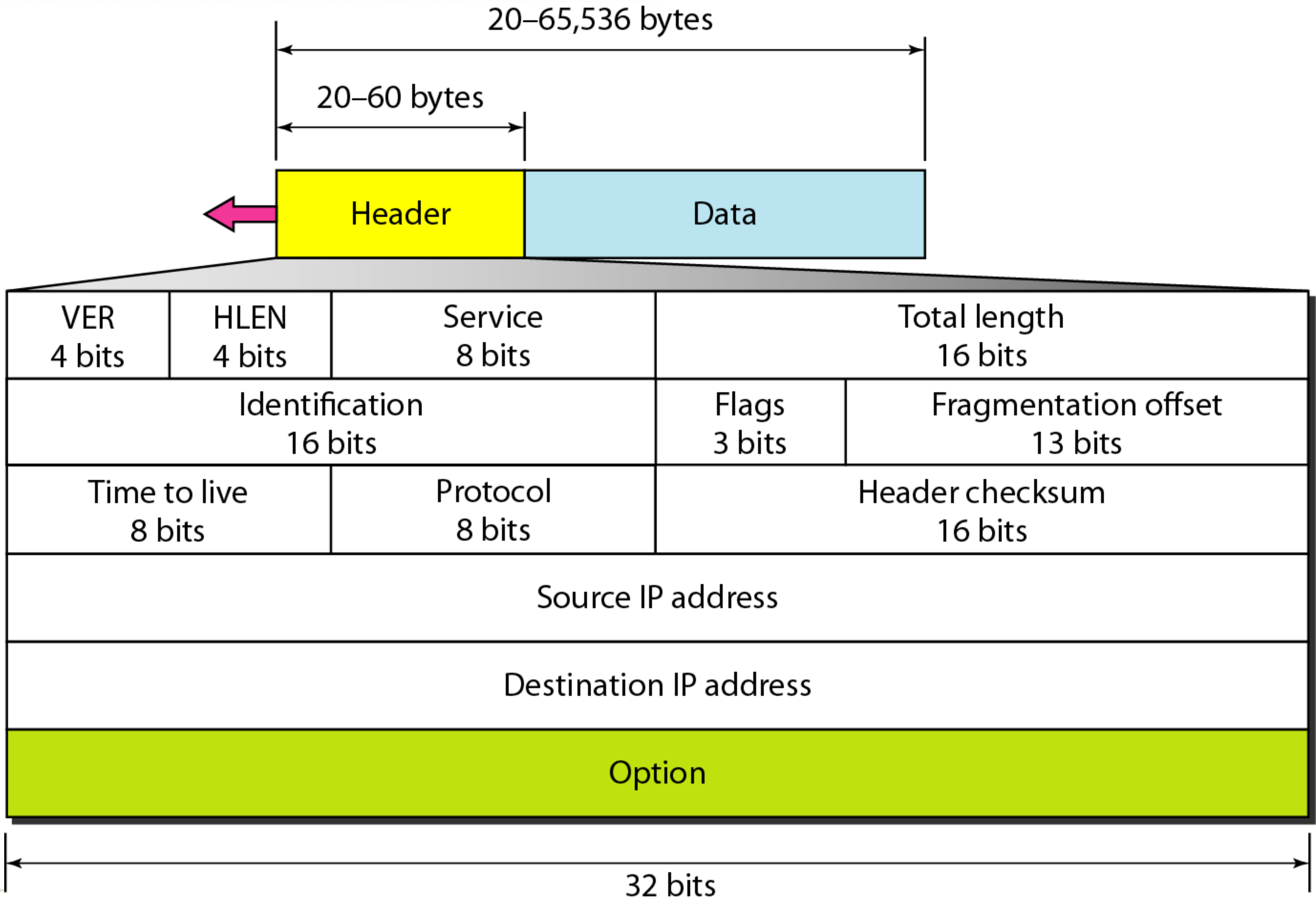
IP Version 4 Protocol

- Packets in IPv4 layer are called datagrams
- Datagram is a variable-length packet consisting of two parts: header and data.
- Header is 20 to 60 bytes in length and contains information essential to routing and delivery.
- Customary in *TCP/IP* to show header in 4-byte sections.



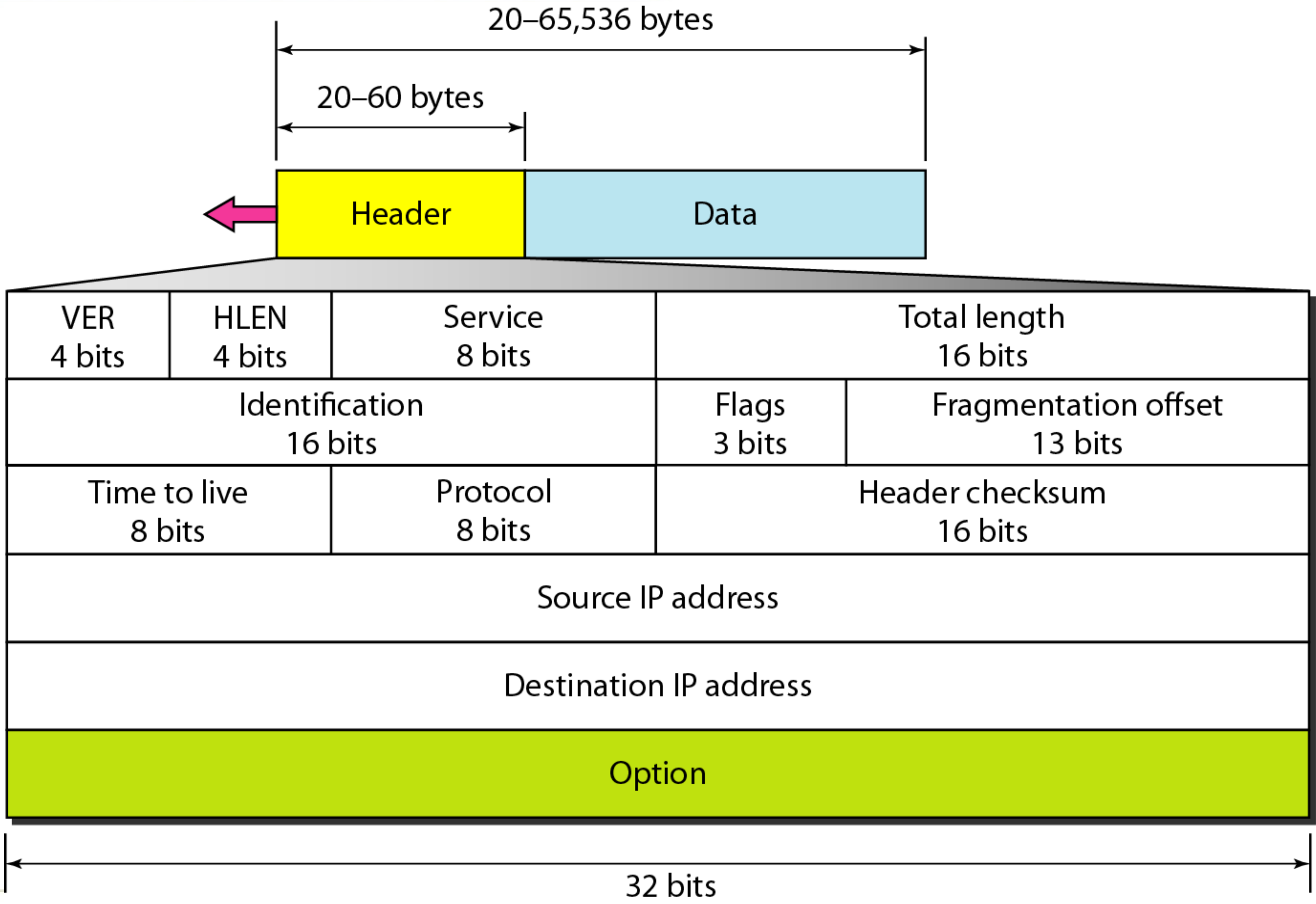
IP Version 4 Protocol - VER

- 4-bit field defines version of IPv4 protocol
- Version 4 dominates Internet today
- VER tells IPv4 software running in processing machine that datagram has format of version 4.
- Including version at start of each datagram, makes possible transition between versions over a long period of time.



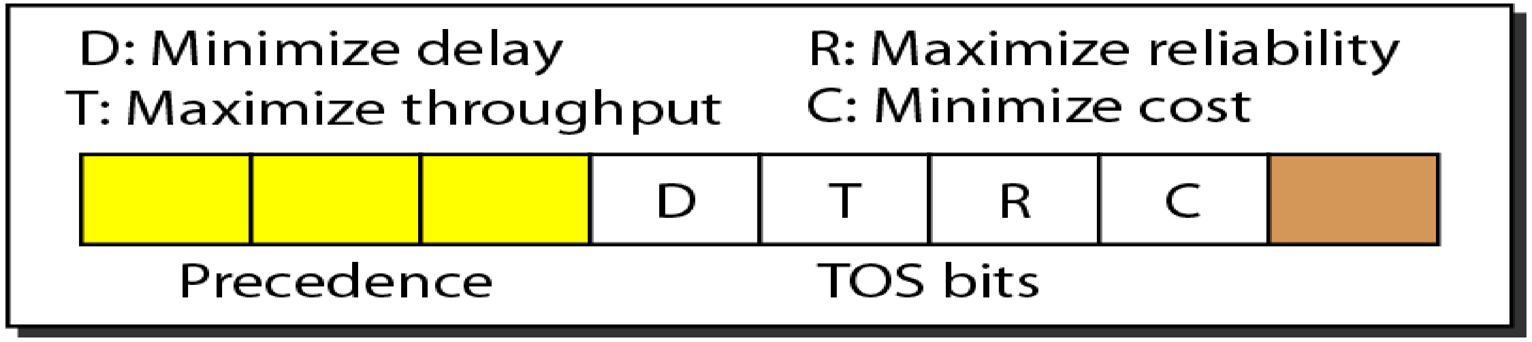
IP Version 4 Protocol - HLEN

- Header length (HLEN).
- 4-bit field defines total length of datagram header in 4-byte words.
- Needed because length of header is variable (between 20 and 60 bytes).
- When there are no options, header length is 20 bytes, and value of this field is 5 (5 x 4 = 20).
- When option field is at its maximum size, value of this field is 15 (15 x 4 = 60).

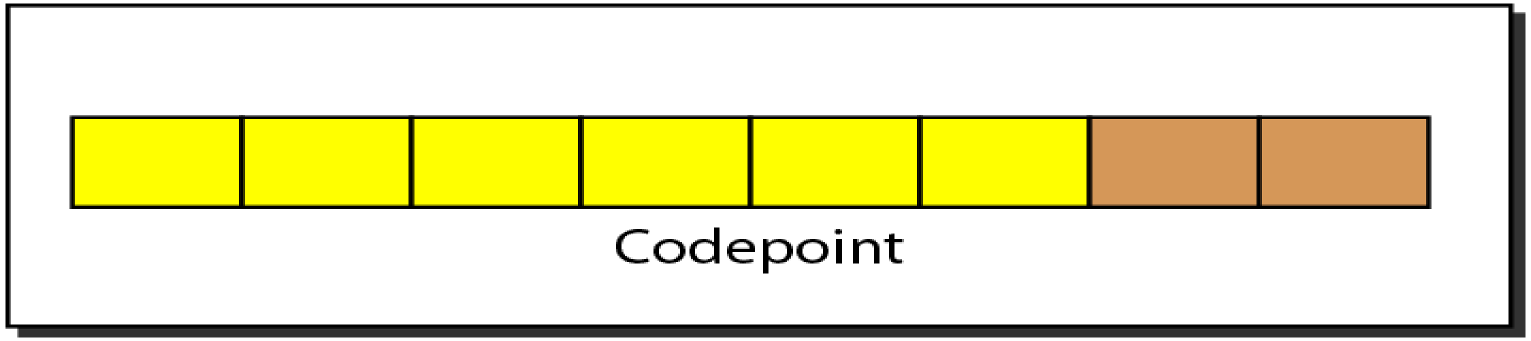


IP Version 4 Protocol - Services

- IETF has changed interpretation and name of this 8-bit field.
- Previously called service type, is now called differentiated services
- First 3 bits are called precedence bits.
- Next 4 bits are called type of service (TOS) bits
- Last bit is not used.



Service type

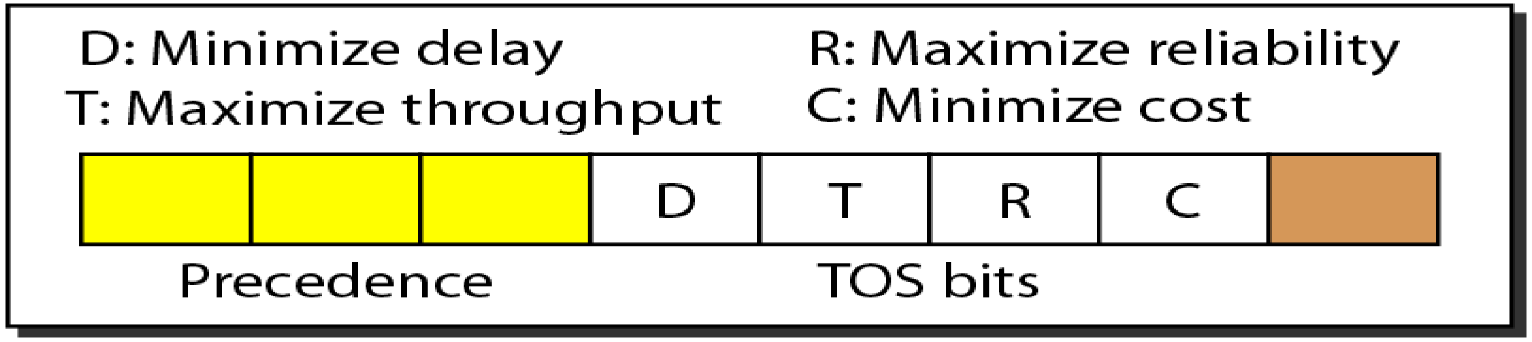


Differentiated services

IP Version 4 Protocol - Services

a. Precedence is a 3-bit subfield ranging from 0 (000 in binary) to 7 (111 in binary).

- Defines priority of datagram in issues such as congestion.
- If router is congested and needs to discard some datagrams, lowest precedence datagrams are discarded first.

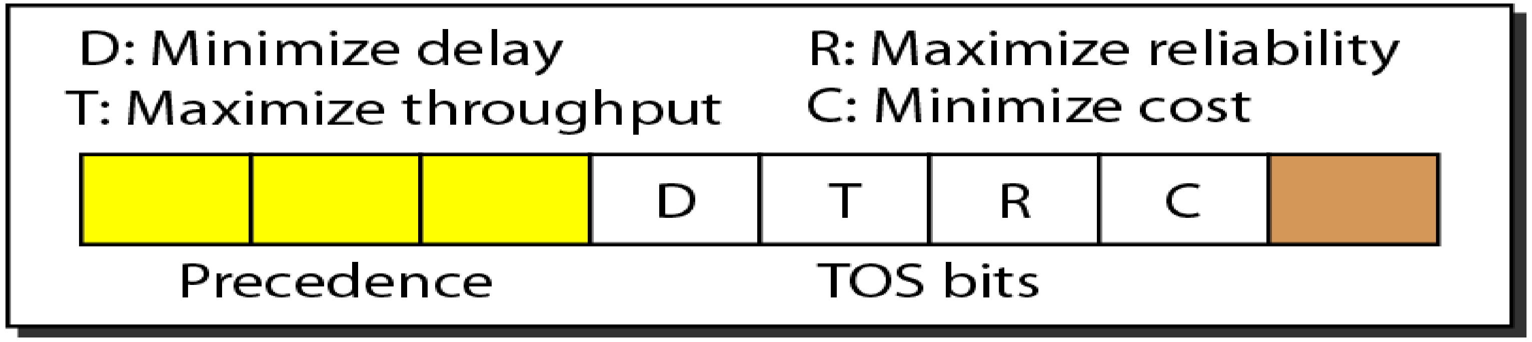


Service type

IP Version 4 Protocol - Services

b. Type of Service (TOS) bits is a 4-bit subfield with each bit having a special meaning.

- Although bits can be either 0 or 1, but only 1 bit set at a time.
- Bit patterns and their interpretations are given in Table.
- With only 1 bit set at a time, we can have five different types of services.



Service type

TOS bits	Description
0000	Normal (Default)
0001	Minimize Cost
0010	Maximize reliability
0100	Maximize throughput
1000	Maximize delay

Default types of service

- Application programs can request a specific type of service.
- Defaults for some applications are shown in table
- Activities requiring immediate attention and response need minimum delay.
- Activities that send bulk data require maximum throughput.
- Management activities need maximum reliability.
- Background activities need minimum cost

Protocol	*TOS Bits*	*Description*
ICMP	0000	Normal
BOOTP	0000	Normal
NNTP	0001	Minimize cost
IGP	0010	Maximize reliability
SNMP	0010	Maximize reliability
TELNET	1000	Minimize delay
FTP (data)	0100	Maximize throughput
FTP (control)	1000	Minimize delay
TFTP	1000	Minimize delay
SMTP (command)	1000	Minimize delay
SMTP (data)	0100	Maximize throughput
DNS (UDP query)	1000	Minimize delay
DNS (TCP query)	0000	Normal
DNS (zone)	0100	Maximize throughput

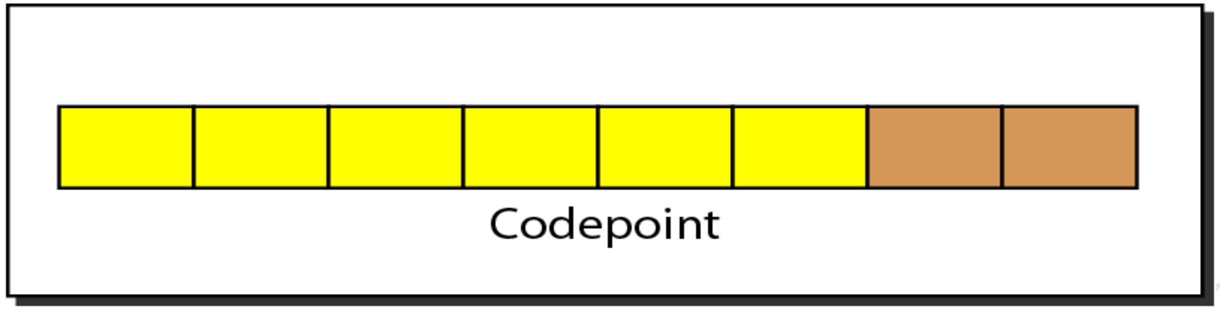
Default types of service

- First 6 bits denote codepoint subfield
- Last 2 bits are not used.
- Codepoint subfield is used in two ways.

A. When 3 rightmost bits are 0's, 3 leftmost bits are interpreted same as precedence bits in service type interpretation. i.e., it is compatible with old interpretation.

B. When 3 rightmost bits are not all 0's, 6 bits define 64 services based on priority assignment by Internet or local authorities

- First category (numbers 0, 2,4, ... ,62) is assigned by Internet authorities (IETF).
- Second category (3, 7, 11, 15,… 63) assigned by local authorities (organizations).
- Third category (1, 5, 9, ,61) is temporary and can be used for experimental purposes.



Differentiated services

Values for Codepoints

Category	Code Point	Description
1	XXXXX0	Internet
2	XXXX11	Local
3	XXXX01	Temporary or experimental

IPv6

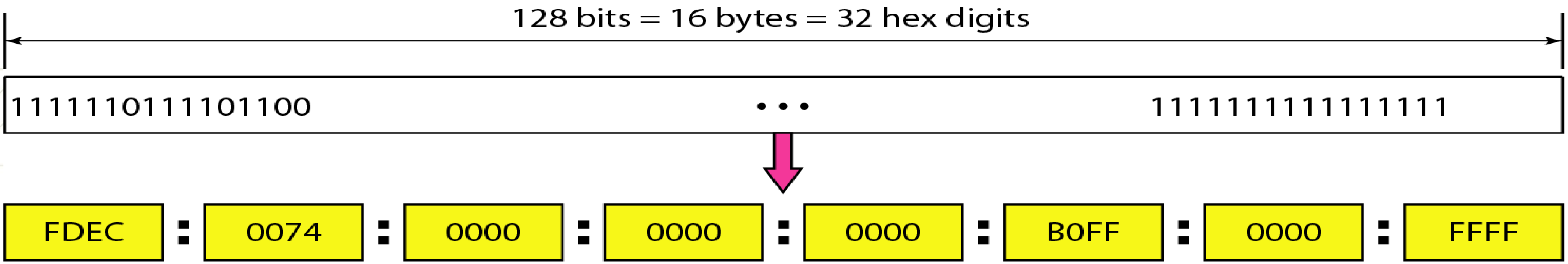
- IPv4 provides host-to-host communication between systems in Internet.
- IPv4 is well designed, data communication has evolved since inception of IPv4 in 1970s.
- IPv4 has some deficiencies that make it unsuitable for the fast-growing Internet.
  - Despite all short-term solutions, such as subnetting, classless addressing, and NAT, address depletion is still a long-term problem in Internet.
  - Internet must accommodate real-time audio and video transmission which requires minimum delay strategies and reservation of resources not provided in IPv4 design.
  - No encryption or authentication is provided by IPv4.
- To overcome these deficiencies, IPv6 was proposed
- IPv6 was extensively modified to accommodate unforeseen growth of Internet.
- Format and length of IP address were changed along with packet format.

IPv6 Advantages over IPv4

- **Larger address space**: IPv6 address is 128 bits long while IPv4 is 32-bit long
- **Better header format.** In, IPv6 options are separated from base header and inserted, when needed, between base header and upper-layer data. This simplifies and speeds up routing process because most options do not need to be checked by routers.
- **New options.** IPv6 has new options to allow for additional functionalities.
- **Allowance for extension.** IPv6 is designed to allow extension of protocol if required by new technologies or applications.
- **Support for resource allocation.** In IPv6, type-of-service field has been removed, but a mechanism (called jlow *label)* has been added to enable source to request special handling of packet. This mechanism can be used to support traffic such as real-time audio and video.
- **Support for more security.** Encryption and authentication options in IPv6 provide confidentiality and integrity of packet.

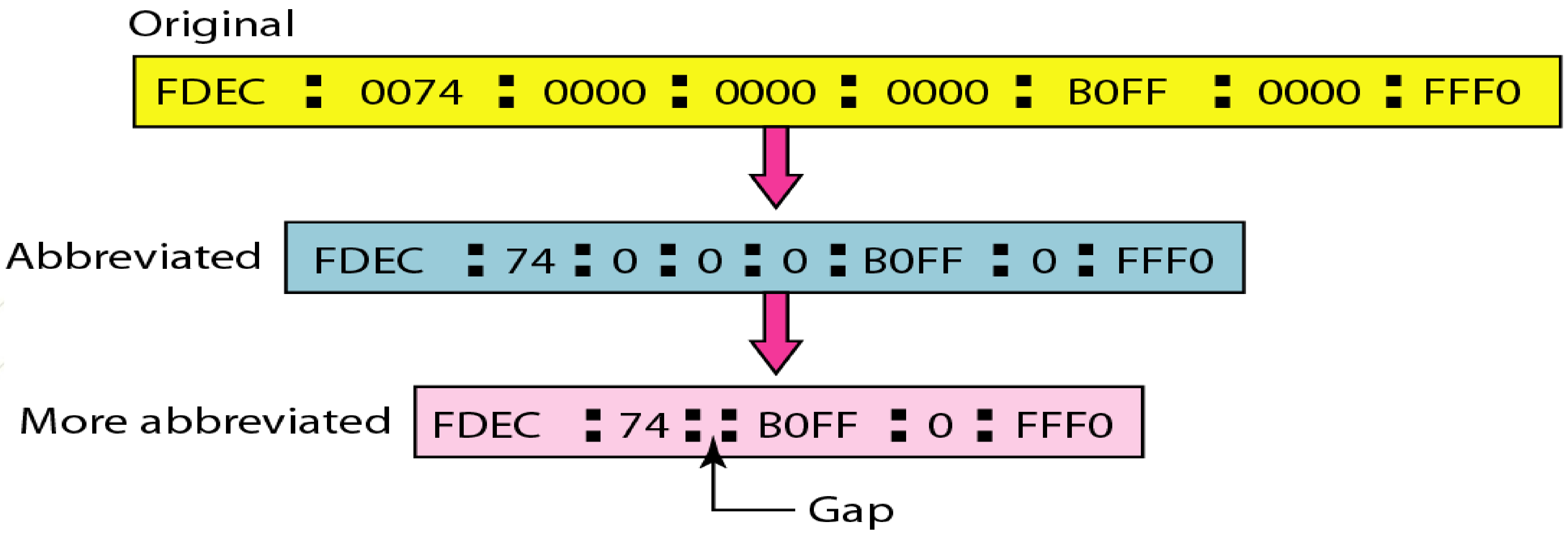
IPv6 Structure

- IPv6 address consists of 16 bytes (octets); it is 128 bits long.
- To make addresses more readable, IPv6 specifies hexadecimal colon notation.
- In this , 128 bits is divided into eight sections, each 2 bytes in length.
- Two bytes in hexadecimal notation requires four hexadecimal digits.
- Thus, address consists of 32 hexadecimal digits, with every four digits separated by a colon



Abbreviated IPv6 addresses

- IP address, in hexadecimal format, is very long, many of its digits are zeros.
- Leading zeros of a section (four digits between two colons) can be omitted. Only leading zeros can be dropped, not trailing zeros



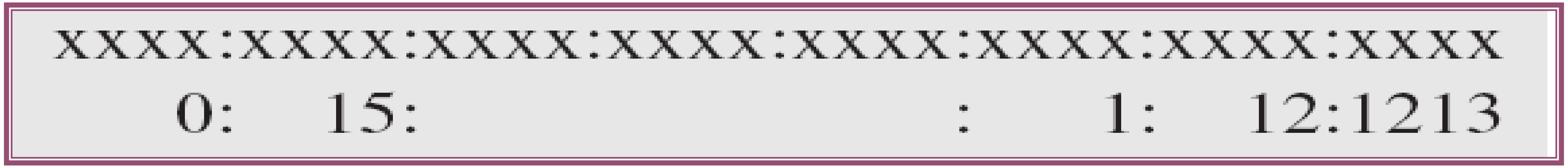
Original
FDEC : 0074 : 0000 : 0000 : 0000 : B0FF : 0000 : FFF0
Abbreviated
FDEC : 74 : 0 : 0 : 0 : B0FF : 0 : FFF0
More abbreviated
FDEC : 74 : : B0FF : 0 : FFF0
Gap

Question

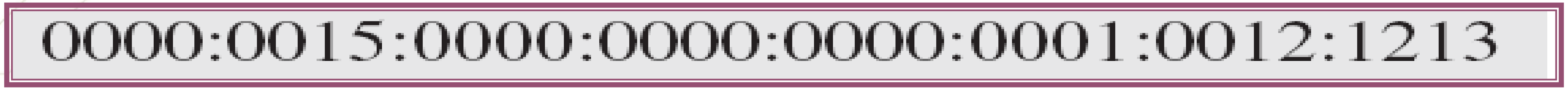
- Expand the address 0:15::1:12:1213 to its original

Solution

- Expand the address 0:15::1:12:1213 to its original

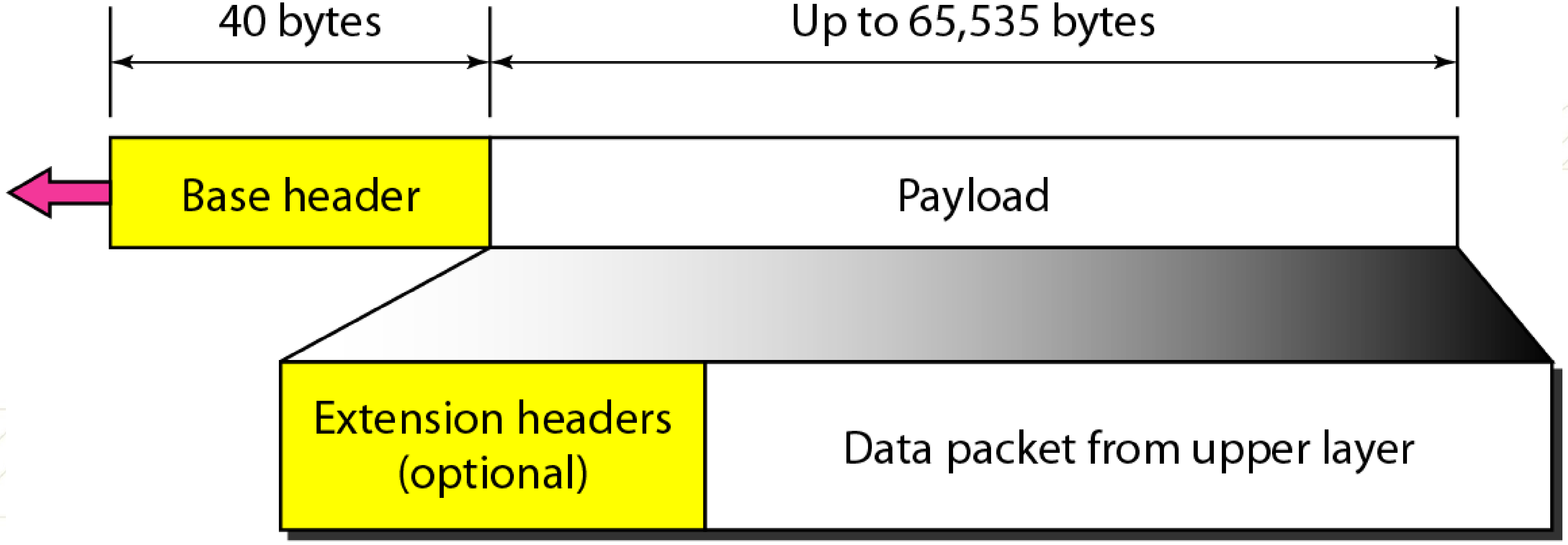


```
xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx
   0:  15:                 :   1:  12:1213
```



```
0000:0015:0000:0000:0000:0001:0012:1213
```

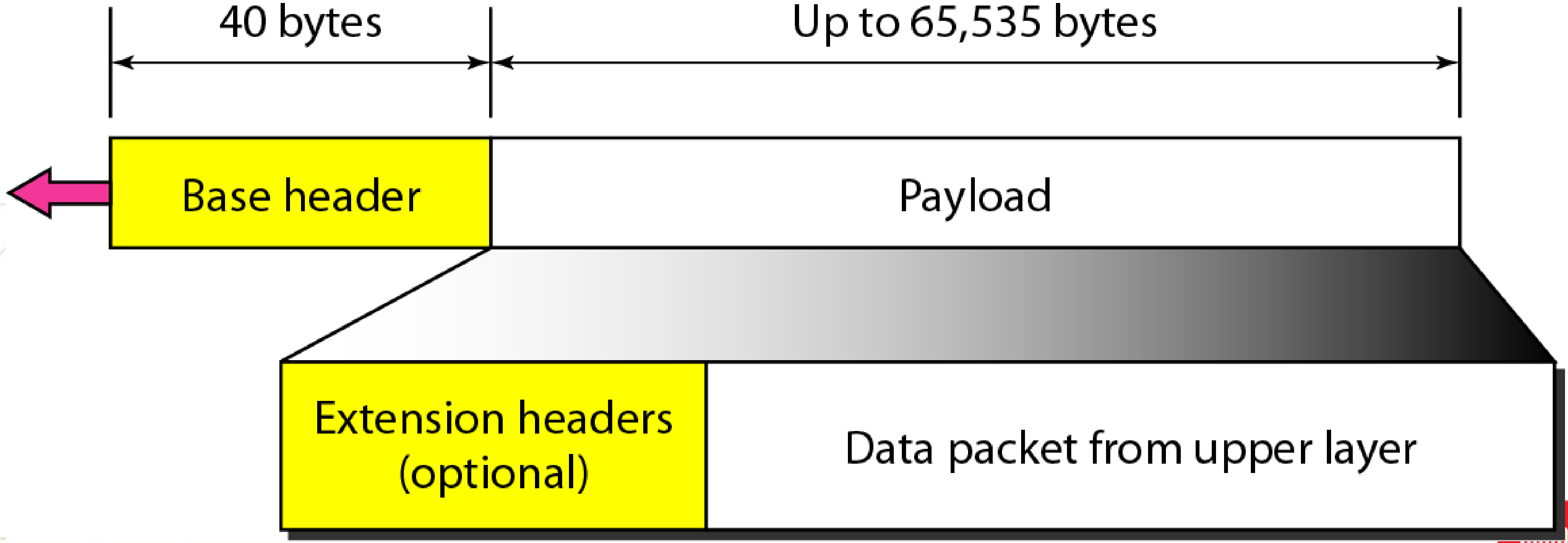
IPv6 datagram header and payload



IPv6 datagram header and payload

Packet Format

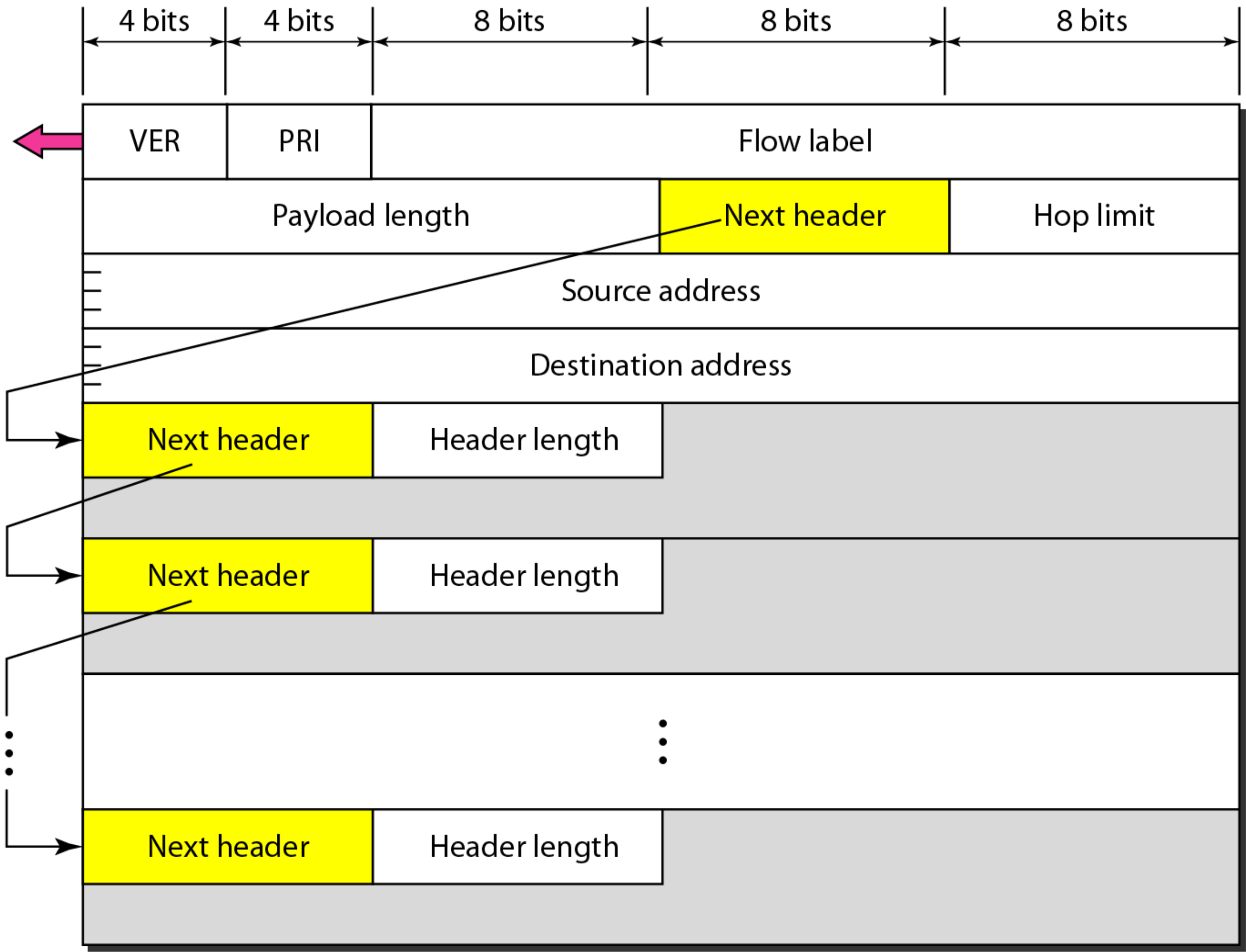
- Each packet is composed of a mandatory base header followed by payload.
- Payload consists of two parts: optional extension headers and data from an upper layer.
- Base header occupies 40 bytes, whereas extension headers and data from upper layer contain up to 65,535 bytes of information.



IPv6 datagram header and payload

Base Header

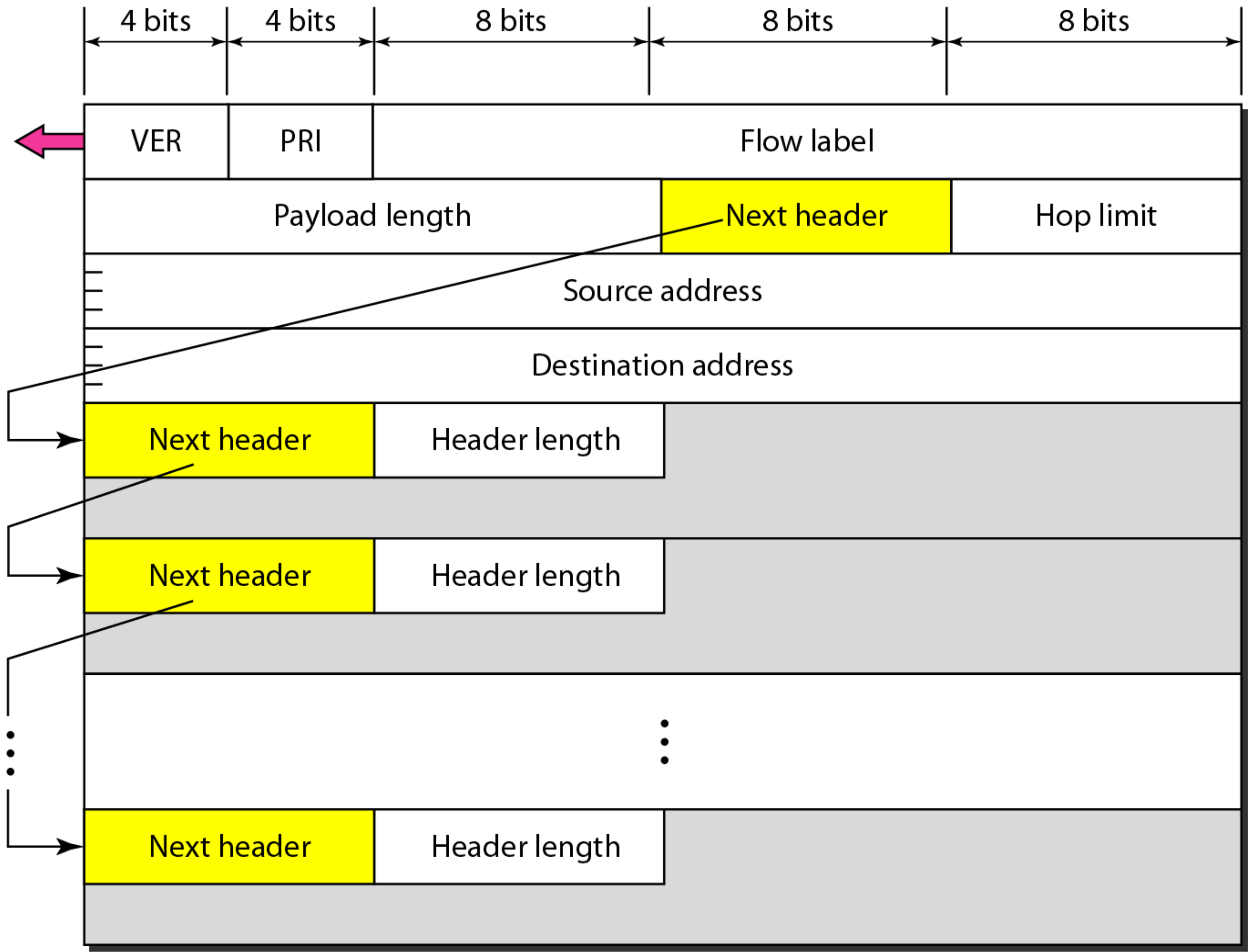
- Base header has eight fields.
  - ❑**Version:** 4-bit field defines version number of IP. For IPv6, value is 6.
  - ❑**Priority:** 4-bit field defines priority of packet w.r.t. traffic congestion.
  - ❑**Flow label:** 3-byte (24-bit) field designed to provide special handling for a particular flow of data.
  - ❑**Payload length:** 2-byte field defines length of IP datagram excluding base header.



IPv6 datagram header and payload

Base Header

- ❑**Next header:** 8-bit field defining header that follows base header in datagram. One of optional extension headers used by IP or header of an encapsulated packet such as UDP or TCP. Each extension header also contains this field. This field in version 4 is called *protocol.*
- ❑**Hop limit:** 8-bit field serves same purpose as TIL field in IPv4.
- ❑**Source address:** 16-byte (128-bit) Internet address that identifies original source of datagram.
- ❑**Destination address:** 16-byte (128-bit) Internet address, identifies final destination of datagram.



Comparison of IPv4 and IPv6

Properties	IPv4	IPv6
Addressing	Provides 32-bit addresses	Provides 128-bit addresses which results in a significantly larger address space
Security	Does not provide security mechanisms, and requires additional security protocols facilitated by network devices	Provides authentication, integrity, and confidentiality during communication of data over a network with the implementation of IPSec protocol suite
Protocol Enhancement	No such protocol enhancement process	Features Hierarchical Addressing, which enables ISPs to allocate a subnet to an organization
Routing	Uses flat routing model which requires routers to store complete table of route identification	Uses hierarchical routing protocol, only store routing information of networks which they are connected to

Comparison of IPv4 and IPv6

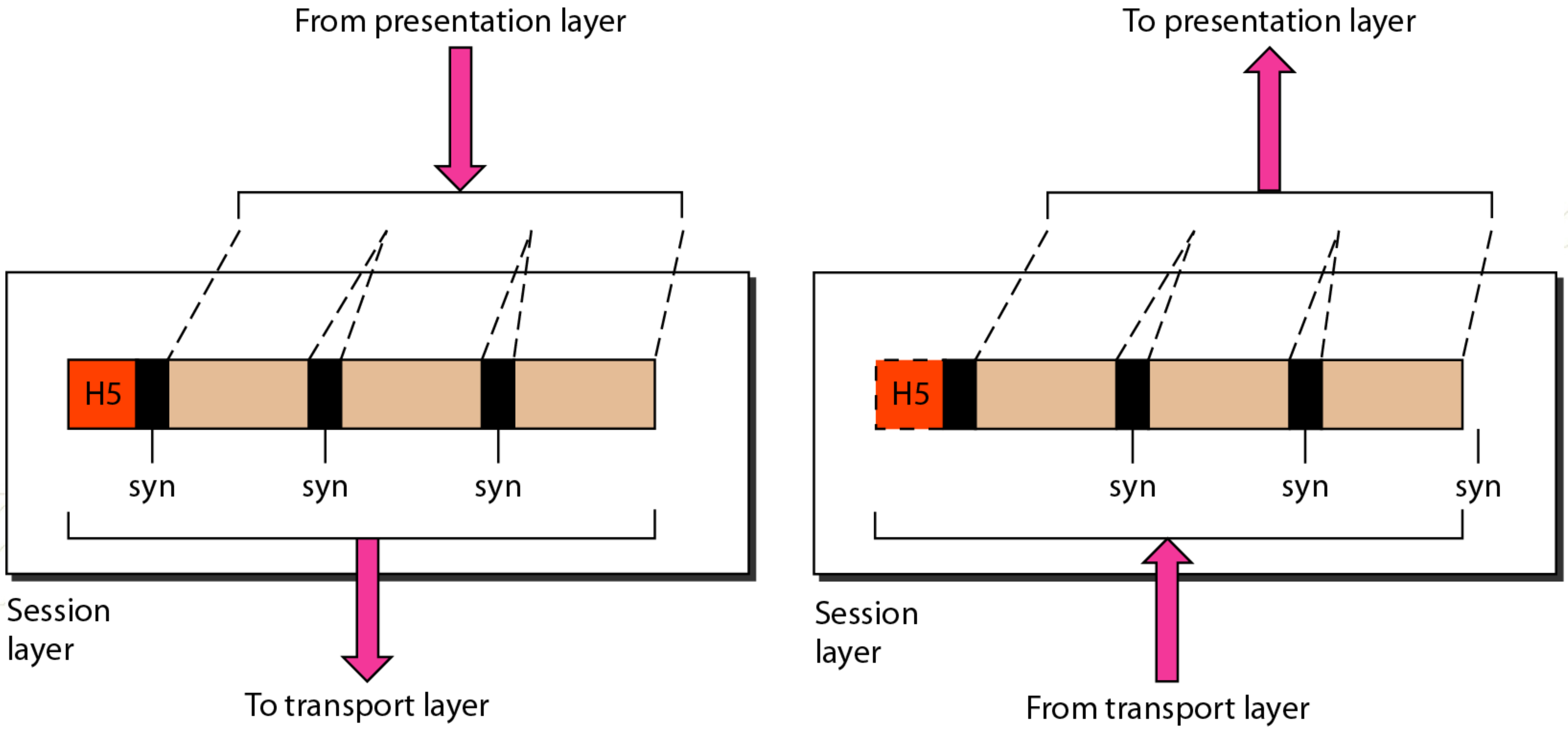
Properties	IPv4	IPv6
Backward Compatibility	Special translation mechanisms like Dual Stack technology, Tunneling, and NAT can be used to enable communication among devices on both networks	Special translation mechanisms like Dual Stack technology, Tunneling, and NAT can be used to enable communication among devices on both networks
Ease of Configuration	Require a separate address configuration protocol	Do not require a separate address configuration protocol

Layer 5 – Session Layer

Data	Session Interhost Communication

- Provides a reliable and secure communication between two devices by establishing, managing and terminating sessions, maintaining session state information, and handling session synchronization and recovery
- Regulates flow of data, Defines format of data sent over connections.
- Manages who can transfer data in a certain amount of time and for how long.
- Reconnects session if it disconnects.
- Reports and logs and upper layer errors.
- **Protocols:** Protocols for session layer are **NetBIOS, Mail Slots, Names Pipes, and RPC**.

Functions of Session Layer:



Functions of Session Layer:

1. **Session Establishment:** Establishes a connection between two devices before data transmission begins. During this process, the session layer determines the type of session required and negotiates session parameters.
2. **Session Management:** Manages session between two devices. Keeps track of session throughout its duration, maintains session state information and ensures that connection remains active. In case of errors during data transfer, ensures session is terminated gracefully.
3. **Session Termination:** After completion of data transmission session is terminated between two devices.

Functions of Session Layer:

4. **Dialog control**: When a device is contacted first, session layer is responsible for determining which device participating in communication will transmit at a given time as well as controlling amount of data that can be sent in a transmission. Types of dialog control that can take place include simplex, half duplex and full duplex.
5. **Dialog Separation or Session Synchronization:** Handles synchronization between incoming and outgoing data streams by adding synchronization points called checkpoints which allow session layer to retransmit lost data more efficiently.
6. **Session Recovery:** Responsible for recovering a session if connection is lost or interrupted. Keeps track of state of session and can recover it from point where connection was lost.
7. **Session Security:** Provide security features like session encryption, authentication, and authorization to ensure secure transmission of data.

Functions of Session Layer:

8. **Quality of Service (QoS)**: Ability of a network to prioritize and deliver data based on its importance and ensure a certain level of performance. QoS is achieved through:
   - **Traffic prioritization:** Ensures higher-priority data is sent first. Achieved by assigning different types of data or applications to different sessions and setting different priorities for each session based on importance of data.
   - **Bandwidth management:** Monitors amount of data being transmitted and allocating bandwidth based on priority.
   - **Congestion Control:** Monitors and control network congestion by managing rate and amount of data being transmitted which prevents network overload, reduces packet loss, and improves overall network performance.
   - **Error Handling and Recovery:** Detects and correct errors that occur during data transmission. If an error is detected, session layer takes corrective action, such as requesting a retransmission or terminating session if error cannot be corrected.

Firewalls

- Security devices that monitor and control network traffic based on predetermined security policies.
- Help to prevent unauthorized access and malicious attacks, and ensure that only authorized sessions are allowed
- Firewalls perform multiple roles in session layer, such as:

  1. **Session Filtering:** Filter traffic based on session-related information such as session ID, source and destination port numbers, protocol type, etc. Identifies and blocks any traffic that does not fit defined policy, to prevent unauthorized access or malicious attacks.
  2. **Session Management:** Manages sessions between two communicating devices, including establishing, maintaining, and terminating sessions. Helps to ensure that only authorized sessions are allowed and that session remains secure and private.

Firewalls Types

1. **Stateful Firewall or Packet-filtering firewalls**:
   - Operate inline at junction points where devices such as routers and switches do their work.
   - They don't route packets; rather compare every received packet with established criteria, such as allowed IP addresses, packet type, port number and packet protocol headers, etc.
   - Flagged packets are dropped.
2. **Circuit-level Firewall or Session-layer gateways:** Operate to monitor and control individual sessions.
   - They establish circuit between communicating devices, to monitor and control flow of data between them.
   - Can filter traffic based on session ID, source and destination port numbers, and protocol type.

Application layer protocols

- Operate at topmost layer of OSI reference model and enable communication between applications on different hosts.
- Designed to communicate with specific applications and provide a variety of services.
- Provide means for end-users to access networked resources and exchange data with other end-users.
- Establish, manage, and terminate communication sessions and enable data exchange for specific applications..

Types of Application Layer Protocols:

1. **HTTP (HyperText Transfer Protocol):** Client-server protocol designed for World Wide Web. Used to transfer hypertext documents and other data between web servers and clients.
2. **FTP (File Transfer Protocol):** Client-server protocol used to transfer files over network. Enables sharing and copying of files between computers located on different networks.
3. **SMTP (Simple Mail Transfer Protocol):** Used to send email messages from one server to another. Most common protocol used for sending emails over Internet.
4. **DNS (Domain Name System):** DNS maps domain names to IP addresses and helps translate human-readable domain names to computer-readable IP addresses.
5. **Telnet:** Used to connect to remote systems over network and enable communication with those systems. Enables operations such as terminal emulation and remote administration.
6. **SSH (Secure Shell):** Used for secure remote administration and data transfer, with encryption of all data transmitted over network.
7. **SNMP (Simple Network Management Protocol):** Used for network management and monitoring. Enables network devices to be monitored, managed, and controlled remotely.

Types of Application Layer Protocols:

1. **HTTP (HyperText Transfer Protocol):** Client-server protocol designed for World Wide Web. Used to transfer hypertext documents and other data between web servers and clients.
2. **FTP (File Transfer Protocol):** Client-server protocol used to transfer files over network. Enables sharing and copying of files between computers located on different networks.
3. **SMTP (Simple Mail Transfer Protocol):** Used to send email messages from one server to another. Most common protocol used for sending emails over Internet.
4. **DNS (Domain Name System):** DNS maps domain names to IP addresses and helps translate human-readable domain names to computer-readable IP addresses.
5. **Telnet:** Used to connect to remote systems over network and enable communication with those systems. Enables operations such as terminal emulation and remote administration.
6. **SSH (Secure Shell):** Used for secure remote administration and data transfer, with encryption of all data transmitted over network.
7. **SNMP (Simple Network Management Protocol):** Used for network management and monitoring. Enables network devices to be monitored, managed, and controlled remotely.

HTTP (HyperText Transfer Protocol)

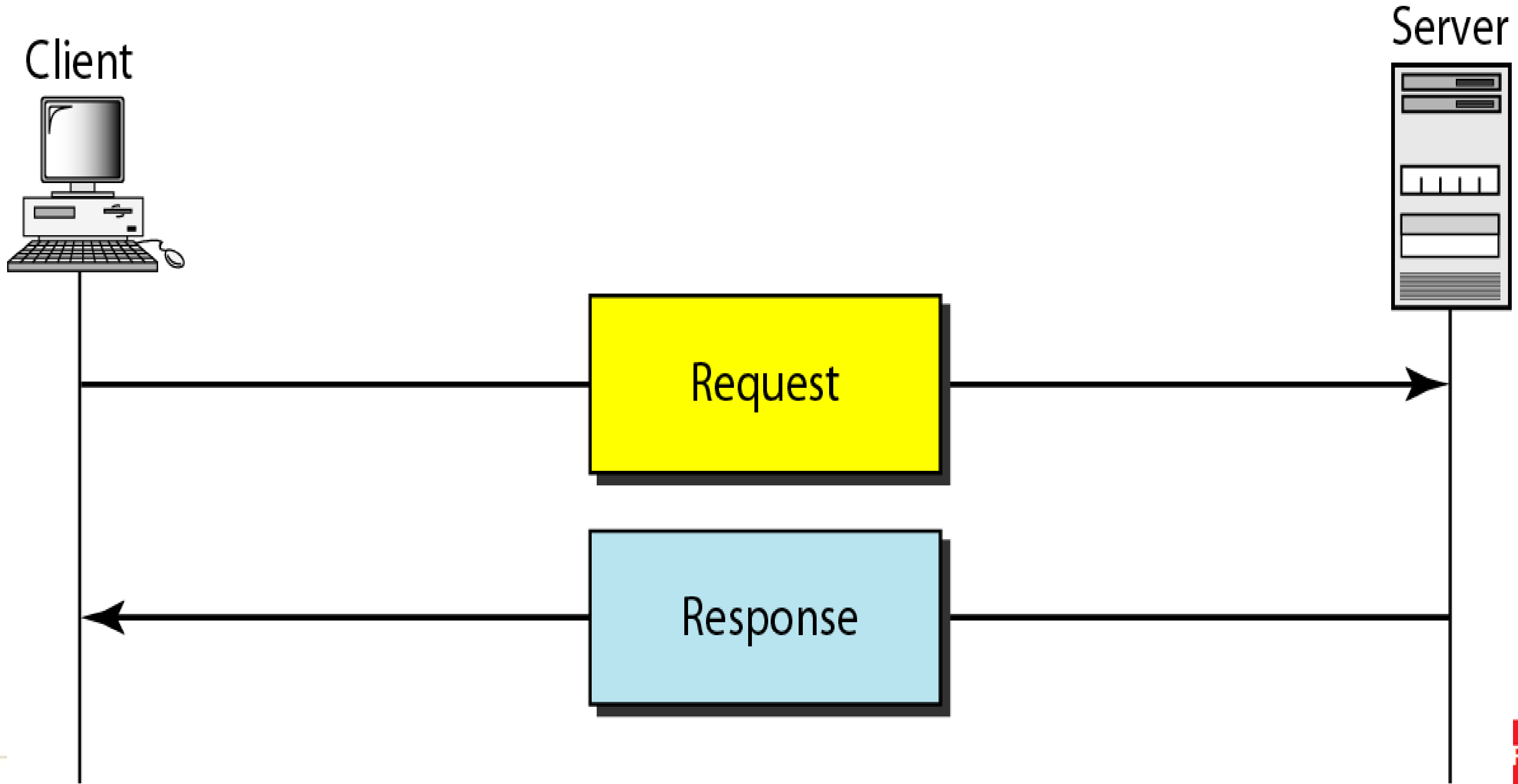
- Used to access data on World Wide Web (www).
- Can be used to transfer data in form of plain text, hypertext, audio, video, and so on.
- Allows to use in a hypertext environment where there are rapid jumps from one document to another document.
- Similar and simpler to FTP as it also transfers files from one host to another host using only one connection, i.e., no control connection to transfer files.
- Used to carry data in form of MIME-like format.
- Similar to SMTP as data is transferred between client and server.
- Differs from SMTP in way messages are sent from client to server and from server to client.
- SMTP messages are stored and forwarded while HTTP messages are delivered immediately.

Features of HTTP:

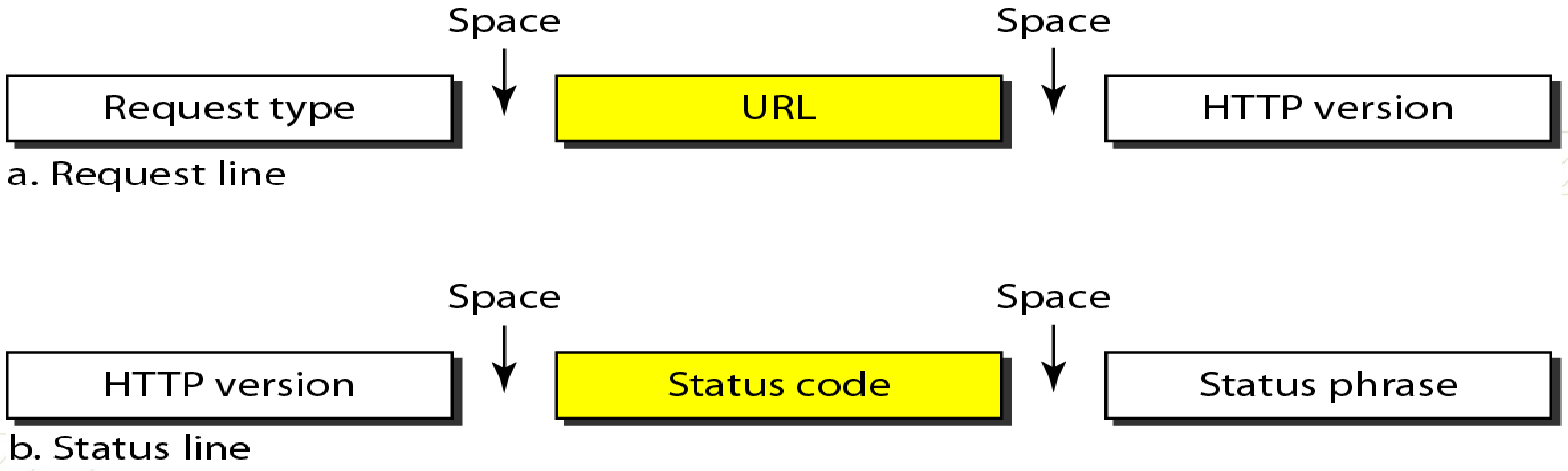
- **Connectionless protocol:** HTTP client initiates request and waits for a response from server. When server receives request, it processes request and sends back response to HTTP client. Connection between client and server exist only during current request and response time.
- **Media independent:** Data can be sent as long as both client and server know how to handle data content. Required for both client and server to specify content type in MIME-type header.
- **Stateless:** Both client and server know each other only during current request. Due to this nature of protocol, both client and server do not retain information between various requests of web pages.

HTTP Transactions

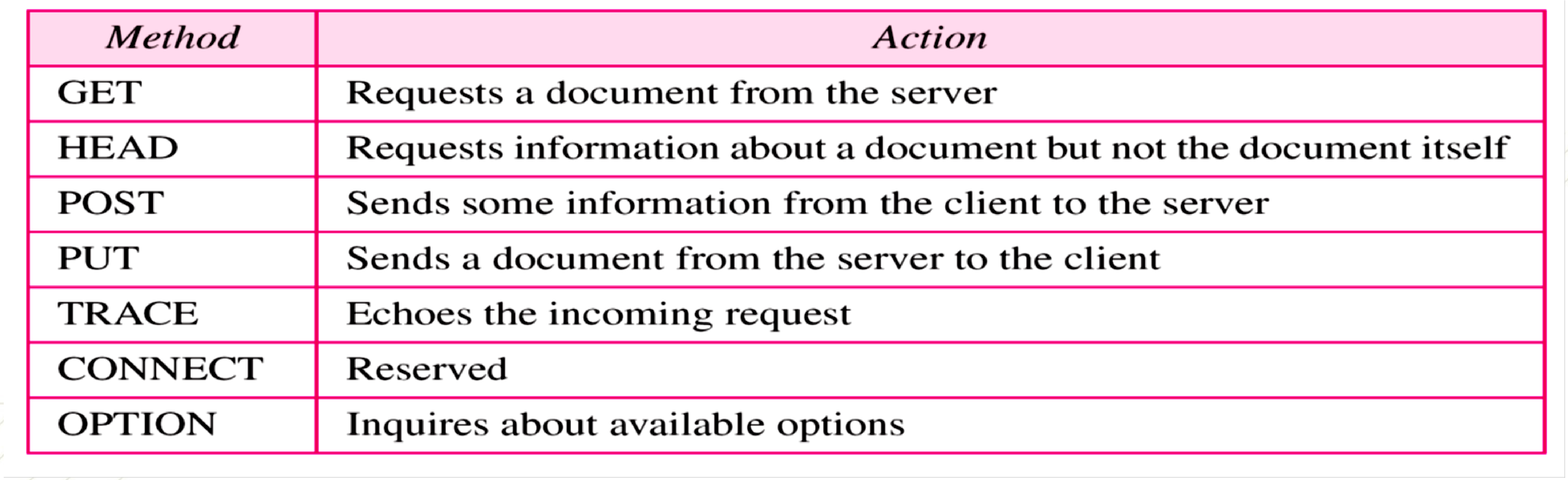
- HTTP client initiates a transaction by sending a request message to server.
- Server replies to request message by sending a response message.



Request and status lines

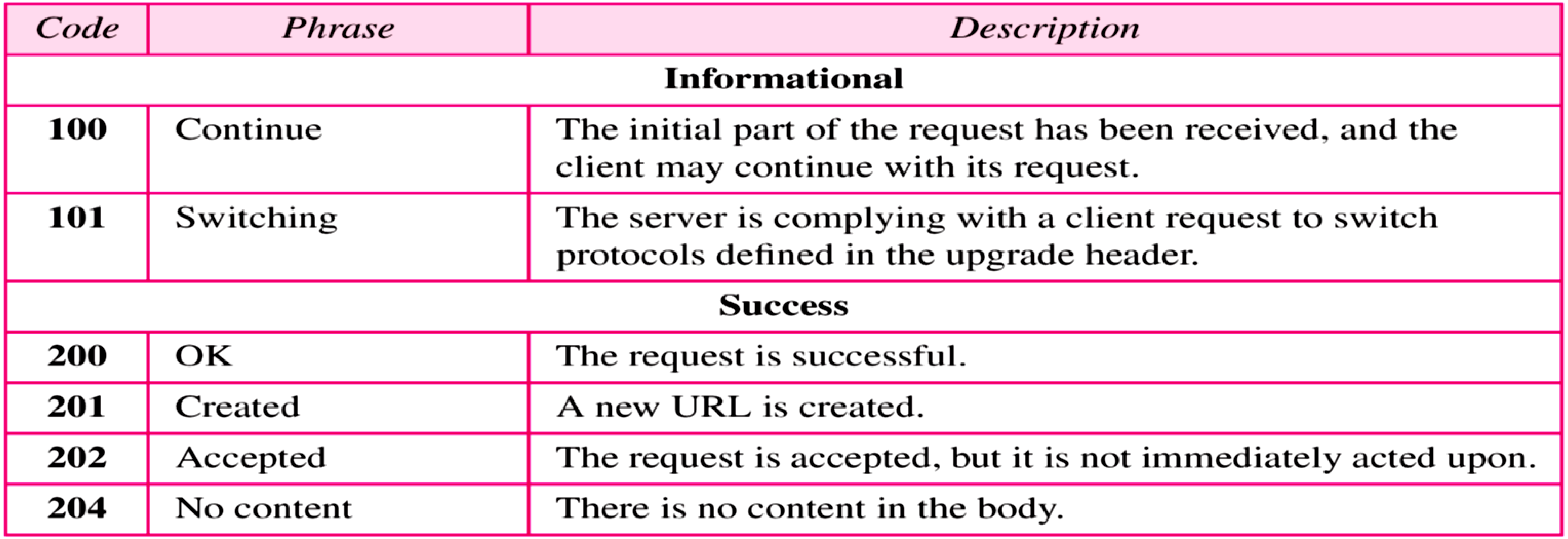


Methods



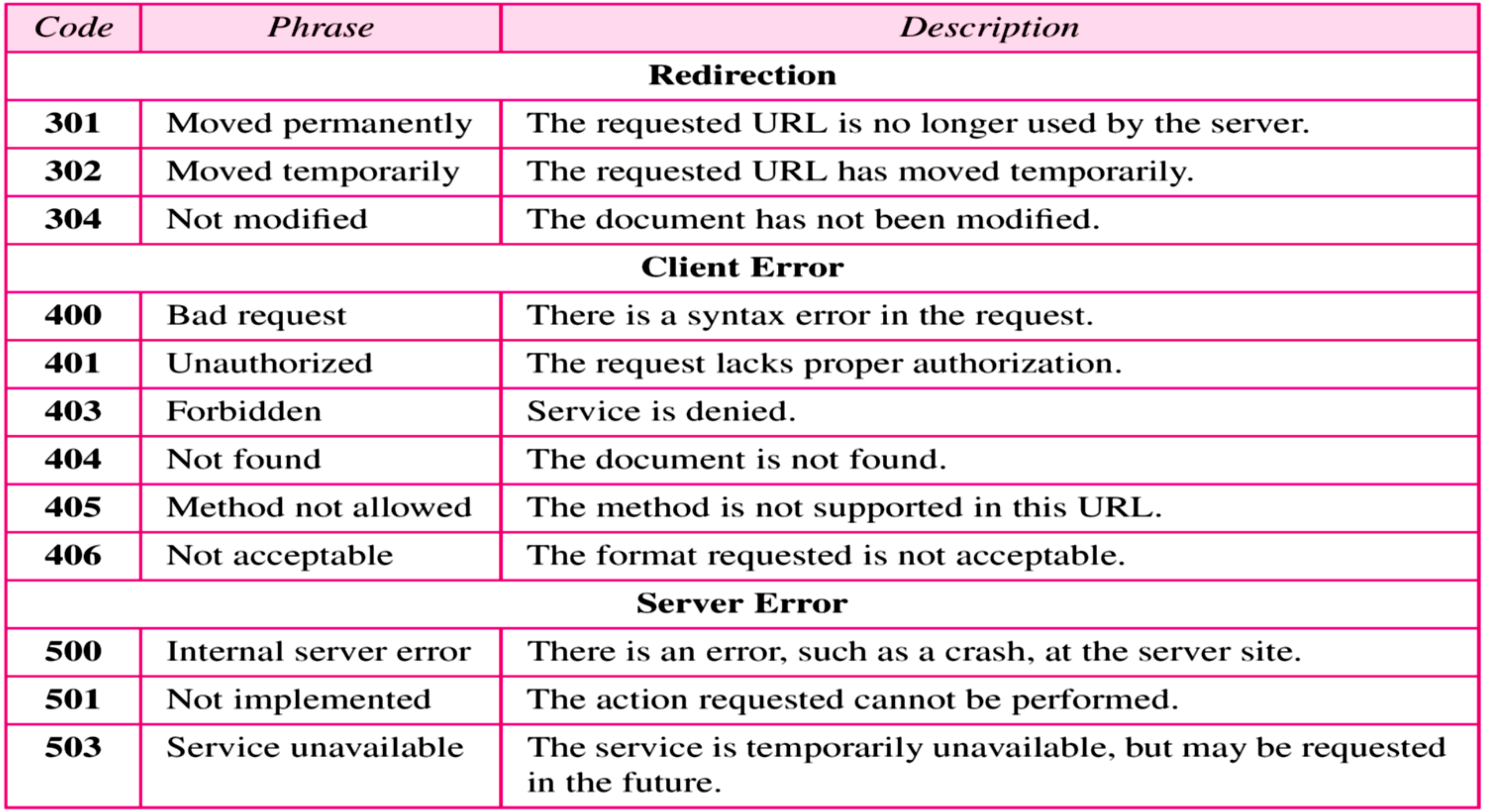
Method	*Action*
GET	Requests a document from the server
HEAD	Requests information about a document but not the document itself
POST	Sends some information from the client to the server
PUT	Sends a document from the server to the client
TRACE	Echoes the incoming request
CONNECT	Reserved
OPTION	Inquires about available options

Status codes



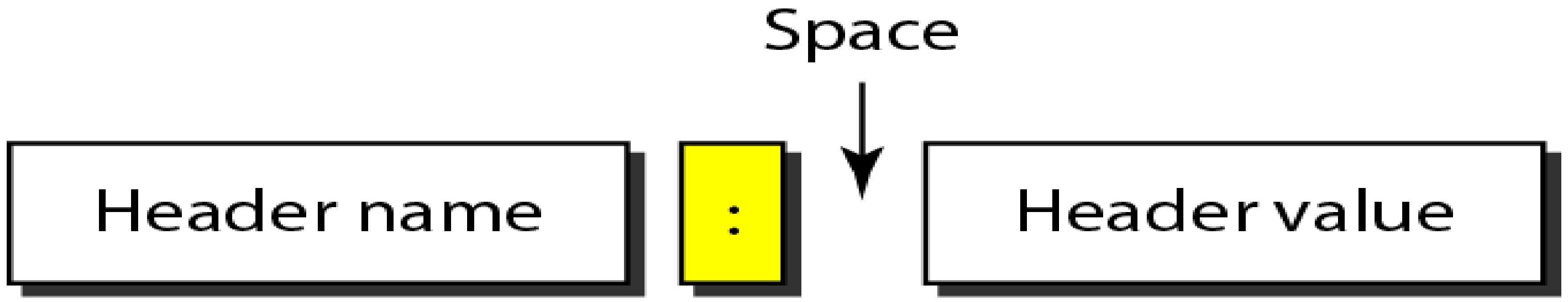
Code	*Phrase*	*Description*
Informational		
100	Continue	The initial part of the request has been received, and the client may continue with its request.
101	Switching	The server is complying with a client request to switch protocols defined in the upgrade header.
Success		
200	OK	The request is successful.
201	Created	A new URL is created.
202	Accepted	The request is accepted, but it is not immediately acted upon.
204	No content	There is no content in the body.

Status codes



Code	*Phrase*	*Description*
		Redirection
301	Moved permanently	The requested URL is no longer used by the server.
302	Moved temporarily	The requested URL has moved temporarily.
304	Not modified	The document has not been modified.
		Client Error
400	Bad request	There is a syntax error in the request.
401	Unauthorized	The request lacks proper authorization.
403	Forbidden	Service is denied.
404	Not found	The document is not found.
405	Method not allowed	The method is not supported in this URL.
406	Not acceptable	The format requested is not acceptable.
		Server Error
500	Internal server error	There is an error, such as a crash, at the server site.
501	Not implemented	The action requested cannot be performed.
503	Service unavailable	The service is temporarily unavailable, but may be requested in the future.

Header format



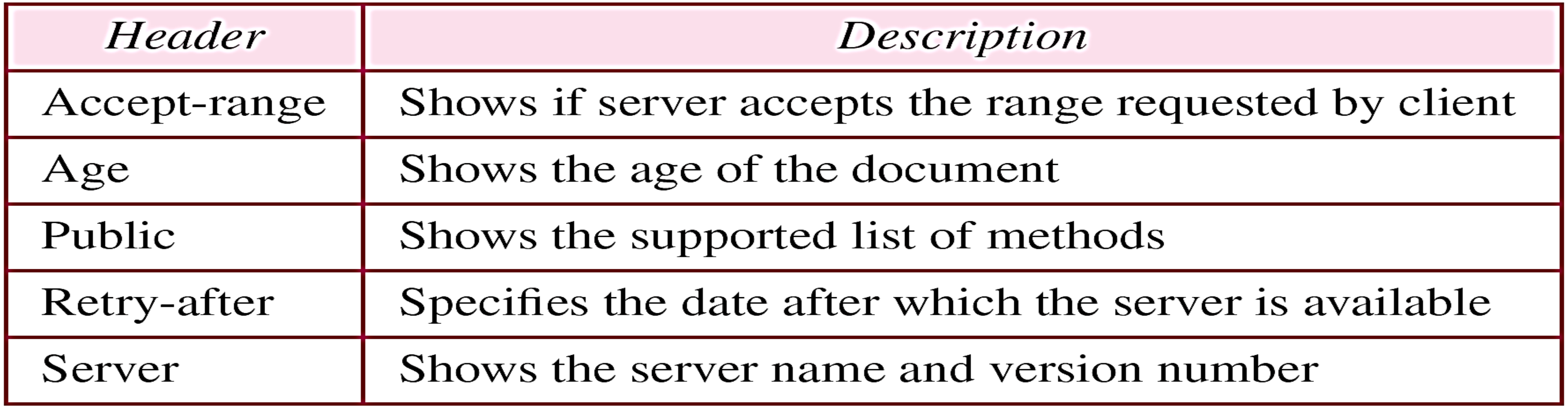
General headers

Header	*Description*
Cache-control	Specifies information about caching
Connection	Shows whether the connection should be closed or not
Date	Shows the current date
MIME-version	Shows the MIME version used
Upgrade	Specifies the preferred communication protocol

Request headers

Header	*Description*
Accept	Shows the medium format the client can accept
Accept-charset	Shows the character set the client can handle
Accept-encoding	Shows the encoding scheme the client can handle
Accept-language	Shows the language the client can accept
Authorization	Shows what permissions the client has
From	Shows the e-mail address of the user
Host	Shows the host and port number of the server
If-modified-since	Sends the document if newer than specified date
If-match	Sends the document only if it matches given tag
If-non-match	Sends the document only if it does not match given tag
If-range	Sends only the portion of the document that is missing
If-unmodified-since	Sends the document if not changed since specified date
Referrer	Specifies the URL of the linked document
User-agent	Identifies the client program

Response headers



Header	*Description*
Accept-range	Shows if server accepts the range requested by client
Age	Shows the age of the document
Public	Shows the supported list of methods
Retry-after	Specifies the date after which the server is available
Server	Shows the server name and version number

Entity Headers

Header	*Description*
Allow	Lists valid methods that can be used with a URL
Content-encoding	Specifies the encoding scheme
Content-language	Specifies the language
Content-length	Shows the length of the document
Content-range	Specifies the range of the document
Content-type	Specifies the medium type
Etag	Gives an entity tag
Expires	Gives the date and time when contents may change
Last-modified	Gives the date and time of the last change
Location	Specifies the location of the created or moved document

Example – To retrieve a document

- Use GET method to retrieve an image with path /usr/bin/image1.
- Request line shows method (GET), URL, and HTTP version (1.1).
- Header has two lines that show client can accept images in GIF or JPEG format.
- Request does not have a body.
- Response message contains status line and four lines of header.
- Header lines define date, server, MIME version, and length of document.
- Body of document follows header.

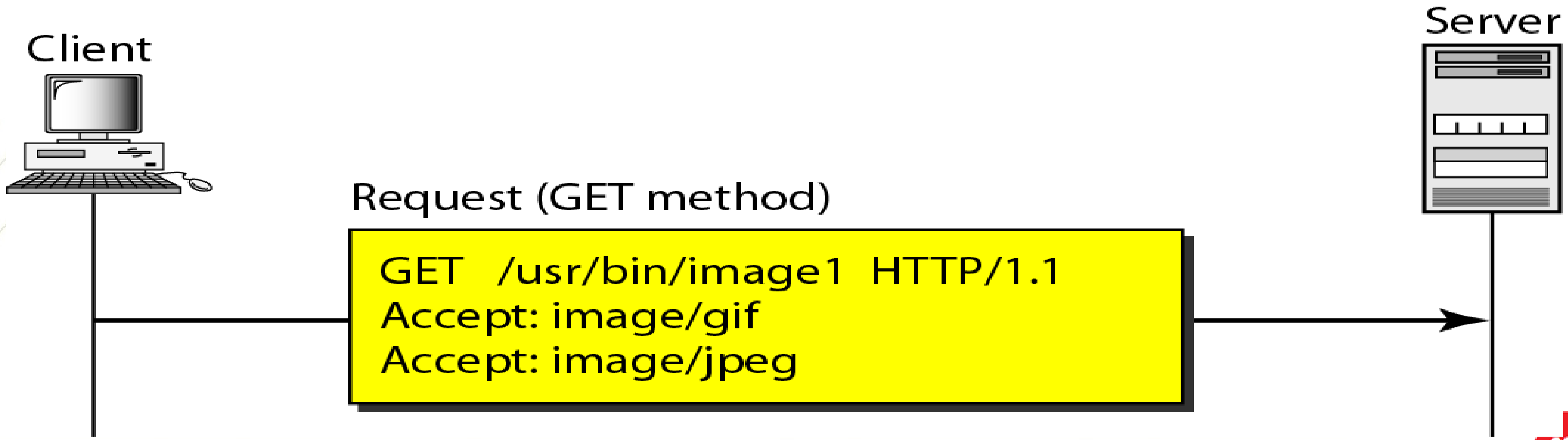
Formulate a HTTP request-response model based on above instructions.

Solution – To retrieve a document

- Use GET method to retrieve an image with path /usr/bin/image1. Request line shows method (GET), URL, and HTTP version (1.1): **GET /usr/bin/image1 HTTP/1.1-------- Under request section**
- Header has two lines that show client can accept images in GIF or JPEG format:

Accept: image/gif

Accept: image/jpeg



```
GET  /usr/bin/image1  HTTP/1.1
Accept: image/gif
Accept: image/jpeg
```

Solution – To retrieve a document

- Response message contains status line and four lines of header. Header lines define date, server, MIME version, and length of document.

  HTTP/1.1 200 OK

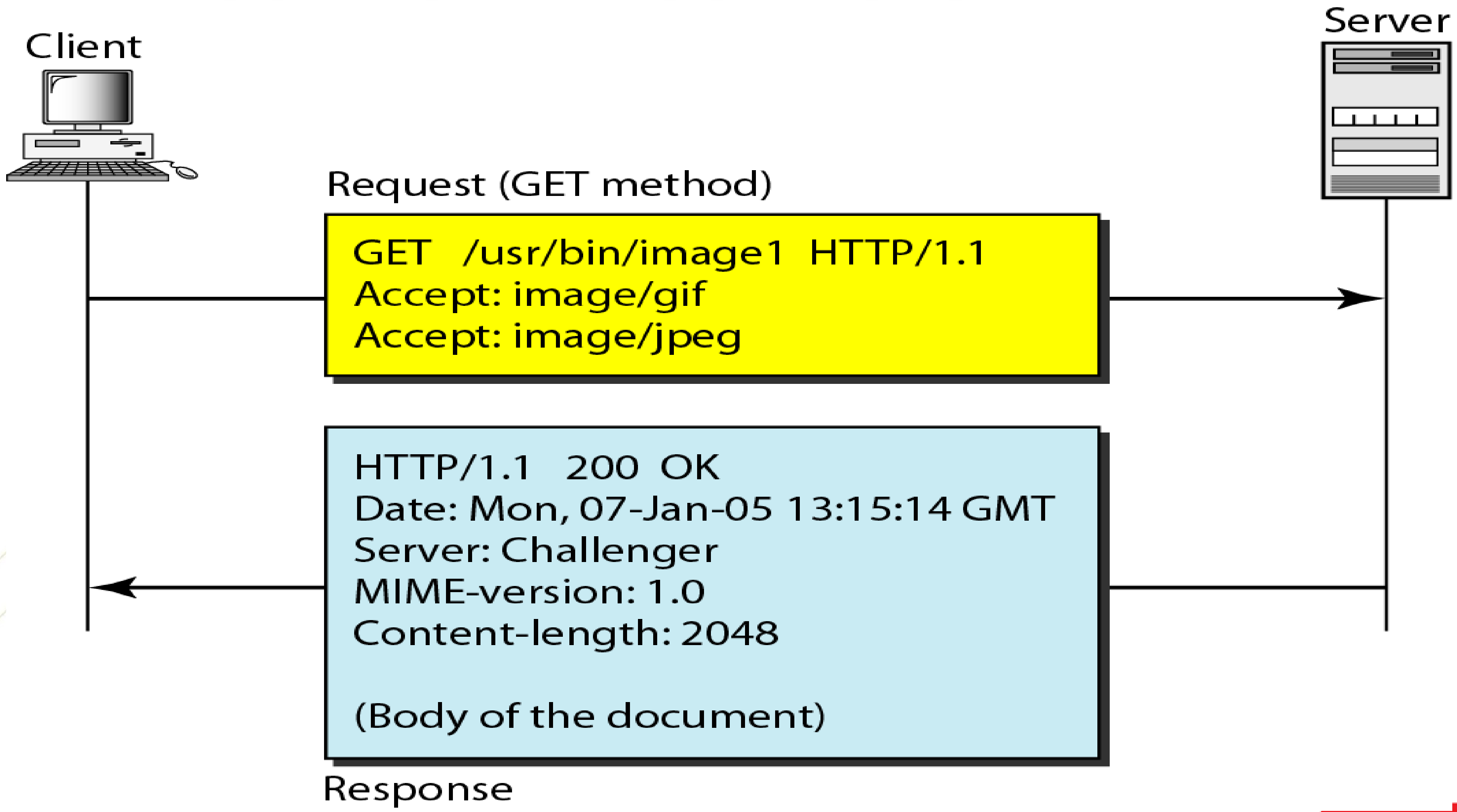
  Date: Mon, 07-Jan-05 13:15:14 GMT

  Server: Challenger

  Content-length: 2048

- Body of document follows header: **(Body of Document)**

Solution – To retrieve a document



```
GET   /usr/bin/image1  HTTP/1.1
Accept: image/gif
Accept: image/jpeg
```

```
HTTP/1.1   200  OK
Date: Mon, 07-Jan-05 13:15:14 GMT
Server: Challenger
MIME-version: 1.0
Content-length: 2048

(Body of the document)
```

Example – Client sending data to server.

- Use POST method.
- Request line shows method (POST), URL, and HTTP version (1.1).
- There are four lines of headers.
- Request body contains input information.
- Response message contains status line and four lines of headers.
- Include created document (CGI document) as body.

Formulate a HTTP request-response model based on above instructions.

Solution – Client sending data to server.

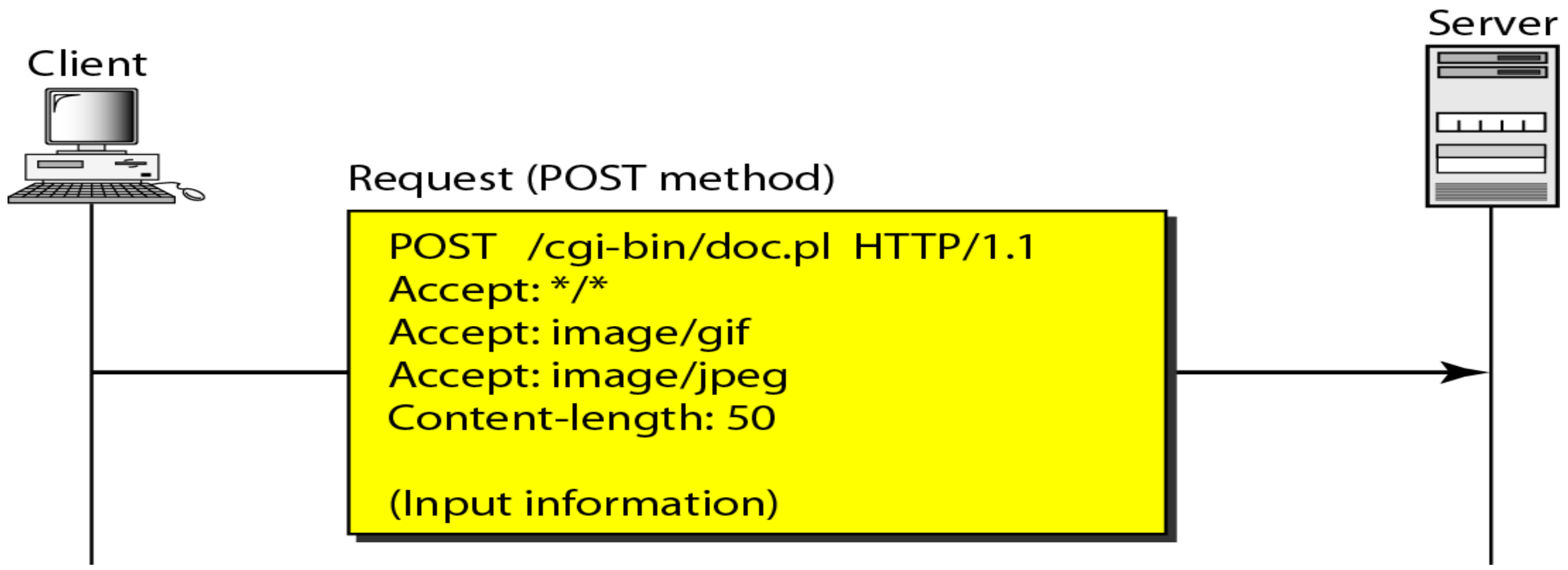
- Use POST method. Request line shows method (POST), URL, and HTTP version (1.1).
- **POST /cgi-bin/doc.pl   HTTP/1.1-------- Under request section**
- There are four lines of headers.

  Accept: */*

  Accept: image/jpeg

  Accept: image/gif

  Content-length: 50



```
POST   /cgi-bin/doc.pl  HTTP/1.1
Accept: */*
Accept: image/gif
Accept: image/jpeg
Content-length: 50

(Input information)
```

- Request body contains input information: **(Input information)**

Solution – Client sending data to server.

- Response message contains status line and four lines of headers.

  HTTP/1.1   200 OK

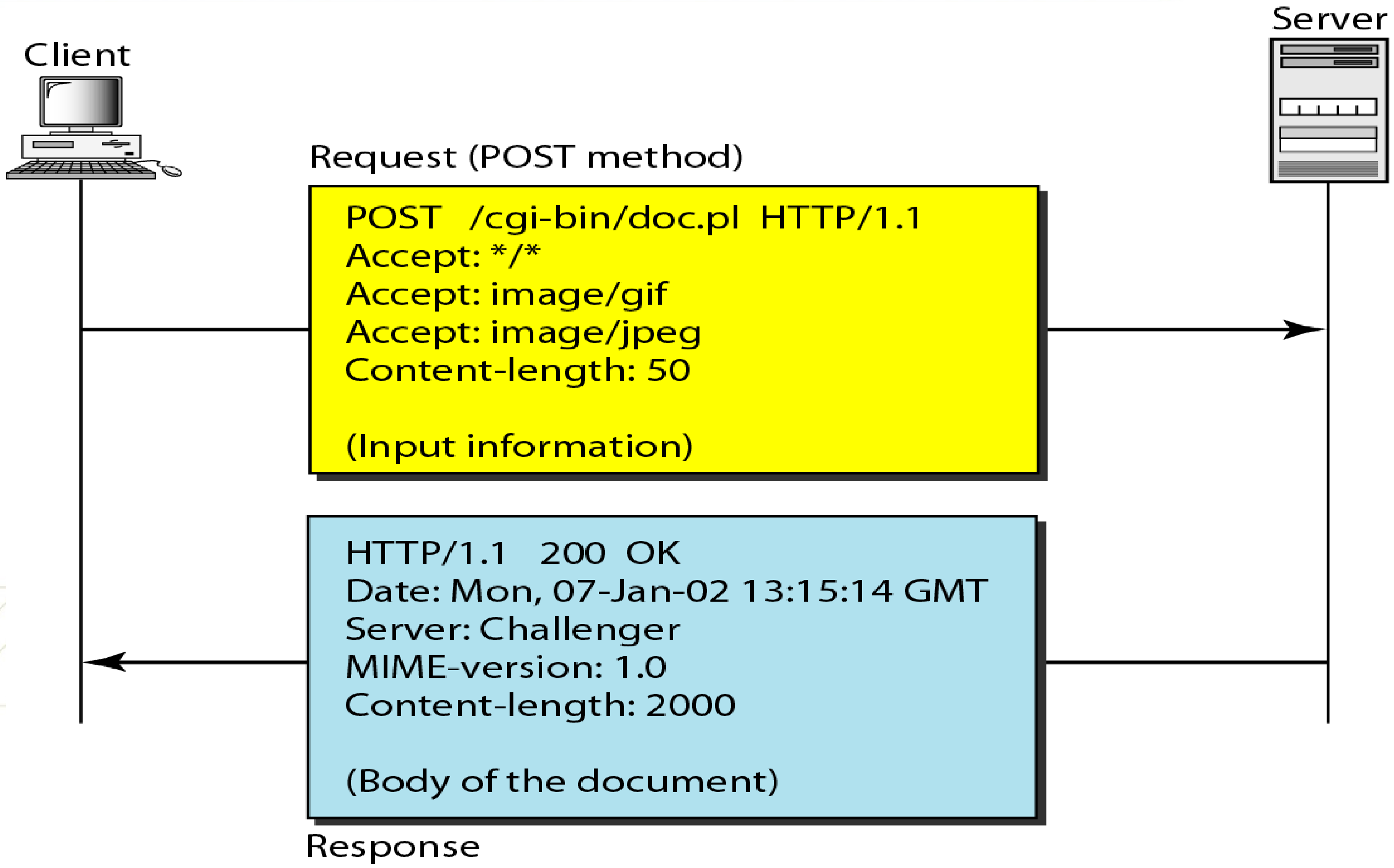
  Date: Mon, 07-Jan-05 13:15:14 GMT

  Server: Challenger

  Content-length: 2000

- Include created document (CGI document) as body: **(Body of Document)**

Solution – Client sending data to server.



```
POST   /cgi-bin/doc.pl  HTTP/1.1
Accept: */*
Accept: image/gif
Accept: image/jpeg
Content-length: 50

(Input information)
```

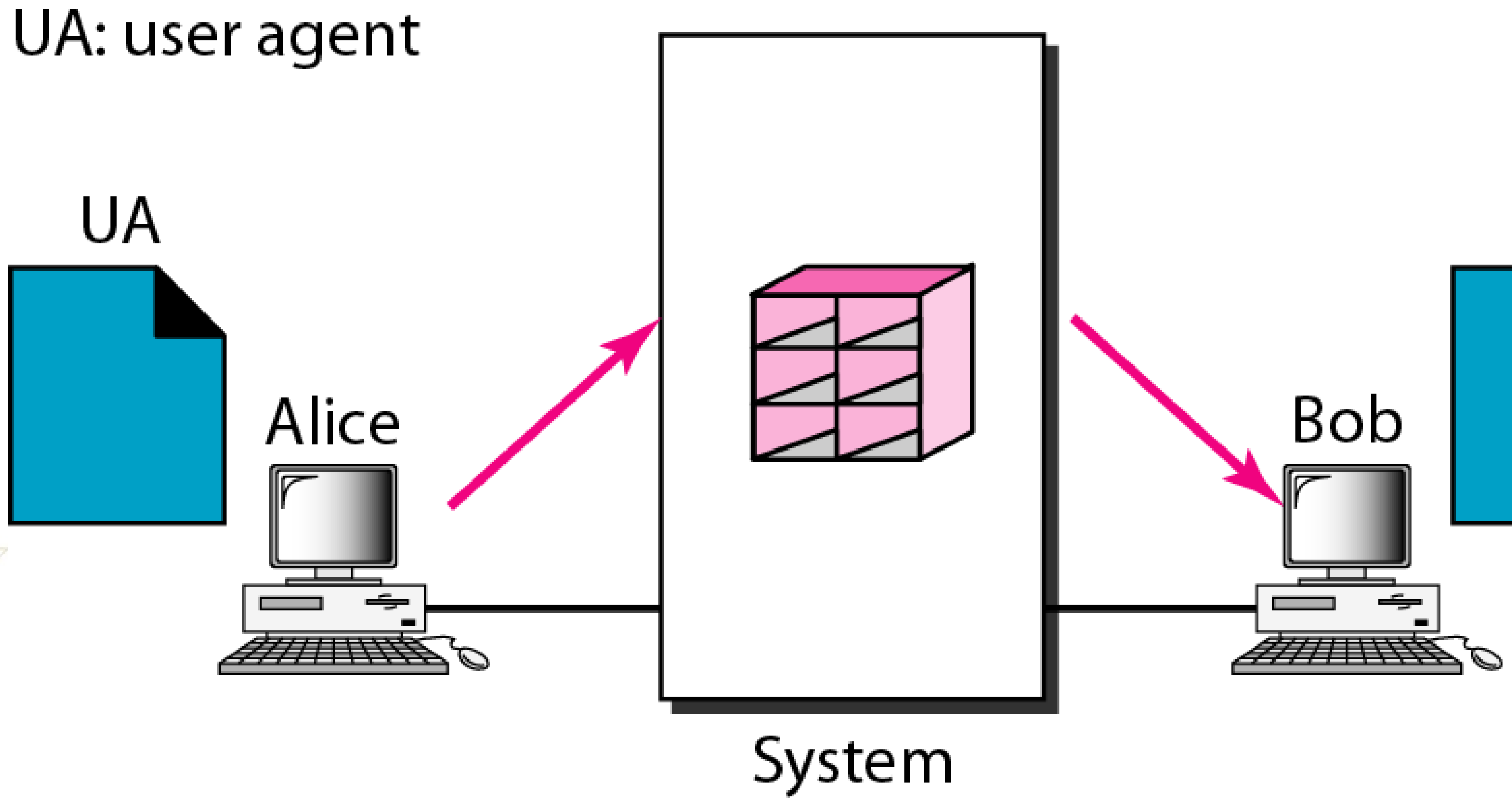
```
HTTP/1.1   200  OK
Date: Mon, 07-Jan-02 13:15:14 GMT
Server: Challenger
MIME-version: 1.0
Content-length: 2000

(Body of the document)
```

SMTP – Simple Mail Transfer Protocol

- Set of communication guidelines that allow software to transmit an e-mail over internet.
- Program used for sending messages to other computer users based on e-mail addresses.
- Provides mail exchange between users on same or different computers
- Can send a single message (like - text, voice, video or graphics) to one or more recipients
- Can send messages on networks outside internet.
- Components of SMTP:
  - ❑ User Agent (UA): prepares message, creates envelope then puts message in envelope
  - ❑ Mail Transfer Agent (MTA): transfers this mail across internet.

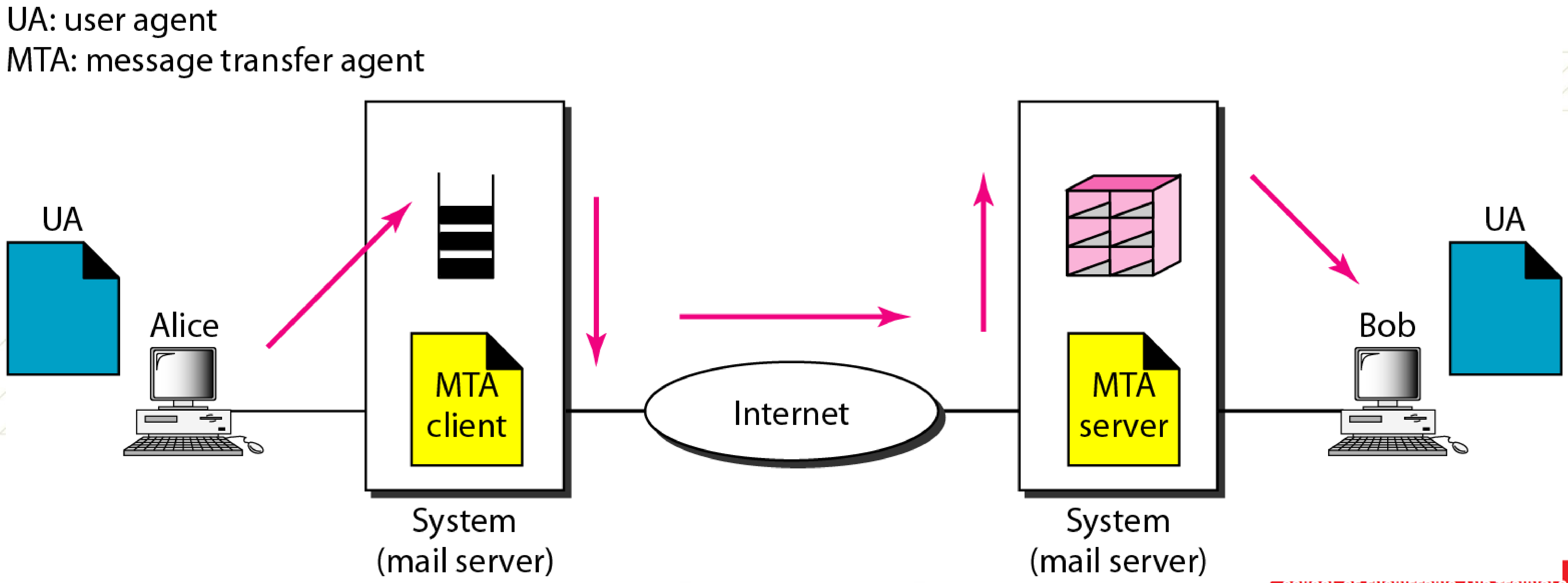
SMTP – Case 1

- When sender and receiver of an e-mail are on same system, it needs only two user agents.



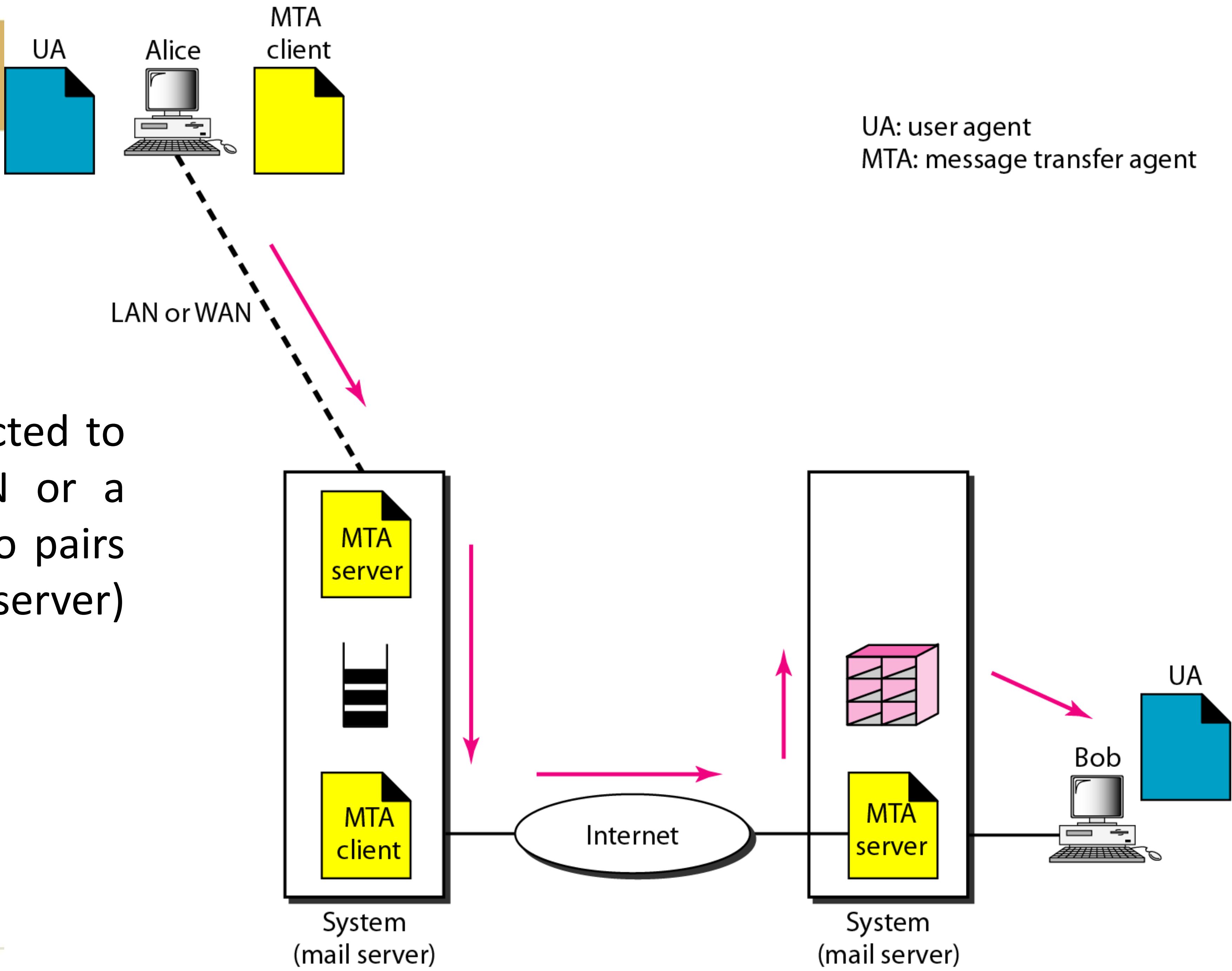
SMTP – Case 2

- When sender and receiver of an e-mail are on different systems, two UAs and a pair of MTAs (client and server) are needed

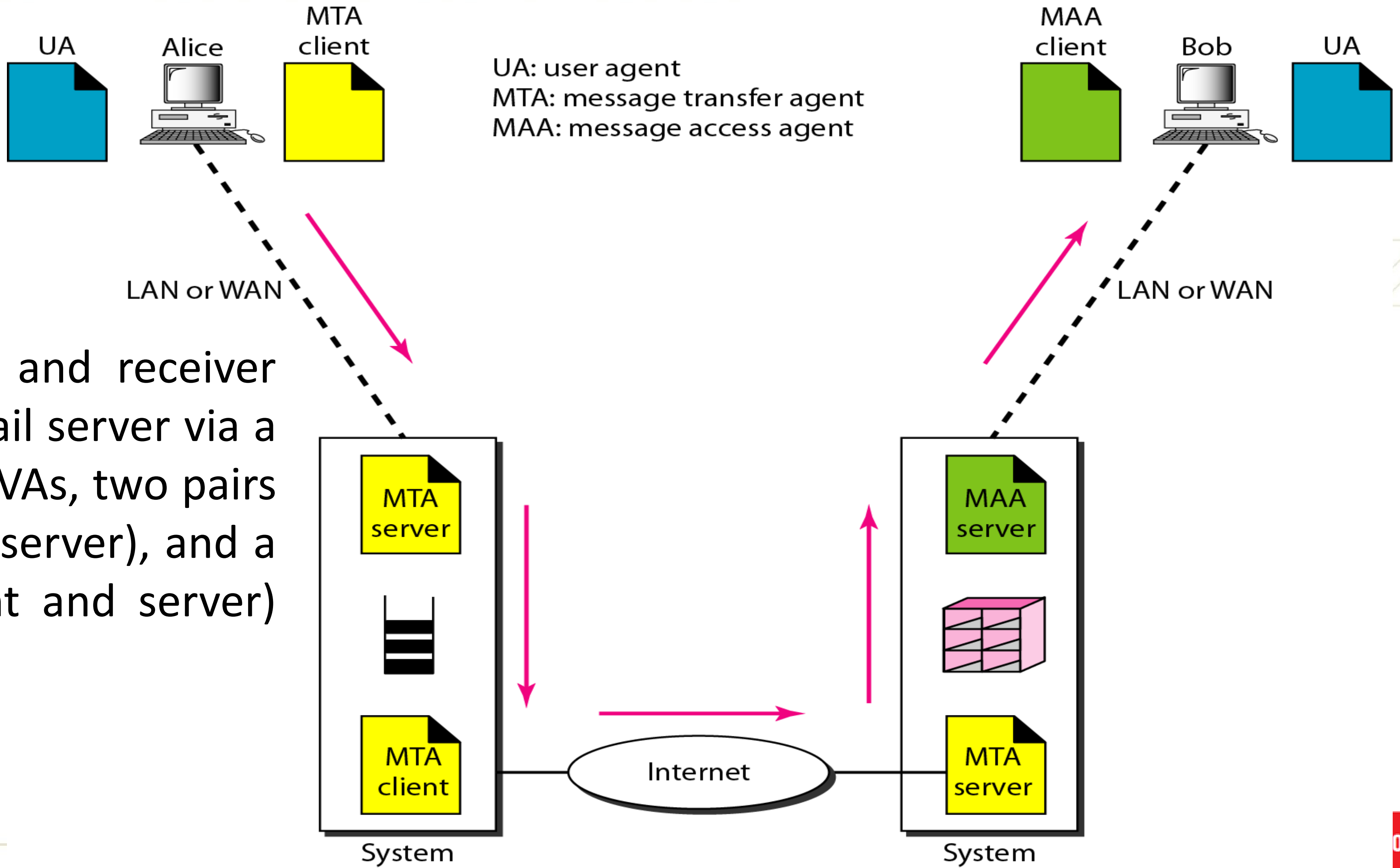


SMTP – Case 3

When sender is connected to mail server via a LAN or a WAN, two UAs and two pairs of MTAs (client and server) are needed



SMTP – Case 4- Most common situation

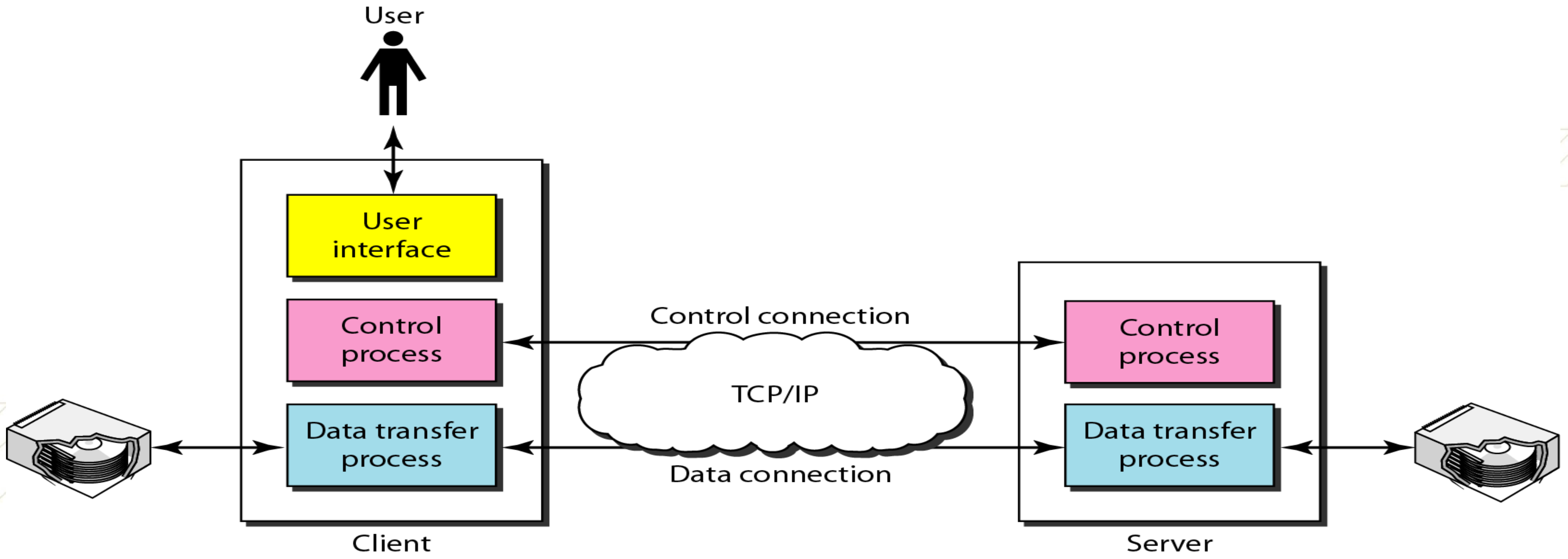


When both sender and receiver are connected to mail server via a LAN or a WAN, two VAs, two pairs of MTAs (client and server), and a pair of MAAs (client and server) are needed.

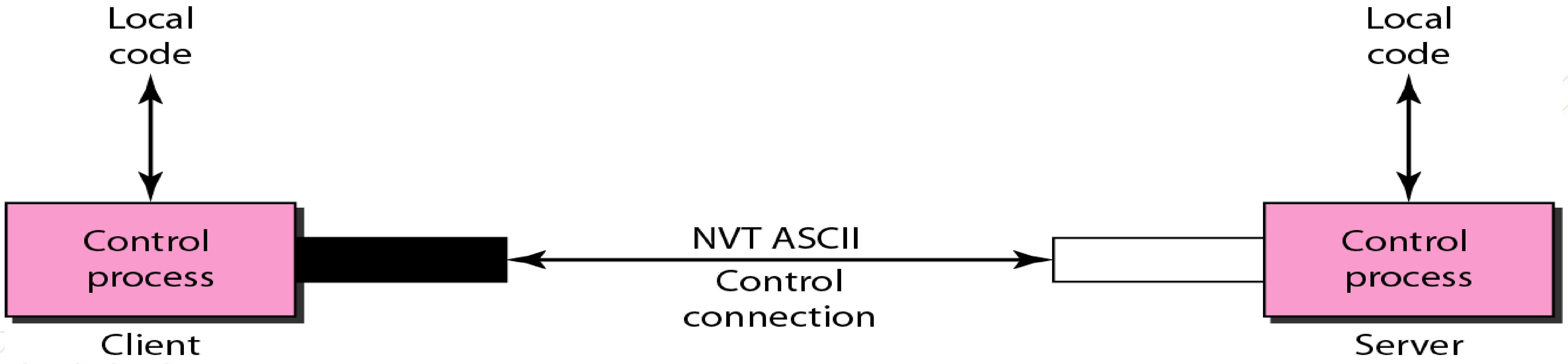
FILE TRANSFER (FTP) Protocol

- Transferring files from one computer to another is most common tasks expected from a networking or internetworking environment.
- Greatest volume of data exchange in Internet today is due to file transfer.
- FTP uses services of TCP.
- It needs two TCP connections.
- Well-known port 21 is used for control connection and well-known port 20 for data connection.

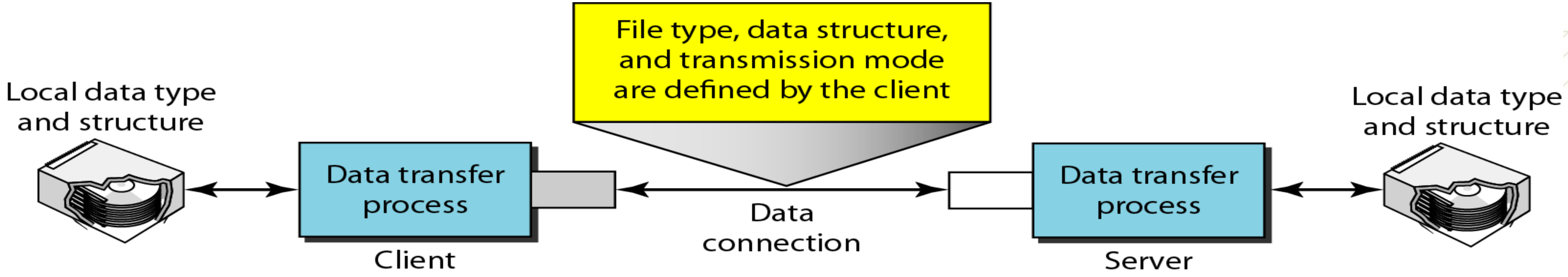
FILE TRANSFER (FTP) Protocol



Using control connection



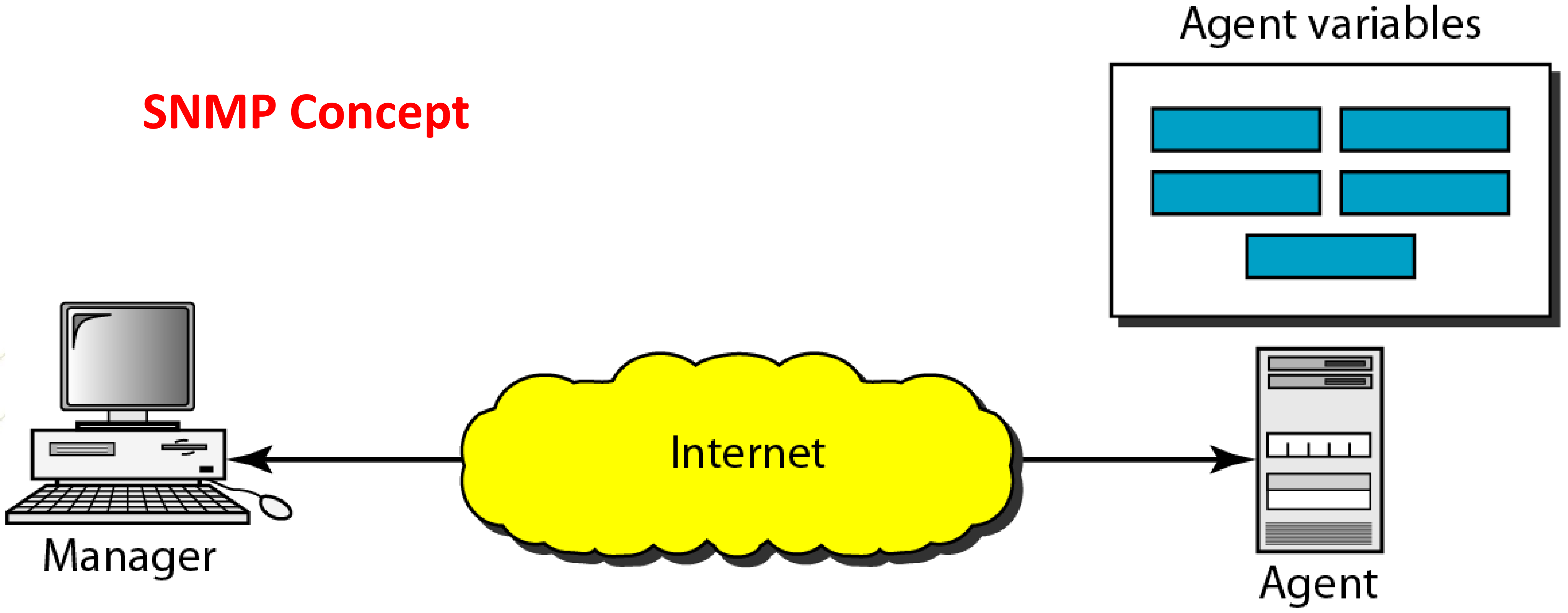
Using data connection


File type, data structure, and transmission mode are defined by the client
Local data type and structure
Data transfer process
Client
Data connection
Data transfer process
Server
Local data type and structure

Simple Network Management Protocol (SNMP)

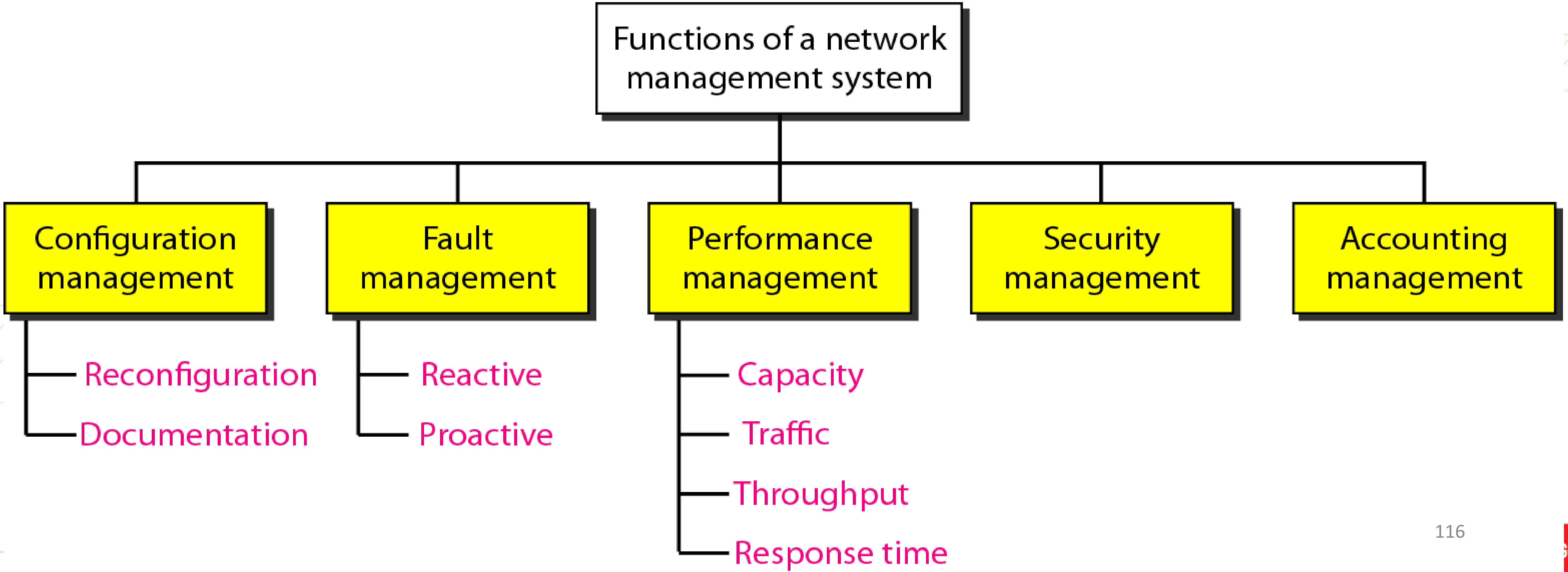
- SNMP is a framework for managing devices in an internet using TCP/IP protocol suite.
- Provides a set of fundamental operations for monitoring and maintaining an internet.

SNMP Concept



Simple Network Management Protocol (SNMP)

- SNMP defines format of packets exchanged between a manager and an agent.
- Reads and changes status (values) of objects (variables) in SNMP packets.



SNMP Management

1. Manager checks an agent by requesting information that reflects behavior of agent.
2. Manager forces an agent to perform a task by resetting values in agent database.
3. Agent contributes to management process by warning manager of an unusual situation.

Syllabus Contents

- **UNIT IV**

❑**Transport Layer Protocols-** Design Issues, Quality of Services. The Internet Transport Protocols. IPV4 vs IPV6.

❑**Session Layer protocol-** Dialog Management, Synchronization, Connection Establishment. Quality of service, security management, Firewalls.

❑**Application layer protocols:** HTTP, SMTP, FTP, SNMP, Etc.

Unit 4- Completed

Thanks

Dr. Divya Agarwal