Chapter 4
Principles of Computer Communications

Network and Transport Layer

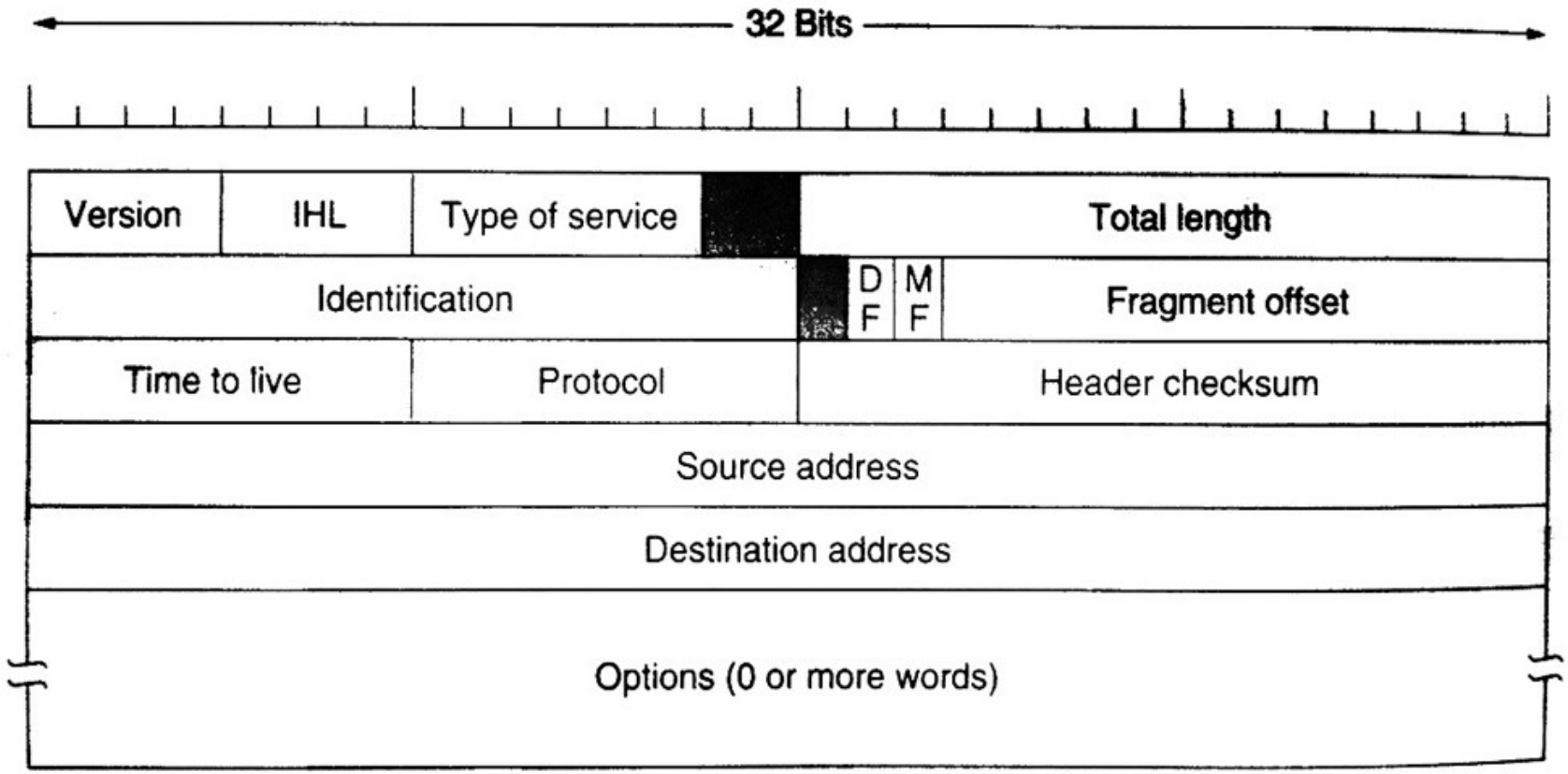
Network and Transport Layer

In this section, the following topics will be covered:

- Address space and addressing
- Network routing
- Congestion control

Internet Protocol (IP)

- IPv4 Header
  - An IP datagram consists of a header part and a text part.
  - The header has a 20-byte fixed part and a variable length optional part.



Internet Protocol (IP)

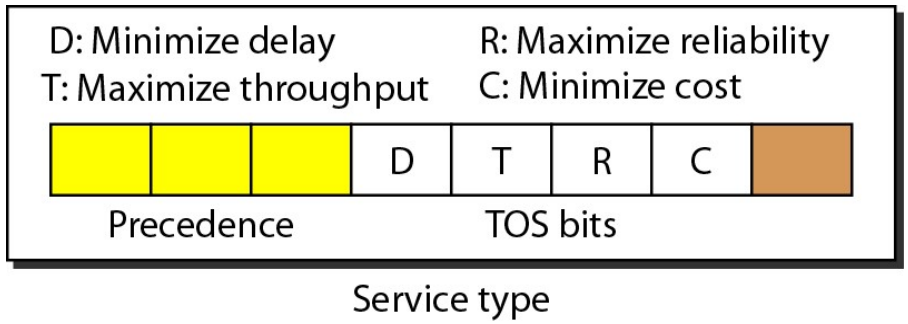
- Version: to keep track of which version of the protocol the datagram belongs to.
- IHL: to provide the length of the header, in 32-bit words. When the value is 5, which applies no options are present.
- Type of service: to distinguish between different classes of services.
- Three-bit precedence field and three flags, D, T, and R {Delay, Throughput and Reliability}.
- Total Length: to provide the length of the datagram (including header and data).

Internet Protocol (IP)

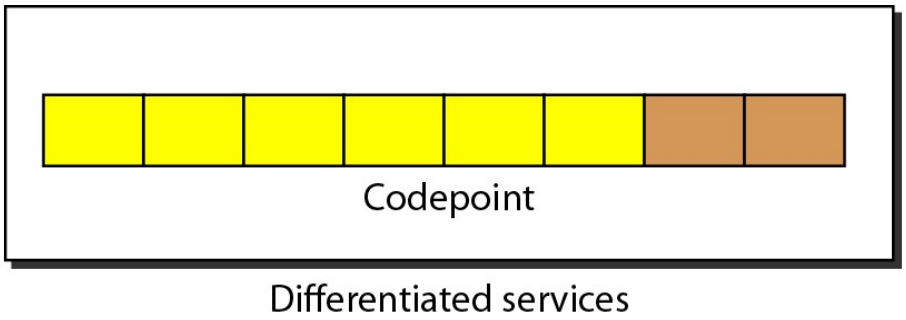
- Identification: to identify each datagram.
- DF and MF: don’t fragment, and more fragments.
- Fragment offset: To tell where in the current datagram this fragment belongs.
- Time to live: a counter used to limit packet lifetimes.
- Protocol: to tell which protocol process to give it to (e.g., TCP or UDP).
- Header checksum: to detect errors in the datagram.
- Source address and destination address

Internet Protocol (IP)

- Service type or differentiated services
- The precedence subfield was part of version 4, but never used.



Service type



Differentiated services

Internet Protocol (IP)

Types of service

TOS Bits	*Description*
0000	Normal (default)
0001	Minimize cost
0010	Maximize reliability
0100	Maximize throughput
1000	Minimize delay

Internet Protocol (IP)

- Default types of service

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

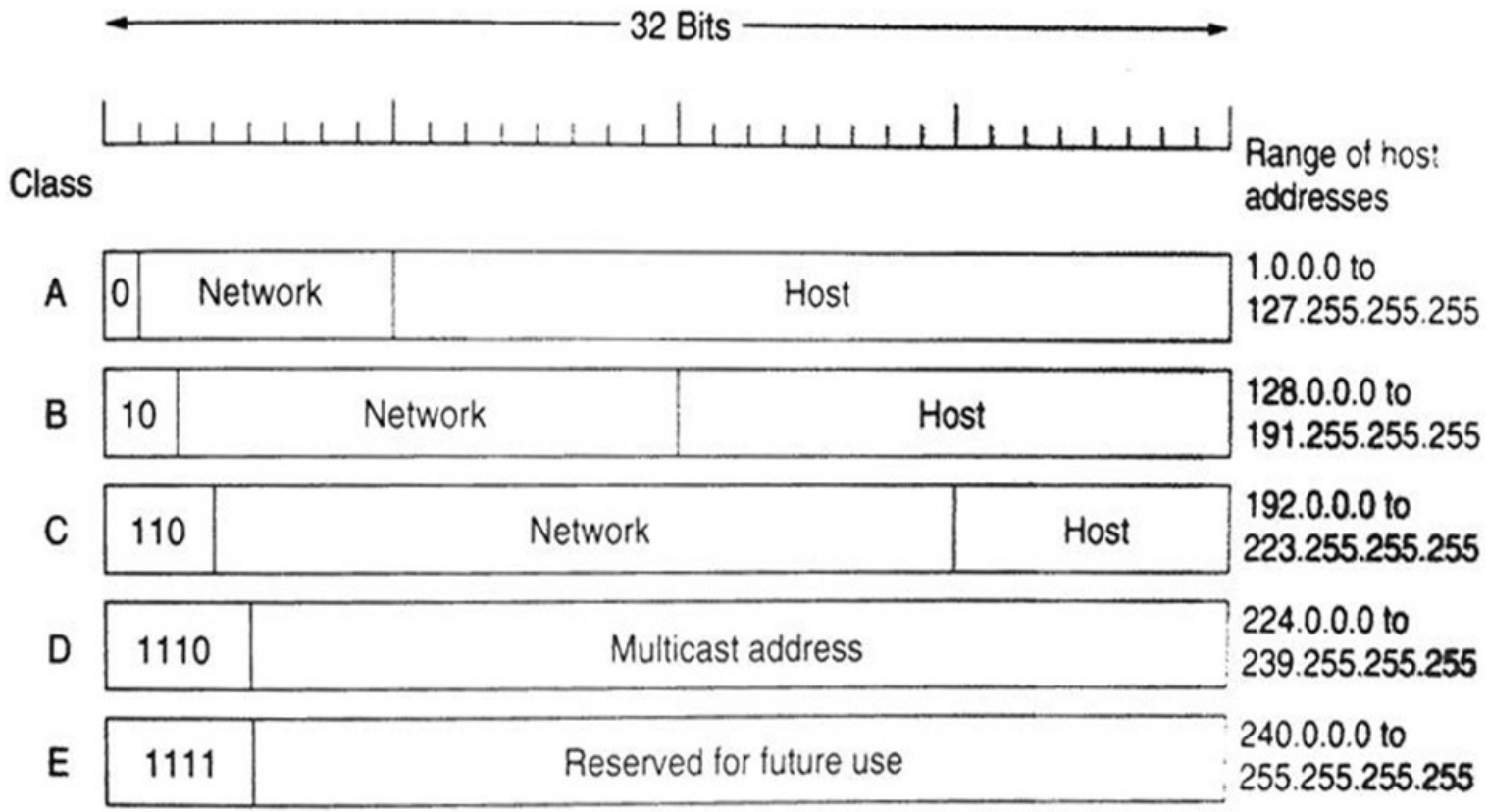
Internet Protocol (IP)

Values for codepoints

Value	*Protocol*
1	ICMP
2	IGMP
6	TCP
17	UDP
89	OSPF

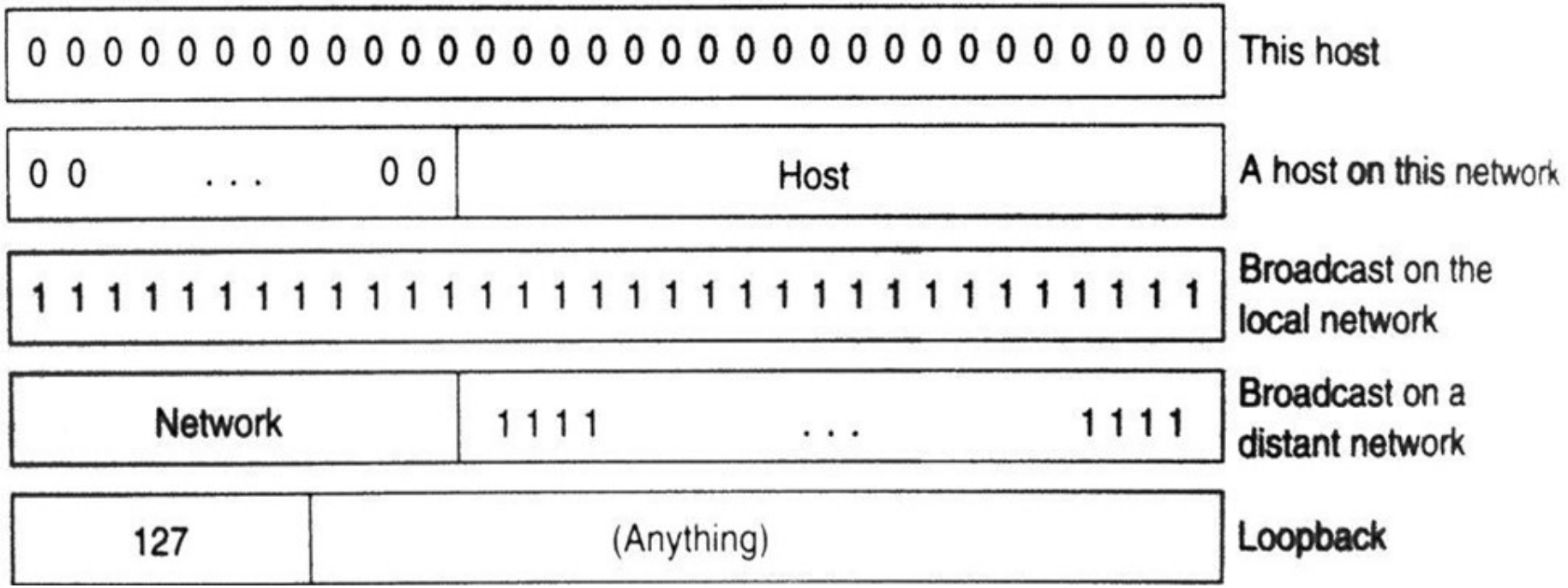
IP Address

- An IP address consists of two parts: network number and host number. (Why?)
- Unique: no two machines on the Internet have the same IP address.
- 32-bit long.



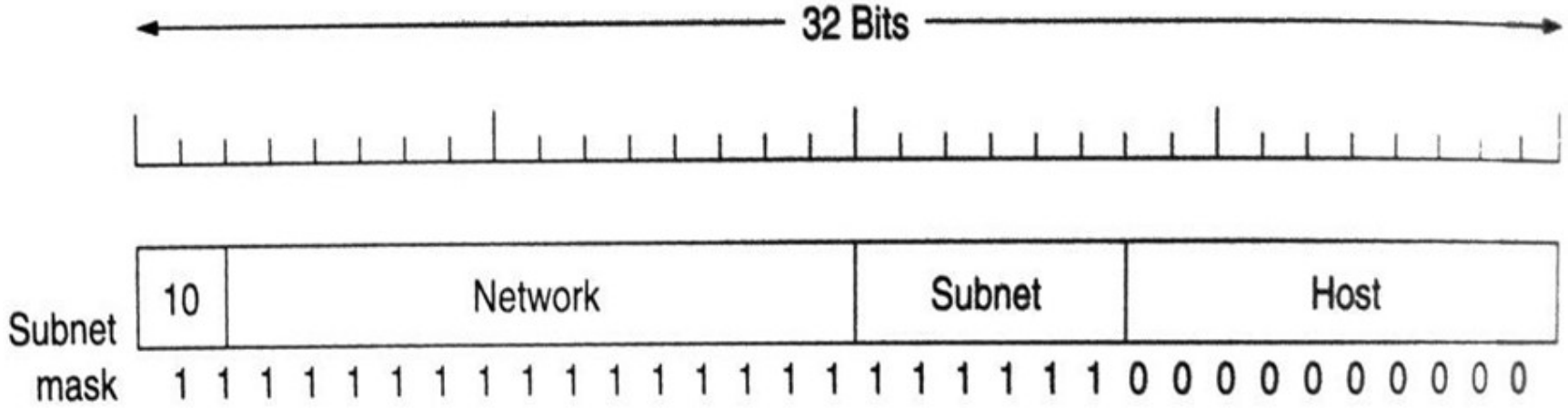
IP Address

- Classful addressing: IP addresses are divided into five categories.
- Usually a 32-bit IP address is written in **dotted decimal notation**. Each of 4 bytes is written in decimal (e.g., 158.182.7.1).
- Special IP addresses



IP Address

- Subnet
  - It is possible to split a large network into subnets.
  - A subnet mask is used to indicate the split of network.



A class B network subnetted into 64 subnets.

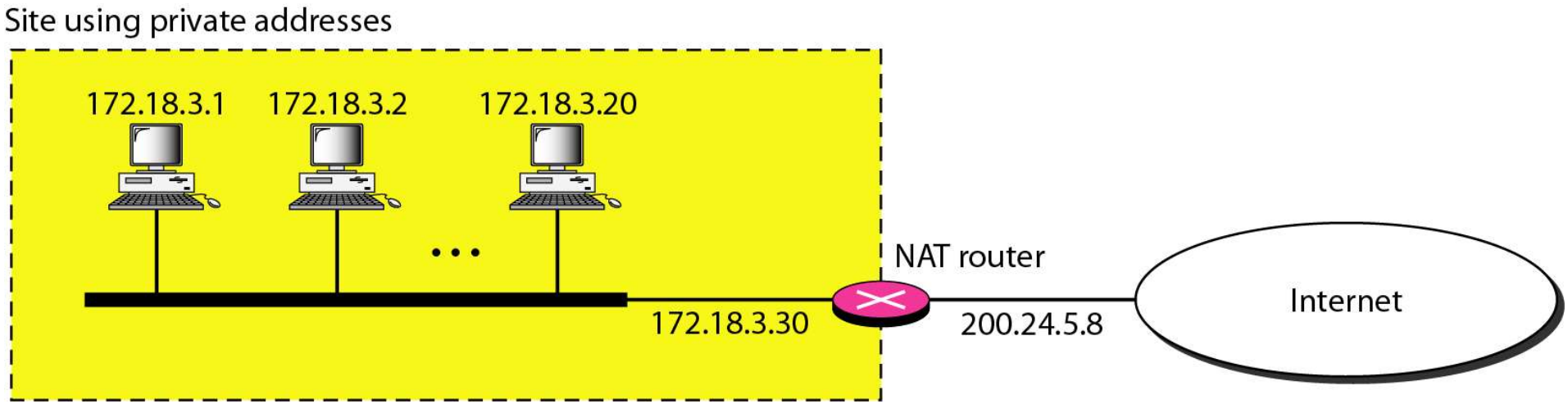
IP Address

- To increase address space by using NAT
- Addresses for private networks

Range			*Total*
10.0.0.0	to	10.255.255.255	2^{24}
172.16.0.0	to	172.31.255.255	2^{20}
192.168.0.0	to	192.168.255.255	2^{16}

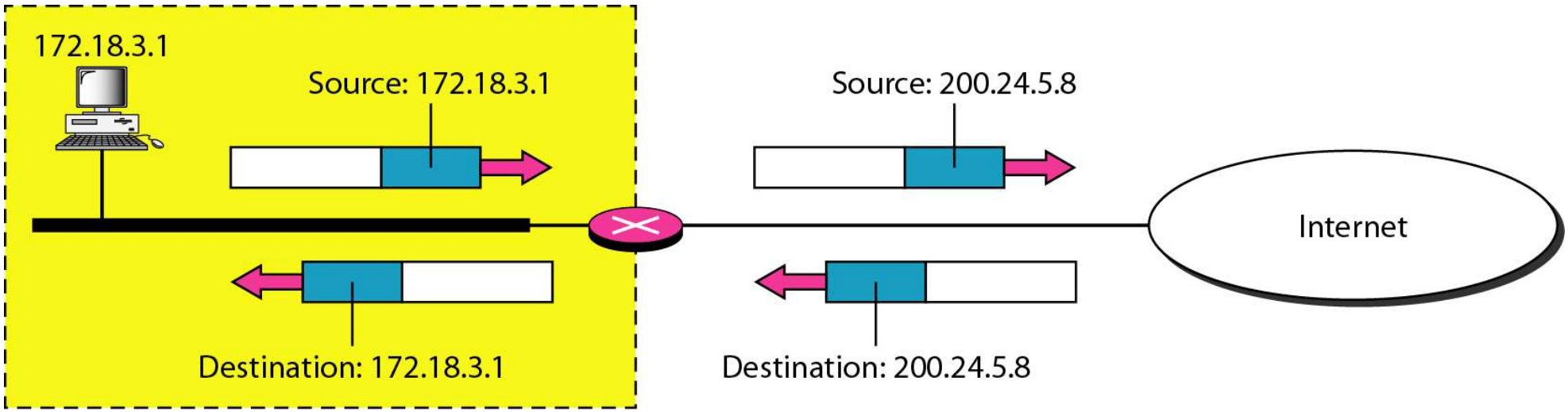
IP Address

- A NAT implementation



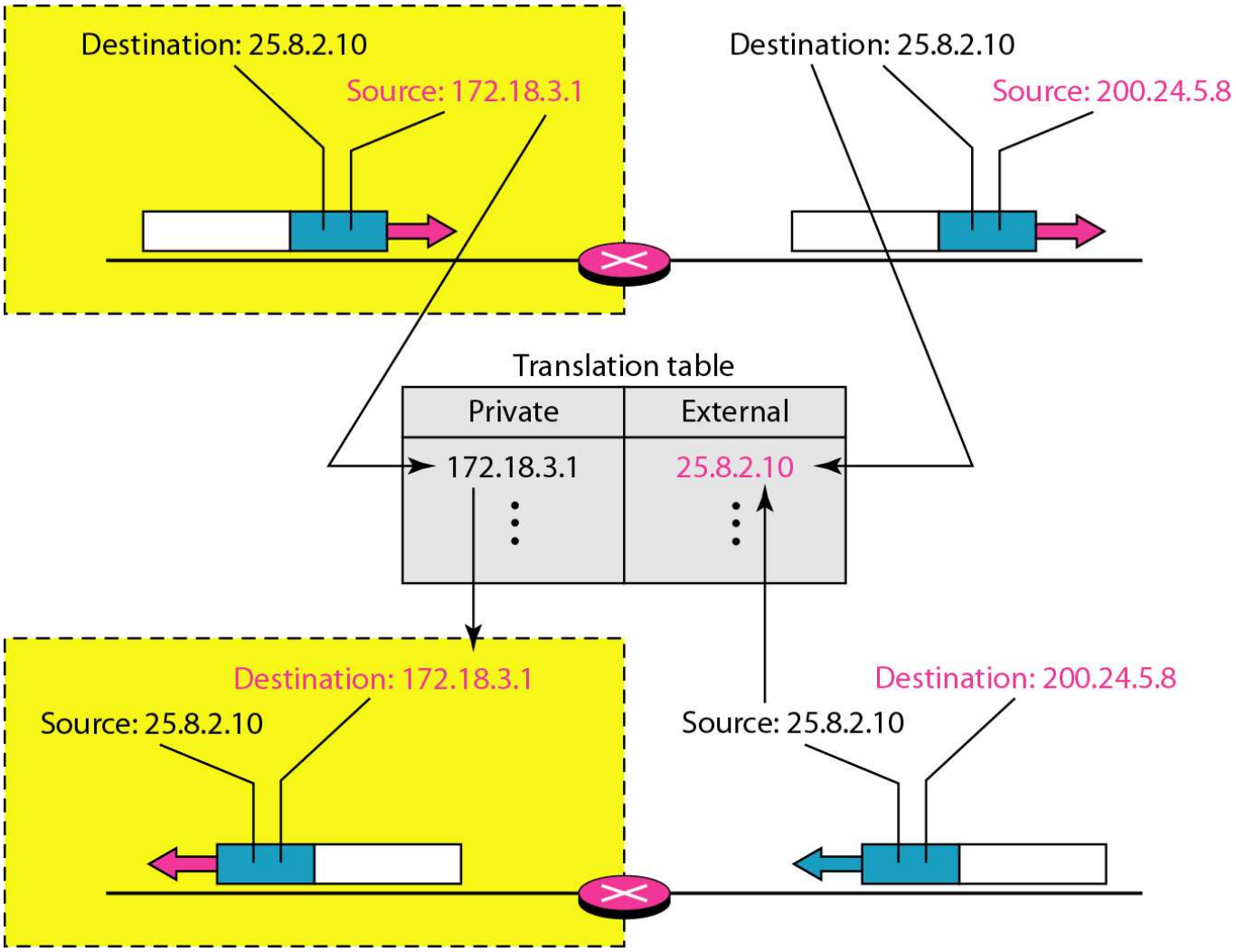
IP Address

Addresses in a NAT



IP Address

- NAT address translation

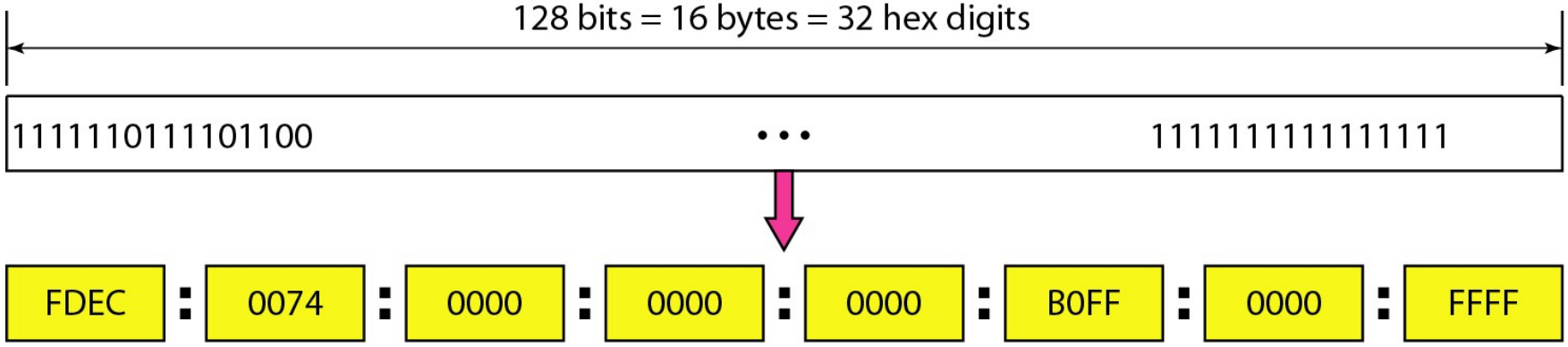


IPv6

- The network layer protocol in the TCP/IP protocol suite is currently IPv4.
- Although IPv4 is well designed, data communication has evolved since the inception of IPv4 in the 1970s.
- IPv4 has some deficiencies that make it unsuitable for the fast-growing Internet.
- The new IP version is to
  - Support more hosts on the Internet;
  - Provide better quality of service.

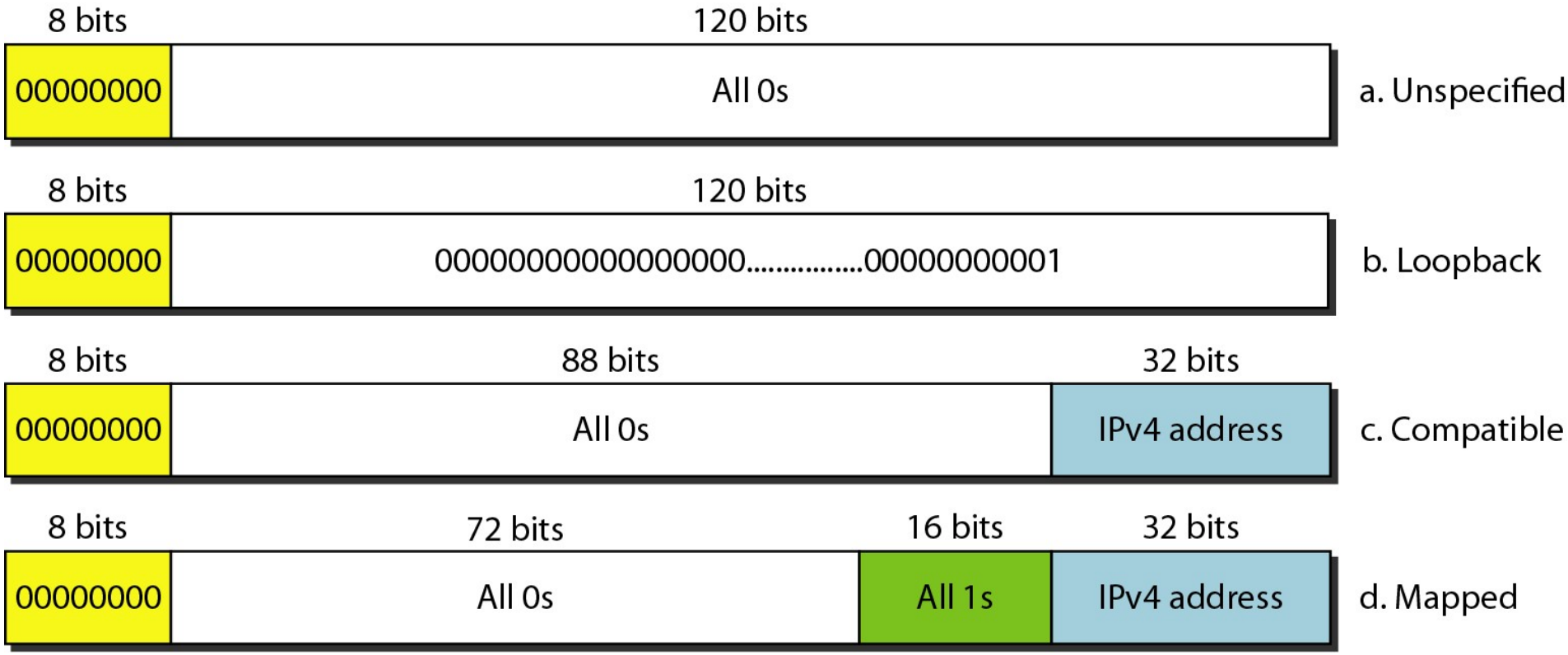
IPv6 Header

- 40-byte header
- 16-byte address
- Address are written as eight groups of four hexadecimal digits with colons between groups: e.g., 8000:0000:89AB:CDEF: 8000:0000:89AB:CDEF
- IPv6 address in binary and hexadecimal colon notation

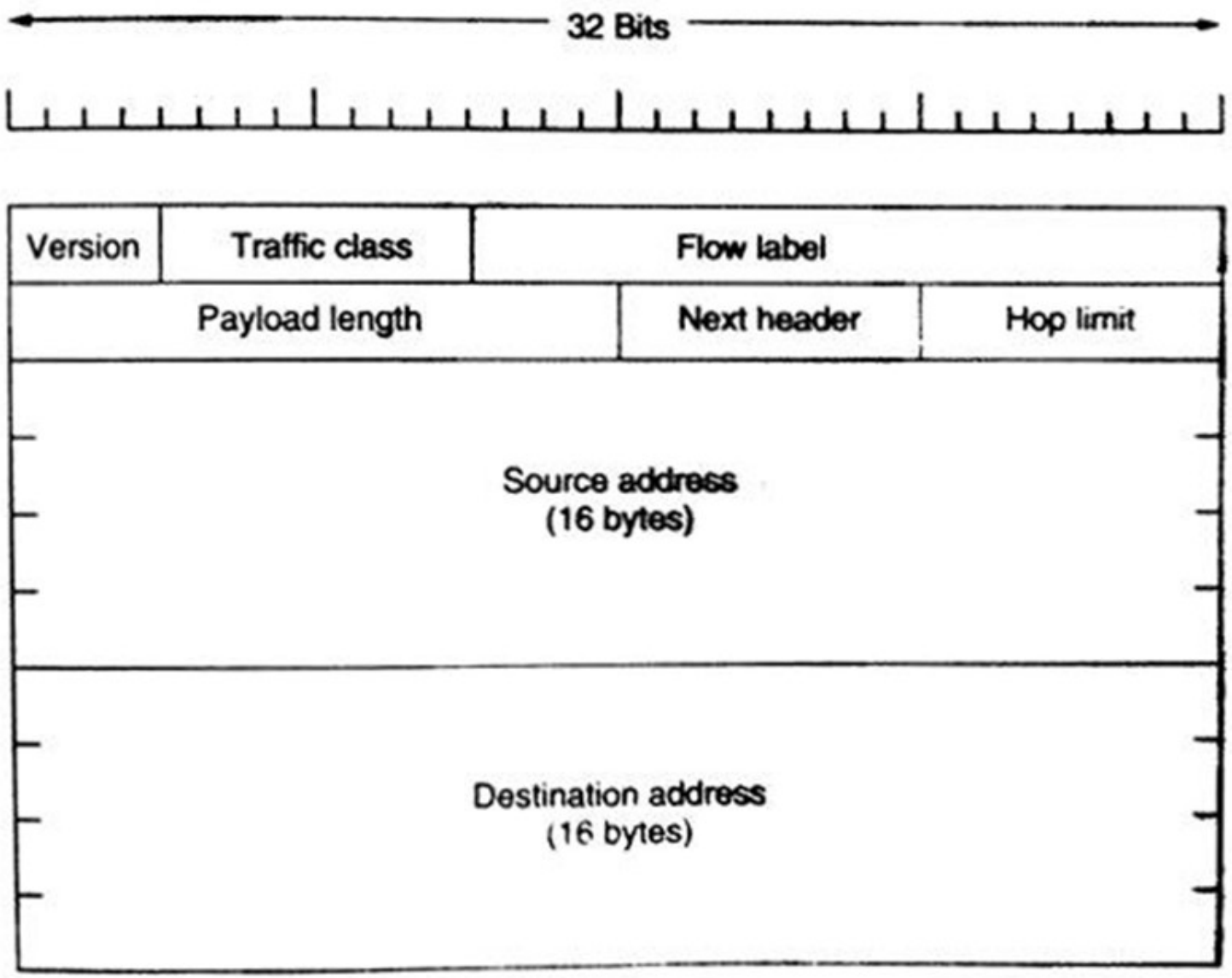


IPv6 Header

Reserved addresses in IPv6

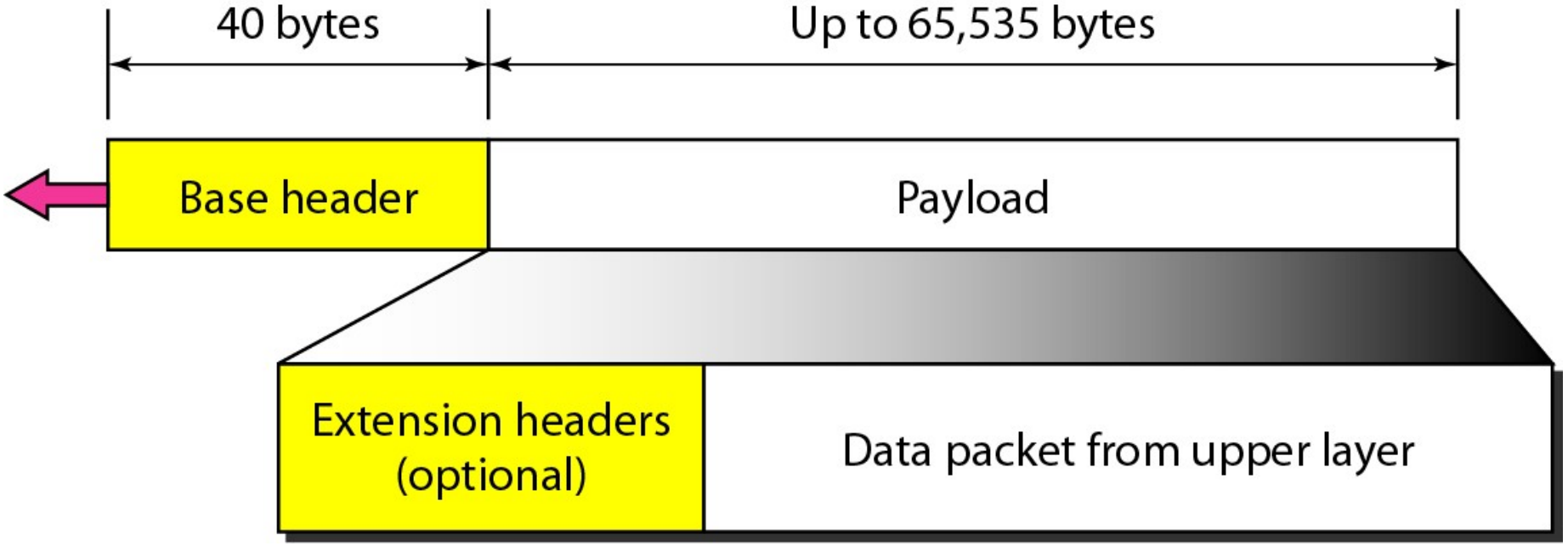


IPv6 Header

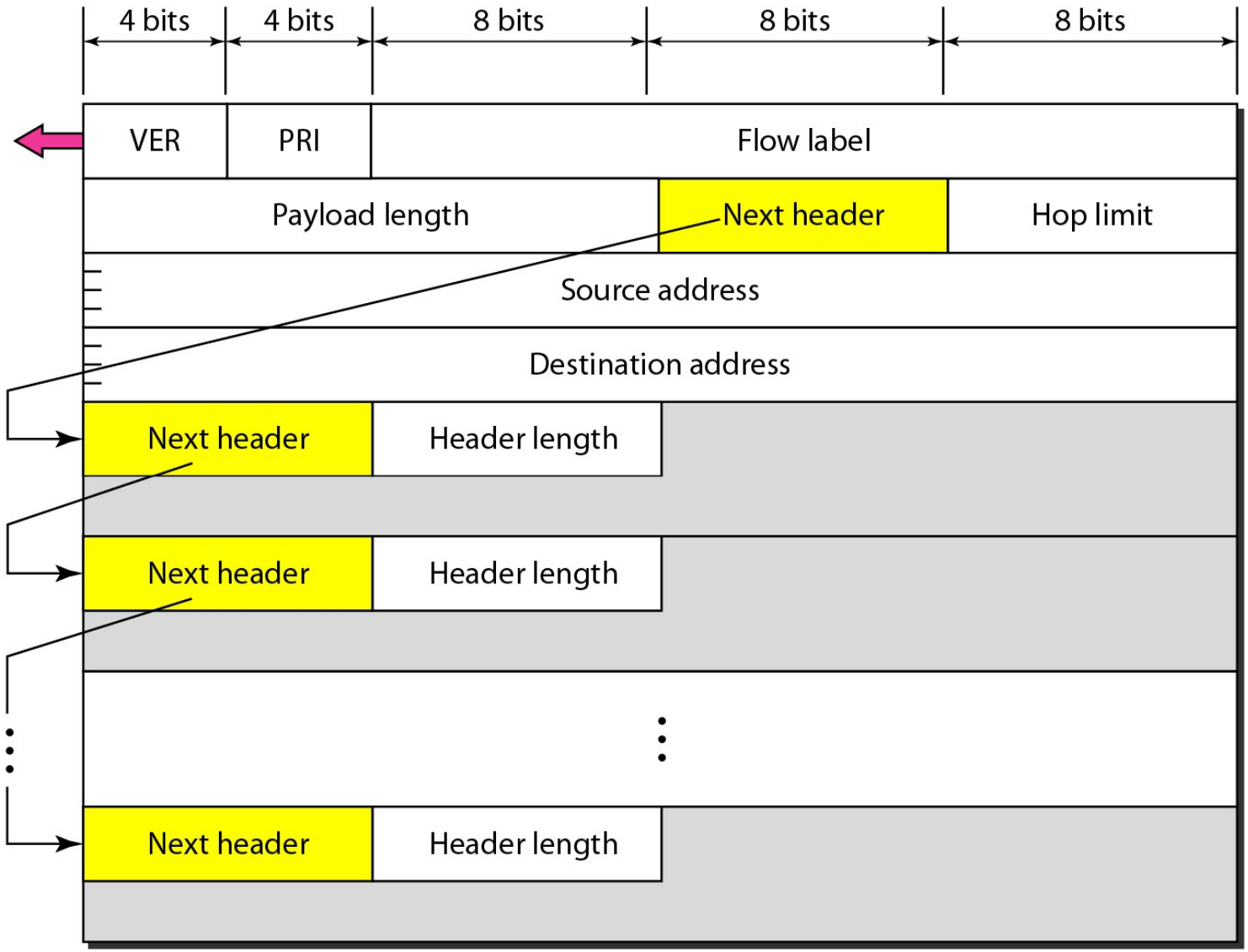

32 Bits
Version
Traffic class
Flow label
Payload length
Next header
Hop limit
Source address
(16 bytes)
Destination address
(16 bytes)

IPv6 Header

IPv6 datagram header and payload



Format of an IPv6 datagram



Comparison between IPv4 and IPv6 Packet Headers

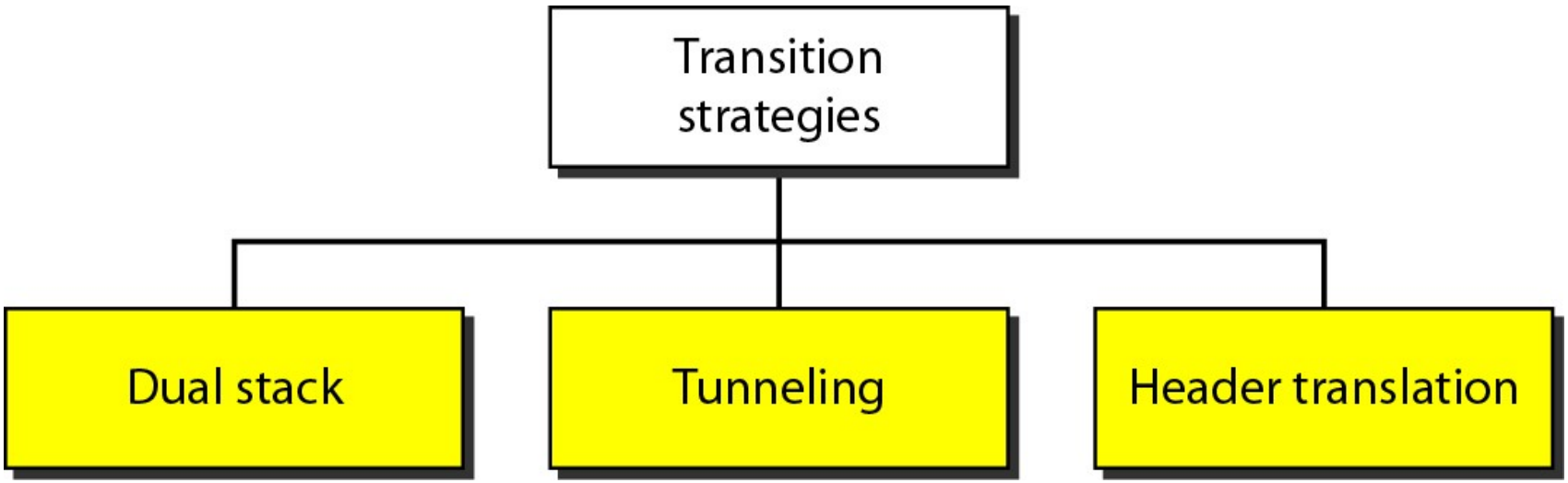
Comparison
1. The header length field is eliminated in IPv6 because the length of the header is fixed in this version.
2. The service type field is eliminated in IPv6. The priority and flow label fields together take over the function of the service type field.
3. The total length field is eliminated in IPv6 and replaced by the payload length field.
4. The identification, flag, and offset fields are eliminated from the base header in IPv6. They are included in the fragmentation extension header.
5. The TTL field is called hop limit in IPv6.
6. The protocol field is replaced by the next header field.
7. The header checksum is eliminated because the checksum is provided by upper-layer protocols; it is therefore not needed at this level.
8. The option fields in IPv4 are implemented as extension headers in IPv6.

Transition from IPv4 to IPv6

- Because of the huge number of systems on the Internet, the transition from IPv4 to IPv6 cannot happen suddenly.
- It takes a considerable amount of time before every system in the Internet can move from IPv4 to IPv6.
- The transition must be smooth to prevent any problems between IPv4 and IPv6 systems.

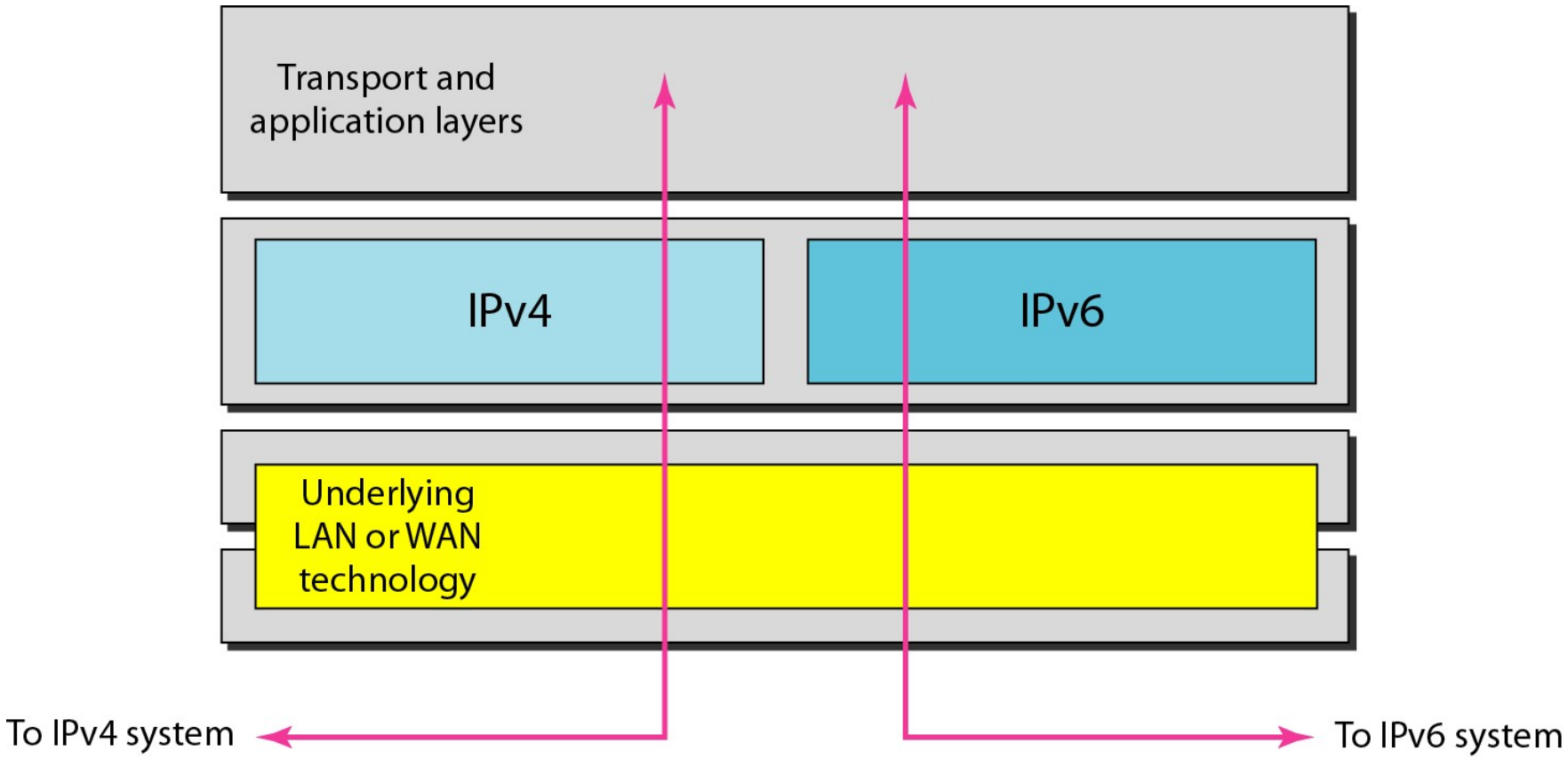
Transition from IPv4 to IPv6

- Three transition strategies



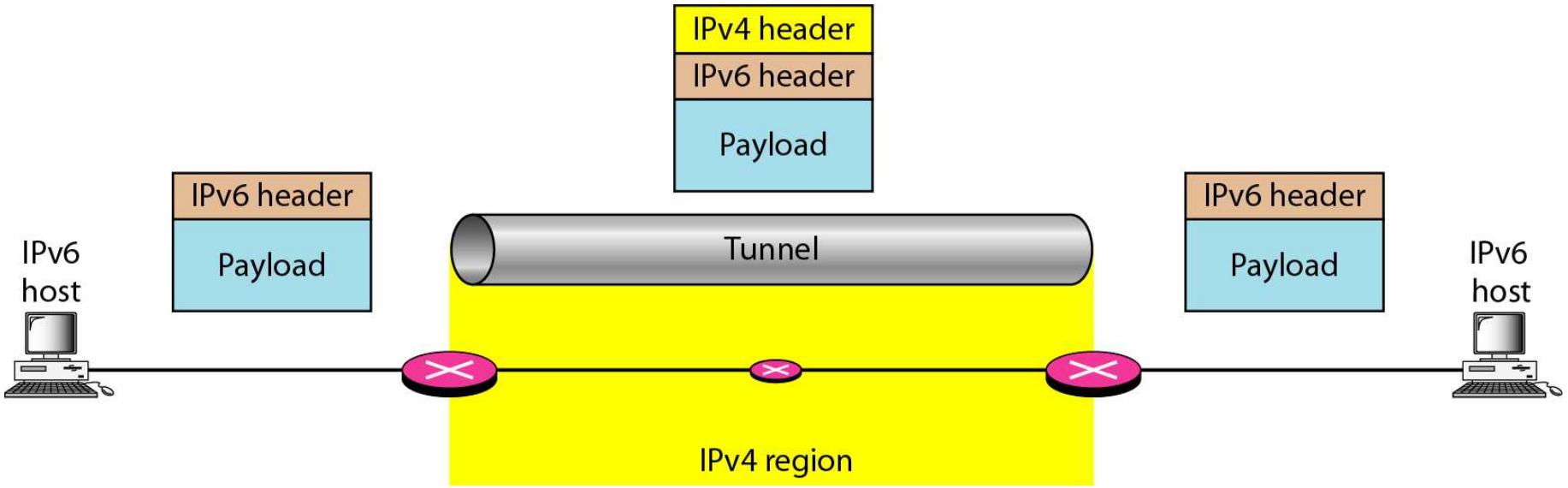
Transition from IPv4 to IPv6

Dual stack



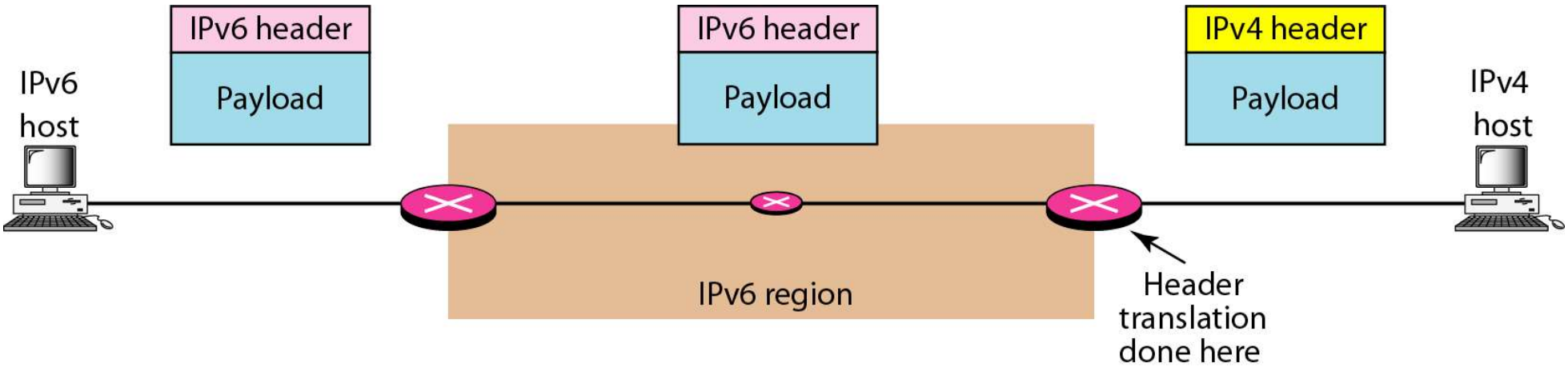
Transition from IPv4 to IPv6

Tunneling strategy



Transition from IPv4 to IPv6

Header translation strategy



Transition from IPv4 to IPv6

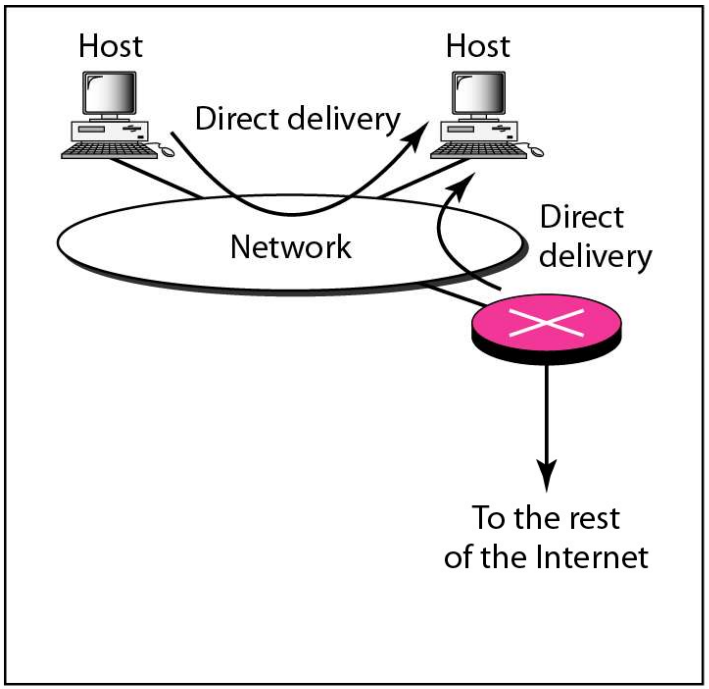
Header translation

Header Translation Procedure
1. The IPv6 mapped address is changed to an IPv4 address by extracting the rightmost 32 bits.
2. The value of the IPv6 priority field is discarded.
3. The type of service field in IPv4 is set to zero.
4. The checksum for IPv4 is calculated and inserted in the corresponding field.
5. The IPv6 flow label is ignored.
6. Compatible extension headers are converted to options and inserted in the IPv4 header. Some may have to be dropped.
7. The length of IPv4 header is calculated and inserted into the corresponding field.
8. The total length of the IPv4 packet is calculated and inserted in the corresponding field.

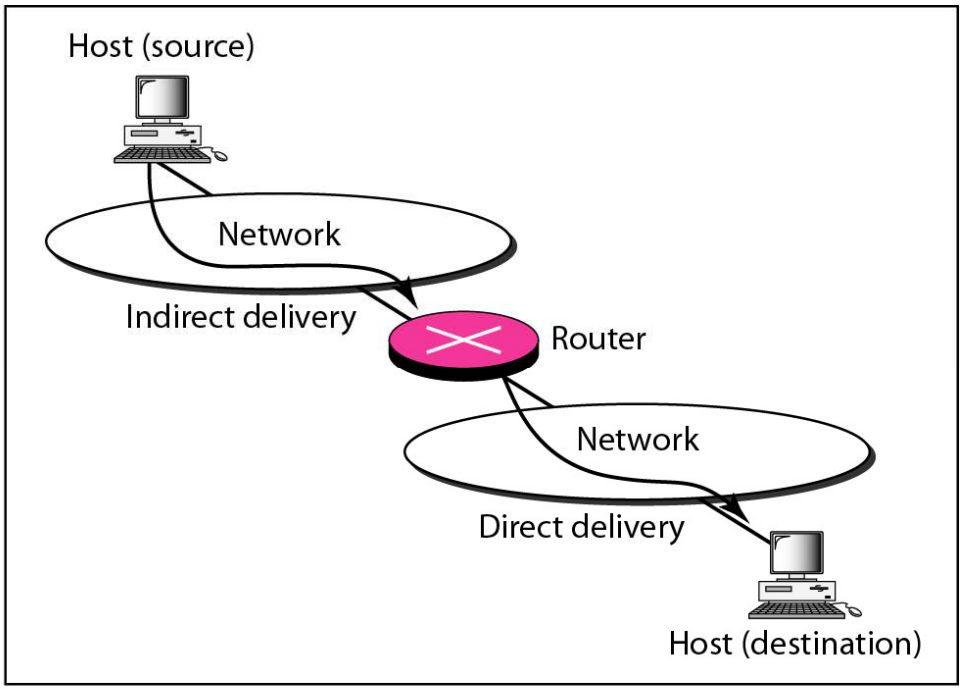
Introduction to Network Routing

Delivery

- The network layer supervises the handling of the packets by the underlying physical networks.
- We define this handling as the delivery of a packet.
- Direct and indirect delivery



a. Direct delivery



b. Indirect and direct delivery

Forwarding

- Forwarding means to place the packet in its route to its destination.
- Forwarding requires a host or a router to have a routing table.
- When a host has a packet to send or when a router has received a packet to be forwarded, it looks at this table to find the route to the final destination.

Routing Algorithms

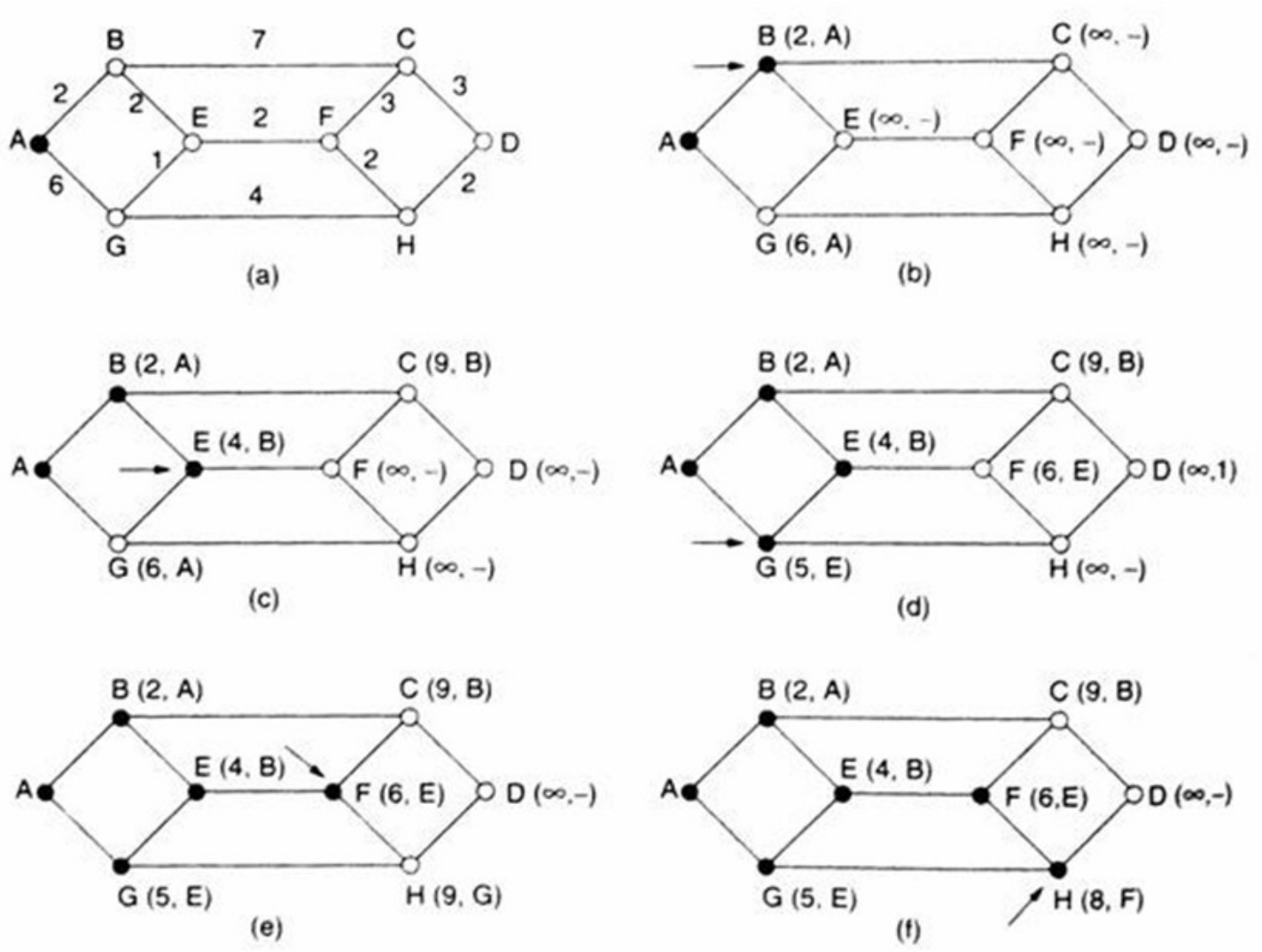
- Shortest Path Routing
  - To study the routing algorithms, a graph is commonly used to represent a subnet.
  - Each node of the graph represents a router and each arc of the graph represents a communication line.
  - The path length can be measured in terms of the number of hops, the geographic distance, etc.

Dijkstra Algorithm

- Set the value of the initial node to zero and set all other nodes to infinity.
- Mark all nodes unvisited. Set the initial node as starting point.
- For the current node, consider all of its unvisited neighbors and calculate their tentative distances.
- Mark the current node as visited and remove it from the unvisited set. A visited node will never be checked again.
- If the destination node has been marked visited or if the smallest tentative distance among the nodes in the unvisited set is infinity, then stop. The algorithm has finished.
- Set the unvisited node marked with the smallest tentative distance as the next **current node** and go back to step 3.

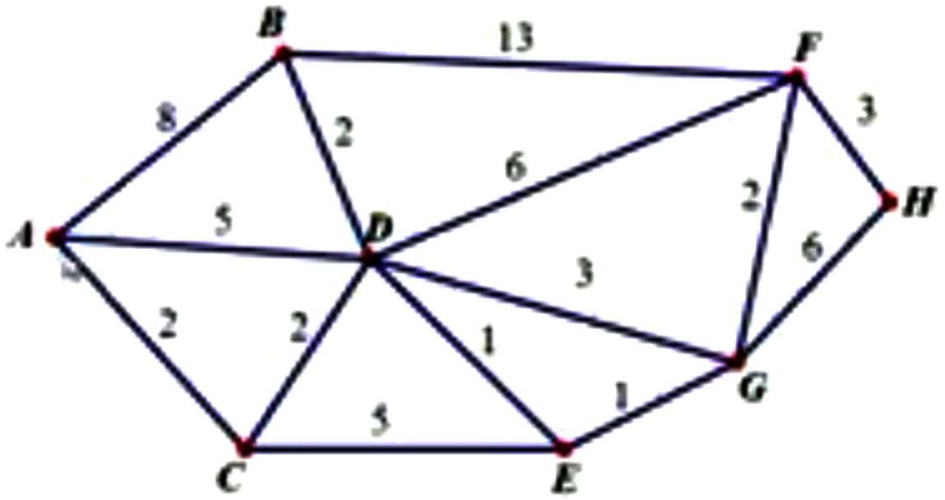
Dijkstra Algorithm

- Illustrated by example



Dijkstra Algorithm

- Illustrated by example



V	A	B	C	D	E	F	G	H
A	$\boxed{0_A}$	8_A	2_A	5_A	∞	∞	∞	∞
C		8_A	$\boxed{2_A}$	4_C	7_C	∞	∞	∞
D		6_D		$\boxed{4_C}$	5_D	10_D	7_D	∞
E		6_D			$\boxed{5_D}$	10_D	6_E	∞
B		$\boxed{6_D}$				10_D	6_E	∞
G						8_G	$\boxed{6_E}$	12_G
F						$\boxed{8_G}$		11_F
H								$\boxed{11_F}$

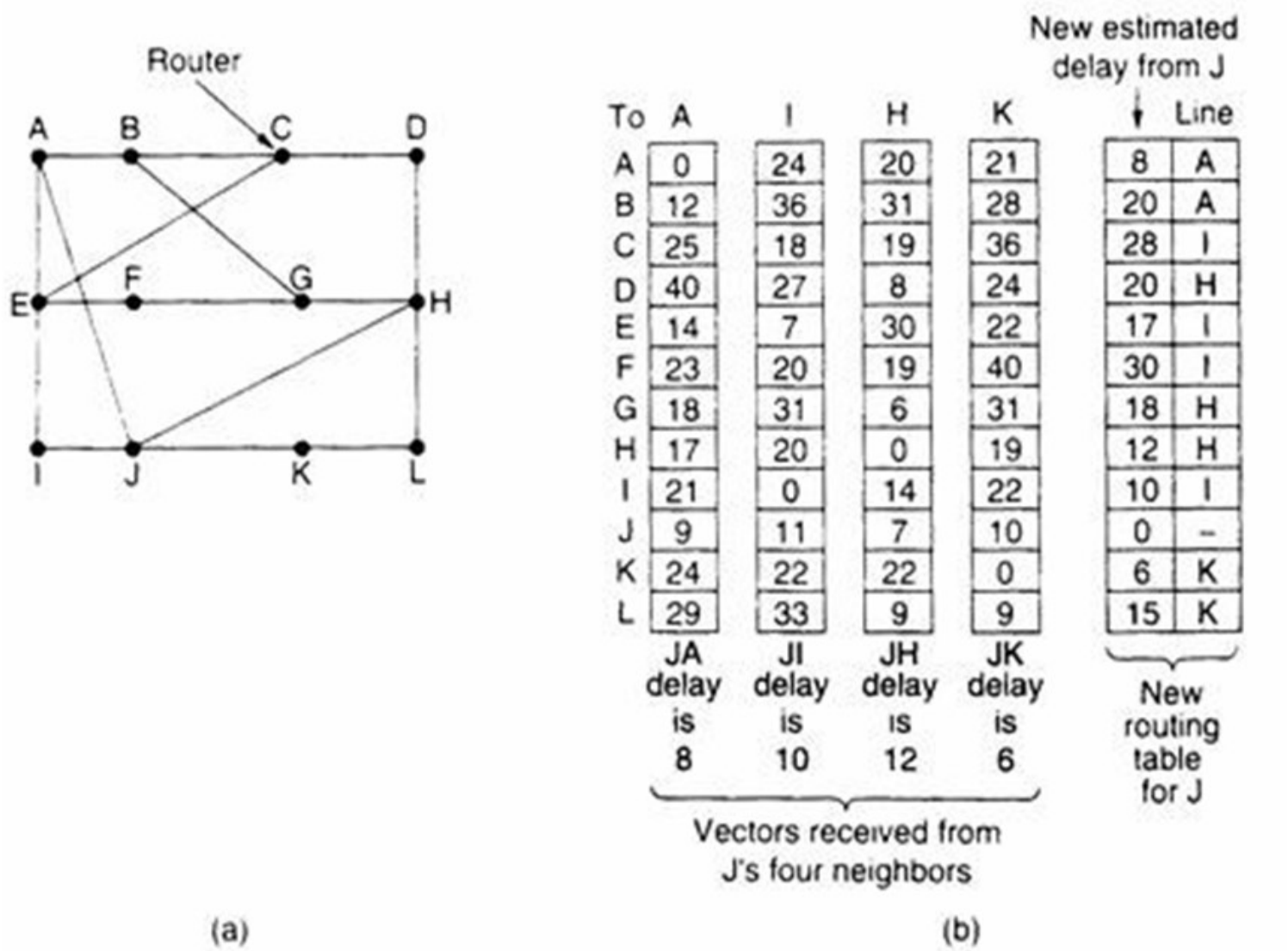
Flooding Algorithm

- Every incoming packet is sent out on every outgoing line except the one it arrived on.
- To solve the vast numbers of duplicate packets, a hop counter is introduced.
- The hop counter is contained in the header of each packet.
- It decremented at each hop. The packet will be discarded when the counter reached zero.

Distance Vector Routing

- It is a dynamic routing algorithm.
- Each router maintains a table giving the best known distance to each destination and which line to use to get there.
- These tables are updated by exchanging information with their neighbors.
- An entry of routing tables contains two parts: the preferred outgoing line to be used for the destination, and an estimate of the distance to that destination.

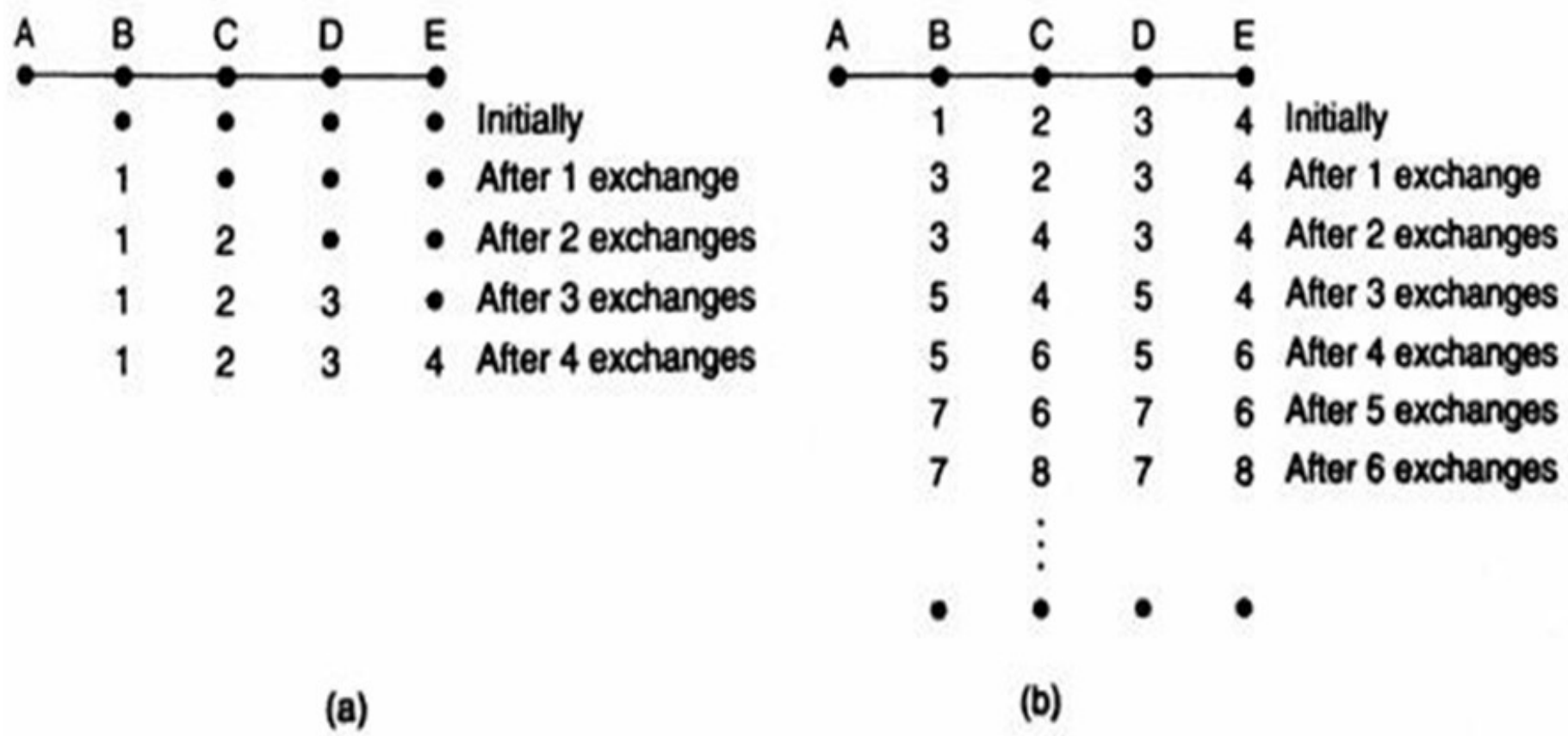
Distance Vector Routing



(a) A subnet. (b) Input from *A*, *I*, *H*, *K*, and the new routing table for *J*.

Distance Vector Routing

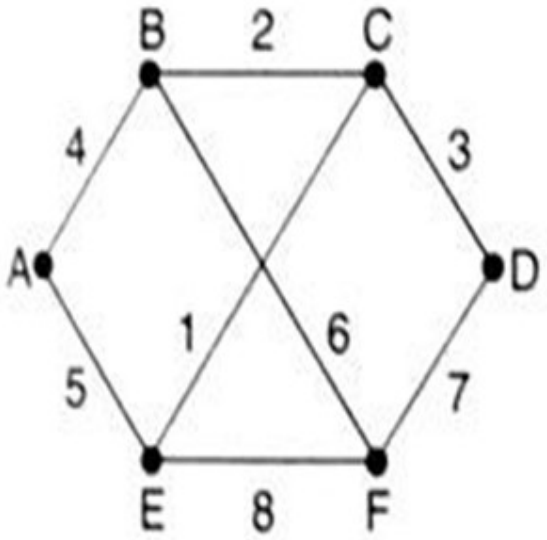
- Count-to-Infinity Problem: this algorithm reacts rapidly to good news, but leisurely to bad news.



Link State Routing

- Learning about the neighbors: When a router is booted, it sends a HELLO packet on each point-to-point line. The neighbors reply with their IDs.
- Measuring Line Cost: the router sends an ECHO packet to each neighbor to measure the round-trip time, and then to estimate the delay.
- Building Link State Packets: each router build a packet containing the sender ID, a sequence number, age (expire time), and a list of neighbors.
- Distributing the Link State Packets: each router flood the packet to its neighbors.
- Computing the new routes: each router uses the set of link state packets to construct the subnet graph.

Link State Routing



(a)

Link State Packets

A	B	C	D	E	F
Seq.	Seq.	Seq.	Seq.	Seq.	Seq.
Age	Age	Age	Age	Age	Age
B 4	A 4	B 2	C 3	A 5	B 6
E 5	C 2	D 3	F 7	C 1	D 7
	F 6	E 1		F 8	E 8

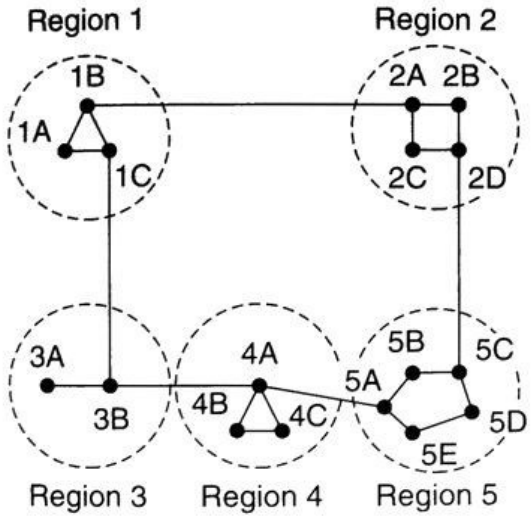
(b)

Broadcast and Multicast Routing

- Broadcast: sending a packet to all destinations simultaneously.
  - **Flooding:** Each router sends the packet to all the out-going line except the one the packet arrives.
- Multicast: sending a packet to a group of nodes. Two approaches are used:
  - Multicast is emulated using multiple point-to-point **unicast** connections.
  - Each router contains either a list of destinations or a bit map indicating the desired destinations. When a packet arrives at a router, the router checks all the destinations to determine the set of route lines that will be needed.

Hierarchical Routing

- Routers are divided into **regions**.
- Each router knows the internal structure within its own region, but only knows the inter-connection points of other regions.



(a)

Full table for 1A

Dest.	Line	Hops
1A	–	–
1B	1B	1
1C	1C	1
2A	1B	2
2B	1B	3
2C	1B	3
2D	1B	4
3A	1C	3
3B	1C	2
4A	1C	3
4B	1C	4
4C	1C	4
5A	1C	4
5B	1C	5
5C	1B	5
5D	1C	6
5E	1C	5

(b)

Hierarchical table for 1A

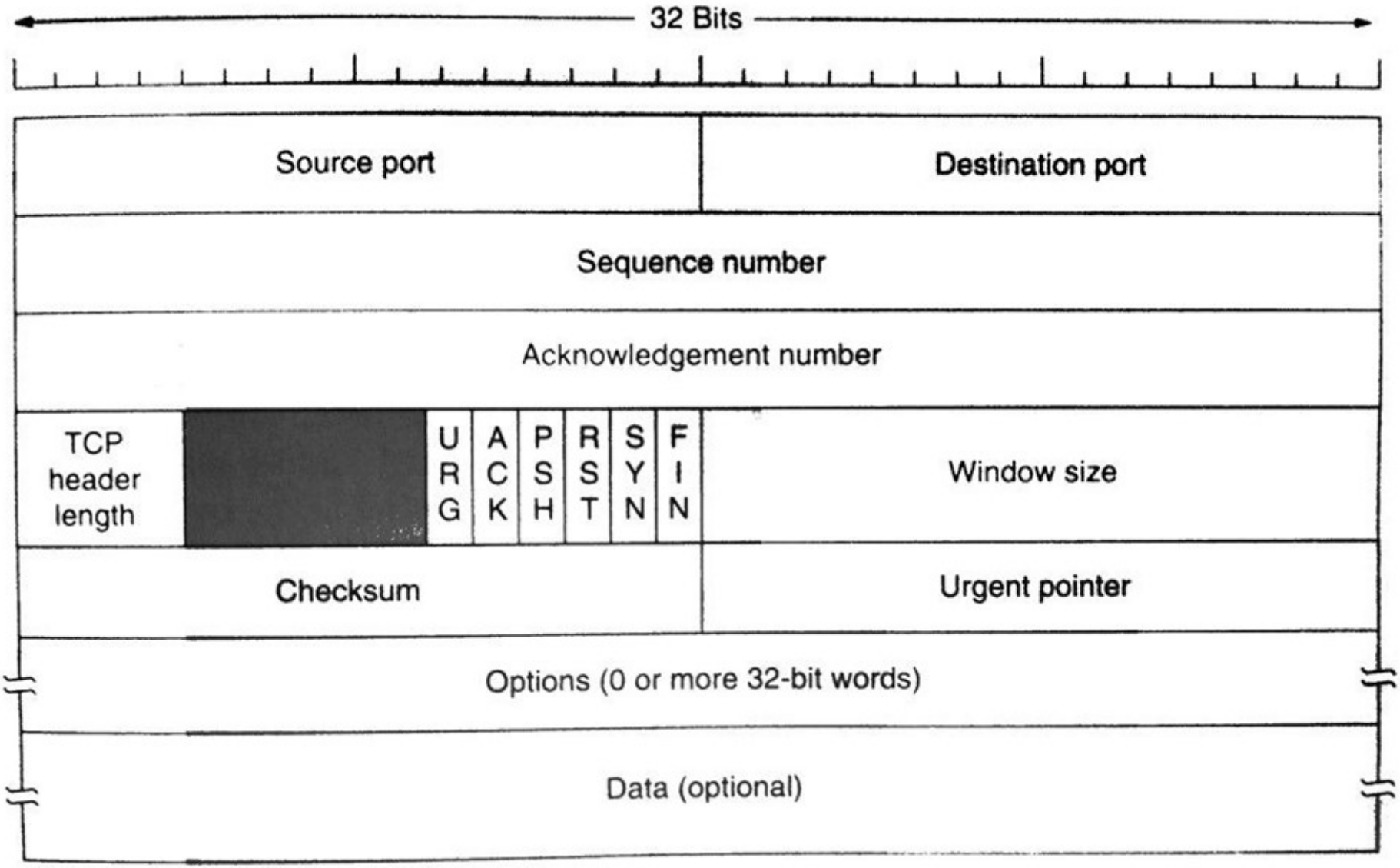
Dest.	Line	Hops
1A	–	–
1B	1B	1
1C	1C	1
2	1B	2
3	1C	2
4	1C	3
5	1C	4

(c)

Transport Control Protocol (TCP)

- Operation of TCP
  - When a sender transmits a segment, it also starts a timer.
  - When the segment arrives at the destination, the receiving TCP entity sends back a segment (with data if any) bearing an acknowledgement number equal to the next sequence number it expects to receive.
  - If the sender's timer goes off before the acknowledgement is received, the sender transmits the segment again.

TCP Segment Header

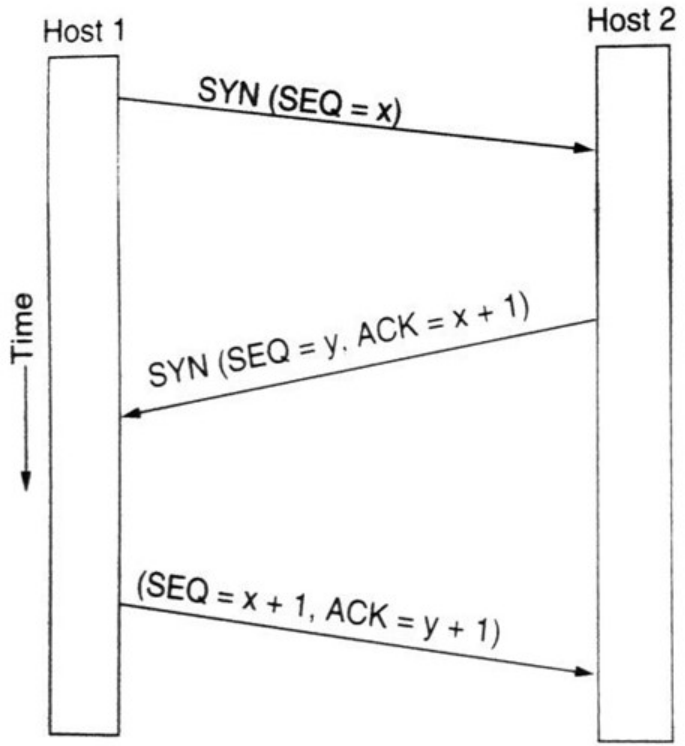


TCP Segment Header

- Source port and destination port: to identify the local end points of the connections.
- Sequence number and acknowledgement number
- TCP header length: to tell how many 32-bit words are contained in the header
- Six 1-bit flags
- Window size: to tell how many bytes may be sent starting at the byte acknowledged.
- Checksum
- Options: to provide a way to extend the header

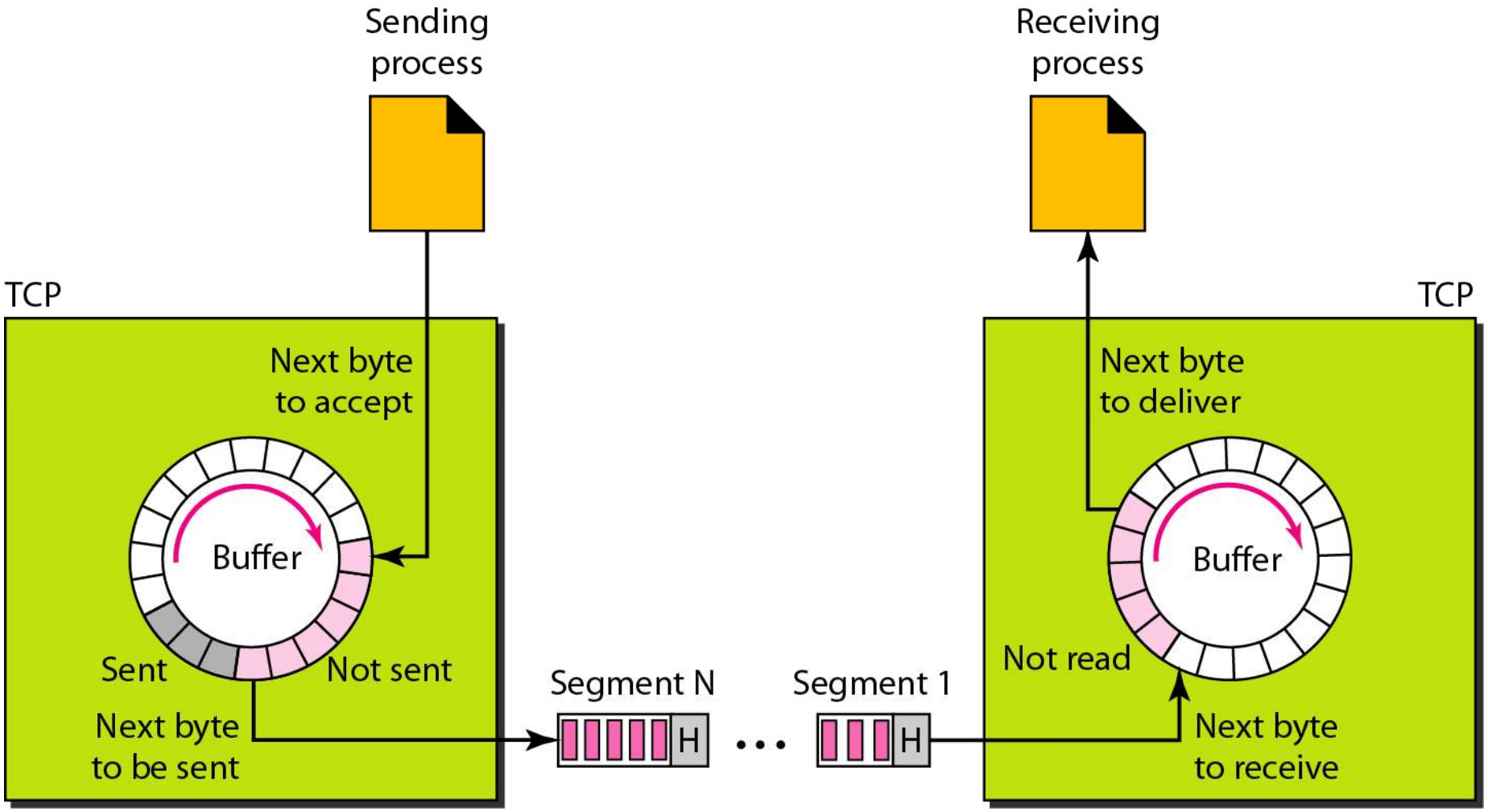
Connection Establishment and Release

- Three-way handshaking



- To release a connection, either party can send a TCP segment with the FIN bit set.

TCP Segments

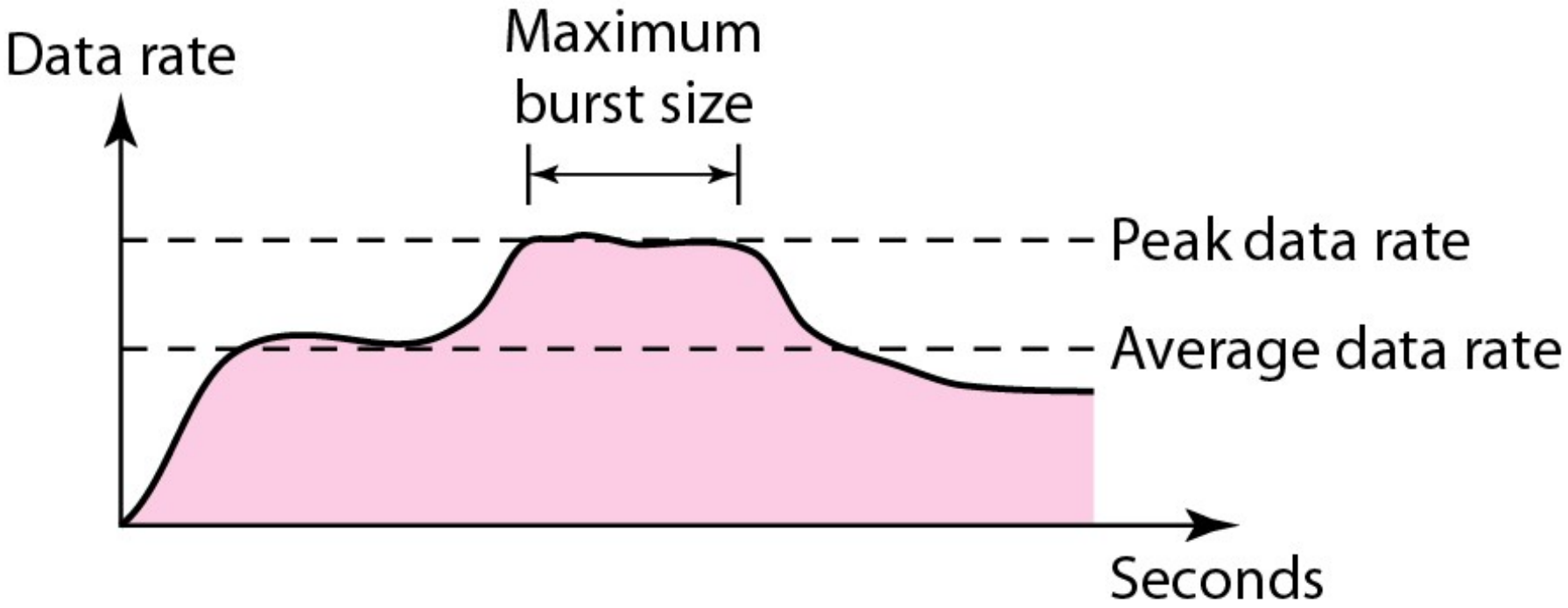


Data Traffic

- The main focus of congestion control and quality of service is data traffic.
- In congestion control we try to avoid traffic congestion.
- In quality of service, we try to create an appropriate environment for the traffic.
- So, before talking about congestion control and quality of service, we discuss the data traffic itself.

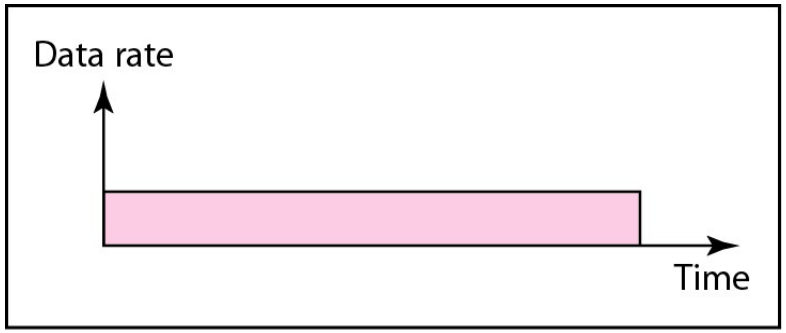
Data Traffic

- Traffic descriptors

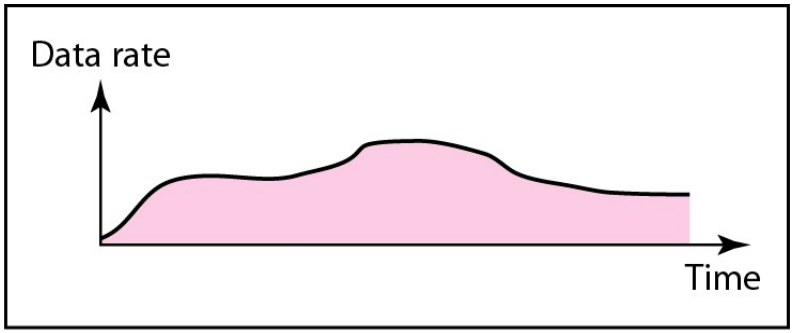


Data Traffic

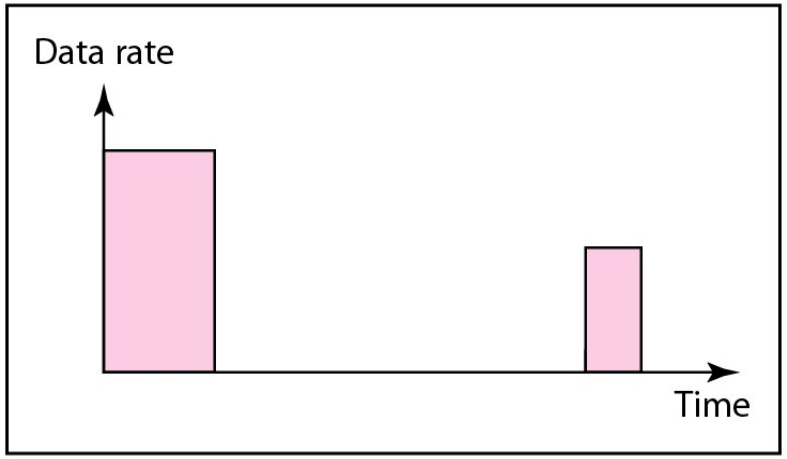
- Three traffic profiles



a. Constant bit rate



b. Variable bit rate

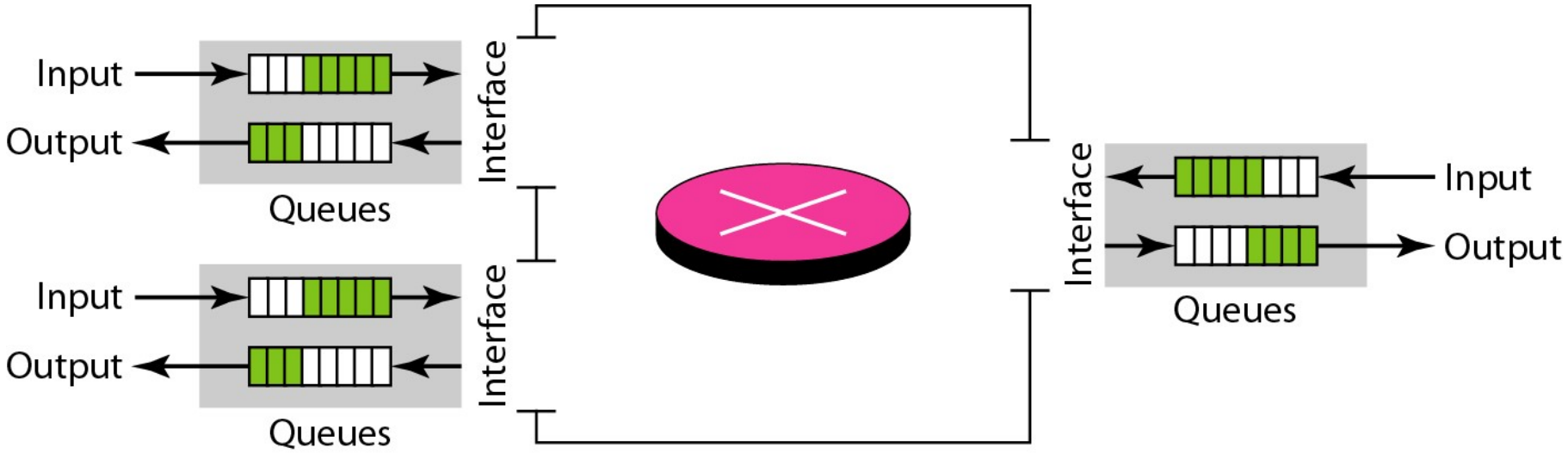


c. Bursty

Congestion Control

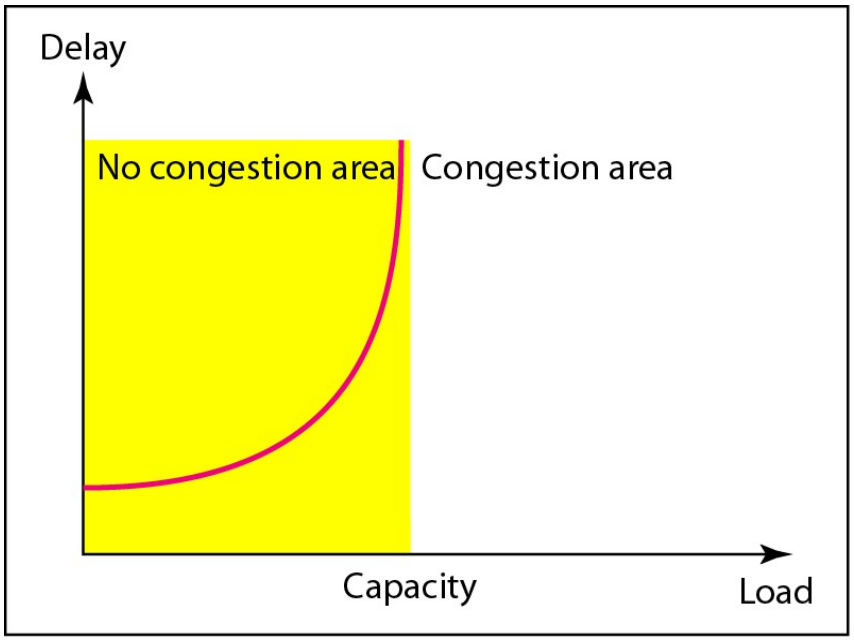
- Congestion in a network may occur if the load on the network is greater than the capacity of the network.
- Congestion control refers to the mechanisms and techniques to control the congestion and keep the load below the capacity.

- Queues in a router

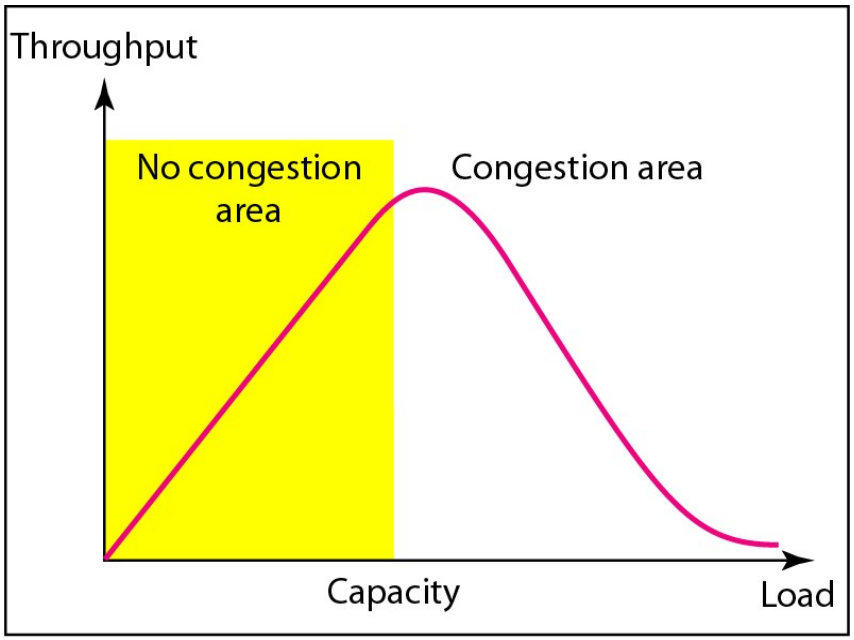


Congestion Control

- Packet delay and throughput as functions of load



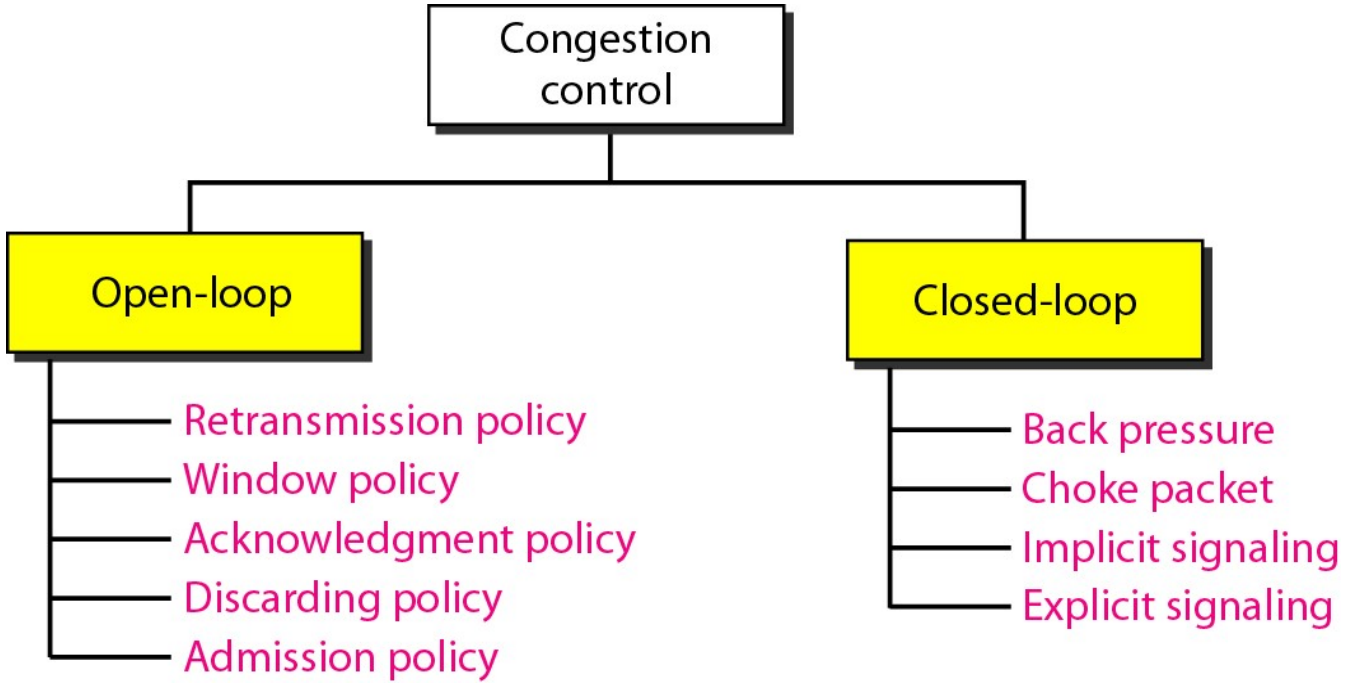
a. Delay as a function of load



b. Throughput as a function of load

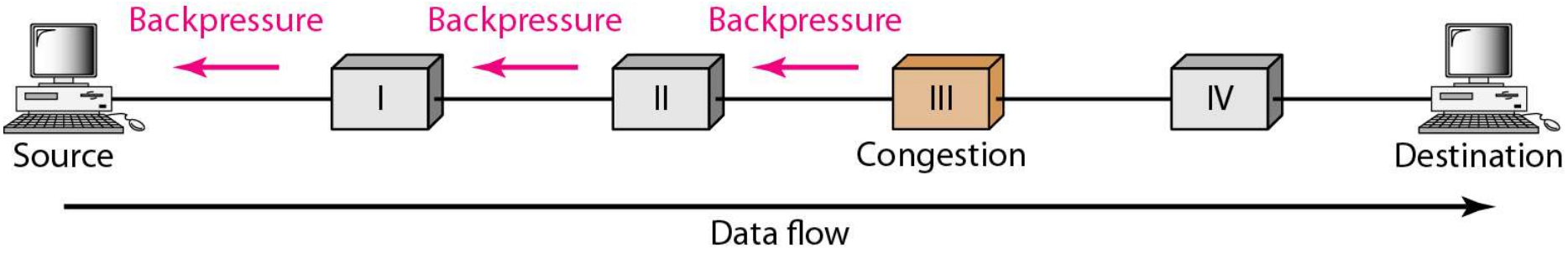
Congestion Control

- Congestion control refers to techniques and mechanisms that can either prevent congestion, before it happens, or remove congestion, after it has happened.
- In general, we can divide congestion control mechanisms into two broad categories: open-loop congestion control (prevention) and closed-loop congestion control (removal).

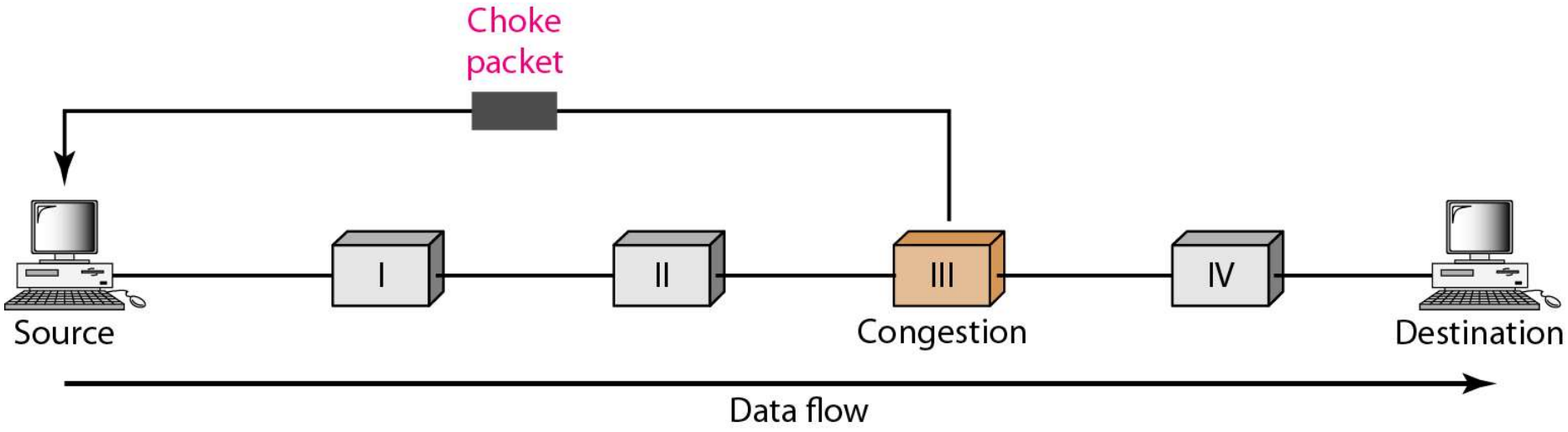


Congestion Control

- Backpressure method for alleviating congestion



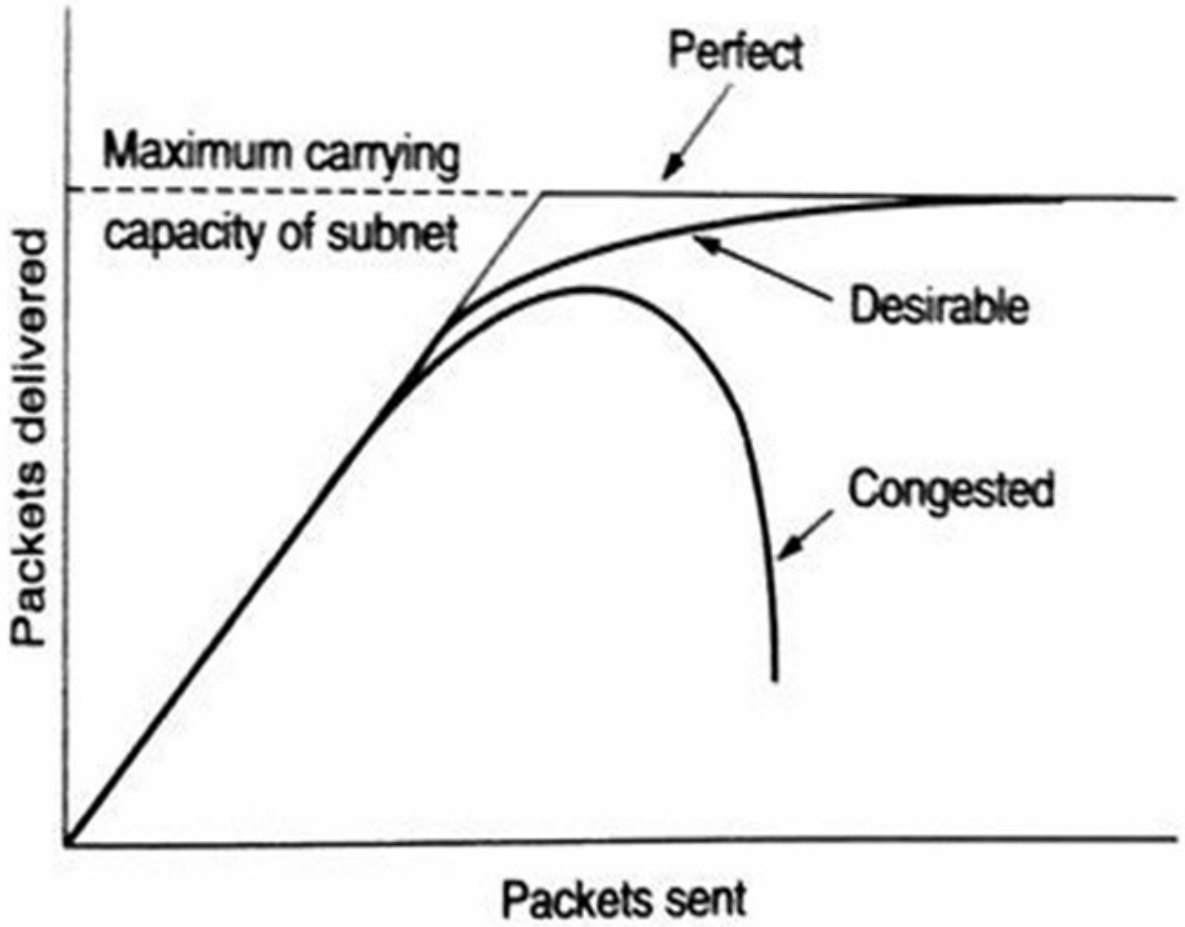
- Choke packet



Congestion Control in TCP

- When traffic increases too far due to many reasons, the routers are not longer able to cope and they begin losing packets. So that the performance degrades.
- To better understand the concept of congestion control, let us give one example in TCP.

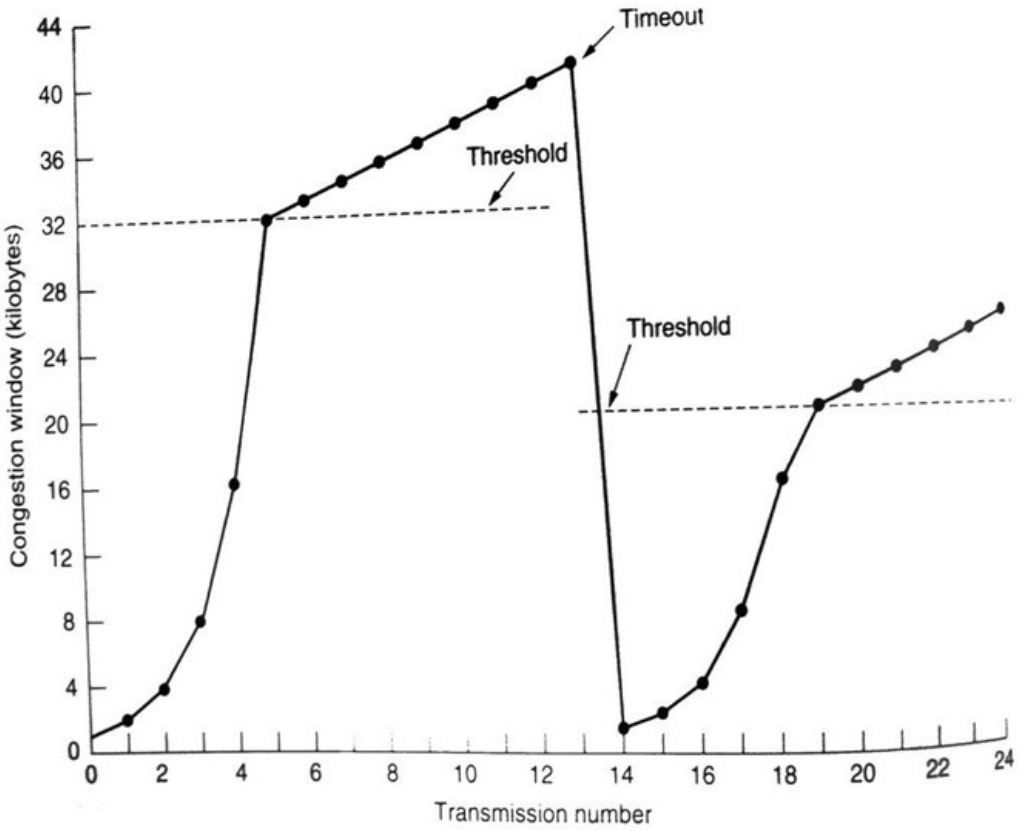
Congestion Control in TCP



When too much traffic is offered, congestion sets in and performance degrades sharply.

Congestion Control: **Exponential and then Linear**

- A threshold is used in the congestion control algorithm, initially 64 KB.
- At the beginning, the window size grows exponentially until the threshold is hit.
- Then successful transmissions grow the window linearly.
- If a timeout occurs, the threshold is set half of the current window, and the starting window is reset to one.



Metrics Used to Monitor a Subnet for Congestion

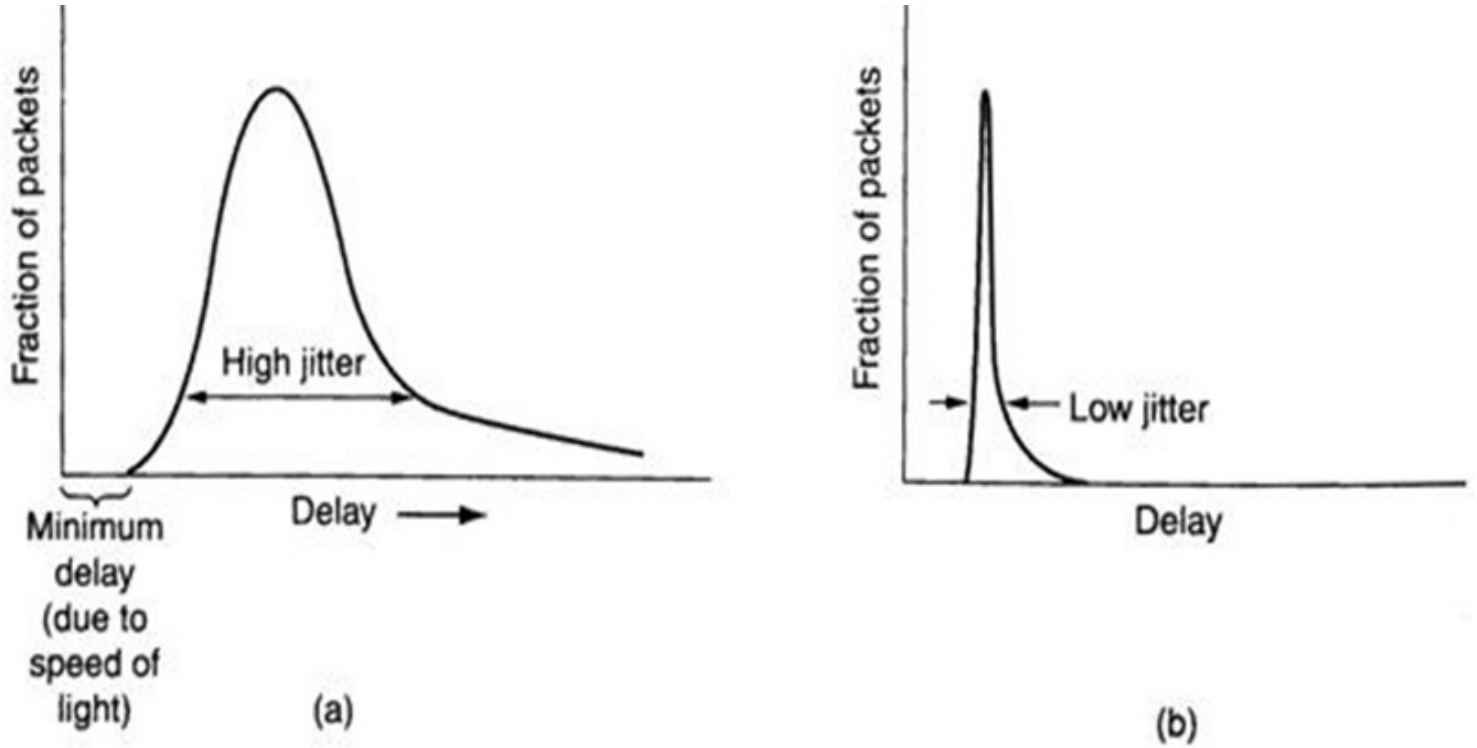
- Percentage of all packets discarded for lack of buffer space;
- Average of queue lengths;
- Number of packets that time out and are retransmitted;
- Average packet delay;
- Standard deviation of packet delay, etc.

Load Shedding

- When routers are being overloaded by packets, they just throw the packets away.
- Many approaches can be used to discard packets:
  - Random manner;
  - Application-based or algorithm-based;
- Random Early Detection (RED) Algorithm
  - To drop packets before the congestion situation has become worse.
  - For a particular line, when the average queue length exceeds a threshold, the line is said to be congested and action is taken.

Jitter Control

- The variation in the packets arrival times is called jitter.
- A small jitter is desirable for some multimedia applications such as video, audio, etc.
- Jitter can be eliminated by buffering. But it only applicable for the non real-time applications such as video-on-demand.



(a) High jitter. (b) Low jitter.

Quality of Service

- Different applications have different requirements on quality of service

Application	**Reliability**	**Delay**	**Jitter**	**Bandwidth**
E-mail	High	Low	Low	Low
File Transfer	High	Low	Low	Medium
Web access	High	Medium	Low	Medium
Audio on demand	Low	Low	High	Medium
Video on demand	Low	Low	High	High
Telephony	Low	High	High	Low
Videoconferencing	Low	High	High	High

Quality of Service

- Techniques for achieving good quality of service
  - Buffering;
  - Traffic shaping;
  - Resource reservation;
  - Admission control.

Timer Management

- For each connection, TCP maintains the timeout interval as:

  $$\text{Timeout} = \mu + 4\delta$$

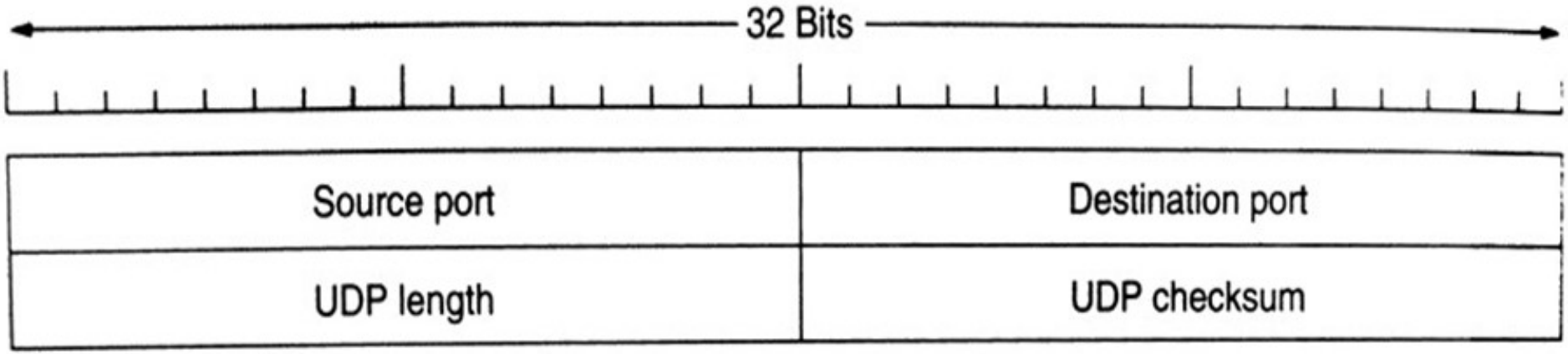
  Where

  μ is the mean *round-trip time* of the observations,

  δ is the standard deviation of the observations.
- TCP constantly adjusts the timeout interval based on continuous measurements of network performance.

User Datagram Protocol (UDP)

- Connectionless protocol
- Provide a way for applications to send encapsulated IP datagrams and send them without having to establish a connection.
- 8-byte header
- UDP Datagram Header



Packet Encapsulation