

# STUDYNAMA.COM

India's Mega Online Education Hub for Class 9-12 Students,  
Engineers, Managers, Lawyers and Doctors.

## Free Resources for Class 9-12 Students

- [Lecture Notes](#)
- [Project Reports](#)
- [Solved Papers](#)

[View More »](#)

## Free Resources for Engineering Students

- [Lecture Notes](#)
- [Project Reports](#)
- [Solved Papers](#)

[View More »](#)

## Free Resources for MBA/BBA Students

- [Lecture Notes](#)
- [Project Reports](#)
- [Solved Papers](#)

[View More »](#)

## Free Resources for LLB/LLM Students

- [Lecture Notes](#)
- [Project Reports](#)
- [Solved Papers](#)

[View More »](#)

## Free Resources for MBBS/BDS Students

- [Lecture Notes](#)
- [Project Reports](#)
- [Solved Papers](#)

[View More »](#)



▼▼ Scroll Down to View your Downloaded File! ▼▼

## **Disclaimer**

Please note none of the content or study material in this document or content in this file is prepared or owned by Studynama.com. This content is shared by our student partners and we do not hold any copyright on this content.

Please let us know if the content in this file infringes any of your copyright by writing to us at: [info@studynama.com](mailto:info@studynama.com) and we will take appropriate action.

# GATE SOLVED PAPER - CSE

## COMPUTER NETWORK

**YEAR 2003**

**ONE MARK**

Q. 1

- Which of the following assertions is false about the internet Protocol (IP) ?
- (A) It is possible for a computer to have multiple IP addresses
  - (B) IP packets from the same source to the same destination can take different routes in the network
  - (C) IP ensures that a packet is forwarded if it is unable to reach its destination within a given number of hops
  - (D) The packet source cannot set the route of an outgoing packets; the route is determined only by the routing tables in the routers on the way.

Q. 2

- Which of the following functionalities must be implemented by a transport protocol over and above the network protocol ?
- (A) Recovery from packet losses
  - (B) Detection of duplicate packets
  - (C) Packet delivery in the correct order
  - (D) End to end connectivity

**YEAR 2003**

**TWO MARKS**

Q. 3

- The subnet mask for a particular network is 255.255.31.0 Which of the following pairs of IP addresses could belong to this network ?
- (A) 172.57.88.62 and 172.56.87.23.2
  - (B) 10.35.28.2 and 10.35.29.4
  - (C) 191.203.31.87 and 191.234.31.88
  - (D) 128.8.129.43 and 128.8.161.55

Q. 4

- A 2 km long broadcast LAN has  $10^7$  bps bandwidth and uses CSMA/ CD. The signal travels along the wire at  $2 \times 10^8$ m/s. What is the minimum packet size that can be used on this network ?
- (A) 50 bytes
  - (B) 100 bytes
  - (C) 200 bytes
  - (D) None of the above

Q. 5

- Host A is sending data to host B over a full duplex link. A and B are using the sliding window protocol for flow control. The send and receive window sizes are 5 packets each. Data packets (sent only from A to B) are all 1000 bytes long and the transmission time for such a packet is 50  $\mu s$ . Acknowledgment packets (sent only from B to A), are very small and require negligible transmission time. The propagation delay over the link is 200  $\mu s$ . What is the maximum achievable throughput in this communication ?
- (A)  $7.69 \times 10^6$ bps
  - (B)  $11.11 \times 10^6$ bps
  - (C)  $12.33 \times 10^6$ bps
  - (D)  $15.00 \times 10^6$ bps

Q. 6

Choose the best matching Group 1 and Group 2.

	Group-1		Group-2
P.	P. Data link layer	1.	Ensures reliable transport of data over a physical point-to-point link
Q.	Network layer	2.	Encodes/ decodes data for physical transmission
R.	Transport layer	3.	Allowed-to-end communication between two processes



Q. 7

Which of the following is NOT true with respect to a transparent bridge and a router?

- (A) Both bridge and router selectively forward data packets
  - (B) A bridge uses IP addresses while a router uses MAC addresses
  - (C) A bridge builds up its routing table by inspecting incoming packets
  - (D) A router can connect between a LAN and a WAN.

Q. 8

How many 8-bit characters can be transmitted per second over a 9600 baud serial communication link using asynchronous mode of transmission with one start bit, eight data bits, and one parity bit ?



Q. 9

A and B are the only two stations on an Ethernet. Each has a steady queue of frames to send. Both A and B attempt to transmit a frame, collide, and A wins the first backoff race. At the end of this successful transmission by A, both A and B attempt to transmit and collide. The probability that A wins the second backoff race is



9-10

The routing table of a router is shown below:

Destination	Subnet Mask	Interface
128.75.43.0	255.255.255.0	Eth 0
128.75.43.0	255.255.255.128	Eth 1
192.12.17.5	255.255.255.255	Eth 3
deraulf		Eth 2

On which interface will the router forward packets addressed to destinations 128.75.43.16 and 192.12.17.10 respectively ?

**Common Data For Q. 11 & 12**

Solve the problems and choose the correct answers.

Consider three IP networks A, B and C. Host  $H_A$  in network A send messages each containing 180 bytes of application data to a host  $H_C$  in network C. The TCP layer prefixes a 20 byte header to the message. This passes through an intermediate network B. The maximum packet size, including 20 byte IP header, in each network is

- A : 1000 bytes
  - B : 100 bytes
  - C : 1000 bytes

The network A and B are connected through a 1 Mbps link, while B and C are connected by a 512 Kbps link ( $\text{bps} = \text{bits per second}$ ).






YEAR 2005

ONE MARK

- Q. 13 Packets of the same session may be routed through different paths in  
(A) TCP, but not UDP  
(B) TCP and UDP  
(C) UDP but not TCP  
(D) Neither TCP, nor UDP

- The address resolution protocol (ARP) is used for

  - (A) Finding the IP address from the DNS
  - (B) Finding the IP address of the default gateway
  - (C) Finding the IP address that corresponds to a MAC address
  - (D) Finding the MAC address that corresponds to an IP address

Q. 16

In a network of LANs connected by bridges, packets are sent from one LAN to another through intermediate bridges. Since more than one path may exist between two LANs, packets may have to be routed through multiple bridges. Why is the spanning tree algorithm used for bridge-routing ?

- (A) For shortest path routing between LANs
- (B) For avoiding loops in the routing paths
- (C) For fault tolerance
- (D) For minimizing collisions

Q. 17

An organization has a class B network and wishes to form subnets for 64 departments. The subnet mask would be

- (A) 255.255.0.0
- (B) 255.255.64.0
- (C) 255.255.128.0
- (D) 255.255.255.0

**YEAR 2005**

**TWO MARKS**

Q. 18

In a packet switching network, packets are routed from source to destination along a single path having two intermediate nodes. If the message size is 24 bytes and each packet contains a header of 3 bytes, then the optimum packet size is

- (A) 4
- (B) 6
- (C) 7
- (D) 9

Q. 19

Suppose the round trip propagation delay for a 10 Mbps Ethernet having 48-bit jamming signal is  $46.4 \mu s$ . The minimum frame size is :

- (A) 94
- (B) 416
- (C) 464
- (D) 512

**YEAR 2006**

**ONE MARK**

Q. 20

For which one of the following reason: does Internet Protocol (IP) use the time-to-live (TTL) field in the IP datagram header?

- (A) Ensure packets reach destination within that time
- (B) Discard packets that reach later than that time
- (C) Prevent packets from looping indefinitely
- (D) Limit the time for which a packet gets queued in intermediate routers

**YEAR 2006**

**TWO MARKS**

Q. 21

Station A uses 32 byte packets to transmit messages to Station B using a sliding window protocol. The round trip delay between A and B is 80 milliseconds and the bottleneck bandwidth on the path between A and B is 128 kbps. What is the optimal window size that A should use ?

- (A) 20
- (B) 40
- (C) 160
- (D) 320

Q. 22

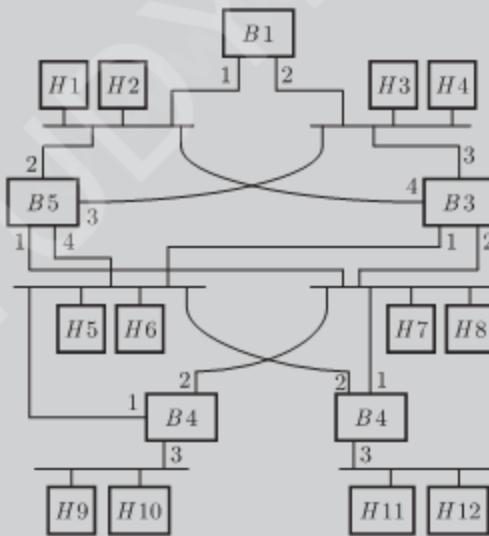
- Two computers C1 and C2 are configured as follows. C1 has IP address 203.197.2.53 and netmask 255.255.128.0. C2 has IP address 203.197.75.201 and netmask 255.255.192.0. Which one of the following statements is true?
- (A) C1 and C2 both assume they are on the same network
  - (B) C2 assumes C1 is on same network, but C1 assumes C2 is on a diff. network
  - (C) C1 assumes C2 is on same network, but C2 assumes C1 is on a diff. network
  - (D) C1 and C2 both assume they are on different networks

Q. 23

- Station A needs to send a message consisting of 9 packets to Station B using a sliding window (window size 3) and go-back-n error control strategy. All packets are ready and immediately available for transmission. If every 5th packet that A transmits gets lost (but no acks from B ever get lost), then what is the number of packets that A will transmit for sending the message to B ?
- (A) 12
  - (B) 14
  - (C) 16
  - (D) 18

Statement For Linked Answer Q. 24 and 25 :

Consider the diagram shown below where a number of LANs are connected by (transparent) bridges. In order to avoid packets looping through circuits in the graph, the bridges organize themselves in a spanning tree. First, the root bridge is identified as the bridge with the least serial number. Next, the root sends out (one or more) data units to enable the setting up of the spanning tree of shortest paths from the root bridge to each bridge. Each bridge identifies a port (the root port) through which it will forward frames to the root bridge. Port conflicts are always resolved in favour of the port with the lower index value. When there is a possibility of multiple bridges forwarding to the same LAN (but not through the root port), ties are broken as follows: bridges closest to the root get preference and between such bridges, the one with the lowest serial number is preferred.



Q. 24

For the given connection of LANs by bridges, which one of the following choices represents the depth first traversal of the a panning tree of bridges?

- (A) B1,B5,B3,B4,B2
- (B) B1,B3,B5,B2,B4
- (C) B1,B5,B2,B3,B4
- (D) B1,B3,B4,B5,B2

Q. 25

Consider the correct spanning tree for the previous question. Let host H1 send out a broadcast ping packet. Which of the following options represents the correct forwarding table on B3?

(A)

Hosts	Ports
H1,H2,H3,H4	3
H5,H6,H9,H10	1
H7,H8,H11,H12	2

(B)

Hosts	Port
H1, H2	4
H3, H4	3
H5, H6	1
H7, H8, H9, H10	2
H11, H12	

(C)

Hosts	Port
H1, H2, H3, H4	3
H5, H6, H9, H10	1
H7,H8, H11, H12	2

(D)

Hosts	Port
H1, H2, H3, H4	3
H5, H7, H9, H10	1
H7, H8, H11, H12	4

YEAR 2007

### **ONE MARK**

Q. 26

In Ethernet when Manchester encoding is used, the bit rate is



Q. 27

Which one of the following uses UDP as the transport protocol?

- (A) HTTP
  - (B) Telnet
  - (C) DNS
  - (D) SMTP

YEAR 2007

TWO MARKS

There are  $n$  stations in a slotted LAN. Each station attempts to transmit with a probability  $p$  in each time slot. What is the probability that ONLY one station transmits in a given time slot?

- (A)  $np(1-p)^{n-1}$       (B)  $(1-p)^{n-1}$   
 (C)  $p(1-p)^{n-1}$       (D)  $1 - (1-p)^{n-1}$

9-29

In a token ring network the transmission speed is 10 bps and the propagation speed is 200 metres/ $\mu$ s. The 1-bit delay in this network is equivalent to:



8 30

The address of a class B host is to be split into subnets with a 6-bit subnet number. What is the maximum number of subnets and the maximum number of hosts in each subnet?

- (A) 62 subnets and 262142 hosts
  - (B) 64 subnets and 262142 hosts
  - (C) 62 subnets and 1022 hosts
  - (D) 64 subnets and 1024 hosts

YEAR 2008

ONE MARK



YEAR 2008

TWO MARKS

Q. 38

A client process  $P$  needs to make a TCP connection to a server Process  $S$ . Consider the following situation; the server process  $S$  executes a socket (), a bind() and a listen () system call in that order, following which it is preempted. Subsequently, the client Process  $P$  executes a socket () system call followed by connect () system call to connect to the server process  $S$ . The server process has not executed any accept() system call. Which one of the following events could take place?

- (A) connect () system call returns successfully
- (B) connect () system call blocks
- (C) connect () system call returns an error
- (D) connect () system call results in a core dump

**YEAR 2009**

**TWO MARKS**

Q. 39

In the RSA public key cryptosystem, the private and the public keys are  $(e, n)$  and  $(d, n)$  respectively, where  $n = p^*$  and  $p$  and  $q$  are large primes. Besides,  $n$  is public and  $p$  and  $q$  are private. Let  $M$  be an integer such that  $0 < M < n$  and  $\phi(n) = (p - 1)(q - 1)$ . Now consider the following equations.

- I.  $M' = M^e \text{ mod } n; M = (M')^d \text{ mod } n$
- II.  $ed \equiv 1 \text{ mod } n$
- III.  $ed \equiv 1 \text{ mod } \phi(n)$
- IV.  $M' = M^e \text{ mod } \phi(n); M = (M')^d \text{ mod } \phi(n)$

Which of the above equations correctly represent RSA cryptosystem ?

- (A) I and II
- (B) I and III
- (C) II and IV
- (D) III and IV

Q. 40

While opening a TCP connection, the initial sequence number is to be derived using a time-of-day (ToD) clock that keeps running even when the host is down. The low order 32 bits of the counter of TOD clock is to be used for the initial sequence numbers. The clock counter increments once per millisecond. The maximum packet lifetime is given to be 64s.

Which one of the choices given below is closest to the minimum permissible rate at which sequence numbers used for packets of a connection can increase ?

- (A) 0.015/s
- (B) 0.064/s
- (C) 0.135/s
- (D) 0.327/s

Q. 41

Let  $G(x)$  be the generator polynomial used for CRC checking. What is the condition that should be satisfied by  $G(x)$  to detect odd number of bits in error ?

- (A)  $G(x)$  contains more than two terms
- (B)  $G(x)$  does not divide  $1 + x^k$ , for any  $K$  not exceeding the frame length
- (C)  $1 + x$  is a factor of  $G(x)$
- (D)  $G(x)$  has an odd number of terms

Statement For Linked Answer Q. 42 & 43 :

Frames of 1000 bits are sent over a  $10^6$  bps duplex link between two hosts. The propagation time is 25 ms. Frames are to be transmitted into to maximally pack them in transit (within the link).

Q. 42

What is the minimum number of bits ( $l$ ) that will be required to represent the sequence numbers distinctly? Assume that no time gap needs to be given between transmission of two frames.



Q. 43

Suppose that the sliding window protocol is used with the sender window size of  $2^l$ , where  $l$  is the number of bits identified in the earlier part and acknowledgements are always piggy backed. After sending  $2^l$  frames, what is the minimum time the sender will have to wait before starting transmission of the next frame ? (Identify the closest choice ignoring the frame processing time)



YEAR 2010

### **ONE MARK**

Q. 44

One of the header fields in an IP datagram is the Time-to-Live (TTL) field. Which of the following statements best explains the need for this field ?

- (A) It can be used to prioritize packets
  - (B) It can be used to reduce delays
  - (C) It can be used to optimize throughput
  - (D) It can be used to prevent packet looping

Q. 45

Which one of the following is not a client-server application ?



YEAR 2010

TWO MARKS

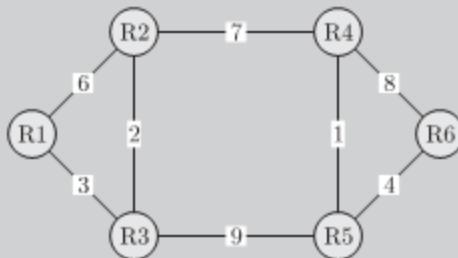
Q. 46

Suppose computers A and B have IP addresses 10.105.1.113 and 10.105.1.91 respectively and they both use the same netmask N. Which of the values of N given below should not be used if A and B should belong to the same network ?



**Statement For Linked Answer Q. 47 & 48 :**

Consider a network with 6 routers R1 and R6 connected with links having weights as shown in the following diagram.



Q. 47

All the routers use the distance vector based routing algorithm to update their routing tables. Each starts with its routing table initialized to contain an entry for each neighbour with the weight of the respective connecting link. After all the routing tables stabilize, how many links in the network will never be used for carrying any data ?

- (A) 4 (B) 3  
(C) 2 (D) 1

Q. 48

Suppose the weights of all unused links in the previous question are changed to 2 and the distance vector algorithm is used again until all routing tables stabilize. How many links will now remain unused ?

- (A) 0 (B) 1  
(C) 2 (D) 3

**YEAR 2011**

**ONE MARK**

Q. 49

A layer-4 firewall (a device that can look at all protocol headers up to the transport layer) CANNOT

- (A) block entire HTTP traffic during 9:00 pm and 5:00 am  
(B) block all ICMP traffic  
(C) stop incoming traffic from a specific IP address but allow outgoing traffic to the same IP address  
(D) block TCP traffic from a specific user on a multi-user system during 9:00 pm and 5:00 am

Q. 50

Consider different activities related to email

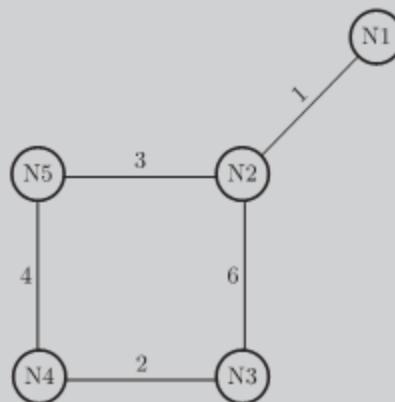
- m1* : Send an email from a mail client to a mail server  
*m2* : Download an email from mailbox server to a mail client  
*m3* : Checking email in a web browser  
(A) *m1*:HTTP *m2*:SMTP *m3*:POP (B) *m1*:SMTP *m2*:FTP *m3*:HTTP  
(C) *m1*:SMTP *m2*:FTP *m3*:HTTP (D) *m1*:POP *m2*:SMTP *m3*:IMAP

**YEAR 2011**

**TWO MARKS**

Statement For Linked Answer Q. 51 and 52 :

Consider a network with five nodes,  $N_1$  to  $N_5$ , as shown below.



The network uses a Distance Vector Routing protocol. Once the routes have stabilized, the distance vectors at different nodes are as following.

$$\begin{array}{lll} N1 : (0, 1, 7, 8, 4) & N2 : (1, 0, 6, 7, 3) & N3 : (7, 6, 0, 2, 6) \\ N4 : (8, 7, 2, 0, 4) & N5 : (4, 3, 6, 4, 0) \end{array}$$

Each distance vector is the distance of the best known path at that instance to nodes,  $N1$  to  $N5$ , where the distance to itself is 0. Also, all links are symmetric and the cost is identical in both directions. In each round, all nodes exchange their distance vectors with their respective neighbors. Then all nodes update their distance vectors. In between two rounds, any change in cost of a link will cause the two incident nodes to change only that entry in their distance vectors.

Q. 51 The cost of link  $N2 - N3$  reduces to 2(in both directions). After the next round of updates, what will be the new distance vector at node,  $N3$ ?

- (A) (3, 2, 0, 2, 5) (B) (3, 2, 0, 2, 6)  
(C) (7, 2, 0, 2, 5) (D) (7, 2, 0, 2, 6)

Q. 52 After the update in the previous question, the link  $N1 - N2$  goes down.  $N2$  will reflect this change immediately in its distance vector as cost  $\infty$ . After the NEXT ROUND of update, what will be the cost to  $N1$  in the distance vector of  $N3$ ?

- (A) 3 (B) 9  
(C) 10 (D)  $\infty$

**YEAR 2012**

**ONE MARK**

Q. 53 The Protocol Data Unit (PDU) for the application layer in the Internet stack is  
(A) Segment (B) Datagram  
(C) Message (D) Frame

Q. 54 Which of the following transport layer protocols is used to support electronic mail?  
(A) SMTP (B) IP  
(C) TCP (D) UDP

Q. 55 In the IPv4 addressing format, the number of networks allowed under Class C addresses is  
(A)  $2^{14}$  (B)  $2^7$   
(C)  $2^{21}$  (D)  $2^{24}$

**YEAR 2012**

**TWO MARKS**

Q. 56 An Internet Service Provider (ISP) has the following chunk of CIDR-based IP addresses available with it : 254.248.128.0/20. The ISP wants to give half of this chunk of addresses to Organization A, and a quarter of Organization B, while retaining the remaining with itself. Which of the following is a valid allocation of addresses to A and B?  
(A) 245.248.136./21 and 245.248.128.0/22 (B) 245.248.128.0/21 and 245.248.128.0/22  
(C) 245.248.132.0/22 245.248.132.0/21 (D) 245.248.136.0/24 245.248.132.0/21

Q. 57

Consider a source computer ( $S$ ) transmitting a file of size  $10^6$  bits to a destination computer ( $D$ ) over a network of two routers ( $R_1$  and  $R_2$ ) and three links ( $L_1$ ,  $L_2$  and  $L_3$ ).  $L_1$  connects  $S$  to  $R_1$ ;  $L_2$  connects  $R_1$  to  $R_2$ ; and  $L_3$  connects  $R_2$  to  $D$ . Let each link be of length 100 km. Assume signals travel over each link at a speed of  $10^8$  meters per second. Assume that the link bandwidth on each link is 1 Mbps. Let the file be broken down into 1000 packets each of size 1000 bits. Find the total sum of transmission and propagation delays in transmitting the file from  $S$  to  $D$ ?



Q. 58

Consider an instance of TCP's Additive Increase Multiplicative Decrease (AIMD) algorithm where the windows size at the start of the slow start phase is 2 MSS and the threshold at the start of the first transmission is 8 MSS. Assume that a time-out occurs during the fifth transmission. Find the congestion window size at the end of the tenth transmission.



本本本本本本本本本

## ANSWER KEY

Computer Network									
1	2	3	4	5	6	7	8	9	10
(D)	(D)	(D)	(C)	(B)	(A)	(B)	(A)	(B)	(A)
11	12	13	14	15	16	17	18	19	20
(D)	(B)	(B)	(D)	(B)	(B)	(D)	(D)	(D)	(B)
21	22	23	24	25	26	27	28	29	30
(B)	(C)	(C)	(A)	(A)	(B)	(C)	(A)	(C)	(C)
31	32	33	34	35	36	37	38	39	40
(B)	(C)	(B)	(B)	(D)	(C)	(B)	(C)	(B)	(B)
41	42	43	44	45	46	47	48	49	50
(C)	(D)	(B)	(D)	(D)	(D)	(C)	(B)	(A)	(C)
51	52	53	54	55	56	57	58		
(A)	(C)	(C)	(C)	(C)	(A)	(D)	(C)		

STUDYNAMA

[Show All Answers](#)[Hide All Answers](#)

## 6.02 Practice Problems: MAC protocols

Please read Chapter 15 before trying to solve these problems. Please also solve the problems at the end of Chapter 15.

---

### Problem 1.

Which of these statements are true for correctly implemented versions of stabilized unslotted Aloha, stabilized slotted Aloha, and Time Division Multiple Access (TDMA)? Assume that the slotted and unslotted versions of Aloha use the same stabilization method and parameters.

- A. When the number of nodes is large, unslotted Aloha has a lower maximum throughput than slotted Aloha.

[Hide Answer](#)

True. By a factor of 2:  $1/(2e)$  instead of  $1/e$ .

- B. When the number of nodes is large and nodes transmit data according to a Poisson process, there exists *some* offered load for which the throughput of unslotted Aloha is higher than the throughput of slotted Aloha.

[Hide Answer](#)

False.

- C. TDMA has no packet collisions.

[Hide Answer](#)

True. TDMA eliminates collisions by explicitly allotting time slots.

- D. There exists some offered load pattern for which TDMA has lower throughput than slotted Aloha.

[Hide Answer](#)

True. For example, a skewed workload in which some nodes have much more traffic to send than others.

---

### Problem 2.

Binary exponential backoff is a mechanism used in some MAC protocols. Which of the following statements is correct?

- A. It ensures that two nodes that experience a collision in a time slot will *never* collide with each other when they each retry that packet.
- B. It ensures that two or more nodes that experience a collision in a time slot will experience a lower probability of colliding with each other when they each retry that packet.

- C. It can be used with slotted Aloha but not with carrier sense multiple access.
- D. Over short time scales, it improves the fairness of the throughput achieved by different nodes compared to not using the mechanism.

[Hide Answer](#)

B is true. The others are false.

---

### Problem 3.

In the Aloha stabilization protocols we studied, when a node experiences a collision, it decreases its transmission probability, but sets a lower bound,  $p_{\min}$ . When it transmits successfully, it increases its transmission probability, but sets an upper bound,  $p_{\max}$ .

- A. Why would we set a lower bound on  $p_{\min}$  that is not too close to 0?

[Hide Answer](#)

To avoid starvation where some nodes are denied access to the medium for long periods of time.

- B. Why would we set  $p_{\max}$  to be significantly smaller than 1?

[Hide Answer](#)

To avoid the capture effect, in which a successful node hogs the medium for multiple time slots even when other nodes are backlogged.

- C. Let  $N$  be the average number of backlogged nodes. What happens if we set  $p_{\min} \gg 1/N$ ?

[Hide Answer](#)

The rate of collisions will be high and the utilization close to 0.

---

### Problem 4.

Consider a shared medium with  $N$  backlogged nodes running the slotted Aloha MAC protocol without any backoffs. An idle slot is one in which no node sends data. We will refer to the fraction of time during which no node uses the medium as the "idle time" of the protocol.

- A. If each node has a sending probability of  $p$ , what is the idle time? What are the smallest and largest possible values of the idle time?

[Hide Answer](#)

$(1-p)^N$ , 0 and 1.

- B. Assume  $N$  is large. If the Aloha sending probability,  $p$ , for each node is picked so as to maximize the utilization, what is the corresponding idle time?

[Hide Answer](#)

1/e --> same as the utilization!

---

### Problem 5.

True or false?

Assume that the shared medium has N nodes and they are always backlogged.

- A. In a slotted Aloha MAC protocol using binary exponential backoff, the probability of transmission will always eventually converge to some value p, and all nodes will eventually transmit with probability p.

[Hide Answer](#)

False - In a binary exponential backoff, the probability of transmission constantly changes. If the transmission succeeds, then the probability goes up. If the transmission fails, the probability goes down.

- B. Using carrier sense multiple access (CSMA), suppose that a node "hears" that the channel is busy at time slot t. To maximize utilization, the node should not transmit in slot t and instead transmit the packet in the next time slot with probability 1.

[Hide Answer](#)

False - The node should not transmit at time t+1 with 100% probability. Other nodes may have also "heard" that the channel is busy and would want to send a packet at time t+1. So the node should send with a probability less than 100% to reduce the possibility of a collision.

- C. There is some workload for which an unslotted Aloha with perfect CSMA will not achieve 100% utilization.

[Hide Answer](#)

True - Multiple nodes may think that the channel is idle and send a packet at the same time, resulting in a collision. Thus, the utilization can never reach 100%.

---

### Problem 6.

Eight Cell Processor cores are connected together with a shared bus. To simplify bus arbitration, Ben Bittdidle, young IBM engineer, suggested time-domain multiplexing (TDM) as an arbitration mechanism. Using TDM each of the processors is allocated a equal-sized time slots on the common bus in a round-robin fashion. He's been asked to evaluate the proposed scheme on two types of applications: 1) core-to-core streaming, 2) random loads.

- 1) Core-to-core streaming setup: Assume each core has the same stream bandwidth requirement.
- 2) Random loads setup: core 1 load = 20%, core 2 load = 30%, core 3 load = 10%, core 4 = 5%, core 5 = 1%, core 6 = 3%, core 7 = 1%, core 8 load = 30%

Help Ben out by evaluating the effectiveness (bus utilization) of TDM under these two traffic scenarios.

[Hide Answer](#)

- 1) All cores have same bandwidth requirements during streaming, so, bus utilization is 100%

2) Each core is given 12.5% of bus bandwidth, but not all cores can use it, and some need more than that. So bus utilization is:  $12.5\% + 12.5\% + 10\% + 5\% + 1\% + 3\% + 1\% + 12.5\% = 57.5\%$

---

### Problem 7.

Randomized exponential backoff is a mechanism used to stabilize contention MAC protocols. Which of the following statements is correct?

- A. It ensures that two nodes that experience a collision in a time-slot will *never* collide with each other when they each retry that packet.

**Hide Answer**

False. They might get unlucky and back-off the same amount.

- B. It ensures that two or more nodes that experience a collision in a time-slot will experience a lower probability of colliding with each other when they each retry that packet.

**Hide Answer**

True. The probability of a repeat collision is smaller in each subsequent backoff.

- C. It can be used with slotted Aloha but not with CSMA.

**Hide Answer**

False. It can be used in any contention protocol.

---

### Problem 8.

Three users X, Y and Z use a shared link to connect to the Internet. Only one of X, Y or Z can use the link at a given time. The link has a capacity of 1 Megabit/s. There are two possible strategies for accessing the shared link:

- TDMA: equal slots of 0.1 seconds.
- "Taking turns": adds a latency of 0.05 seconds before taking the turn. The user can then use the link for as long as it has data to send. A user requests the link only when it has data to send.

In each of the following two cases, which strategy would you pick and why?

- A. X, Y and Z send a 40 Kbytes file every 1sec.

**Hide Answer**

TDMA. Why: Each of the users generate a load of  $40\text{KB/s} = 0.32\text{ Megabits/s}$ , which can be fully transmitted given the share of  $0.33\text{ Megabits/s}$  available per user when partitioning the channel with TDMA. Taking turns on the other hand does not offer enough capacity for all the files to be transmitted:  $3 \times 0.32 + 3 \times 0.05 = 1.11\text{s} > 1\text{s}$ , and would incur extra overhead.

- B. X sends 80 Kbytes files every 1sec, while Y and Z send 10 Kbytes files every 1sec.

[Hide Answer](#)

Taking Turns Why: First, by using TDMA, X does not have enough capacity to transmit,  $80 \text{ Kbytes/s} = 0.640 \text{ Megabits/s} > 0.33 \text{ Megabits/s}$ . Second, with TDMA, Y and Z waste 3 out of 4 slots. On the other hand, when taking turns, there is enough capacity to transmit all the data:

$$0.64 + 0.05 + 0.08 + 0.05 + 0.08 + 0.05 = 0.95 \text{ s.}$$

---

### Problem 9.

Alyssa P. Hacker is setting up an 8-node broadcast network in her apartment building in which all nodes can hear each other. Nodes send packets of the same size. If packet collisions occur, both packets are corrupted and lost; no other packet losses occur. All nodes generate equal load on average.

Alyssa observes a utilization of 0.5. Which of the following are consistent with the observed utilization?

- A. True/False: Four nodes are backlogged on average, and the network is using Slotted Aloha with stabilization, and the fairness is close to 1.

[Hide Answer](#)

False. With four backlogged nodes and fairness close to 1, the probability of transmission is 1/4. That would put the utilization at  $4 * (1/4) * (1 - 1/4)^3 = 0.42 < 0.5$ .

- B. True/False: Four nodes are backlogged on average, and the network is using TDMA, and the fairness is close to 1.

[Hide Answer](#)

True. TDMA gives each node an equal share of the network, but 4 nodes have nothing to send, giving a utilization of 0.5.

Now suppose Alyssa's 8-node network runs the Carrier Sense Multiple Access (CSMA) MAC protocol. The maximum data rate of the network is 10 Megabits/s. Including retries, each node sends traffic according to some unknown random process at an average rate of 1 Megabit/s per node. Alyssa measures the network's utilization and finds that it is 0.75. No packets get dropped in the network except due to collisions.

- C. What fraction of packets sent by the nodes (including retries) experience a collision?

[Hide Answer](#)

The offered load presented to the network is 8 Megabits/s in aggregate. The throughput of the protocol is  $0.75 * 10 = 7.5 \text{ Megabits/s}$ . The packet collision rate is therefore equal to

$$1 - 7.5 / 8 = 1 / 16 = 6.25\%.$$

---

### Problem 10.

*Note: this problem is useful to review how to set up and solve problems related to Aloha-like access protocols, but the calculations shown in the answer are more complex than we would ask on a quiz.*

Consider a network with four nodes, where each node has a dedicated channel to each other node. The

probability that any node transmits is  $p$ . Each node can only send OR receive one packet at a time. What is the utilization of the network? Assume each packet takes 1 time slot. Assume each node has 3 queues -- one for each other node -- and that each is backlogged.

[Hide Answer](#)

Utilization is the expected number of packets that get through in each slot divided by the maximum (2, i.e. when A always transmits to B and C always transmits to D).

$$P[4 \text{ nodes transmit}] = p^4$$

$$P[3 \text{ nodes transmit}] = 4p^3(1-p)$$

$$P[2 \text{ nodes transmit}] = 6p^2(1-p)^2$$

$$P[1 \text{ node transmits}] = 4p(1-p)^3$$

$$P[0 \text{ nodes transmit}] = (1-p)^4$$

For each case, we compute the expected number of packets that get through, then remove conditioning using the total probability theorem.

Expected packets through if 4 nodes transmit = 0

No one is free to receive a packet.

Expected packets through if 3 nodes transmit =  $1*(3/27) = 1/9$

Without loss of generality, assume that A,B and C transmit. There are 27 combinations of destinations that they can transmit to (each can choose one of 3). Of these, only 3 result in the successful transmission of one packet (e.g. A->D,B->C,C->B) or (B->D,A<->C) or (C->D,A<->B)

Expected packets through if 2 nodes transmit =  $2*(2/9) = 4/9$

Similar to above, assume that A and B transmit. There are 9 combinations of destinations. Of these, only 2 result in the successful transmission of 2 packets (A->D,B->C) or (A->C,B->D).

Expected packets through if 1 node transmits = 1

No matter who the node transmits to, it will get through.

Expected packets through if 0 nodes transmit = 0

No packets sent.

Thus, the expected number of packets through is

$$(1/9)*4p^3(1-p) + (4/9)*6p^2(1-p)^2 + (1)*4p(1-p)^3 = 4/9*p^3(1-p) + (4/3)p^2(1-p)^2 + 4p(1-p)^3$$

The utilization is half of that.

### Problem 11.

Suppose that there are three nodes seeking access to a shared medium using slotted Aloha, where each packet takes one slot to transmit. Assume that the nodes are always backlogged, and that each has probability  $p_i$  of sending a packet in each slot, where  $i = 1, 2$  and  $3$  indexes the node. Suppose that we assign more the sending probabilities so that

$$p_1 = 2(p_2) \text{ and } p_2 = p_3$$

- A. What is the utilization of the shared medium?

[Hide Answer](#)

$$U = (p_1)(1 - p_2)(1 - p_3) + (1 - p_1)(p_2)(1 - p_3) + (1 - p_1)(1 - p_2)(p_3)$$

If  $p = p_3$

$$U = 2p(1-p)^2 + 2(1-2p)p(1-p) = 4p - 10p^2 + 6p^3$$

- B. What are the probabilities that maximize the utilization and the corresponding utilization?

[Hide Answer](#)

Differentiating  $U$  and set the result equal to zero we obtain

$$\frac{dU}{dp} = 4 - 20p + 18p^2 = 0$$

has two roots at  $p=0.2616$  and  $0.8495$ . However, only the root at  $0.2616$  is feasible since the other leads to a value of  $p_1$  that is greater than 1. Thus,

$$p_1 = 0.5232, p_2 = 0.2616 \text{ and } p_3 = 0.2616.$$

The corresponding utilization is 0.4695.

---

## Problem 12.

Suppose that two nodes are seeking access to a shared medium using slotted Aloha with binary exponential backoff subject to maximum and minimum limits of the probability  $p_{\max} = 0.8$  and  $p_{\min} = 0.1$ . Suppose that both nodes are backlogged, and at slot  $n$ , the probabilities the two nodes transmit packets are  $p_1 = 0.5$  and  $p_2 = 0.3$ .

- A. What are the possible values of  $p_1$  at slot  $n+1$ ? What are the probabilities associated with each possible value?

[Hide Answer](#)

$p_1$  increases to 0.8 if node 1 transmits a packet and node 2 does not transmit a packet. Probability of this event:

$$(p_1)(1 - p_2) = 0.5 * 0.7 = 0.35$$

$p_1$  decreases to 0.25 if node 1 transmits a packet and node 2 also transmits a packet. Probability of this event:

$$(p_1)(p_2) = 0.15$$

Otherwise  $p_1$  stays the same with probability  $1 - 0.35 - 0.15 = 0.5$ .

- B. What are the possible values of  $p_2$  at slot  $n+1$ ? What are the probabilities associated with each possible value?

[Hide Answer](#)

$p_2$  increases to 0.6 if node 2 transmits a packet and node 1 does not transmit a packet. Probability of this event:

$$(p_2)(1 - p_1) = 0.3 * 0.5 = 0.15$$

$p_2$  decreases to 0.15 if node 2 transmits a packet and node 1 also transmits a packet. Probability of this event:

$$(p_1)(p_2) = 0.15$$

Otherwise  $p_2$  stays the same with probability  $1 - 0.15 - 0.15 = 0.7$ .

---

**Problem 13.** Bluetooth is a wireless technology found on many mobile devices, including laptops, mobile phones, GPS navigation devices, headsets, and so on. It uses a MAC protocol called Time Division Duplex (TDD). In TDD, the shared medium network has 1 master and N slaves. You may assume that the network has already been configured with one device as the master and the others as slaves. Each slave has a unique identifier (ID) that serves as its address, an integer between 1 and N. Assume that no devices ever turn off during the operation of the protocol. Unless otherwise mentioned, assume that no packets are lost.

The MAC protocol works as follows. Time is slotted and each packet is one time slot long.

In every *odd* time slot (1, 3, 5, ...,  $2t-1$ , ...), the master sends a packet addressed to some slave for which it has packets backlogged, in round-robin order (i.e., cycling through the slaves in numeric order).

In every *even* time slot (2, 4, 6, ...,  $2t$ , ...), the slave that received a packet from the master in the immediately preceding time slot gets to send a packet to the master, if it has a packet to send. If it has no packet to send, then that time slot is left unused, and the slot is wasted.

- A. Alyssa P. Hacker finds a problem with the TDD protocol described above, and implements the following rule in addition:

From time to time, in an odd time slot, the master sends a "dummy" packet addressed to a slave even if it has no other data packets to send to the slave (and even if it has packets for other slaves).

Why does Alyssa's rule improve the TDD protocol?

[Hide Answer](#)

Because it will prevent a slave from being starved; without it, a slave that has packets to send will never send data packets if the master never has packets to send to it.

Henceforth, the term "TDD" will refer to the protocol described above, augmented with Alyssa's rule. Moreover, whenever a "dummy" packet is sent, that time slot will be considered a wasted slot.

- B. Alyssa's goal is to emulate a round-robin TDMA scheme amongst the N slaves. Propose a way to achieve this goal by specifying the ID of the slave that the master should send a data or dummy packet to, in time slot  $2t-1$  (note that  $1 \leq t \leq \infty$ ).

[Hide Answer](#)

Send to the slave whose ID is  $(t \bmod N)+1$ . This is almost exactly the same problem as in PSet #6, Task

1 (TDMA). The only difference is that the nodes begin with ID 1 here, not 0.

Henceforth, assume that the TDD scheme implements round-robin TDMA amongst the slaves. Suppose the master always has data packets to send only to an arbitrary (but fixed) subset of the  $N$  slaves. In addition, a (possibly different) subset of the slaves always has packets to send to the master. Each subset is of size  $r$ , a fixed value. Answer the questions below (you may find it helpful to think about different subsets of slaves).

- C. What is the *maximum possible* utilization of such a configuration?

**Hide Answer**

The protocol is TDMA, so the master sends useful data packets in  $r$  of the  $2N$  slots in an epoch, and receives useful packets in the  $r$  of the  $2N$  slots. So the utilization is  $r/N$ . If one  $N$  seeks to maximize that over  $r$ , it's clear that the maximum happens when  $r = N$ , giving us a maximum of 1.

- D. What is the *minimum possible* utilization (for a given value of  $r$ ) of such a configuration? Assume that  $r > N/2$ . Note that if the master does not have a data packet to send to a slave in a round, it sends a "dummy" packet to that slave instead. A dummy packet does not count toward the utilization of the medium.

**Hide Answer**

$r/N$ . When minimizing over all  $r$ , the smallest value becomes  $1/2 + 1/N$  when  $N$  is even and  $1/2 + 1/2N$  when  $N$  is odd.

---

6.02 Introduction to EECS II: Digital Communication Systems  
Fall 2012

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.



# COMSATS Institute of Information Technology, Lahore

## Department of Computer Science

**Course:** Advanced Topics in Computer Networks

**Class:** MS-CS

**Instructor:** Rab Nawaz Khan Jadoon

**Max Marks:** 50

**Exam:** Final Term

**Time:** 3Hrs

**Semester:** Fall 2010

**Date:** 17-02-2011

### Special instructions:

1. First of all write down your name and registration# clearly on your question paper as well as answer paper.
2. You must write Question Paper Serial# in the circle at top left side of title page of your Answer-book.
3. While answering your questions, you must indicate on your Answer-book the same question No. as appears in your question paper.
4. Candidates are required to give their answers in their own words as far as practicable. Marks allotted to each question are indicated against it.
5. All the sections are compulsory.
6. If any helping material is found during the exam, the student is directly punished according to the CIIT rules.

---

### (Solution Manual)

#### Question 1: (10)

- a) A block of addresses is granted to a small organization. We know that one of the addresses is 205.16.37.39/28. What is the (the first address, the last address, and total number of addresses in the block?

#### Solution:

1. The first address in the block can be found by setting the rightmost 32 - n bits to Os.

OR

- a. The first address can be found by ANDing the given addresses with the mask.

2. The last address in the block can be found by setting the rightmost 32 - n bits to Is.

OR

- a. The last address can be found by ORing the given addresses with the complement of the mask.

3. The number of addresses in the block can be found by using the formula  $2^{32-n}$ .

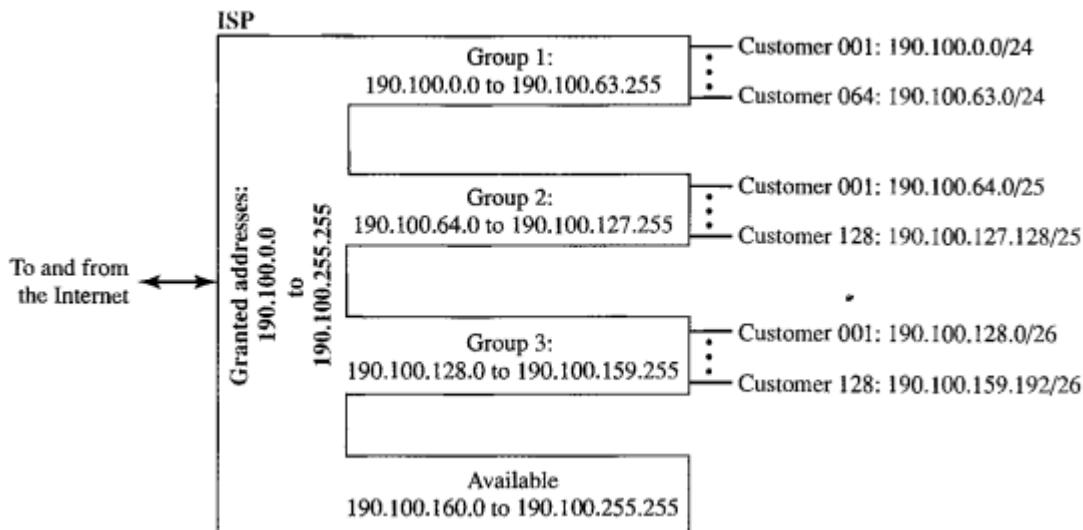
OR

- a. The number of addresses can be found by complementing the mask, interpreting it as a decimal number, and adding 1 to it.

- b) An ISP is granted a block of addresses starting with 190.100.0.0/16 (65,536 addresses). The ISP needs to distribute these addresses to three groups of customers as follows:

- 1) The first group has 64 customers; each needs 256 addresses.
- 2) The second group has 128 customers; each needs 128 addresses.
- 3) The third group has 128 customers; each needs 64 addresses.
- 4) Design the subblocks and find out how many addresses are still available after these allocations.

### Solution:



### 1. Group 1

For this group, each customer needs 256 addresses. This means that 8 ( $\log_2 256$ ) bits are needed to define each host. The prefix length is then  $32 - 8 = 24$ . The addresses are

1st Customer: 190.100.0.0/24      190.100.0.255/24

2nd Customer: 190.100.1.0/24      190.100.1.255/24

.....

64th Customer: 190.100.63.0/24      190.100.63.255/24

$$\text{Total} = 64 \times 256 = 16,384$$

### 2. Group 2

For this group, each customer needs 128 addresses. This means that 7 ( $\log_2 128$ ) bits are needed to define each host. The prefix length is then  $32 - 7 = 25$ . The addresses are,

1st Customer: 190.100.64.0/25      190.100.64.127/25

2nd Customer: 190.100.64.128/25      190.100.64.255/25

.....  
128th Customer: 190.100.127.128/25                  190.100.127.255/25 3.

Total =  $128 \times 128 = 16,384$

### 3. Group 3

For this group, each customer needs 64 addresses. This means that 6 ( $\log_2 64$ ) bits are needed to each host. The prefix length is then  $32 - 6 = 26$ . The addresses are,

1st Customer: 190.100.128.0/26

2nd Customer: 190.100.128.64/26

.....  
128th Customer: 190.100.159.192/26

Total =  $128 \times 64 = 8192$

Number of granted addresses to the ISP: 65,536

Number of allocated addresses by the ISP: 40,960

Number of available addresses: 24,576

### Question 2: (10)

- a. Contrast TDMA with CDMA in terms of advantages regarding medium access?

#### Solution:

##### **TDMA**

In time-division multiple access (TDMA), the stations share the bandwidth of the channel in time. Each station is allocated a time slot during which it can send data. Each station transmits its data in its assigned time slot. In TDMA, the bandwidth is just one channel that is timeshared between different stations.

##### **CDMA**

Code-division multiple access (CDMA) was conceived several decades ago. Recent advances in electronic technology have finally made its implementation possible. CDMA differs from FDMA because only one channel occupies the entire bandwidth of the link. It differs from TDMA because all stations can send data simultaneously; there is no timesharing.

## b. How Chipping sequences are created in CDMA?

### Solution

To generate chip sequences, we use a Walsh table, which is a two-dimensional table with an equal number of rows and columns, as shown in Figure,

$$W_1 = \begin{bmatrix} +1 \end{bmatrix}$$
$$W_{2N} = \begin{bmatrix} W_N & W_N \\ W_N & \overline{W_N} \end{bmatrix}$$

a. Two basic rules

$$W_1 = \begin{bmatrix} +1 \end{bmatrix}$$
$$W_2 = \begin{bmatrix} +1 & +1 \\ +1 & -1 \end{bmatrix}$$
$$W_4 = \begin{bmatrix} +1 & +1 & +1 & +1 \\ +1 & -1 & +1 & -1 \\ +1 & +1 & -1 & -1 \\ +1 & -1 & -1 & +1 \end{bmatrix}$$

b. Generation of  $W_1$ ,  $W_2$ , and  $W_4$

### c. Find the chips for a network with,

a. Two stations

b. Four stations

### Solution

a. For a two-station network, we have  $[+1 +1]$  and  $[+1 -1]$ .

b. For a four-station network we have  $[+1 +1 +1 +1]$ ,  $[+1 -1 +1 -1]$ ,  
 $[+1 +1 -1 -1]$ , and  $[+1 -1 -1 +1]$ .

### d. What is the number of sequences if we have 90 stations in our network?

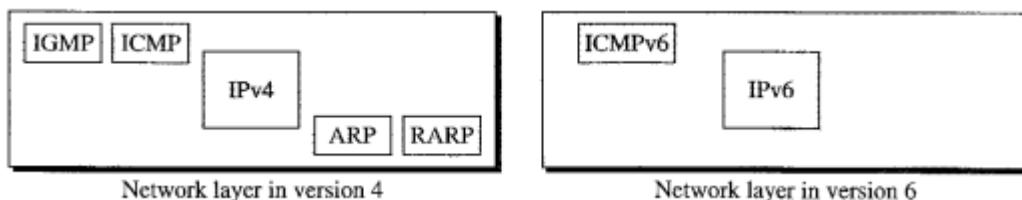
### Solution:

The number of sequences needs to be  $2^m$ . We need to choose  $m = 7$  and  $N = 2^7$  or 128. We can then use 90 of the sequences as the chips.

**Question 3:** (10)

### a. Discuss the difference between ICMPv4 and ICMPv6?

### Solution:



The ARP and IGMP protocols in version 4 are combined in ICMPv6. The RARP

Protocol is dropped from the suite because it was rarely used and BOOTP has the same functionality.

**b. Discuss the following error reporting messages in ICMP,**

- a. Type 3
- b. Type 4
- c. Type 5

**Solution:**

**Type 3 (Destination Unreachable)**

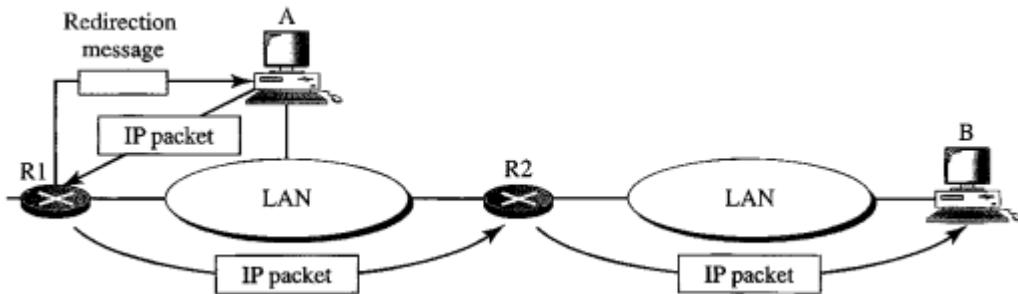
When a router cannot route a datagram or a host cannot deliver a datagram, the datagram is discarded and the router or the host sends a destination-unreachable message back to the source host that initiated the datagram.

**Type 4 (Source Quench)**

The source-quench message in ICMP was designed to add a kind of flow control to the IP. When a router or host discards a datagram due to congestion, it sends a source-quench message to the sender of the datagram. This message has two purposes. First, it informs the source that the datagram has been discarded.

**Second**, it warns the source that there is congestion somewhere in the path and that the Source should slow down (quench) the sending process.

**Type 5 (Redirection)**



**c. Discuss the basic operation of IGMP? In IGMP, a membership report is sent twice, one after the other, why?**

**Solution:**

IGMP 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.

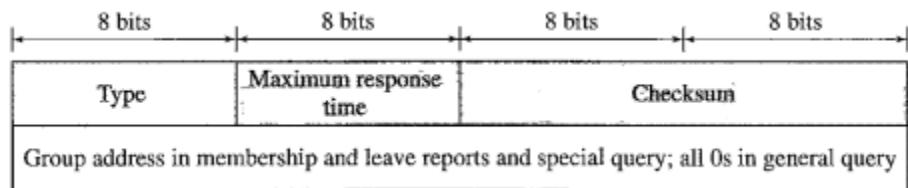
### **Operation:**

IGMP operates locally. A multicast router connected to a network has a list of multicast addresses of the groups with at least one loyal member in that network.

For each group, there is one router that has the duty of distributing the multicast packets destined for that group. This means that if there are three multicast routers connected to a network, their lists of groupids are mutually exclusive. A host or multicast router can have membership in a group. When a host has membership, it means that one of its processes (an application program) receives multicast packets from some group. When a router has membership, it means that a network connected to one of its other interfaces receives these multicast packets. We say that the host or the router has an interest in the group. In both cases, the host and the router keep a list of groupids and relay their interest to the distributing router

#### **d. Discuss the IGMP message format?**

#### **Solution:**



### **Question 4:** (10)

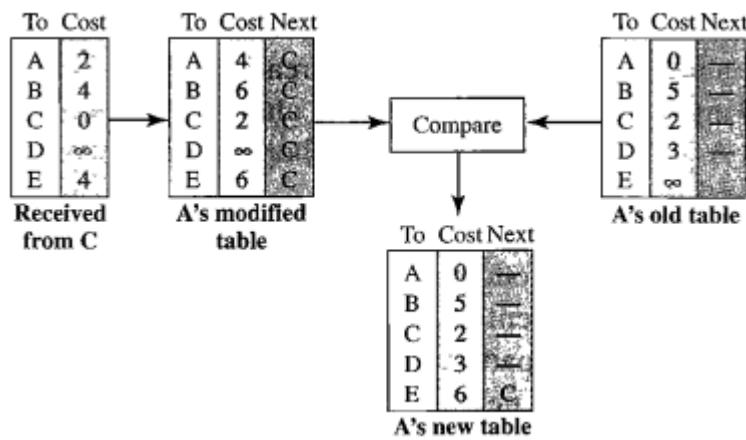
#### **a. How routing tables are updated in Distance vector routing?**

#### **Solution:**

When a node receives a two-column table from a neighbor, it needs to update its routing table. Updating takes three steps:

1. The receiving node needs to add the cost between itself and the sending node to each value in the second column. The logic is clear. If node C claims that its distance to a destination is  $x$  mi, and the distance between A and C is  $y$  mi, then the distance between A and that destination, via C, is  $x + y$  mi.
2. The receiving node needs to add the name of the sending node to each row as the third column if the receiving node uses information from any row. The sending node is the next node in the route.
3. The receiving node needs to compare each row of its old table with the corresponding row of the modified version of the received table.

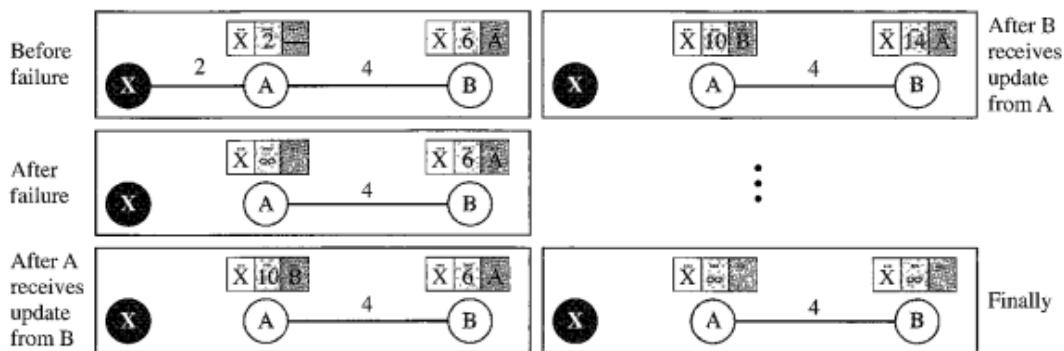
- a. If the next-node entry is different, the receiving node chooses the row with the smaller cost. If there is a tie, the old one is kept.
- b. If the next-node entry is the same, the receiving node chooses the new row.
- For example, suppose node C has previously advertised a route to node X with distance
4. Suppose that now there is no path between C and X; node C now advertises this route with a distance of infinity. Node A must not ignore this value even though its old entry is smaller. The old route does not exist any more. The new route has a distance of infinity.



- b. What is the “two node loop instability” in DVR? What different strategies are advised by researcher to solve this problem?**

#### Solution:

A problem with distance vector routing is instability, which means that a network using this protocol can become unstable.

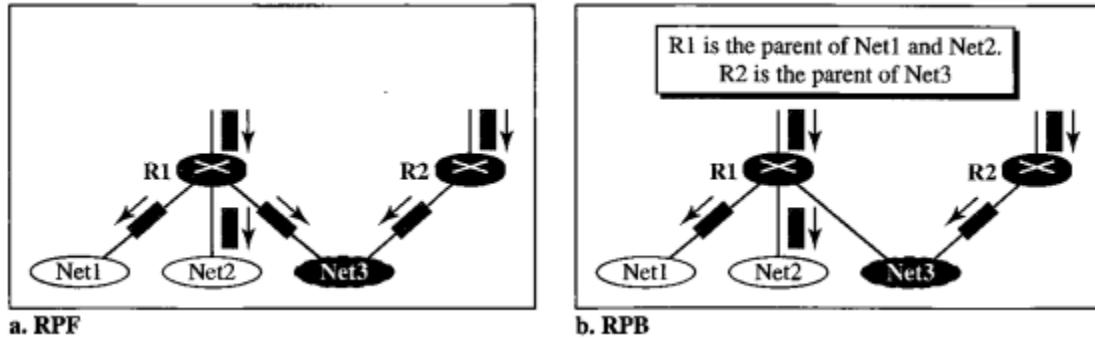


Following are some well known strategies use to overcome the above problem,

- Defining infinity
- Split Horizon
- Split horizon with reverse poison

c. Pictorially represents the difference between RPF and RPB?

**Solution:**



**Question 5:**

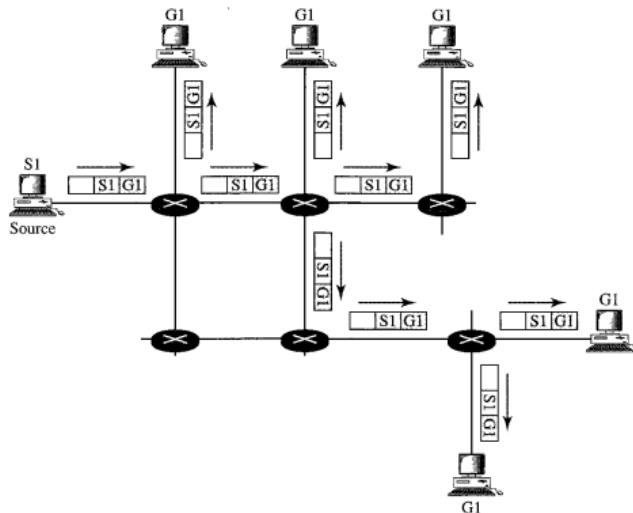
(10)

a. What is multicasting and multiple unicasting?

**Solution:**

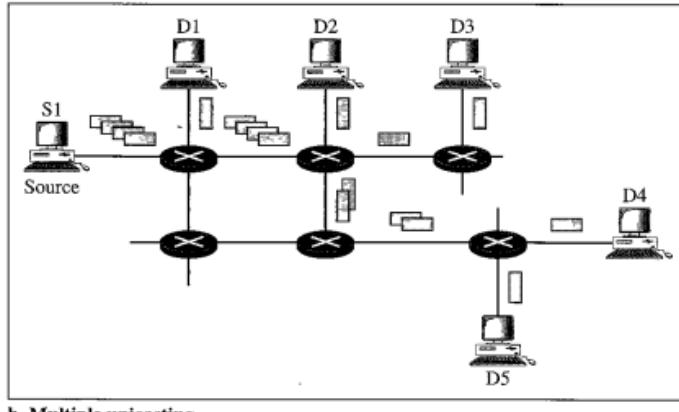
**Multicasting**

In multicast communication, there is one source and a group of destinations. The relationship is one-to-many. In this type of communication, the source address is a unicast address, but the destination address is a group address, which defines one or more destinations. The group address identifies the members of the group.



**Multiple Unicasting**

In multiple unicasting, several packets start from the source. If there are five destinations, for example, the source sends five packets, each with a different unicast destination address.



b. Multiple unicasting

**b. What is source based tree? Discuss its major bottleneck?**

**Solution:**

### Source Based Tree

In the source-based tree approach, each router needs to have one shortest path tree for each group. The shortest path tree for a group defines the next hop for each network that has loyal member(s) for that group.

### Bottleneck

We can imagine the complexity of the routing table if we have hundreds or thousands of groups. However, we will show how different protocols manage to alleviate the situation.

**c. Briefly discuss the core of your research paper that you have done?**

**Solution:**

This is answered by students

\*\*\*Best of Luck\*\*\*

## **Chapter 4: Sample Questions, Problems and Solutions**

### **Bölüm 4: Örnek Sorular, Problemler ve Çözümleri**

#### **Örnek Sorular (Sample Questions):**

- What are broadcast channels or multi-access channels or random access channels?
- What is a function of MAC (Medium Access Control) sublayer?
- Which are the static channel allocation methods?
- What are the lacks of the static channel allocation methods?
- What is a pure ALOHA protocol?
- What is a slotted ALOHA protocol?
- What is a contention system?
- What are Persistent and Nonpersistent CSMA (Carrier Sense Multiple Access) protocols?
- What is CSMA/CD (Carrier Sense Multiple Access with Collision Detection) protocol?
- What is a Collision-Free protocol?
- What is a Bit Map protocol?
- What is a Binary Countdown protocol?
- What is a Limited-Contention protocol?
- What is an Adaptive Tree Walk Protocol?
- What is a Wavelength Division Multiple Access Protocols?
- What is MACA (Multiple Access with Collision Avoidance)?
- What is MACA for Wireless (MACAW)?
- What are IEEE 802.11 Services?
- What is a Broadband Wireless?
- What is a IEEE 802.16 Protocol Stack?
- The IEEE 802.16 Physical Layer
- What is the IEEE 802.16 MAC Sublayer protocol?
- What is the IEEE 802.16 Frame Structure?
- What is a BLUETOOTH?
- What is an IEEE 802.15 Protocol Stack?
- What is the IEEE 802.15 Frame Structure?
- What is a Bridge?
- Bridges from 802.x to 802.y
- Local Internetworking
- Spanning Tree Bridges
- Remote Bridges
- What are Repeaters?
- What are Hubs?
- What are Switches?
- What are Routers?
- What are Gateways?
- What are Virtual LANs?
- What is the IEEE 802.1Q Standard?

## Örnek Problemler ve Çözümleri (Sample Problems and Solutions):

### (Chapter 4, Problem 2-1)

A group of  $N$  stations shares a  $56\text{-kbps}$  pure ALOHA channel. Each station outputs a  $1000\text{-bit}$  frame on an average of once every  $100\text{ sec}$ , even if the previous one has not yet been sent (e.g., the stations can buffer outgoing frames). What is the maximum value of  $N$ ?

ANS:

With pure ALOHA the usable bandwidth is  $0.184 \times 56\text{ kbs} = 10.3\text{ kbps}$ . Each station requires  $10\text{ bps}$ , so  $N = 10300 / 10 = 1030$  stations.

### (Chapter 4, Problem 4)

1000 airline reservation stations are competing for the use of a single slotted ALOHA channel. The average station makes 36 requests per hour. A slot is  $100\text{ }\mu\text{sec}$ . What is the approximate total channel load?

ANS:

Each terminal makes one request every  $3600\text{ sec} / 36\text{ request} = 100\text{ sec}$ .

Total load is 1000 requests per 100 sec or 10 requests per sec.

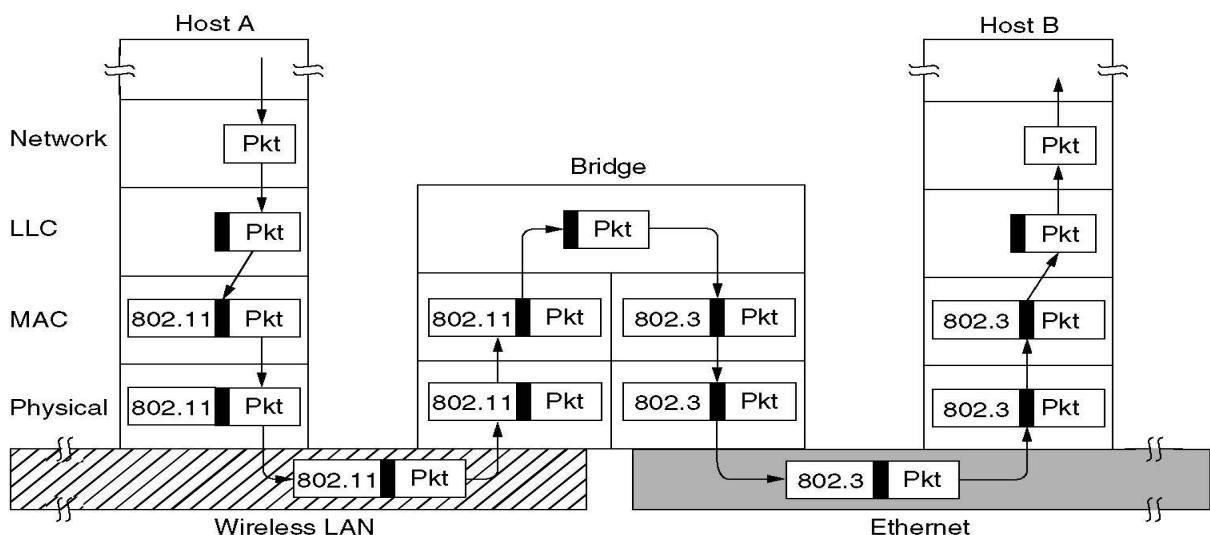
There are  $1\text{ sec}/100\text{ }\mu\text{sec} = 1000000\text{ }\mu\text{sec} / 100\text{ }\mu\text{sec} = 10000$  slots in one second.

Hence,  $G = 10/10000 = 1/1000 = 0.1\%$

### (Chapter 4, Problem 7.1)

Sketch and describe the operation of a LAN bridge from 802.11 to 802.3.

ANS:



**Host A** on a wireless (802.11) LAN has a packet to send to a fixed host, **B**, on an (802.3) Ethernet to which the wireless LAN is connected. The packet descends into the LLC sublayer and acquires an LLC header (shown in black in the figure). Then it passes into the MAC sublayer and an 802.11 header is prepended to it (also a trailer, not shown in the figure). This unit goes out over the air and is picked up by the base station, which sees that it needs to go to the fixed Ethernet. When it hits the bridge connecting the 802.11 network to the 802.3 network, it starts in the physical layer and works its way upward. In the MAC sublayer in the bridge, the 802.11 header is stripped off. The bare packet (with LLC header) is then handed off to the LLC sublayer in the bridge. In this example, the packet is destined for an 802.3 LAN, so it works its way down the 802.3 side of the bridge and off it goes on the Ethernet. Note that a bridge connecting  $k$  different LANs will have  $k$  different MAC sublayers and  $k$  different physical layers, one for each type.

**(Chapter 4, Problem 10-1)**

Sixteen stations, numbered 1 through 16, are contending for the use of a shared channel by using the adaptive tree walk protocol. If all the stations whose addresses are odd numbers suddenly become ready at once, how many bit slots are needed to resolve the contention?

**ANS:**

**Stations 1, 3, 5, 7, 9, 11, 13, and 15 want to send:**

**Slot 1: 1, 3, 5, 7, 9, 11, 13, 15**

**Slot 2: 1, 3, 5, 7**

**Slot 3: 1, 3**

**Slot 4: 1**

**Slot 5: 3**

**Slot 6: 5, 7**

**Slot 7: 5**

**Slot 8: 7**

**Slot 9: 9, 11, 13, 15**

**Slot 10: 9, 11**

**Slot 11: 9**

**Slot 12: 11**

**Slot 13: 13, 15**

**Slot 14: 13**

**Slot 15: 15**

**15 bit slots are needed.**

**(Chapter 4, Problem 10-2)**

Sixteen stations, numbered 1 through 16, are contending for the use of a shared channel by using the adaptive tree walk protocol. If stations 2, 6, 8, 11, and 15 suddenly become ready at once, how many bit slots are needed to resolve the contention?

**ANS:**

**Stations 2, 6, 8, 11, and 15 want to send:**

### Slot 1: 2, 6, 8, 11, 15

### Slot 2: 2, 6, 8

### Slot 3: 2

## Slot 4: 6, 8

## Slot 5: 6

## Slot 6: 8

Slot 7: 1

**Slot 8: 11**

**Slot 9: 15**

**Slot 9: 15  
9 bit slots**

9 bit slots are needed.

### (Chapter 4, Problem 16-1)

**What is the baud rate of the standard 10-Mbps Ethernet which uses Manchester encoding?**

**ANS:**

The Ethernet uses Manchester encoding, which means it has two signal periods per bit sent. The data rate of the standard Ethernet is 10 Mbps, so the baud rate is twice that, 20 Megabaud.

**(Chapter 4, Problem 16-2)**

**What is the baud rate of the Gigabit Ethernet which uses Manchester encoding?**

**ANS:**

The Ethernet uses Manchester encoding, which means it has two signal periods per bit sent. The data rate of the Gigabit Ethernet is 1Gbps, so the baud rate is twice that, 1Gigabaud.

(Chapter 4, Problem 17)

**Sketch the Manchester encoding for the bit stream: 1010110000110100**

**ANS:**

### (Chapter 4, Problem 19)

A 1-km-long, 10-Mbps CSMA/CD (Carrier Sense Multiple Access with Collision Detection) LAN (not 802.3) has a propagation speed of 200 m/ $\mu$ sec. Repeaters are not allowed in this system. Data frames are 256 bits long, including 32 bits of header, checksum, and other overhead. The first bit slot after a successful transmission is reserved for the receiver to capture the channel in order to send a 32-bit acknowledgement frame. What is the effective data rate, excluding overhead, assuming that there are no collisions?

ANS:

The round-trip propagation time of the cable is 10  $\mu$ sec. A complete transmission has six phases:

- 1) Transmitter seizes cable (10  $\mu$ sec)
- 2) Transmit data (25.6  $\mu$ sec)
- 3) Delay for last bit to get to the end (5.0  $\mu$ sec)
- 4) Receiver seizes cable (10  $\mu$ sec)
- 5) Acknowledgement sent (3.2  $\mu$ sec)
- 6) Delay for last bit to get to the end (5.0  $\mu$ sec)

The sum of these is 58.8  $\mu$ sec. In this period, 224 data bits are sent, for a rate of about 3.8 Mbps.

### (Chapter 4, Problem 21-1)

Consider building a CSMA/CA (Carrier Sense Multiple Access with Collision Detection) network running at 1 Gbps over a 1-km cable with no repeaters. The signal speed in the cable is 200,000 km/sec. What is the minimum frame size?

ANS:

For a 1-km cable, the one-way propagation time is 5  $\mu$ sec, so  $2\tau = 10 \mu$ sec. To make CSMA/CD work, it must be impossible to transmit an entire frame in this interval. At 1 Gbps, all frames shorter than 10,000 bits can be completely transmitted in under 10  $\mu$ sec, so the minimum frame is 10,000 bits or 1250 bytes.

### (Chapter 4, Problem 21-2)

Consider building a CSMA/CD network running at 10 Gbps over a 2.5-km cable with no repeaters. The signal speed in the cable is 200,000 km/sec. What is the minimum frame size?

ANS:

For 2.5-km cable, the one-way propagation time  $\tau = 12.5 \mu$ sec, so  $2\tau = 25 \mu$ sec. At 10 Gbps, 250,000 bits can be completely transmitted in under 25  $\mu$ sec. So minimum frame is 250,000 bits or 3125 bytes.

(Chapter 4, Problem 30)

An 802.16 network has a channel width of 10 MHz. How many bits/sec can be sent to a subscriber station? (Evaluate for QAM-64, QAM-16 and QPSK methods).

ANS:

It depends how far away subscriber is.

If the subscriber is close in, QAM-64 is used for 60 Mbps.

For medium distance, QAM-16 is used for 40 Mbps.

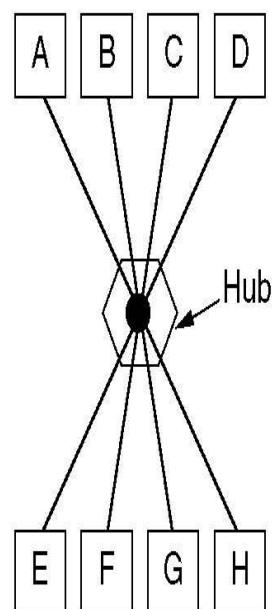
For distant stations, QPSK is used for 20 Mbps.

(Chapter 4, Pages 326-327)

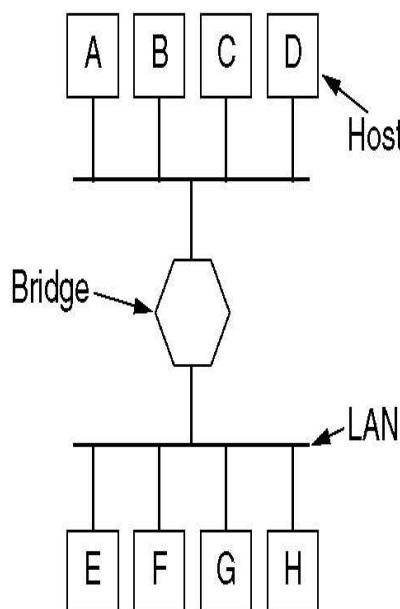
Define which devices are in which layers and give a short definition for each of them.

ANS:

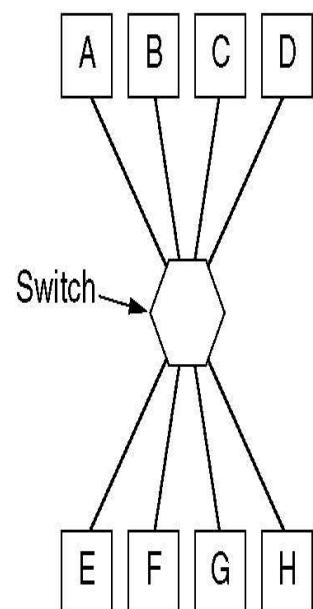
Application layer	Application gateway
Transport layer	Transport gateway
Network layer	Router
Data link layer	Bridge, switch
Physical layer	Repeater, hub



(a)



(b)



(c)

### (Chapter 4, Problem 40)

A switch designed for use with fast Ethernet has a backplane that can move 10 Gbps. How many frames/sec can it handle in the worst case? Hint: the worst case is an endless stream of 64-byte frames.

ANS:

The worst case is an endless stream of 64-byte 64 byte = 512 bits frames.

If the backplane can handle 10 Gbps =  $10^9$  bps.

The number of frames it can handle is  $10^9$  bit / 512 bit

This is 1,953,125 frames / sec.

### (Chapter 4)

There are 4 contention slots. In 1st cycle of frame transmission the stations 1 and 3, in the 2nd cycle of frame transmission the station 0 and 2, in the 3rd cycle of frame transmission the station 3 and in the 4th cycle of frame transmission all stations have frames to send. Sketch these cycles for the basic bit-map protocol.

ANS:

Cycles:	1st cycle	2nd cycle	3rd cycle	4th cycle
Slots:	0 1 2 3	0 1 2 3	0 1 2 3	0 1 2 3
Frames:	1 1 F1 F3	1 1 F0 F2	1 F3	1 1 1 1 F0 F1 F2 F3

### (Chapter 4)

Given 8 stations and their data to send using single channel. Find the number of station and data sent. Use binary countdown protocol.

ANS:

Station 1	0	1	0	1	0	1	0	1
Station 2	1	1	0	0	1	1	0	0
Station 3	0	0	0	1	1	1	0	1
Station 4	1	1	1	0	0	0	1	0
Station 5	1	1	1	1	0	1	0	1
Station 6	0	0	0	0	1	0	1	0
Station 7	1	0	1	1	1	1	0	0
Station 8	0	0	1	1	0	0	1	1

Bit time							
0	1	2	3	4	5	6	7
0							
1	1	0					
0							
1	1	1	0				
1	1	1	1	0	1	0	1
0							
1	0						
0							

The 5<sup>th</sup> station sends the data 11110101

**(Chapter 4)**

**A LAN uses Mok and Ward's version of binary countdown. At a certain instant, the ten stations have the virtual station numbers 8, 2, 4, 5, 1, 7, 3, 6, 9, and 0. The next three stations to send are 4, 3, and 9, in that order. What are the new virtual station numbers after all three have finished their transmissions?**

**ANS:**

**Initial virtual station numbers are:**

**8 2 4 5 1 7 3 6 9 0**

**When station 4 sends, it becomes 0, and 0, 1, 2, 3 are increased by 1.**

**8 3 0 5 2 7 4 6 9 1**

**When station 3 sends, it becomes 0, and 0, 1, 2, 3 are increased by 1.**

**8 0 1 5 3 7 4 6 9 2**

**Finally, when station 9 sends, it becomes 0 and all the other stations are incremented by**

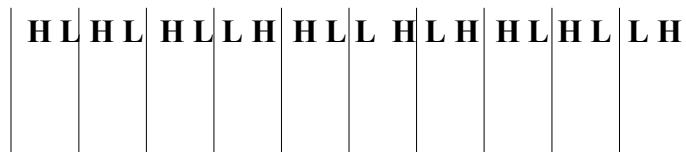
**1. The result is**

**9 1 2 6 4 8 5 7 0 3**

**(Chapter 4)**

**Sketch the differential Manchester encoding for the bit stream 0001110101. Assume the line is initially in low state.**

**ANS:**



# **COMPUTER NETWORKS**

## **[R15A0513]**

# **LECTURE NOTES**

**B.TECH III YEAR – II SEM (R15)**

**(2019-20)**



**DEPARTMENT OF**  
**COMPUTER SCIENCE AND ENGINEERING**

**MALLA REDDY COLLEGE OF ENGINEERING &**  
**TECHNOLOGY**

**(Autonomous Institution – UGC, Govt. of India)**

Recognized under 2(f) and 12 (B) of UGC ACT 1956

(Affiliated to JNTUH, Hyderabad, Approved by AICTE - Accredited by NBA & NAAC – ‘A’ Grade - ISO 9001:2015 Certified)  
Maisammaguda, Dhulapally (Post Via. Hakimpet), Secunderabad – 500100, Telangana State, India

**(R15A0513) Computer Networks****Objectives:**

- To introduce the fundamental types of computer networks.
- To demonstrate the TCP/IP & OSI model merits & demerits.
- To know the role of various protocols in Networking

**UNIT - I:**

Introduction: Network, Uses of Networks, Types of Networks, Reference Models: TCP/IP Model, The OSI Model, Comparison of the OSI and TCP/IP reference model. Architecture of Internet.

Physical Layer: Guided transmission media, Wireless transmission media, Switching

**UNIT - II:**

Data Link Layer - Design issues, Error Detection & Correction, Elementary Data Link Layer Protocols, Sliding window protocols

Multiple Access Protocols - ALOHA, CSMA,CSMA/CD, CSMA/CA, Collision free protocols, Ethernet- Physical Layer, Ethernet Mac Sub layer, Data link layer switching: Use of bridges, learning bridges, spanning tree bridges, repeaters, hubs, bridges, switches, routers and gateways.

**UNIT - III:**

Network Layer: Network Layer Design issues, store and forward packet switching connection less and connection oriented networks-routing algorithms-optimality principle, shortest path, flooding, Distance Vector Routing, Count to Infinity Problem, Link State Routing, Path Vector Routing, Hierarchical Routing; Congestion control algorithms, IP addresses, CIDR, Subnetting, SuperNetting, IPv4, Packet Fragmentation, IPv6 Protocol, Transition from IPv4 to IPv6, ARP, RARP.

**UNIT - IV:**

Transport Layer: Services provided to the upper layers elements of transport protocol addressing connection establishment, Connection release, Error Control & Flow Control, Crash Recovery.

The Internet Transport Protocols: UDP, Introduction to TCP, The TCP Service Model, The TCP Segment Header, The Connection Establishment, The TCP Connection Release, The TCP Sliding Window, The TCP Congestion Control Algorithm.

UNIT - V: Application Layer- Introduction, providing services, Applications layer paradigms: Client server model, HTTP, E-mail, WWW, TELNET, DNS; RSA algorithm,

**TEXT BOOKS:**

1. Computer Networks - Andrew S Tanenbaum, 4th Edition, Pearson Education.
2. Data Communications and Networking - Behrouz A. Forouzan, Fifth Edition TMH, 2013.

**REFERENCES BOOKS:**

1. An Engineering Approach to Computer Networks - S. Keshav, 2nd Edition, Pearson Education.
2. Understanding communications and Networks, 3rd Edition, W. A. Shay, Cengage Learning.
3. Computer Networking: A Top-Down Approach Featuring the Internet, James F. Kurose, K. W. Ross, 3rd Edition, Pearson Education.

**Outcomes:**

- Students should understand and explore the basics of Computer Networks and Various Protocols.
- Student will be in a position to understand the World Wide Web concepts.
- Students will be in a position to administrate a network and flow of information further
- Student can understand easily the concepts of network security, Mobile

## INDEX

<b>UNIT NO</b>	<b>TOPIC</b>	<b>PAGE NO</b>
<b>I</b>	<b>INTRODUCTION TO NETOWRKS</b>	<b>1</b>
	<b>TYPES OF NETWORKS,</b>	<b>4</b>
	<b>INTRODUCTION TO PHYSICAL LAYER</b>	<b>18</b>
	<b>COMPARISON OF OSI AND TCP/IP PROTOCOLS</b>	<b>25</b>
<b>II</b>	<b>DATA LINK LAYER DESIGN ISSUES</b>	<b>35</b>
	<b>SLIDING WINDOW PROTOCOLS</b>	<b>41</b>
	<b>MULTIPLE ACCESS PROTOCOLS</b>	<b>49</b>
<b>III</b>	<b>NETWORK LAYER DESIGN ISSUES</b>	<b>78</b>
	<b>CONNECTION LESS AND CONNECTION ORIENTED PROTOCOLS</b>	<b>80</b>
	<b>ROUTING PROTOCOLS,IP ADDRESS</b>	<b>81</b>
<b>IV</b>	<b>TRANSPORT LAYER SERVICES PROVIDED</b>	<b>101</b>
	<b>THE INTERNET TRANSPORT PROTOCOLS</b>	<b>130</b>
<b>V</b>	<b>APPLICATION LAYER SERVICES</b>	<b>249</b>
	<b>APPLICATIONS LAYER PARADIGMS</b>	<b>256</b>



# **UNIT - I**

## **NETWORKS**

A network is a set of devices (often referred to as *nodes*) connected by communication links. A node can be a computer, printer, or any other device capable of sending and/or receiving data generated by other nodes on the network.

"Computer network" to mean a collection of autonomous computers interconnected by a single technology. Two computers are said to be interconnected if they are able to exchange information.

The connection need not be via a copper wire; fiber optics, microwaves, infrared, and communication satellites can also be used.

Networks come in many sizes, shapes and forms, as we will see later. They are usually connected together to make larger networks, with the **Internet** being the most well-known example of a network of networks.

There is considerable confusion in the literature between a **computer network** and a **distributed system**. The key distinction is that in a distributed system, a collection of independent computers appears to its users as a single coherent system. Usually, it has a single model or paradigm that it presents to the users. Often a layer of software on top of the operating system, called **middleware**, is responsible for implementing this model. A well-known example of a distributed system is the **World Wide Web**. It runs on top of the Internet and presents a model in which everything looks like a document (Web page).

## **USES OF COMPUTER NETWORKS**

### **1. Business Applications**

- to distribute information throughout the company (**resource sharing**). sharing physical resources such as printers, and tape backup systems, is sharing information
- **client-server model**. It is widely used and forms the basis of much network usage.
- **communication medium** among employees. **email (electronic mail)**, which employees generally use for a great deal of daily communication.
- Telephone calls between employees may be carried by the computer network instead of by the phone company. This technology is called **IP telephony** or **Voice over IP (VoIP)** when Internet technology is used.
- **Desktop sharing** lets remote workers see and interact with a graphical computer screen
- doing business electronically, especially with customers and suppliers. This new model is called **e-commerce (electronic commerce)** and it has grown rapidly in recent years.

### **2 Home Applications**

- **peer-to-peer** communication
- person-to-person communication

- electronic commerce
- entertainment.(game playing,)

### **3 Mobile Users**

- Text messaging or texting
- Smart phones,
- GPS (Global Positioning System)
- m-commerce
- NFC (Near Field Communication)

### **4 Social Issues**

With the good comes the bad, as this new-found freedom brings with it many unsolved social, political, and ethical issues.

Social networks, message boards, content sharing sites, and a host of other applications allow people to share their views with like-minded individuals. As long as the subjects are restricted to technical topics or hobbies like gardening, not too many problems will arise.

The trouble comes with topics that people actually care about, like politics, religion, or sex. Views that are publicly posted may be deeply offensive to some people. Worse yet, they may not be politically correct. Furthermore, opinions need not be limited to text; high-resolution color photographs and video clips are easily shared over computer networks. Some people take a live-and-let-live view, but others feel that posting certain material (e.g., verbal attacks on particular countries or religions, pornography, etc.) is simply unacceptable and that such content must be censored. Different countries have different and conflicting laws in this area. Thus, the debate rages.

Computer networks make it very easy to communicate. They also make it easy for the people who run the network to snoop on the traffic. This sets up conflicts over issues such as **employee rights versus employer rights**. Many people read and write email at work. Many employers have claimed the right to read and possibly censor employee messages, including messages sent from a home computer outside working hours. Not all employees agree with this, especially the latter part.

Another conflict is centered around government versus citizen's rights. A new twist with mobile devices is location privacy. As part of the process of providing service to your mobile device the network operators learn where you are at different times of day. This allows them to track your movements. They may know which nightclub you frequent and which medical center you visit.

**Phishing ATTACK:** **Phishing** is a type of social engineering **attack** often used to steal user data, including login credentials and credit card numbers. It occurs when an attacker, masquerading as a trusted entity, dupes a victim into opening an email, instant message, or text message.

**BOTNET ATTACK:** Botnets can be used to perform [distributed denial-of-service attack](#) (DDoS attack), steal data, send spam, and allows the attacker to access the device and its connection.

The effectiveness of a data communications system depends on four fundamental characteristics: delivery, accuracy, timeliness, and jitter.

1. **Delivery.** The system must deliver data to the correct destination. Data must be received by the intended device or user and only by that device or user.

2 **Accuracy.** The system must deliver the data accurately. Data that have been altered in transmission and left uncorrected are unusable.

3. **Timeliness.** The system must deliver data in a timely manner. Data delivered late are useless. In the case of video and audio, timely delivery means delivering data as they are produced, in the same order that they are produced, and without significant delay. This kind of delivery is called *real-time* transmission.

4. **Jitter.** Jitter refers to the variation in the packet arrival time. It is the uneven delay in the delivery of audio or video packets. For example, let us assume that video packets are sent every 30 ms. If some of the packets arrive with 30-ms delay and others with 40-ms delay, an uneven quality in the video is the result.

A data communications system has five components

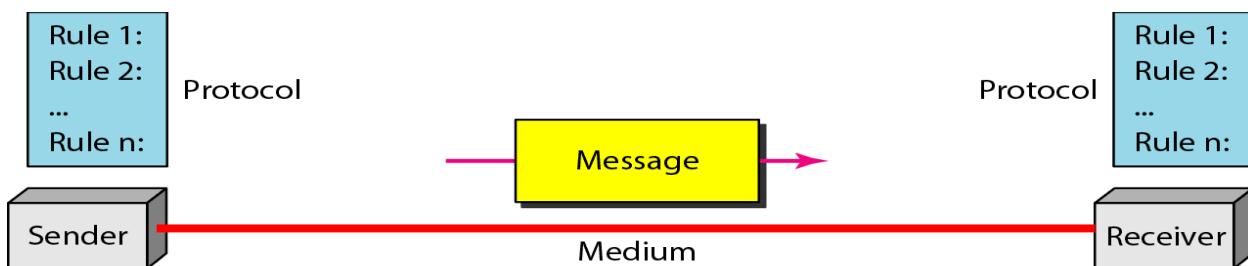
1. **Message.** The message is the information (data) to be communicated. Popular forms of information include text, numbers, pictures, audio, and video.

2 **Sender.** The sender is the device that sends the data message. It can be a computer, workstation, telephone handset, video camera, and so on.

3. **Receiver.** The receiver is the device that receives the message. It can be a computer, workstation, telephone handset, television, and so on.

4. **Transmission medium.** The transmission medium is the physical path by which a message travels from sender to receiver. Some examples of transmission media include twisted-pair wire, coaxial cable, fiber-optic cable, and radio waves.

5. **Protocol.** A protocol is a set of rules that govern data communications. It represents an agreement between the communicating devices. Without a protocol, two devices may be connected but not communicating, just as a person speaking French cannot be understood by a person who speaks only Japanese.



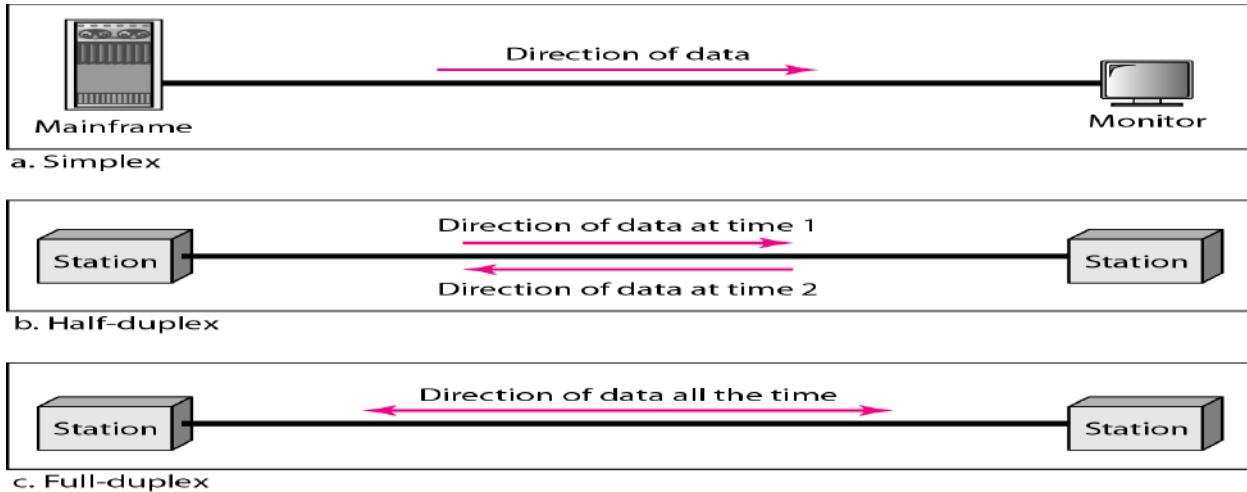
## Data Representation

Text  
Numbers  
Images  
Audio  
Video

---

## Data Flow

Communication between two devices can be simplex, half-duplex, or full-duplex as shown in Figure.



**Simplex** In simplex mode, the communication is unidirectional, as on a one-way street. Only one of the two devices on a link can transmit; the other can only receive (Figure a). Keyboards and traditional monitors are examples of simplex devices.

## Half-Duplex

In half-duplex mode, each station can both transmit and receive, but not at the same time. When one device is sending, the other can only receive, and vice versa (Figure b). Walkie-talkies and CB (citizens band) radios are both half-duplex systems.

## Full-Duplex

In full-duplex, both stations can transmit and receive simultaneously (Figure c). One common example of full-duplex communication is the telephone network. When two people are communicating by a telephone line, both can talk and listen at the same time. The full-duplex mode is used when communication in both directions is required all the time.

---

## Network Criteria

A network must be able to meet a certain number of criteria. The most important of these are performance, reliability, and security.

## Performance

Performance can be measured in many ways, including transit time and response time. Transit time is the amount of time required for a message to travel from one device to another. Response time is the elapsed time between

an inquiry and a response. The performance of a network depends on a number of factors, including the number of users, the type of transmission medium, the capabilities of the connected hardware, and the efficiency of the software.

Performance is often evaluated by two networking metrics: **throughput and delay**. We often need more throughput and less delay. However, these two criteria are often contradictory. If we try to send more data to the network, we may increase throughput but we increase the delay because of traffic congestion in the network.

**Reliability:** In addition to accuracy of delivery, network reliability is measured by the frequency of failure, the time it takes a link to recover from a failure, and the network's robustness in a catastrophe.

**Security:** Network security issues include protecting data from unauthorized access, protecting data from damage and development, and implementing policies and procedures for recovery from breaches and data losses.

---

## **Physical Structures**

Before discussing networks, we need to define some network attributes.

### **Type of Connection**

A network is two or more devices connected through links. A link is a communications pathway that transfers data from one device to another.

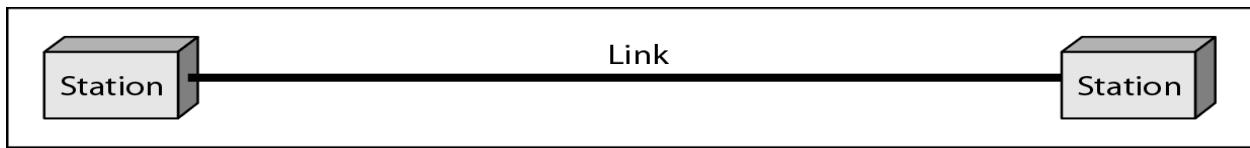
There are two possible types of connections: point-to-point and multipoint.

**Point-to-Point** A point-to-point connection provides a dedicated link between two devices. The entire capacity of the link is reserved for transmission between those two devices. Most point-to-point connections use an actual length of wire or cable to connect the two ends, but other options, such as microwave or satellite links, are also possible

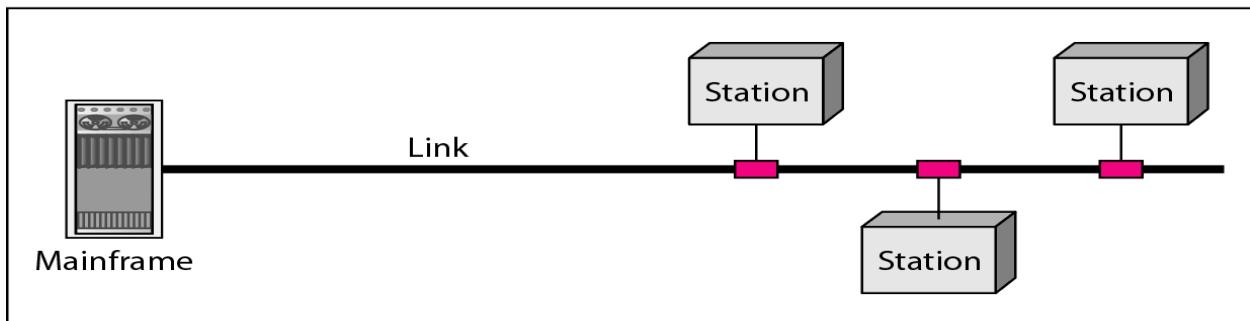
When you change television channels by infrared remote control, you are establishing a point-to-point connection between the remote control and the television's control system.

**Multipoint** A multipoint (also called multi-drop) connection is one in which more than two specific devices share a single link

In a multipoint environment, the capacity of the channel is shared, either spatially or temporally. If several devices can use the link simultaneously, it is a *spatially shared* connection. If users must take turns, it is a *timeshared* connection.



a. Point-to-point



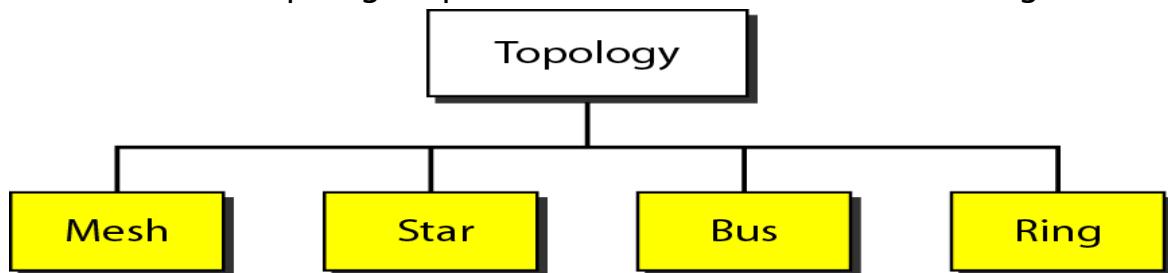
b. Multipoint

## **Physical Topology**

The term *physical topology* refers to the way in which a network is laid out physically.

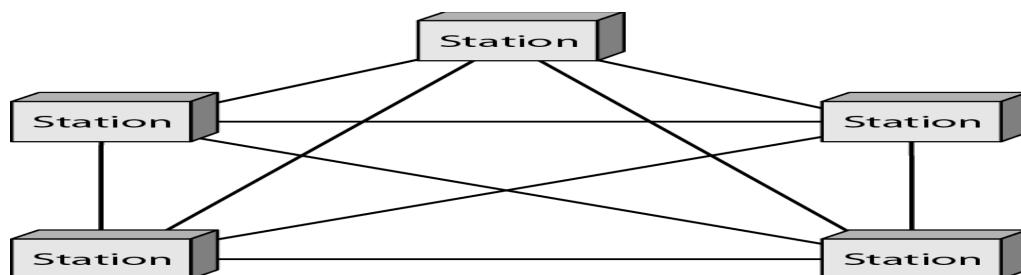
Two or more devices connect to a link; two or more links form a topology. The topology of a network is the geometric representation of the relationship of all the links and linking devices (usually called nodes) to one another.

There are four basic topologies possible: mesh, star, bus, and ring



### **MESH:**

A mesh topology is the one where every node is connected to every other node in the network.



A mesh topology can be a **full mesh topology** or a **partially connected mesh topology**.

In a *full mesh topology*, every computer in the network has a connection to each of the other computers in that network. The number of connections in this

network can be calculated using the following formula ( $n$  is the number of computers in the network):  $n(n-1)/2$

In a *partially connected mesh topology*, at least two of the computers in the network have connections to multiple other computers in that network. It is an inexpensive way to implement redundancy in a network. In the event that one of the primary computers or connections in the network fails, the rest of the network continues to operate normally.

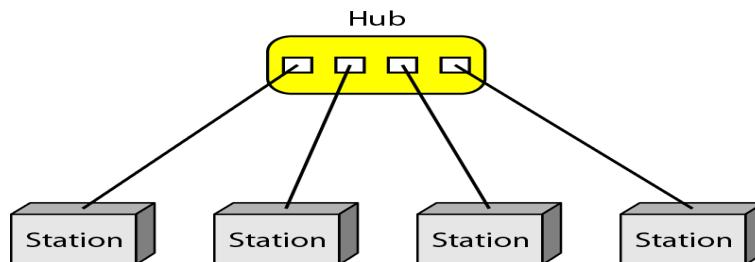
### Advantages of a mesh topology

- Can handle high amounts of traffic, because multiple devices can transmit data simultaneously.
- A failure of one device does not cause a break in the network or transmission of data.
- Adding additional devices does not disrupt data transmission between other devices.

### Disadvantages of a mesh topology

- The cost to implement is higher than other network topologies, making it a less desirable option.
- Building and maintaining the topology is difficult and time consuming.
- The chance of redundant connections is high, which adds to the high costs and potential for reduced efficiency.

## STAR:



A **star network, star topology** is one of the most common network setups. In this configuration, every node connects to a central network device, like a hub, switch, or computer. The central network device acts as a server and the peripheral devices act as clients. Depending on the type of network card used in each computer of the star topology, a coaxial cable or a RJ-45 network cable is used to connect computers together.

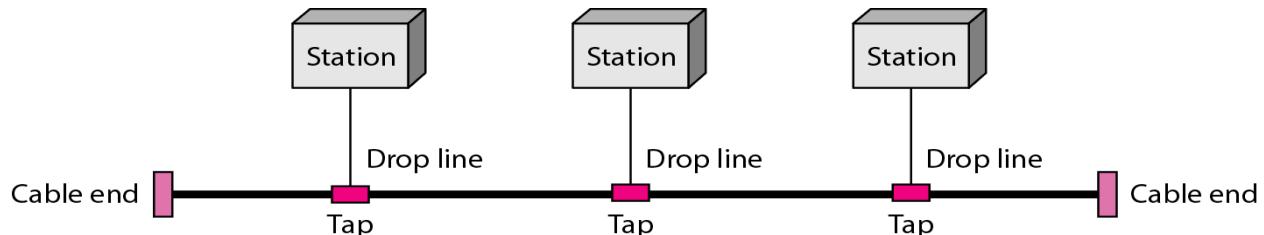
### Advantages of star topology

- Centralized management of the network, through the use of the central computer, hub, or switch.
- Easy to add another computer to the network.
- If one computer on the network fails, the rest of the network continues to function normally.
- The star topology is used in local-area networks (LANs). High-speed LANs often use a star topology with a central hub.

### Disadvantages of star topology

- Can have a higher cost to implement, especially when using a switch or router as the central network device.
- The central network device determines the performance and number of nodes the network can handle.
- If the central computer, hub, or switch fails, the entire network goes down and all computers are disconnected from the network

## BUS:



a **line topology**, a **bus topology** is a network setup in which each computer and network device are connected to a single cable or [backbone](#).

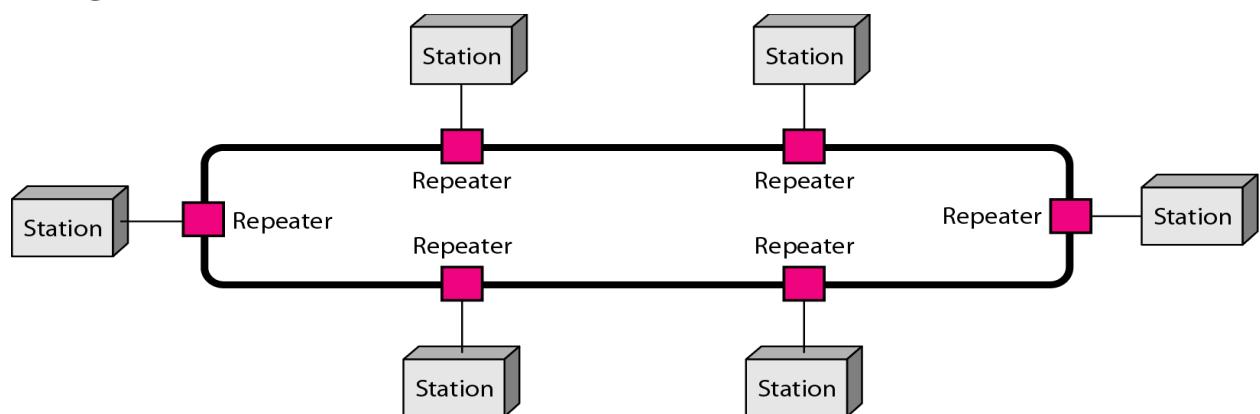
### Advantages of bus topology

- It works well when you have a small network.
- It's the easiest network topology for connecting computers or peripherals in a linear fashion.
- It requires less cable length than a star topology.

### Disadvantages of bus topology

- It can be difficult to identify the problems if the whole network goes down.
- It can be hard to troubleshoot individual device issues.
- Bus topology is not great for large networks.
- Terminators are required for both ends of the main cable.
- Additional devices slow the network down.
- If a main cable is damaged, the network fails or splits into two.

## RING:



A **ring topology** is a network configuration in which device connections create a circular data path. In a ring network, packets of data travel from one device to the next until they reach their destination. Most ring topologies allow packets to travel only in one direction, called a **unidirectional** ring network. Others permit data to move in either direction, called **bidirectional**.

The major disadvantage of a ring topology is that if any individual connection in the ring is broken, the entire network is affected.

Ring topologies may be used in either local area networks (LANs) or wide area networks (WANs).

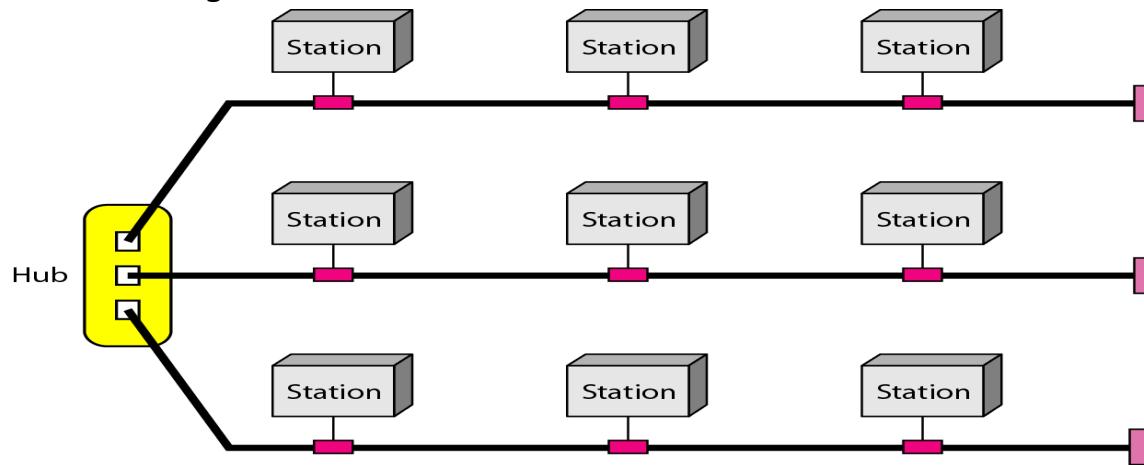
### Advantages of ring topology

- All data flows in one direction, reducing the chance of packet collisions.
- A network server is not needed to control network connectivity between each workstation.
- Data can transfer between workstations at high speeds.
- Additional workstations can be added without impacting performance of the network.

### Disadvantages of ring topology

- All data being transferred over the network must pass through each workstation on the network, which can make it slower than a star topology.
- The entire network will be impacted if one workstation shuts down.
- The hardware needed to connect each workstation to the network is more expensive than Ethernet cards and hubs/switches.

**Hybrid Topology** A network can be hybrid. For example, we can have a main star topology with each branch connecting several stations in a bus topology as shown in Figure



### Types of Network based on size

The types of network are classified based upon the size, the area it covers and its physical architecture. The three primary network categories are LAN, WAN and MAN. Each network differs in their characteristics such as distance, transmission speed, cables and cost.

Basic types

## **LAN (Local Area Network)**

Group of interconnected computers within a small area. (room, building, campus)

Two or more pc's can from a LAN to share files, folders, printers, applications and other devices.

Coaxial or CAT 5 cables are normally used for connections.

Due to short distances, errors and noise are minimum.

Data transfer rate is 10 to 100 mbps.

Example: A computer lab in a school.

## **MAN (Metropolitan Area Network)**

Design to extend over a large area.

Connecting number of LAN's to form larger network, so that resources can be shared.

Networks can be up to 5 to 50 km.

Owned by organization or individual.

Data transfer rate is low compare to LAN.

Example: Organization with different branches located in the city.

## **WAN (Wide Area Network)**

Are country and worldwide network.

Contains multiple LAN's and MAN's.

Distinguished in terms of geographical range.

Uses satellites and microwave relays.

Data transfer rate depends upon the ISP provider and varies over the location.

Best example is the internet.

## **Other types**

### **WLAN (Wireless LAN)**

A LAN that uses high frequency radio waves for communication.

Provides short range connectivity with high speed data transmission.

### **PAN (Personal Area Network)**

Network organized by the individual user for its personal use.

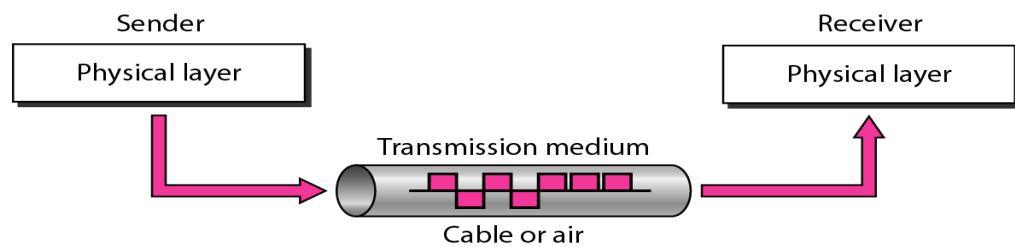
### **SAN (Storage Area Network)**

Connects servers to data storage devices via fiber-optic cables.

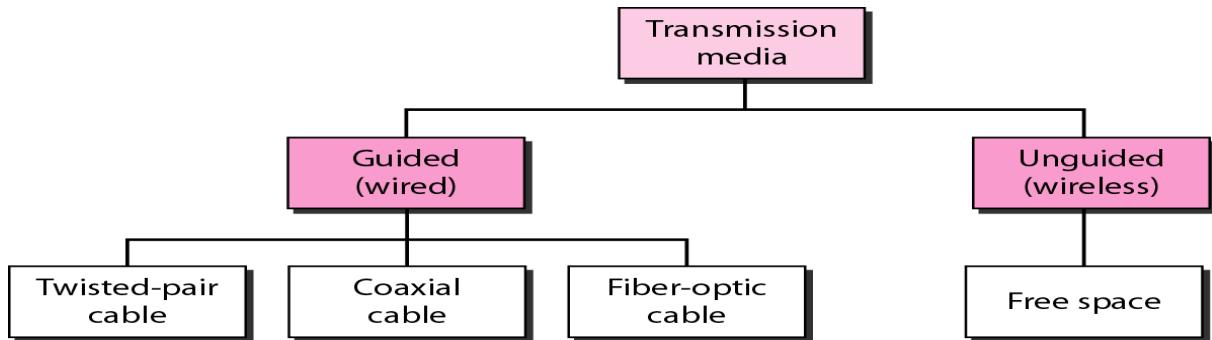
E.g.: Used for daily backup of organization or a mirror copy

---

A **transmission medium** can be broadly defined as anything that can carry information from a source to a destination.

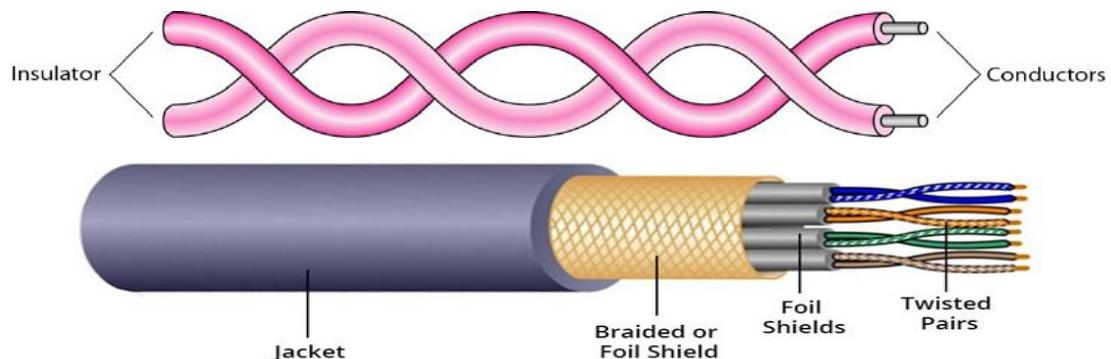


## Classes of transmission media



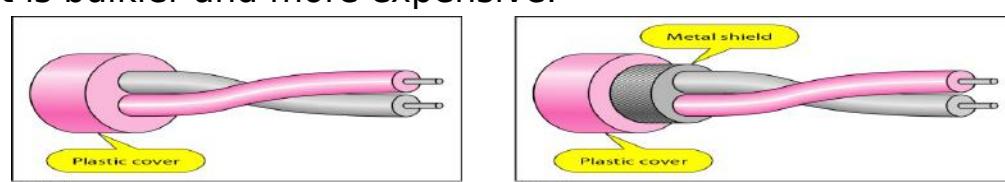
**Guided Media:** Guided media, which are those that provide a medium from one device to another, include twisted-pair cable, coaxial cable, and fiber-optic cable.

**Twisted-Pair Cable:** A twisted pair consists of two conductors (normally copper), each with its own plastic insulation, twisted together. One of the wires is used to carry signals to the receiver, and the other is used only as a ground reference.



### Unshielded Versus Shielded Twisted-Pair Cable

The most common twisted-pair cable used in communications is referred to as unshielded twisted-pair (UTP). STP cable has a metal foil or braided mesh covering that encases each pair of insulated conductors. Although metal casing improves the quality of cable by preventing the penetration of noise or crosstalk, it is bulkier and more expensive.



The most common UTP connector is RJ45 (RJ stands for registered jack)

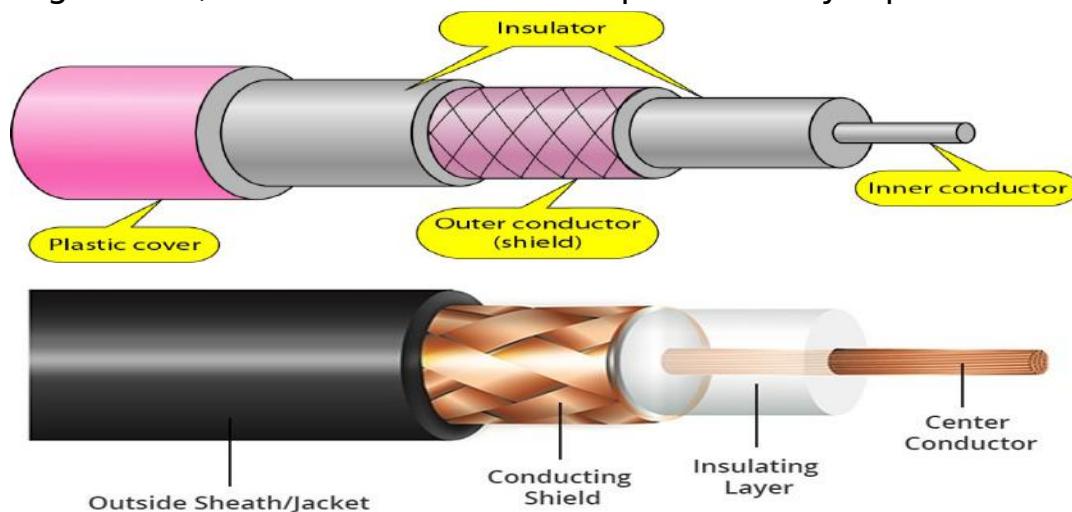
## Applications

Twisted-pair cables are used in telephone lines to provide voice and data channels.

Local-area networks, such as 10Base-T and 100Base-T, also use twisted-pair cables.

## **Coaxial Cable**

Coaxial cable (or *coax*) carries signals of higher frequency ranges than those in twisted pair cable. coax has a central core conductor of solid or stranded wire (usually copper) enclosed in an insulating sheath, which is, in turn, encased in an outer conductor of metal foil, braid, or a combination of the two. The outer metallic wrapping serves both as a shield against noise and as the second conductor, which completes the circuit. This outer conductor is also enclosed in an insulating sheath, and the whole cable is protected by a plastic cover.



The most common type of connector used today is the Bayone-Neill-Concelman (BNC), connector.

## Applications

Coaxial cable was widely used in analog telephone networks, digital telephone networks

Cable TV networks also use coaxial cables.

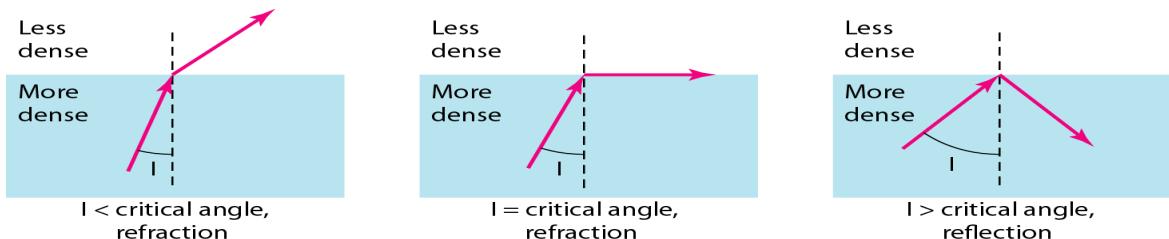
Another common application of coaxial cable is in traditional Ethernet LANs

## **Fiber-Optic Cable**

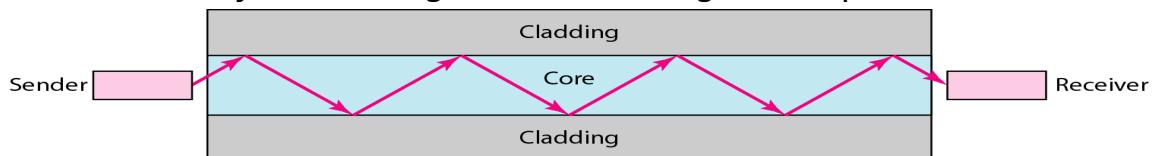
A fiber-optic cable is made of glass or plastic and transmits signals in the form of light. Light travels in a straight line as long as it is moving through a single uniform substance.

If a ray of light traveling through one substance suddenly enters another substance (of a different density), the ray changes direction.

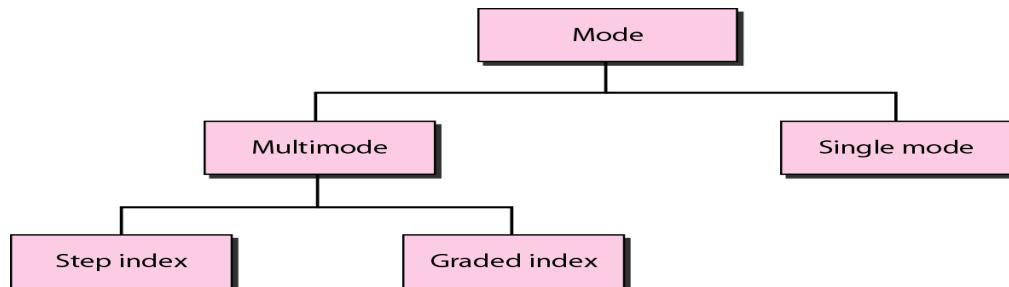
*Bending of light ray*



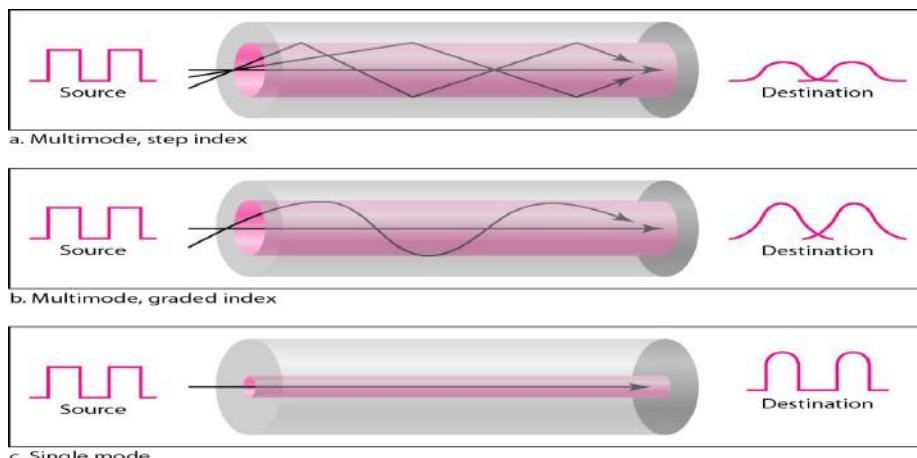
Optical fibers use reflection to guide light through a channel. A glass or plastic core is surrounded by a cladding of less dense glass or plastic.



### Propagation Modes



Multimode is so named because multiple beams from a light source move through the core in different paths. How these beams move within the cable depends on the structure of the core, as shown in Figure.

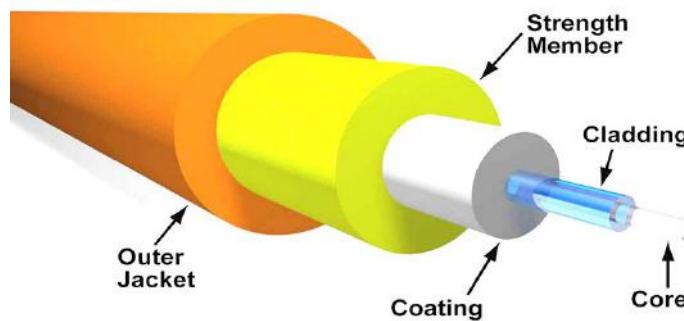


In **multimode step-index fiber**, the density of the core remains constant from the center to the edges. A beam of light moves through this constant density in a straight line until it reaches the interface of the core and the cladding. The term *step index* refers to the suddenness of this change, which contributes to the distortion of the signal as it passes through the fiber.

A second type of fiber, called **multimode graded-index fiber**, decreases this distortion of the signal through the cable. The word *index* here refers to the index of refraction.

**Single-Mode:** Single-mode uses step-index fiber and a highly focused source of light that limits beams to a small range of angles, all close to the horizontal.

## Fiber Construction



The **subscriber channel (SC) connector**, The **straight-tip (ST) connector**, **MT-RJ(mechanical transfer registered jack)** is a connector

### Applications

Fiber-optic cable is often found in backbone networks because its wide bandwidth is cost-effective..

Some cable TV companies use a combination of optical fiber and coaxial cable, thus creating a hybrid network.

Local-area networks such as 100Base-FX network (Fast Ethernet) and 1000Base-X also use fiber-optic cable

### Advantages and Disadvantages of Optical Fiber

**Advantages** Fiber-optic cable has several advantages over metallic cable (twisted pair or coaxial).

- 1 Higher bandwidth.
- 2 Less signal attenuation. Fiber-optic transmission distance is significantly greater than that of other guided media. A signal can run for 50 km without requiring regeneration. We need repeaters every 5 km for coaxial or twisted-pair cable.
- 3 Immunity to electromagnetic interference. Electromagnetic noise cannot affect fiber-optic cables.
- 4 Resistance to corrosive materials. Glass is more resistant to corrosive materials than copper.
- 5 Light weight. Fiber-optic cables are much lighter than copper cables.
- 6 Greater immunity to tapping. Fiber-optic cables are more immune to tapping than copper cables. Copper cables create antenna effects that can easily be tapped.

**Disadvantages** There are some disadvantages in the use of optical fiber.

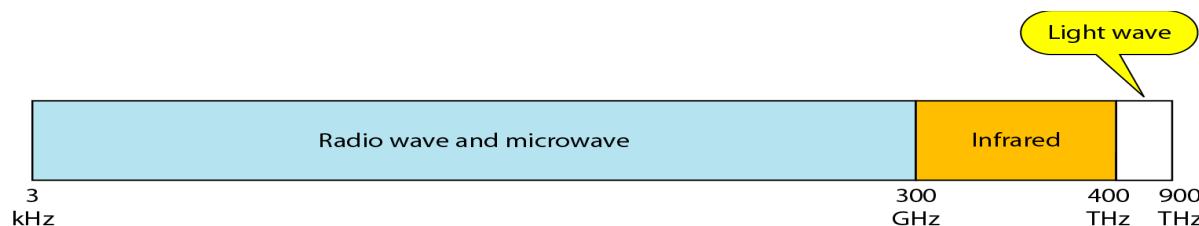
- 1 Installation and maintenance
- 2 Unidirectional light propagation. Propagation of light is unidirectional. If we need bidirectional communication, two fibers are needed.
- 3 Cost. The cable and the interfaces are relatively more expensive than those of other guided media. If the demand for bandwidth is not high, often the use of optical fiber cannot be justified.

Unguided media transport electromagnetic waves without using a physical conductor. This type of communication is often referred to as wireless communication.

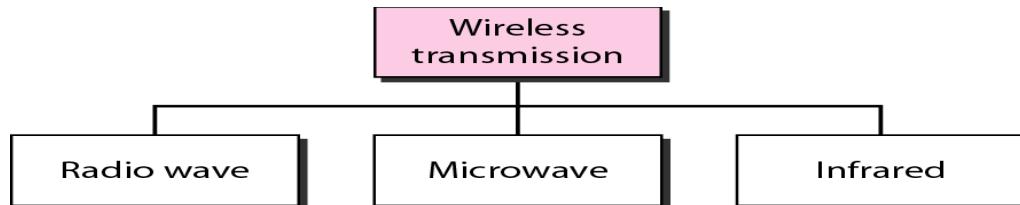
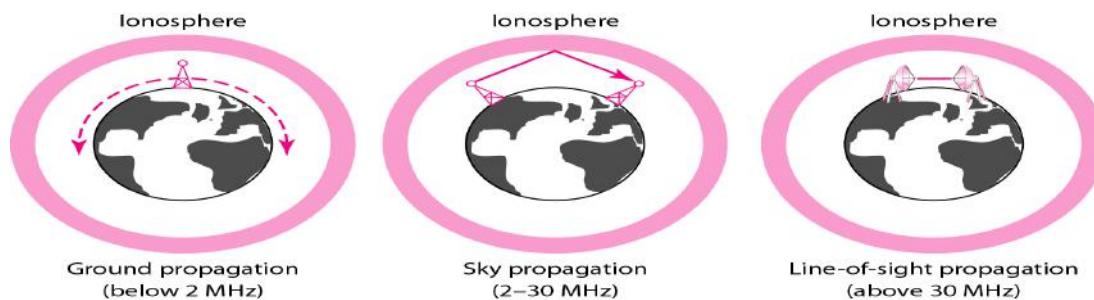
## Radio Waves

## Microwaves

## Infrared



Unguided signals can travel from the source to destination in several ways: ground propagation, sky propagation, and line-of-sight propagation, as shown in Figure

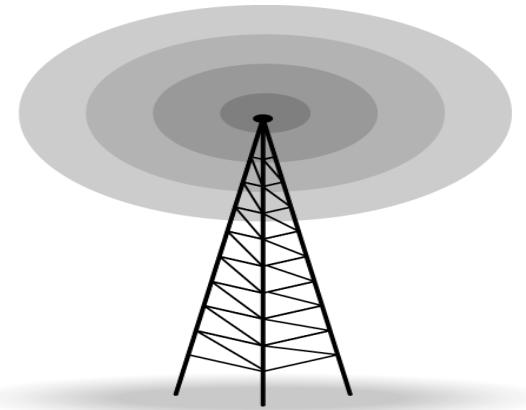


## Radio Waves

Electromagnetic waves ranging in frequencies between 3 kHz and 1 GHz are normally called radio waves. Radio waves are omni directional. When an antenna transmits radio waves, they are propagated in all directions. This means that the sending and receiving antennas do not have to be aligned. A sending antenna sends waves that can be received by any receiving antenna. The omni directional property has a disadvantage, too. The radio waves transmitted by one antenna are susceptible to interference by another antenna that may send signals using the same frequency or band.

### *Omni directional Antenna*

Radio waves use omnidirectional antennas that send out signals in all directions. Based on the wavelength, strength, and the purpose of transmission, we can have several types of antennas. Figure shows an omnidirectional antenna.



### **Applications**

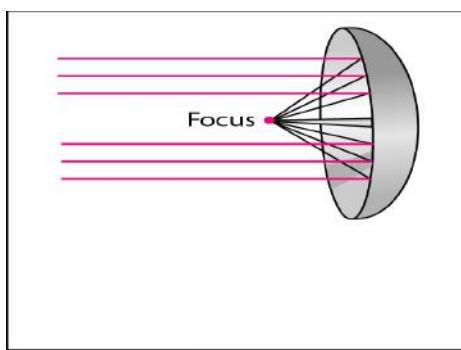
The Omni directional characteristics of radio waves make them useful for multicasting, in which there is one sender but many receivers. AM and FM radio, television, maritime radio, cordless phones, and paging are examples of multicasting.

## **Microwaves**

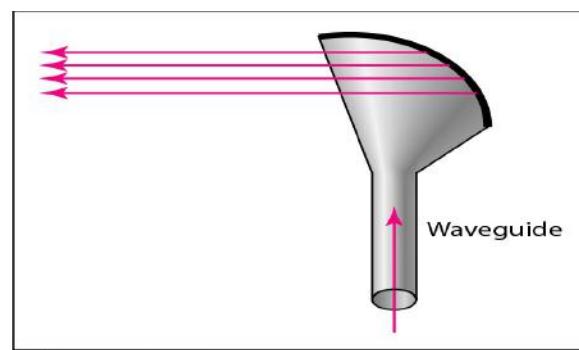
Electromagnetic waves having frequencies between 1 and 300 GHz are called microwaves. Microwaves are unidirectional. The sending and receiving antennas need to be aligned. The unidirectional property has an obvious advantage. A pair of antennas can be aligned without interfering with another pair of aligned antennas.

### **Unidirectional Antenna**

Microwaves need unidirectional antennas that send out signals in one direction. Two types of antennas are used for microwave communications: the parabolic dish and the horn



a. Dish antenna



b. Horn antenna

### **Applications:**

Microwaves are used for unicast communication such as cellular telephones, satellite networks, and wireless LANs

## **Infrared**

Infrared waves, with frequencies from 300 GHz to 400 THz (wavelengths from 1 mm to 770 nm), can be used for short-range communication. Infrared waves, having high frequencies, cannot penetrate walls. This advantageous

characteristic prevents interference between one system and another; a short-range communication system in one room cannot be affected by another system in the next room.

When we use our infrared remote control, we do not interfere with the use of the remote by our neighbors. Infrared signals useless for long-range communication. In addition, we cannot use infrared waves outside a building because the sun's rays contain infrared waves that can interfere with the communication.

### **Applications:**

**Infrared signals can be used for short-range communication in a closed area using line-of-sight propagation.**

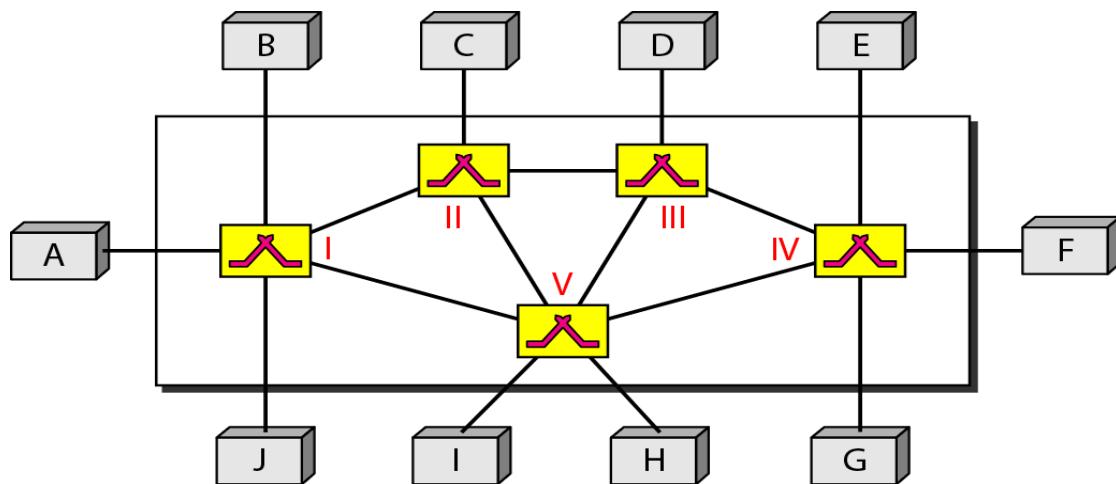
---

### **Switching**

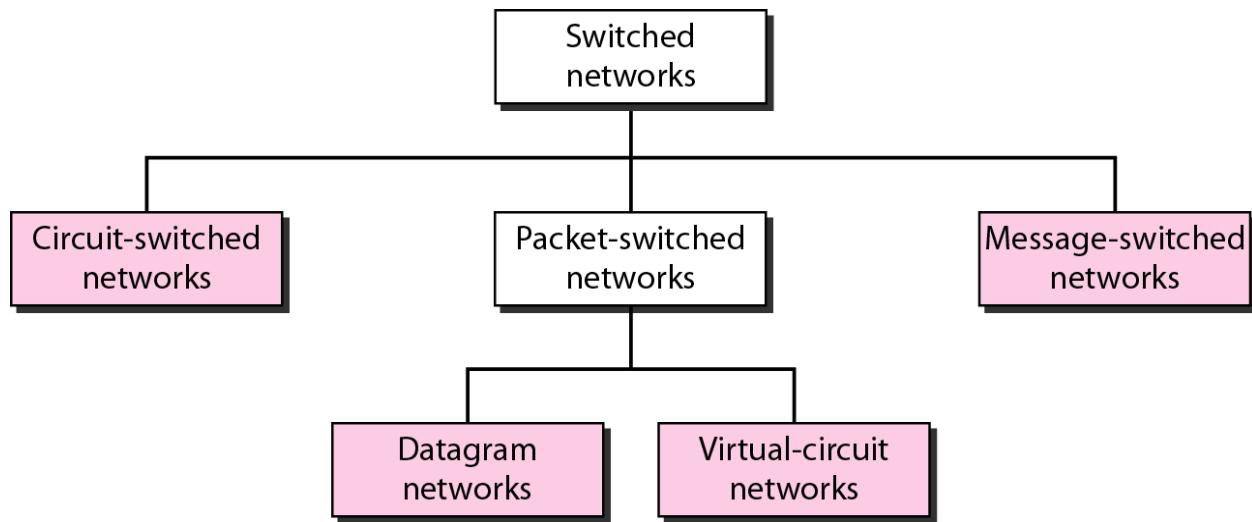
A network is a set of connected devices. Whenever we have multiple devices, we have the problem of how to connect them to make one-to-one communication possible. One solution is to make a point-to-point connection between each pair of devices (a mesh topology) or between a central device and every other device (a star topology). These methods, however, are impractical and wasteful when applied to very large networks.

The number and length of the links require too much infrastructure to be cost-efficient, and the majority of those links would be idle most of the time.

A better solution is switching. A switched network consists of a series of interlinked nodes, called switches. Switches are devices capable of creating temporary connections between two or more devices linked to the switch. In a switched network, some of these nodes are connected to the end systems (computers or telephones, for example). Others are used only for routing. Figure shows a switched network.



We can then divide today's networks into three broad categories: circuit-switched networks, packet-switched networks, and message-switched. Packet-switched networks can further be divided into two subcategories-virtual-circuit networks and datagram networks as shown in Figure.

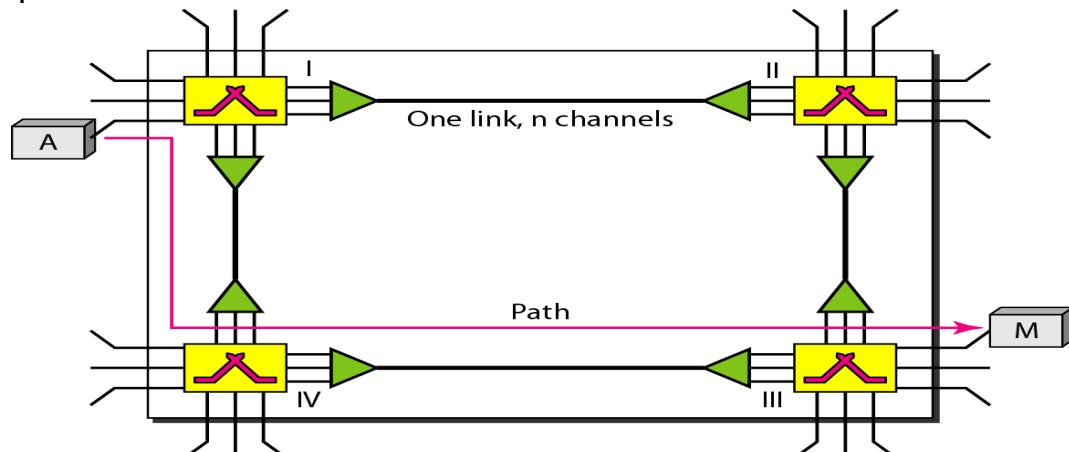


## CIRCUIT-SWITCHED NETWORKS

A circuit-switched network consists of a set of switches connected by physical links. A connection between two stations is a dedicated path made of one or more links. However, each connection uses only one dedicated channel on each link. Each link is normally divided into  $n$  channels by using FDM or TDM.

In circuit switching, the resources need to be reserved during the setup phase;

the resources remain dedicated for the entire duration of data transfer until the teardown phase



### Three Phases

The actual communication in a circuit-switched network requires three phases: connection setup, data transfer, and connection teardown.

### Setup Phase

Before the two parties (or multiple parties in a conference call) can communicate, a dedicated circuit (combination of channels in links) needs to be established. Connection setup means creating dedicated channels between the switches. For example, in Figure, when system A needs to connect to system M, it sends a setup request that includes the address of system M, to switch I. Switch I finds a channel between itself and switch IV that can be dedicated for this purpose. Switch I then sends the request to switch IV, which finds a

dedicated channel between itself and switch III. Switch III informs system M of system A's intention at this time.

In the next step to making a connection, an acknowledgment from system M needs to be sent in the opposite direction to system A. Only after system A receives this acknowledgment is the connection established.

### Data Transfer Phase

After the establishment of the dedicated circuit (channels), the two parties can transfer data.

### Teardown Phase

When one of the parties needs to disconnect, a signal is sent to each switch to release the resources.

### **Efficiency**

It can be argued that circuit-switched networks are not as efficient as the other two types of networks because resources are allocated during the entire duration of the connection. These resources are unavailable to other connections.

### **Delay**

Although a circuit-switched network normally has low efficiency, the delay in this type of network is minimal. During data transfer the data are not delayed at each switch; the resources are allocated for the duration of the connection.

The total delay is due to the time needed to create the connection, transfer data, and disconnect the circuit.

**Switching at the physical layer in the traditional telephone network uses the circuit-switching approach.**

---

## **DATAGRAM NETWORKS**

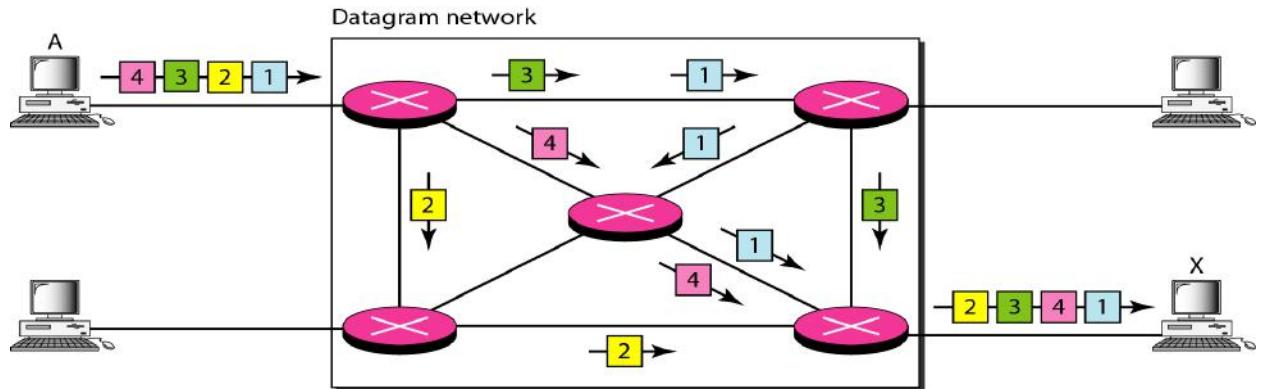
In a packet-switched network, there is no resource reservation; resources are allocated on demand. The allocation is done on a first come, first-served basis. When a switch receives a packet, no matter what is the source or destination, the packet must wait if there are other packets being processed. This lack of reservation may create delay. For example, if we do not have a reservation at a restaurant, we might have to wait.

In a datagram network, each packet is treated independently of all others. Packets in this approach are referred to as datagrams. Datagram switching is normally done at the network layer.

Figure shows how the datagram approach is used to deliver four packets from station A to station X. The switches in a datagram network are traditionally referred to as routers.

The datagram networks are sometimes referred to as connectionless networks. The term **connectionless** here means that the switch (packet switch) does not keep information about the connection state. There are no setup or teardown phases. Each packet is treated the same by a switch regardless of its source or destination.

A switch in a datagram network uses a routing table that is based on the destination address. The destination address in the header of a packet in a datagram network remains the same during the entire journey of the packet.



### Efficiency

The efficiency of a datagram network is better than that of a circuit-switched network; resources are allocated only when there are packets to be transferred.

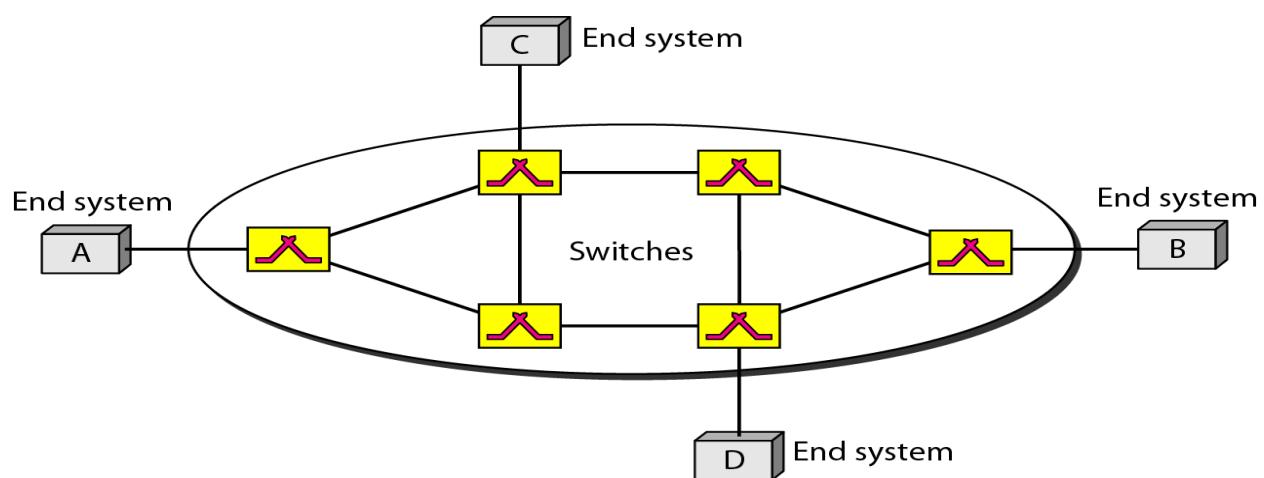
### Delay

There may be greater delay in a datagram network than in a virtual-circuit network. Although there are no setup and teardown phases, each packet may experience a wait at a switch before it is forwarded. In addition, since not all packets in a message necessarily travel through the same switches, the delay is not uniform for the packets of a message.

Switching in the Internet is done by using the datagram approach to packet switching at the network layer.

## VIRTUAL-CIRCUIT NETWORKS

**A virtual-circuit network is a cross between a circuit-switched network and a datagram network. It has some characteristics of both.**



1. As in a circuit-switched network, there are setup and teardown phases in addition to the data transfer phase.
2. Resources can be allocated during the setup phase, as in a circuit-switched network, or on demand, as in a datagram network.
3. As in a datagram network, data are packetized and each packet carries an address in the header. However, the address in the header has local jurisdiction (it defines what should be the next switch and the channel on which the packet is being carried), not end-to-end jurisdiction.
4. As in a circuit-switched network, all packets follow the same path established during the connection.
5. A virtual-circuit network is normally implemented in the data link layer, while a circuit-switched network is implemented in the physical layer and a datagram network in the network layer.

### Addressing

In a virtual-circuit network, two types of addressing are involved: global and local (virtual-circuit identifier).

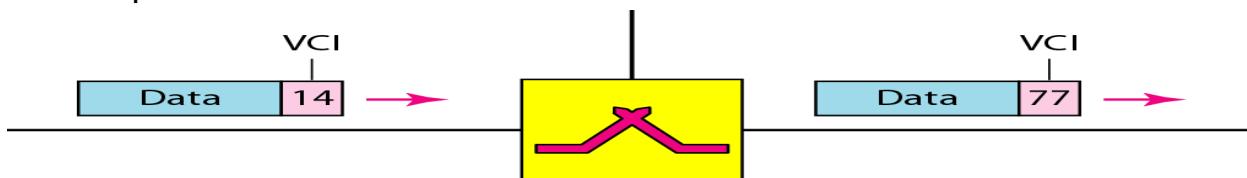
#### Global Addressing

A source or a destination needs to have a global address—an address that can be unique in the scope of the network.

#### Virtual-Circuit Identifier

The identifier that is actually used for data transfer is called the virtual-circuit identifier (VCI). A VCI, unlike a global address, is a small number that has only switch scope; it is used by a frame between two switches. When a frame arrives at a switch, it has a VCI; when it leaves, it has a different VCI.

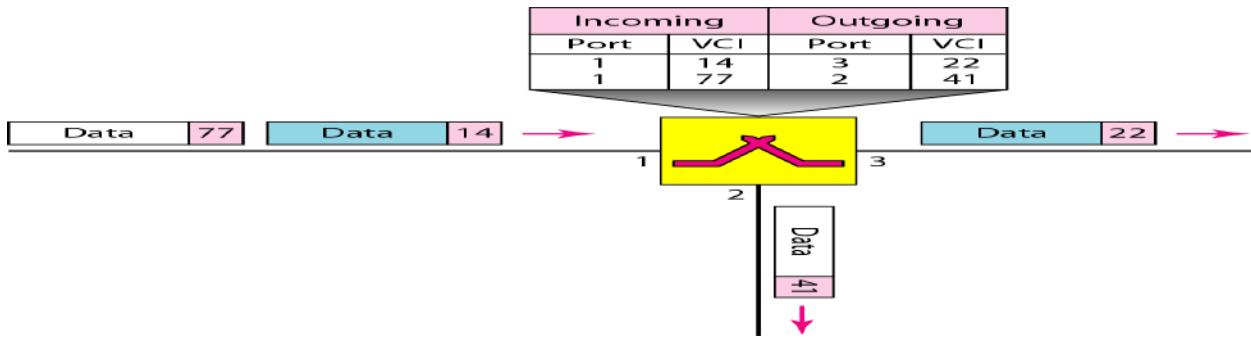
Figure shows how the VCI in a data frame changes from one switch to another. Note that a VCI does not need to be a large number since each switch can use its own unique set of VCIs.



### **Three Phases**

Three phases in a virtual-circuit network: setup, data transfer, and teardown. We first discuss the data transfer phase, which is more straightforward; we then talk about the setup and teardown phases.

### **Data Transfer Phase**

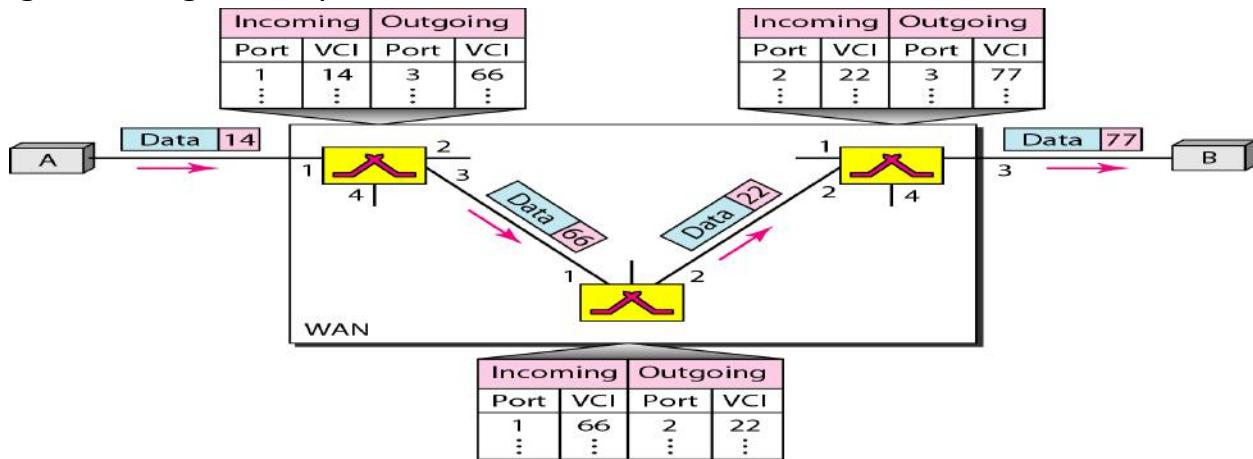


To transfer a frame from a source to its destination, all switches need to have a table entry for this virtual circuit. The table, in its simplest form, has four columns.

We show later how the switches make their table entries, but for the moment we assume that each switch has a table with entries for all active virtual circuits. Figure shows such a switch and its corresponding table.

Figure shows a frame arriving at port 1 with a VCI of 14. When the frame arrives, the switch looks in its table to find port 1 and a VCI of 14. When it is found, the switch knows to change the VCI to 22 and send out the frame from port 3.

Figure shows how a frame from source A reaches destination B and how its VCI changes during the trip.



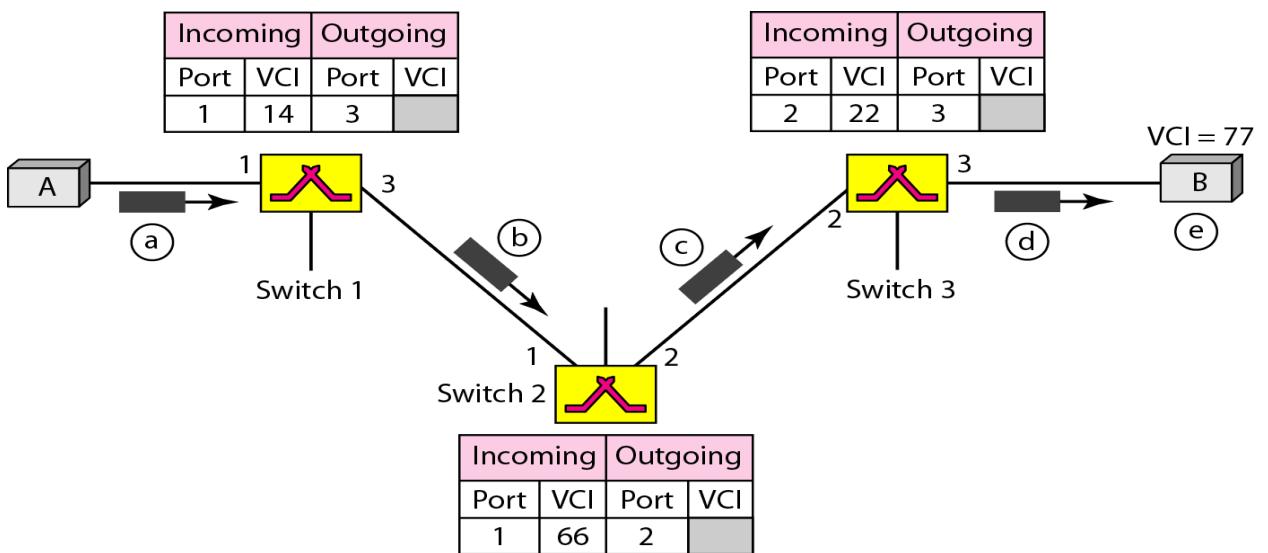
Each switch changes the VCI and routes the frame.

The data transfer phase is active until the source sends all its frames to the destination. The procedure at the switch is the same for each frame of a message. The process creates a virtual circuit, not a real circuit, between the source and destination.

## Setup Phase

In the setup phase, a switch creates an entry for a virtual circuit. For example, suppose source A needs to create a virtual circuit to B. Two steps are required: the setup request and the acknowledgment.

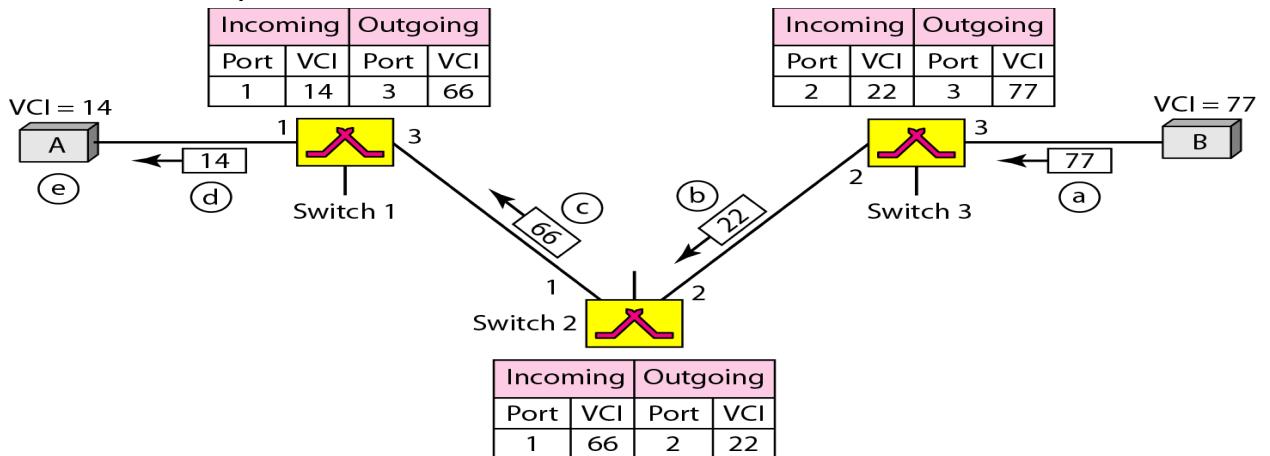
**Setup Request** A setup request frame is sent from the source to the destination. Figure shows the process.



- Source A sends a setup frame to switch 1.
- Switch 1 receives the setup request frame. It knows that a frame going from A to B goes out through port 3. For the moment, assume that it knows the output port. The switch creates an entry in its table for this virtual circuit, but it is only able to fill three of the four columns. The switch assigns the incoming port (1) and chooses an available incoming VCI (14) and the outgoing port (3). It does not yet know the outgoing VCI, which will be found during the acknowledgment step. The switch then forwards the frame through port 3 to switch 2.
- Switch 2 receives the setup request frame. The same events happen here as at switch 1; three columns of the table are completed: in this case, incoming port (1), incoming VCI (66), and outgoing port (2).
- Switch 3 receives the setup request frame. Again, three columns are completed: incoming port (2), incoming VCI (22), and outgoing port (3).
- Destination B receives the setup frame, and if it is ready to receive frames from A, it assigns a VCI to the incoming frames that come from A, in this case 77. This VCI lets the destination know that the frames come from A, and not other sources.

**Acknowledgment** A special frame, called the acknowledgment frame, completes the entries in the switching tables.

Figure shows the process.



- a. The destination sends an acknowledgment to switch 3. The acknowledgment carries the global source and destination addresses so the switch knows which entry in the table is to be completed. The frame also carries VCI 77, chosen by the destination as the incoming VCI for frames from A. Switch 3 uses this VCI to complete the outgoing VCI column for this entry. Note that 77 is the incoming VCI for destination B, but the outgoing VCI for switch 3.
- b. Switch 3 sends an acknowledgment to switch 2 that contains its incoming VCI in the table, chosen in the previous step. Switch 2 uses this as the outgoing VCI in the table.
- c. Switch 2 sends an acknowledgment to switch 1 that contains its incoming VCI in the table, chosen in the previous step. Switch 1 uses this as the outgoing VCI in the table.
- d. Finally switch 1 sends an acknowledgment to source A that contains its incoming VCI in the table, chosen in the previous step.
- e. The source uses this as the outgoing VCI for the data frames to be sent to destination B.

### ***Teardown Phase***

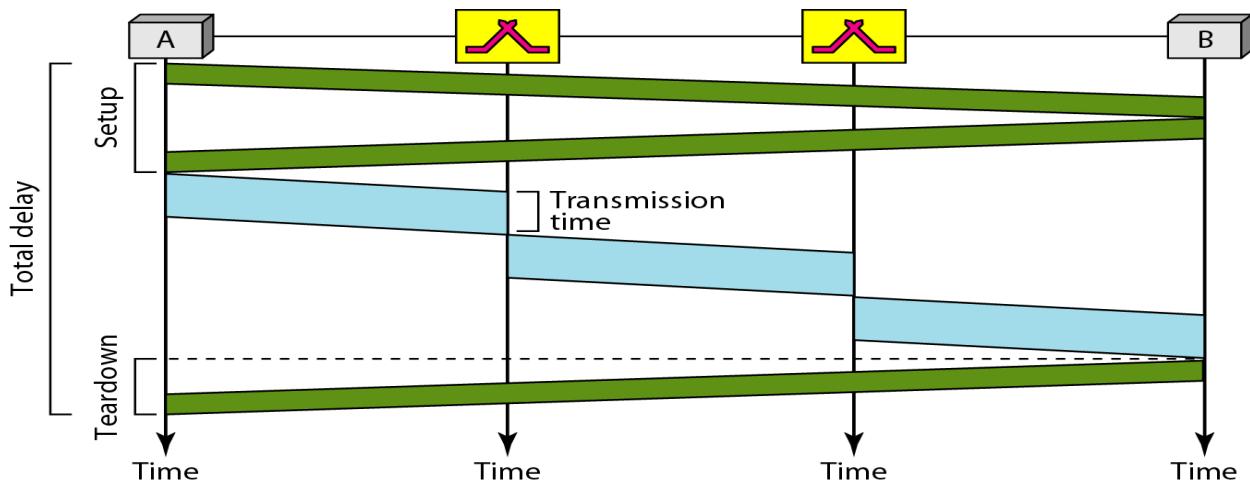
In this phase, source A, after sending all frames to B, sends a special frame called a *teardown request*. Destination B responds with a teardown confirmation frame. All switches delete the corresponding entry from their tables.

### ***Efficiency***

In virtual-circuit switching, all packets belonging to the same source and destination travel the same path; but the packets may arrive at the destination with different delays if resource allocation is on demand.

### ***Delay***

In a virtual-circuit network, there is a one-time delay for setup and a one-time delay for teardown. If resources are allocated during the setup phase, there is no wait time for individual packets. Figure shows the delay for a packet traveling through two switches in a virtual-circuit network



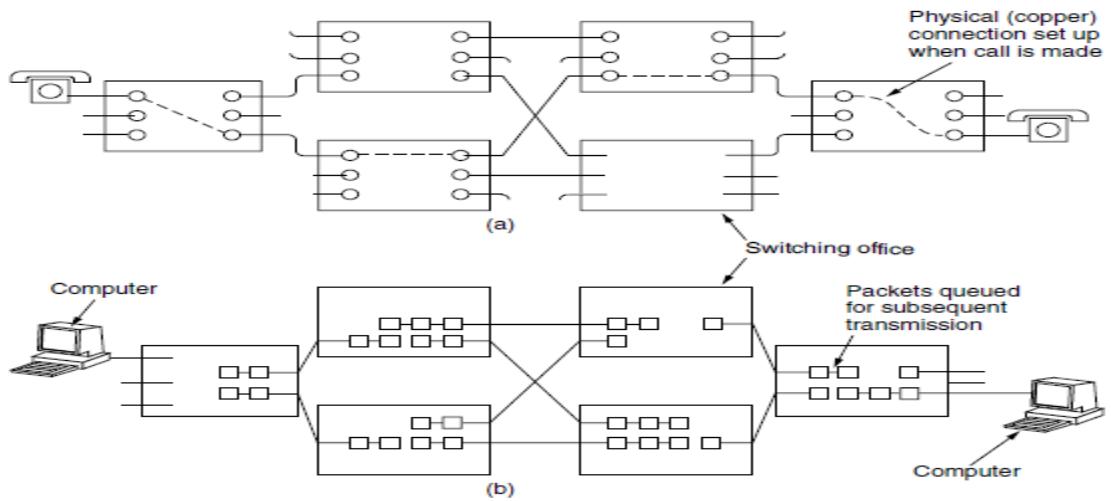
Switching at the data link layer in a switched WAN is normally implemented by using virtual-circuit techniques.

---

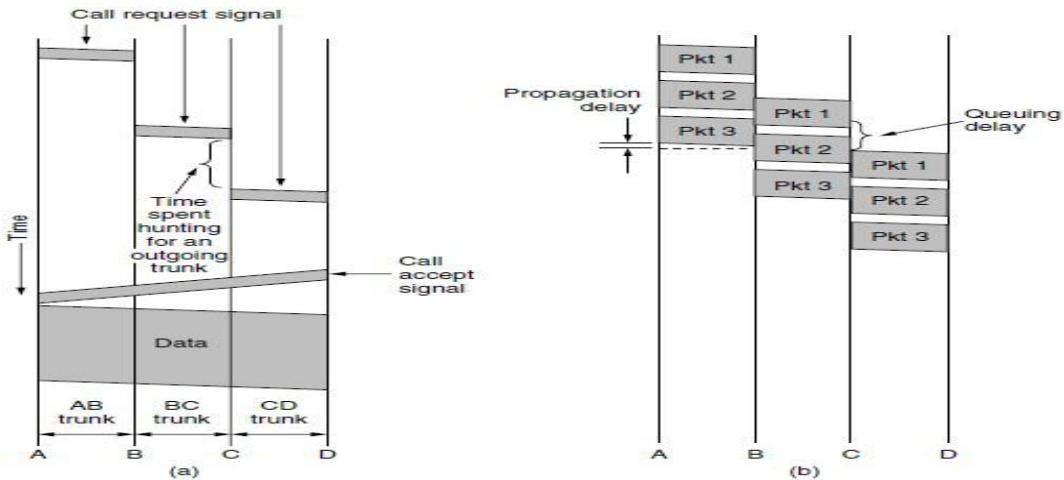
## Comparison

Issue	Datagram network	Virtual-circuit network
Circuit setup	Not needed	Required
Addressing	Each packet contains the full source and destination address	Each packet contains a short VC number
State information	Routers do not hold state information about connections	Each VC requires router table space per connection
Routing	Each packet is routed independently	Route chosen when VC is set up; all packets follow it
Effect of router failures	None, except for packets lost during the crash	All VCs that passed through the failed router are terminated
Quality of service	Difficult	Easy if enough resources can be allocated in advance for each VC
Congestion control	Difficult	Easy if enough resources can be allocated in advance for each VC

## Diagrams from Tanenbaum Textbook



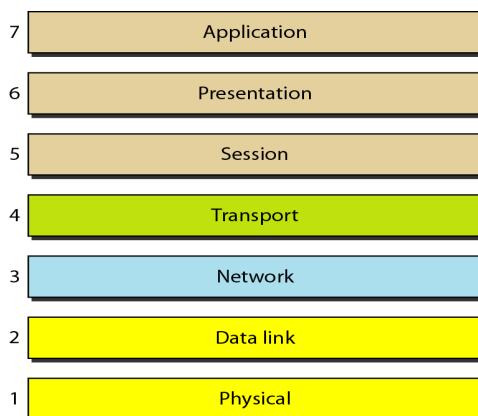
**Figure 2-42.** (a) Circuit switching. (b) Packet switching.



**Figure 2-43.** Timing of events in (a) circuit switching. (b) packet switching.

## OSI

- OSI stands for Open Systems Interconnection
- Created by International Standards Organization (ISO)
- Was created as a framework and reference model to explain how different networking technologies work together and interact
- It is not a standard that networking protocols must follow
- Each layer has specific functions it is responsible for
- All layers work together in the correct order to move data around a network

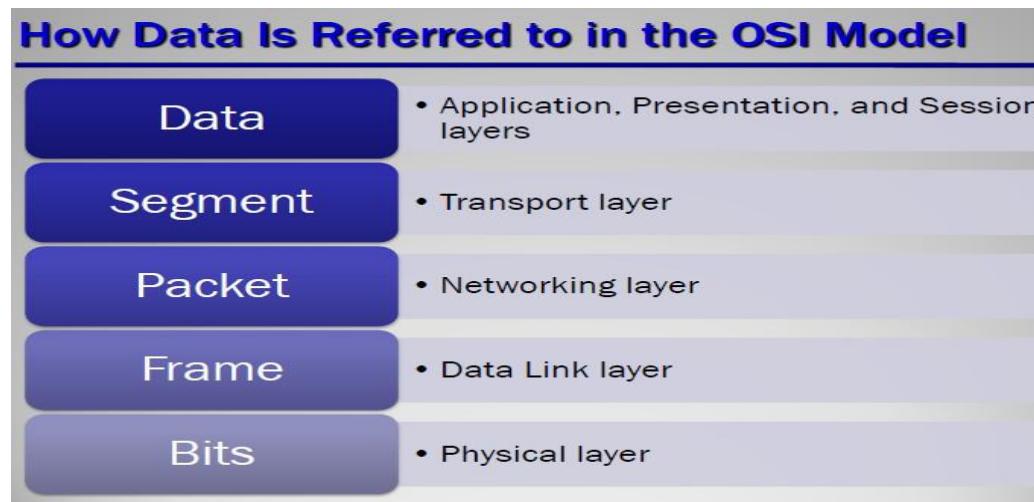


Top to bottom

-All People Seem To Need Data Processing

Bottom to top

-Please Do Not Throw Sausage Pizza Away



## **Physical Layer**

- Deals with all aspects of physically moving data from one computer to the next
- Converts data from the upper layers into 1s and 0s for transmission over media
- Defines how data is encoded onto the media to transmit the data
- Defined on this layer: Cable standards, wireless standards, and fiber optic standards.

Copper wiring, fiber optic cable, radio frequencies, anything that can be used to transmit data is defined on the Physical layer of the OSI Model

- Device example: Hub
- Used to transmit data

## **Data Link Layer**

- Is responsible for moving frames from node to node or computer to computer
- Can move frames from one adjacent computer to another, cannot move frames across routers
- Encapsulation = frame
- Requires MAC address or *physical address*
- Protocols defined include Ethernet Protocol and Point-to-Point Protocol (PPP)
- Device example: Switch
- Two sublayers: Logical Link Control (LLC) and the Media Access Control (MAC)
  - Logical Link Control (LLC)
  - -Data Link layer addressing, flow control, address notification, error control
  - Media Access Control (MAC)
  - -Determines which computer has access to the network media at any given time
  - -Determines where one frame ends and the next one starts, called frame synchronization

## **Network Layer**

- Responsible for moving packets (data) from one end of the network to the other, called *end-to-end communications*
- Requires *logical addresses* such as IP addresses
- Device example: Router
- -Routing is the ability of various network devices and their related software to move data packets from source to destination

### **Transport Layer**

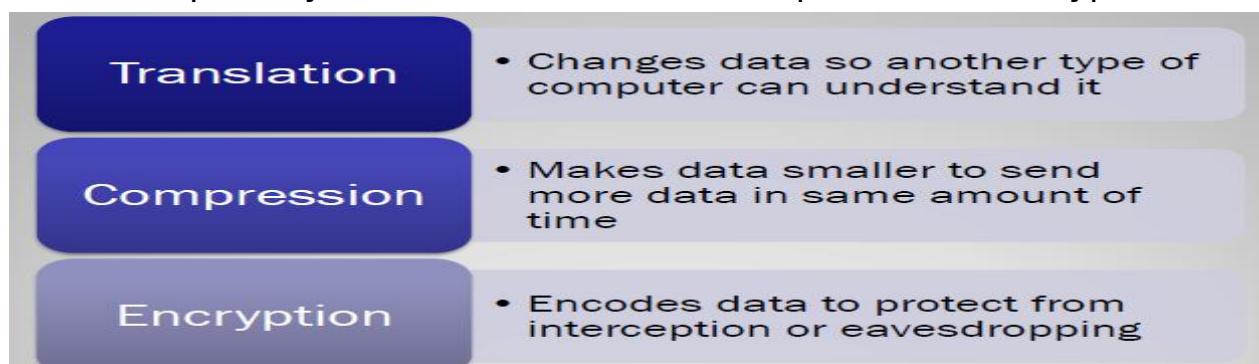
- Takes data from higher levels of OSI Model and breaks it into segments that can be sent to lower-level layers for data transmission
- Conversely, reassembles data segments into data that higher-level protocols and applications can use
- Also puts segments in correct order (called sequencing ) so they can be reassembled in correct order at destination
- Concerned with the reliability of the transport of sent data
- May use a *connection-oriented protocol* such as TCP to ensure destination received segments
- May use a *connectionless protocol* such as UDP to send segments without assurance of delivery
- Uses port addressing

### **Session Layer**

- Responsible for managing the dialog between networked devices
- Establishes, manages, and terminates connections
- Provides duplex, half-duplex, or simplex communications between devices
- Provides procedures for establishing checkpoints, adjournment, termination, and restart or recovery procedures

### **Presentation Layer**

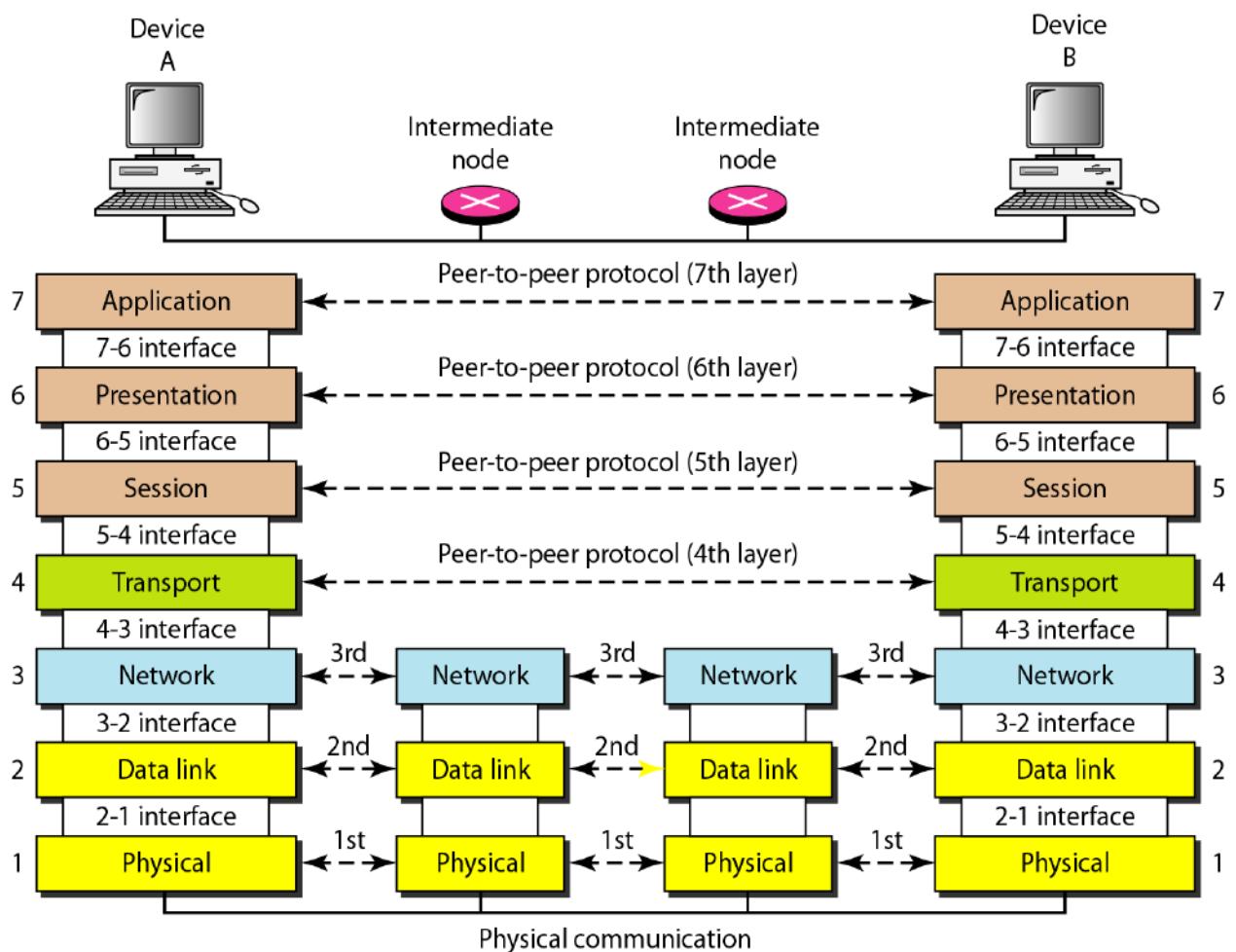
- Concerned with how data is presented to the network
- Handles three primary tasks: -Translation , -Compression , -Encryption



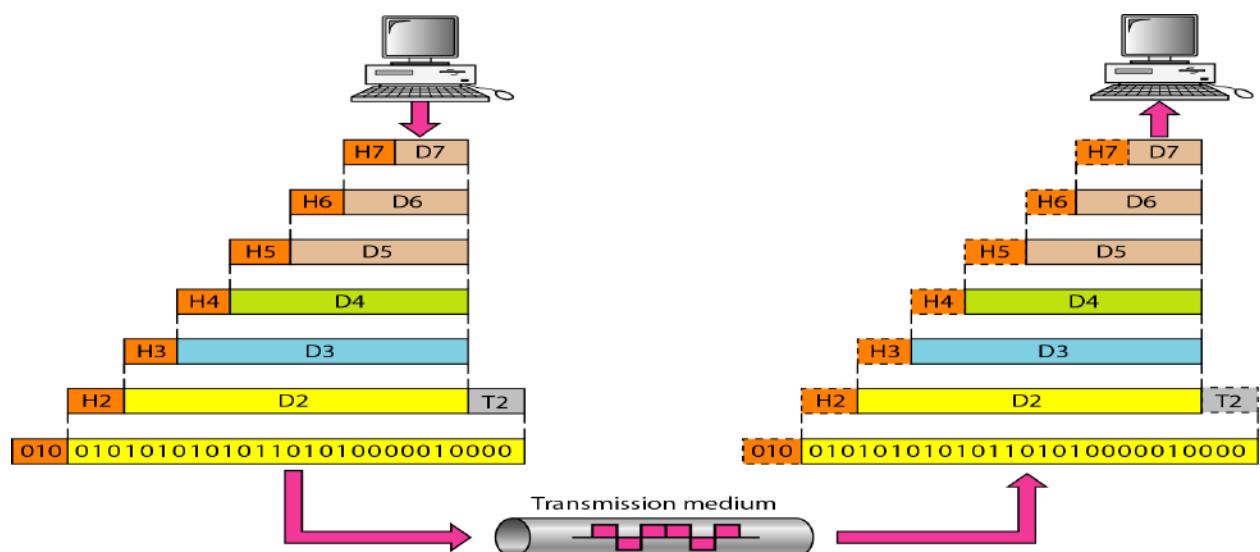
### **Application Layer**

- Contains all services or protocols needed by application software or operating system to communicate on the network
- Examples
  - -Firefox web browser uses HTTP (Hyper-Text Transport Protocol)
  - -E-mail program may use POP3 (Post Office Protocol version 3) to read e-mails and SMTP (Simple Mail Transport Protocol) to send e-mails

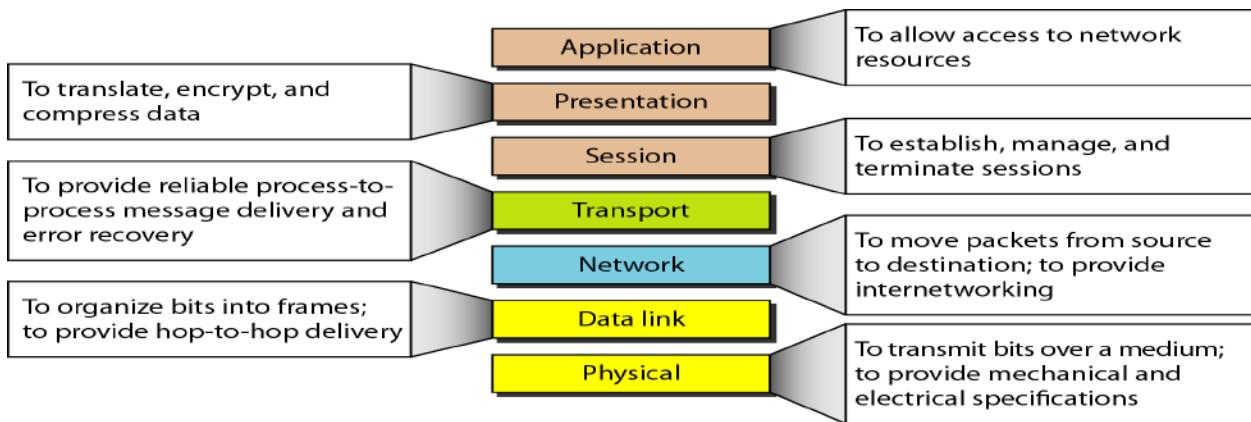
## The interaction between layers in the OSI model



## An exchange using the OSI model

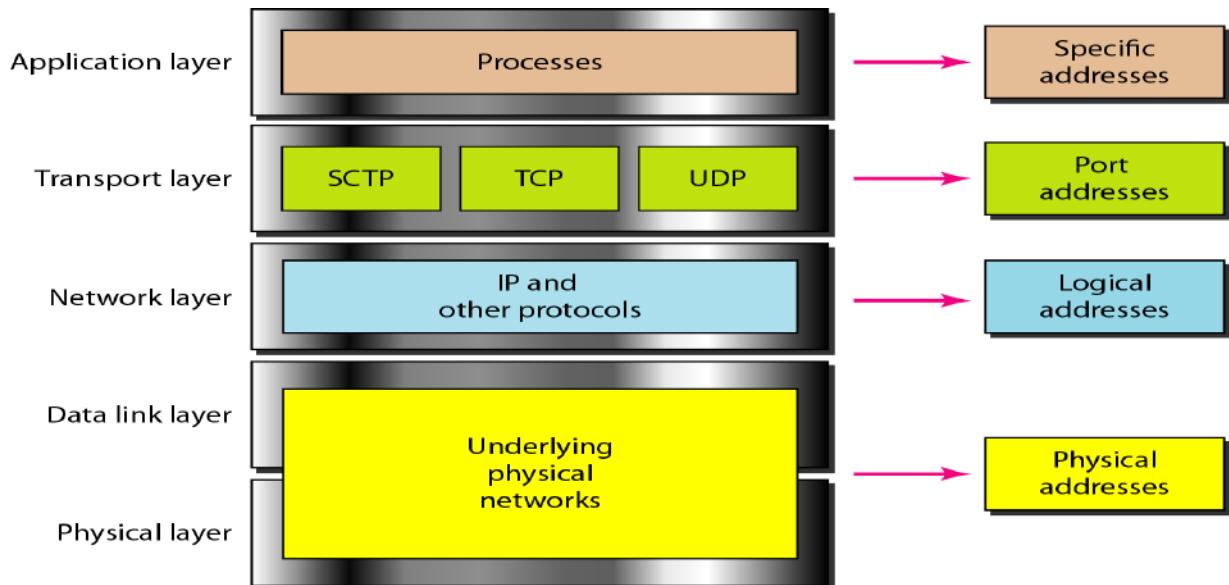


**SUMMARY:**



## **TCP/IP Model (Transmission Control Protocol/Internet Protocol)**

-A *protocol suite* is a large number of related protocols that work together to allow networked computers to communicate



***Relationship of layers and addresses in TCP/IP***

### **Application Layer**

- Application layer protocols define the rules when implementing specific network applications
- Rely on the underlying layers to provide accurate and efficient data delivery
- Typical protocols:
  - FTP – File Transfer Protocol
  - For file transfer
  - Telnet – Remote terminal protocol
  - For remote login on any other computer on the network
  - SMTP – Simple Mail Transfer Protocol
  - For mail transfer
  - HTTP – Hypertext Transfer Protocol
  - For Web browsing
- Encompasses same functions as these OSI Model layers Application Presentation Session

### **Transport Layer**

#### **TCP & UDP**

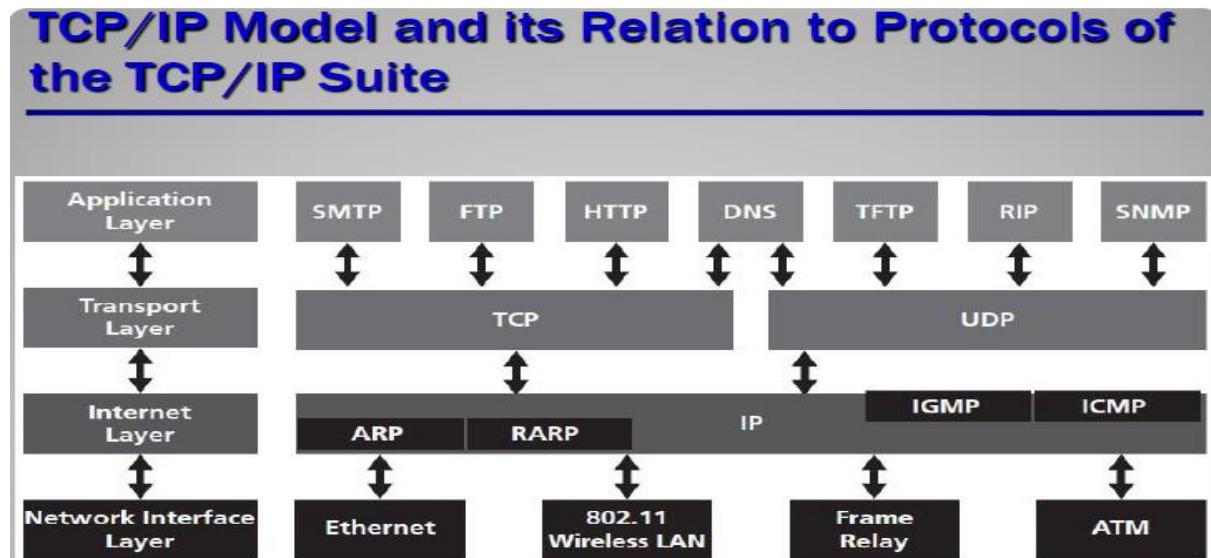
- TCP is a connection-oriented protocol
- Does not mean it has a physical connection between sender and receiver
- TCP provides the function to allow a connection virtually exists - also called virtual circuit
- UDP provides the functions:
  - Dividing a chunk of data into segments
  - Reassembly segments into the original chunk
  - Provide further the functions such as reordering and data resend
- Offering a reliable byte-stream delivery service
- Functions the same as the Transport layer in OSI
- Synchronize source and destination computers to set up the session between the respective computers

### **Internet Layer**

- The network layer, also called the internet layer, deals with packets and connects independent networks to transport the packets across network boundaries. The network layer protocols are the IP and the Internet Control Message Protocol ([ICMP](#)), which is used for error reporting.

### **Host-to-network layer**

The **Host-to-network layer** is the lowest **layer** of the **TCP/IP** reference model. It combines the link **layer** and the physical **layer** of the ISO/OSI model. At this **layer**, data is transferred between adjacent **network** nodes in a WAN or between nodes on the same LAN.



<b>OSI MODEL</b>	<b>TCP/IP MODEL</b>
Contains 7 Layers	Contains 4 Layers
Uses Strict Layering resulting in vertical layers.	Uses Loose Layering resulting in horizontal layers.
Supports both connectionless & connection-oriented communication in the Network layer, but only connection-oriented communication in Transport Layer	Supports only connectionless communication in the Network layer, but both connectionless & connection-oriented communication in Transport Layer
It distinguishes between Service, Interface and Protocol.	Does not clearly distinguish between Service, Interface and Protocol.
Protocols are better hidden and can be replaced relatively easily as technology changes (No transparency)	Protocols are not hidden and thus cannot be replaced easily. (Transparency) Replacing IP by a substantially different protocol would be virtually impossible
OSI reference model was devised before the corresponding protocols were designed.	The protocols came first and the model was a description of the existing protocols

## THE INTERNET

The Internet has revolutionized many aspects of our daily lives. It has affected the way we do business as well as the way we spend our leisure time. Count the ways you've used the Internet recently. Perhaps you've sent electronic mail (e-mail) to a business associate, paid a utility bill, read a newspaper from a distant city, or looked up a local movie schedule-all by using the Internet. Or maybe you researched a medical topic, booked a hotel reservation, chatted with a fellow Trekkie, or comparison-shopped for a car. The Internet is a communication system that has brought a wealth of information to our fingertips and organized it for our use.

### A Brief History

A network is a group of connected communicating devices such as computers and printers. An internet (note the lowercase letter i) is two or more networks that can communicate with each other. The most notable internet is called the Internet (uppercase letter I), a collaboration of more than hundreds of thousands of interconnected networks. Private individuals as well as various organizations such as government agencies, schools, research facilities, corporations, and libraries in more than 100 countries use the Internet. Millions of people are users. Yet this extraordinary communication system only came into being in 1969.

In the mid-1960s, mainframe computers in research organizations were standalone devices. Computers from different manufacturers were unable to communicate with one another. The Advanced Research Projects Agency

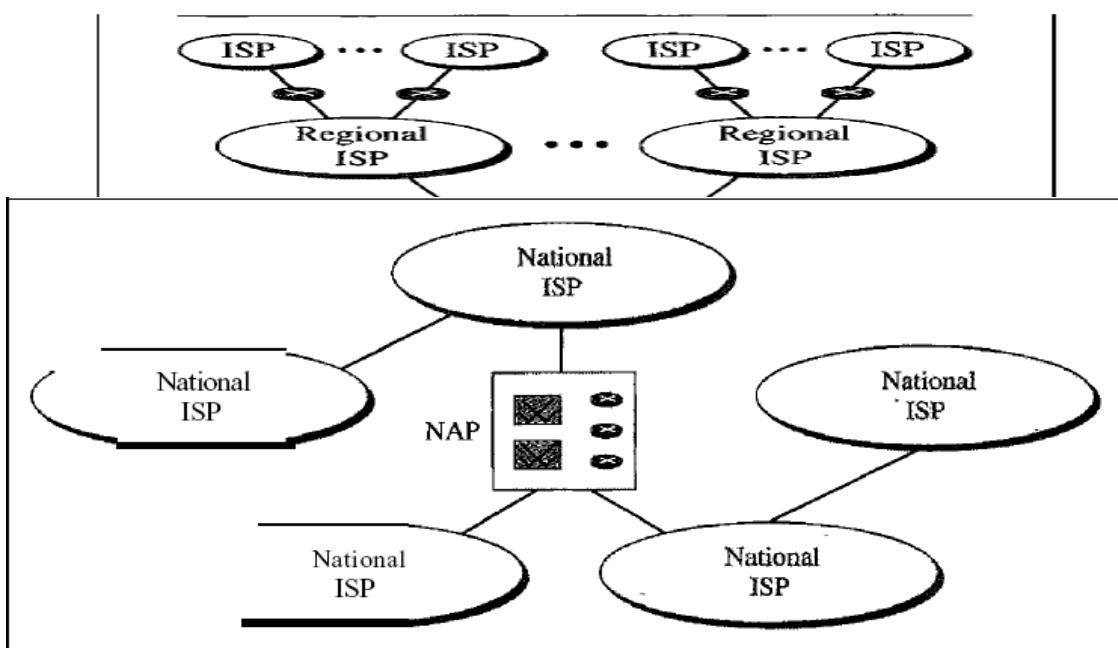
(ARPA) in the Department of Defense (DoD) was interested in finding a way to connect computers so that the researchers they funded could share their findings, thereby reducing costs and eliminating duplication of effort.

In 1967, at an Association for Computing Machinery (ACM) meeting, ARPA presented its ideas for ARPANET, a small network of connected computers. The idea was that each host computer (not necessarily from the same manufacturer) would be attached to a specialized computer, called an *interface message processor* (IMP). The IMPs, in turn, would be connected to one another. Each IMP had to be able to communicate with other IMPs as well as with its own attached host. By 1969, ARPANET was a reality. Four nodes, at the University of California at Los Angeles (UCLA), the University of California at Santa Barbara (UCSB), Stanford Research Institute (SRI), and the University of Utah, were connected via the IMPs to form a network. Software called the *Network Control Protocol* (NCP) provided communication between the hosts.

In 1972, Vint Cerf and Bob Kahn, both of whom were part of the core ARPANET group, collaborated on what they called the *Internetting Project*. Cerf and Kahn's landmark 1973 paper outlined the protocols to achieve end-to-end delivery of packets. This paper on Transmission Control Protocol (TCP) included concepts such as encapsulation, the datagram, and the functions of a gateway. Shortly thereafter, authorities made a decision to split TCP into two protocols: Transmission Control Protocol (TCP) and Internetworking Protocol (IP). IP would handle datagram routing while TCP would be responsible for higher-level functions such as segmentation, reassembly, and error detection. The internetworking protocol became known as TCPIIP.

### The Internet Today

The Internet has come a long way since the 1960s. The Internet today is not a simple hierarchical structure. It is made up of many wide- and local-area networks joined by connecting devices and switching stations. It is difficult to give an accurate representation of the Internet because it is continually changing-new networks are being added, existing networks are adding addresses, and networks of defunct companies are being removed. Today most end users who want Internet connection use the services of Internet service providers (ISPs). There are international service providers, national service providers, regional service providers, and local service providers. The Internet today is run by private companies, not the government. Figure 1.13 shows a conceptual (not geographic) view of the Internet.



b. Interconnection of national ISPs

### **International Internet Service Providers:**

At the top of the hierarchy are the international service providers that connect nations together.

### **National Internet Service Providers:**

The national Internet service providers are backbone networks created and maintained by specialized companies. There are many national ISPs operating in North America; some of the most well known are SprintLink, PSINet, UUNet Technology, AGIS, and internet Mel. To provide connectivity between the end users, these backbone networks are connected by complex switching stations (normally run by a third party) called network access points (NAPs). Some national ISP networks are also connected to one another by private switching stations called *peering points*. These normally operate at a high data rate (up to 600 Mbps).

### **Regional Internet Service Providers:**

Regional internet service providers or regional ISPs are smaller ISPs that are connected to one or more national ISPs. They are at the third level of the hierarchy with a smaller data rate. **Local Internet Service Providers:**

Local Internet service providers provide direct service to the end users. The local ISPs can be connected to regional ISPs or directly to national ISPs. Most end users are connected to the local ISPs. Note that in this sense, a local ISP can be a company that just provides Internet services, a corporation with a network that supplies services to its own employees, or a nonprofit organization, such as a college or a university, that runs its own network. Each of these local ISPs can be connected to a regional or national service provider.

# **UNIT- II**

## **DATA LINK LAYER FUNCTIONS (SERVICES)**

### **1. Providing services to the network layer:**

#### **1 Unacknowledged connectionless service.**

Appropriate for low error rate and real-time traffic. Ex: Ethernet

#### **2. Acknowledged connectionless service.**

Useful in unreliable channels, WiFi. Ack/Timer/Resend

#### **3. Acknowledged connection-oriented service.**

Guarantee frames are received exactly once and in the right order.  
Appropriate over long, unreliable links such as a satellite channel or a long-distance telephone circuit

**2. Framing:** Frames are the streams of bits received from the network layer into manageable data units. This division of stream of bits is done by Data Link Layer.

**3. Physical Addressing:** The Data Link layer adds a header to the frame in order to define physical address of the sender or receiver of the frame, if the frames are to be distributed to different systems on the network.

**4. Flow Control:** A receiving node can receive the frames at a faster rate than it can process the frame. Without flow control, the receiver's buffer can overflow, and frames can get lost. To overcome this problem, the data link layer uses the flow control to prevent the sending node on one side of the link from overwhelming the receiving node on another side of the link. This prevents traffic jam at the receiver side.

**5. Error Control:** Error control is achieved by adding a trailer at the end of the frame. Duplication of frames are also prevented by using this mechanism. Data Link Layers adds mechanism to prevent duplication of frames.

**Error detection:** Errors can be introduced by signal attenuation and noise. Data Link Layer protocol provides a mechanism to detect one or more errors. This is achieved by adding error detection bits in the frame and then receiving node can perform an error check.

**Error correction:** Error correction is similar to the Error detection, except that receiving node not only detects the errors but also determine where the errors have occurred in the frame.

**6. Access Control:** Protocols of this layer determine which of the devices has control over the link at any given time, when two or more devices are connected to the same link.

**7. Reliable delivery:** Data Link Layer provides a reliable delivery service, i.e., transmits the network layer datagram without any error. A reliable delivery service is accomplished with transmissions and acknowledgements. A data link layer mainly provides the reliable delivery

service over the links as they have higher error rates and they can be corrected locally, link at which an error occurs rather than forcing to retransmit the data.

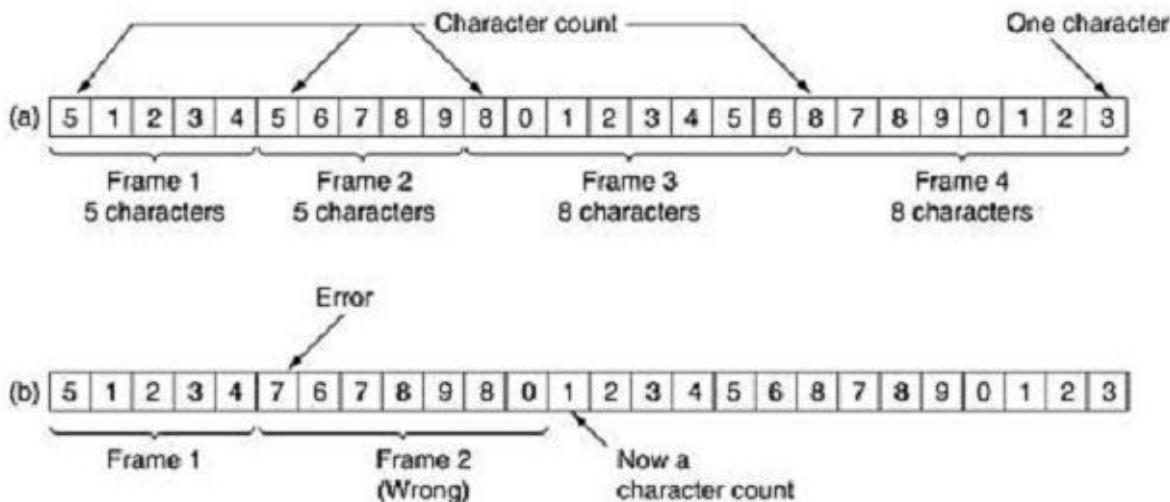
8. **Half-Duplex & Full-Duplex:** In a Full-Duplex mode, both the nodes can transmit the data at the same time. In a Half-Duplex mode, only one node can transmit the data at the same time.

## FRAMING:

To provide service to the network layer, the data link layer must use the service provided to it by the physical layer. What the physical layer does is accept a raw bit stream and attempt to deliver it to the destination. This bit stream is not guaranteed to be error free. The number of bits received may be less than, equal to, or more than the number of bits transmitted, and they may have different values. It is up to the data link layer to **detect and, if necessary, correct errors**. The usual approach is for the data link layer to break the bit stream up into discrete frames and compute the checksum for each frame (**framing**). When a frame arrives at the destination, the checksum is recomputed. If the newly computed checksum is different from the one contained in the frame, the data link layer knows that an error has occurred and takes steps to deal with it (e.g., discarding the bad frame and possibly also sending back an error report). We will look at four framing methods:

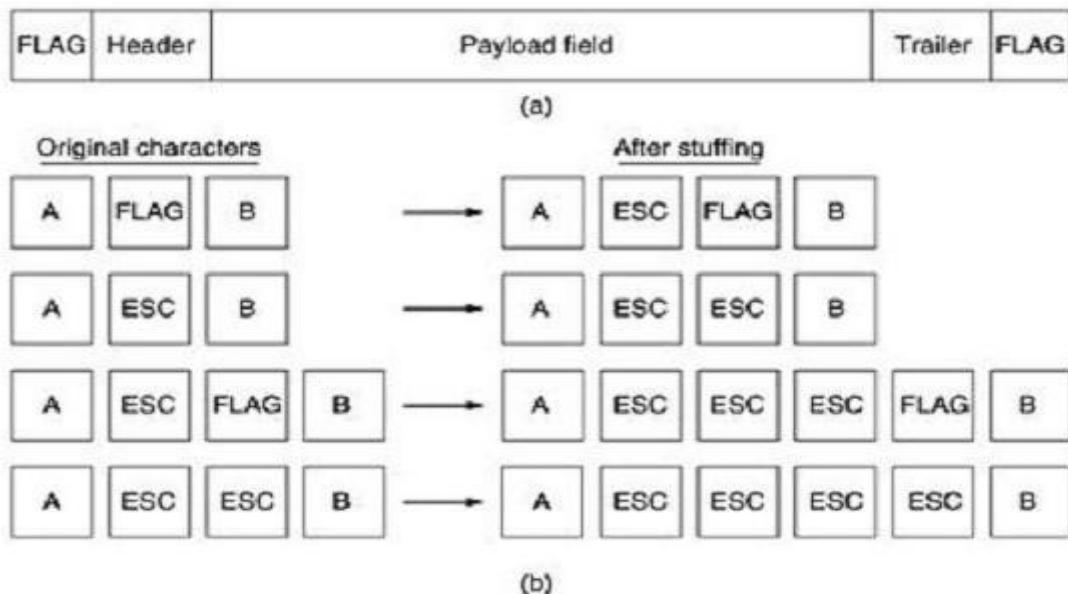
1. Character count.
2. Flag bytes with byte stuffing.
3. Starting and ending flags, with bit stuffing.
4. Physical layer coding violations.

**Character count** method uses a field in the header to specify the number of characters in the frame. When the data link layer at the destination sees the character count, it knows how many characters follow and hence where the end of the frame is. This technique is shown in Fig. (a) For four frames of sizes 5, 5, 8, and 8 characters, respectively.



The trouble with this algorithm is that the count can be garbled by a transmission error. For example, if the character count of 5 in the second frame of Fig. (b) becomes a 7, the destination will get out of synchronization and will be unable to locate the start of the next frame. Even if the checksum is incorrect so the destination knows that the frame is bad, it still has no way of telling where the next frame starts. Sending a frame back to the source asking for a retransmission does not help either, since the destination does not know how many characters to skip over to get to the start of the retransmission. For this reason, the character count method is rarely used anymore.

**Flag bytes with byte stuffing** method gets around the problem of resynchronization after an error by having each frame start and end with special bytes. In the past, the starting and ending bytes were different, but in recent years most protocols have used the same byte, called a flag byte, as both the starting and ending delimiter, as shown in Fig. (a) as FLAG. In this way, if the receiver ever loses synchronization, it can just search for the flag byte to find the end of the current frame. Two consecutive flag bytes indicate the end of one frame and start of the next one.



- (a) A frame delimited by flag bytes (b) Four examples of byte sequences before and after byte stuffing

It may easily happen that the flag byte's bit pattern occurs in the data. This situation will usually interfere with the framing. One way to solve this problem is to have the sender's data link layer insert a special escape byte (ESC) just before each "accidental" flag byte in the data. The data link layer on the receiving end removes the escape byte before the data are given to the network layer. This technique is called byte stuffing or character stuffing.

Thus, a framing flag byte can be distinguished from one in the data by the absence or presence of an escape byte before it.

What happens if an escape byte occurs in the middle of the data? The answer is that, it too is stuffed with an escape byte. Thus, any single escape byte is part of an escape sequence, whereas a doubled one indicates that a single escape occurred naturally in the data. Some examples are shown in Fig. (b). In all cases, the byte sequence delivered after de stuffing is exactly the same as the original byte sequence.

A major disadvantage of using this framing method is that it is closely tied to the use of 8-bit characters. Not all character codes use 8-bit characters. For example UNICODE uses 16-bit characters, so a new technique had to be developed to allow arbitrary sized characters

**Starting and ending flags, with bit stuffing** allows data frames to contain an arbitrary number of bits and allows character codes with an arbitrary number of bits per character. It works like this. Each frame begins and ends with a special bit pattern, 01111110 (in fact, a flag byte). Whenever the sender's data link layer encounters five consecutive 1s in the data, it automatically stuffs a 0 bit into the outgoing bit stream. This bit stuffing is analogous to byte stuffing, in which an escape byte is stuffed into the outgoing character stream before a flag byte in the data.

When the receiver sees five consecutive incoming 1 bits, followed by a 0 bit, it automatically de-stuffs (i.e., deletes) the 0 bit. Just as byte stuffing is completely transparent to the network layer in both computers, so is bit stuffing. If the user data contain the flag pattern, 01111110, this flag is transmitted as 011111010 but stored in the receiver's memory as 01111110.

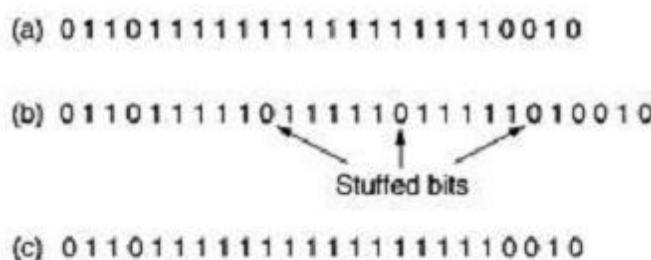
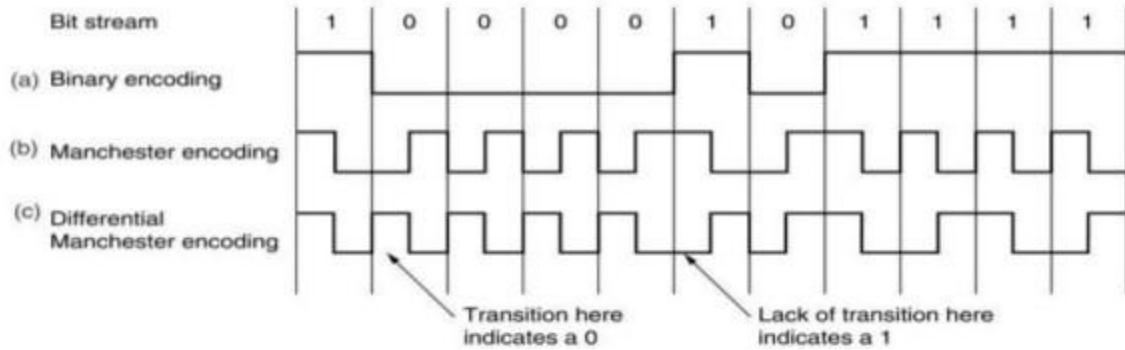


Fig: Bit stuffing. (a) The original data. (b) The data as they appear on the line. (c) The data as they are stored in the receiver's memory after destuffing.

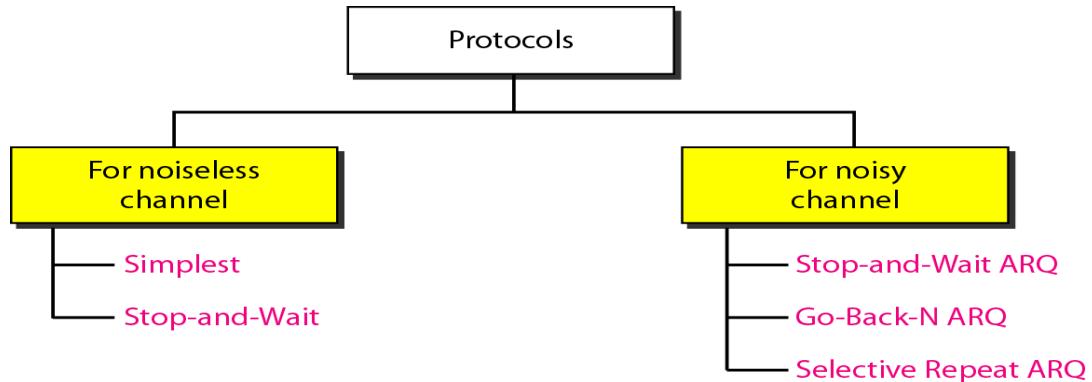
With bit stuffing, the boundary between two frames can be unambiguously recognized by the flag pattern. Thus, if the receiver loses track of where it is, all it has to do is scan the input for flag sequences, since they can only occur at frame boundaries and never within the data.

**Physical layer coding violations** method of framing is only applicable to networks in which the encoding on the physical medium contains some redundancy. For example, some LANs encode 1 bit of data by using 2 physical bits. Normally, a 1 bit is a high-low pair and a 0 bit is a low-high pair. The scheme means that every data bit has a transition in the middle, making it easy for the receiver to locate the bit boundaries. The combinations high-

high and low-low are not used for data but are used for delimiting frames in some protocols.

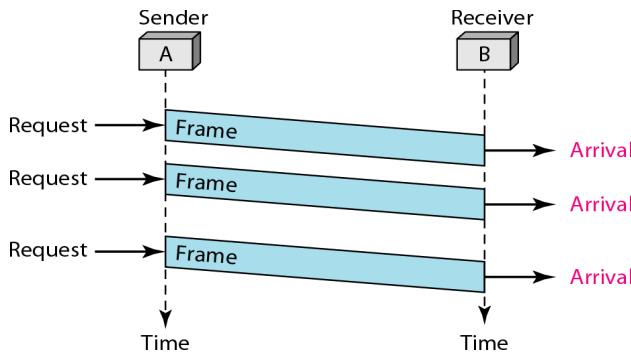


As a final note on framing, many data link protocols use combination of a character count with one of the other methods for extra safety. When a frame arrives, the count field is used to locate the end of the frame. Only if the appropriate delimiter is present at that position and the checksum is correct is the frame accepted as valid. Otherwise, the input stream is scanned for the next delimiter



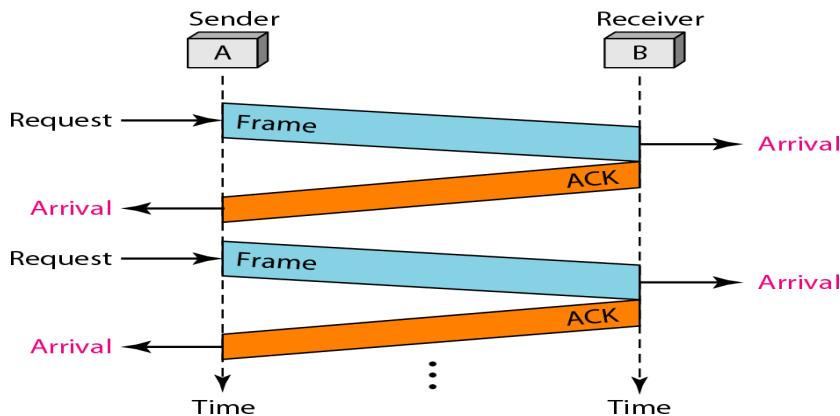
## ELEMENTARY DATA LINK PROTOCOLS

### **Simplest Protocol**



It is very simple. The sender sends a sequence of frames without even thinking about the receiver. Data are transmitted in one direction only. Both sender & receiver always ready. Processing time can be ignored. Infinite buffer space is available. And best of all, the communication channel between the data link layers never damages or loses frames. This thoroughly unrealistic protocol, which we will nickname "Utopia," .The utopia protocol is unrealistic because **it does not handle either flow control or error correction**

### Stop-and-wait Protocol



It is still very simple. The sender sends one frame and waits for feedback from the receiver. When the ACK arrives, the sender sends the next frame  
It is Stop-and-Wait Protocol because the sender sends one frame, stops until it receives confirmation from the receiver (okay to go ahead), and then sends the next frame. We still have unidirectional communication for data frames, but auxiliary ACK frames (simple tokens of acknowledgment) travel from the other direction. We add flow control to our previous protocol.

### NOISY CHANNELS

Although the Stop-and-Wait Protocol gives us an idea of how to add flow control to its predecessor, noiseless channels are nonexistent. We can ignore the error (as we sometimes do), or we need to add error control to our protocols. We discuss three protocols in this section that use error control.

### Sliding Window Protocols:

1 Stop-and-Wait Automatic Repeat Request

2 Go-Back-N Automatic Repeat Request

### **3 Selective Repeat Automatic Repeat Request**

#### **1 Stop-and-Wait Automatic Repeat Request**

To detect and correct corrupted frames, we need to add redundancy bits to our data frame. When the frame arrives at the receiver site, it is checked and if it is corrupted, it is silently discarded. The detection of errors in this protocol is manifested by the silence of the receiver.

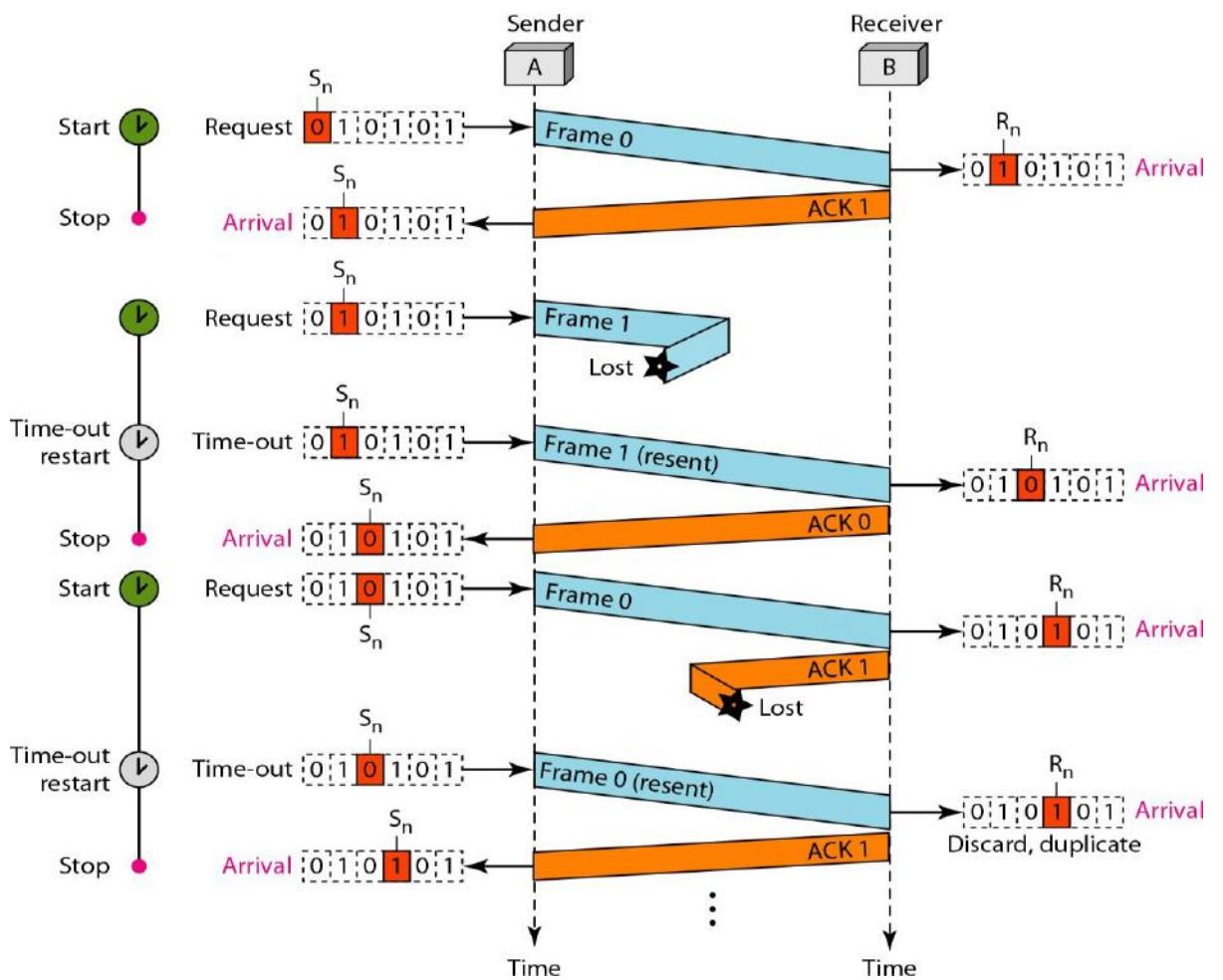
Lost frames are more difficult to handle than corrupted ones. In our previous protocols, there was no way to identify a frame. The received frame could be the correct one, or a duplicate, or a frame out of order. The solution is to number the frames. When the receiver receives a data frame that is out of order, this means that frames were either lost or duplicated

The lost frames need to be resent in this protocol. If the receiver does not respond when there is an error, how can the sender know which frame to resend? To remedy this problem, the sender keeps a copy of the sent frame. At the same time, it starts a timer. If the timer expires and there is no ACK for the sent frame, the frame is resent, the copy is held, and the timer is restarted. Since the protocol uses the stop-and-wait mechanism, there is only one specific frame that needs an ACK

Error correction in Stop-and-Wait ARQ is done by keeping a copy of the sent frame and retransmitting of the frame when the timer expires

**In Stop-and-Wait ARQ, we use sequence numbers to number the frames. The sequence numbers are based on modulo-2 arithmetic.**

**In Stop-and-Wait ARQ, the acknowledgment number always announces in modulo-2 arithmetic the sequence number of the next frame expected.**



### **Bandwidth Delay Product:**

Assume that, in a Stop-and-Wait ARQ system, the bandwidth of the line is 1 Mbps, and 1 bit takes 20 ms to make a round trip. What is the bandwidth-delay product? If the system data frames are 1000 bits in length, what is the utilization percentage of the link?

$$(1 \times 10^6) \times (20 \times 10^{-3}) = 20,000 \text{ bits}$$

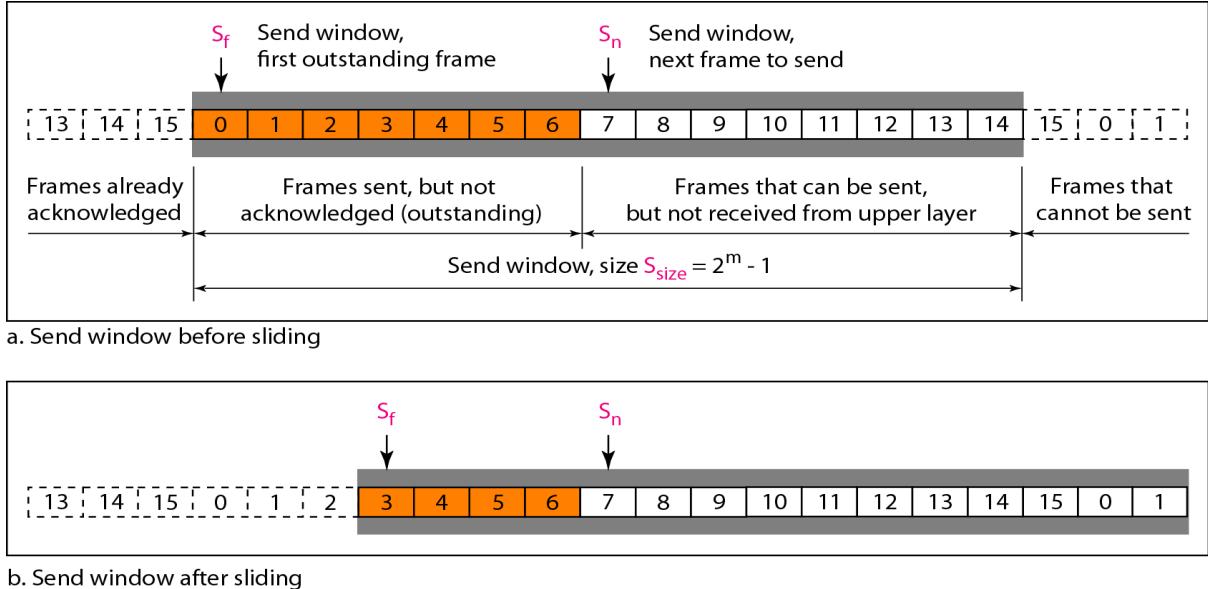
The link utilization is only 1000/20,000, or 5 percent. For this reason, for a link with a high bandwidth or long delay, the use of Stop-and-Wait ARQ wastes the capacity of the link.

### **2. Go-Back-N Automatic Repeat Request**

To improve the efficiency of transmission (filling the pipe), multiple frames must be in transition while waiting for acknowledgment. In other words, we need to let more than one frame be outstanding to keep the channel busy while the sender is waiting for acknowledgment.

The first is called Go-Back-N Automatic Repeat. In this protocol we can send several frames before receiving acknowledgments; we keep a copy of these frames until the acknowledgments arrive.

**In the Go-Back-N Protocol, the sequence numbers are modulo  $2^m$ , where  $m$  is the size of the sequence number field in bits.** The sequence numbers range from 0 to  $2^m - 1$ . For example, if  $m$  is 4, the only sequence numbers are 0 through 15 inclusive.



The **sender window** at any time divides the possible sequence numbers into four regions.

The first region, from the far left to the left wall of the window, defines the sequence numbers belonging to frames that are already acknowledged. The sender does not worry about these frames and keeps no copies of them.

The second region, colored in Figure (a), defines the range of sequence numbers belonging to the frames that are sent and have an unknown status. The sender needs to wait to find out if these frames have been received or were lost. We call these outstanding frames.

The third range, white in the figure, defines the range of sequence numbers for frames that can be sent; however, the corresponding data packets have not yet been received from the network layer.

Finally, the fourth region defines sequence numbers that cannot be used until the window slides.

**The send window is an abstract concept defining an imaginary box of size  $2^m - 1$  with three variables:  $S_f$ ,  $S_n$ , and  $S_{size}$ .** The variable  $S_f$  defines the sequence number of the first (oldest) outstanding frame. The variable  $S_n$  holds the sequence number that will be assigned to the next frame to be sent. Finally, the variable  $S_{size}$  defines the size of the window.

Figure (b) shows how a send window can slide one or more slots to the right when an acknowledgment arrives from the other end. The acknowledgments in this protocol are cumulative, meaning that more than one frame can be acknowledged by an ACK frame. In Figure, frames 0, 1, and 2 are

acknowledged, so the window has slide to the right three slots. Note that the value of  $S_f$  is 3 because frame 3 is now the first outstanding frame. **The send window can slide one or more slots when a valid acknowledgment arrives.**

**Receiver window:** variable  $R_n$  (receive window, next frame expected) .

The sequence numbers to the left of the window belong to the frames already received and acknowledged; the sequence numbers to the right of this window define the frames that cannot be received. Any received frame with a sequence number in these two regions is discarded. Only a frame with a sequence number matching the value of  $R_n$  is accepted and acknowledged. The receive window also slides, but only one slot at a time. When a correct frame is received (and a frame is received only one at a time), the window slides.( see below figure for receiving window)

The receive window is an abstract concept defining an imaginary box of size 1 with one single variable  $R_n$ . The window slides when a correct frame has arrived; sliding occurs one slot at a time

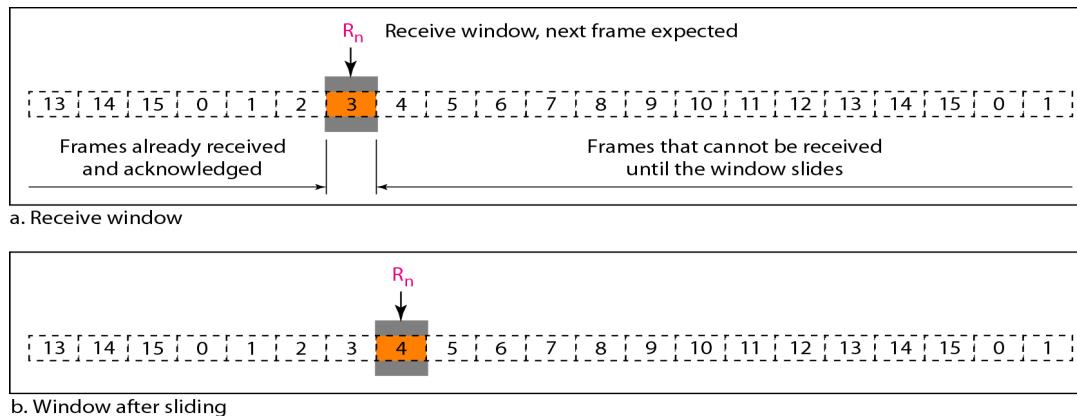


Fig: Receiver window (before sliding (a), After sliding (b))

### Timers

Although there can be a timer for each frame that is sent, in our protocol we use only one. The reason is that the timer for the first outstanding frame always expires first; we send all outstanding frames when this timer expires.

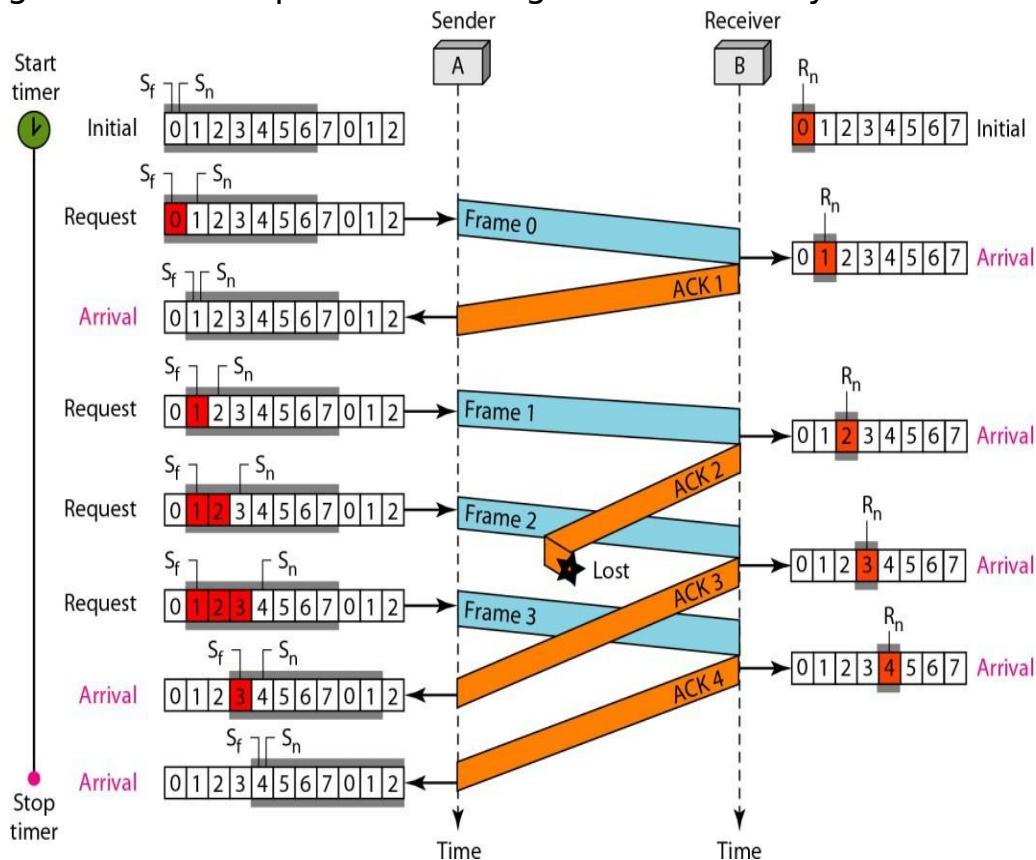
### Acknowledgment

The receiver sends a positive acknowledgment if a frame has arrived safe and sound and in order. If a frame is damaged or is received out of order, the receiver is silent and will discard all subsequent frames until it receives the one it is expecting. The silence of the receiver causes the timer of the unacknowledged frame at the sender side to expire. This, in turn, causes the sender to go back and resend all frames, beginning with the one with the expired timer. The receiver does not have to acknowledge each frame received. It can send one cumulative acknowledgment for several frames.

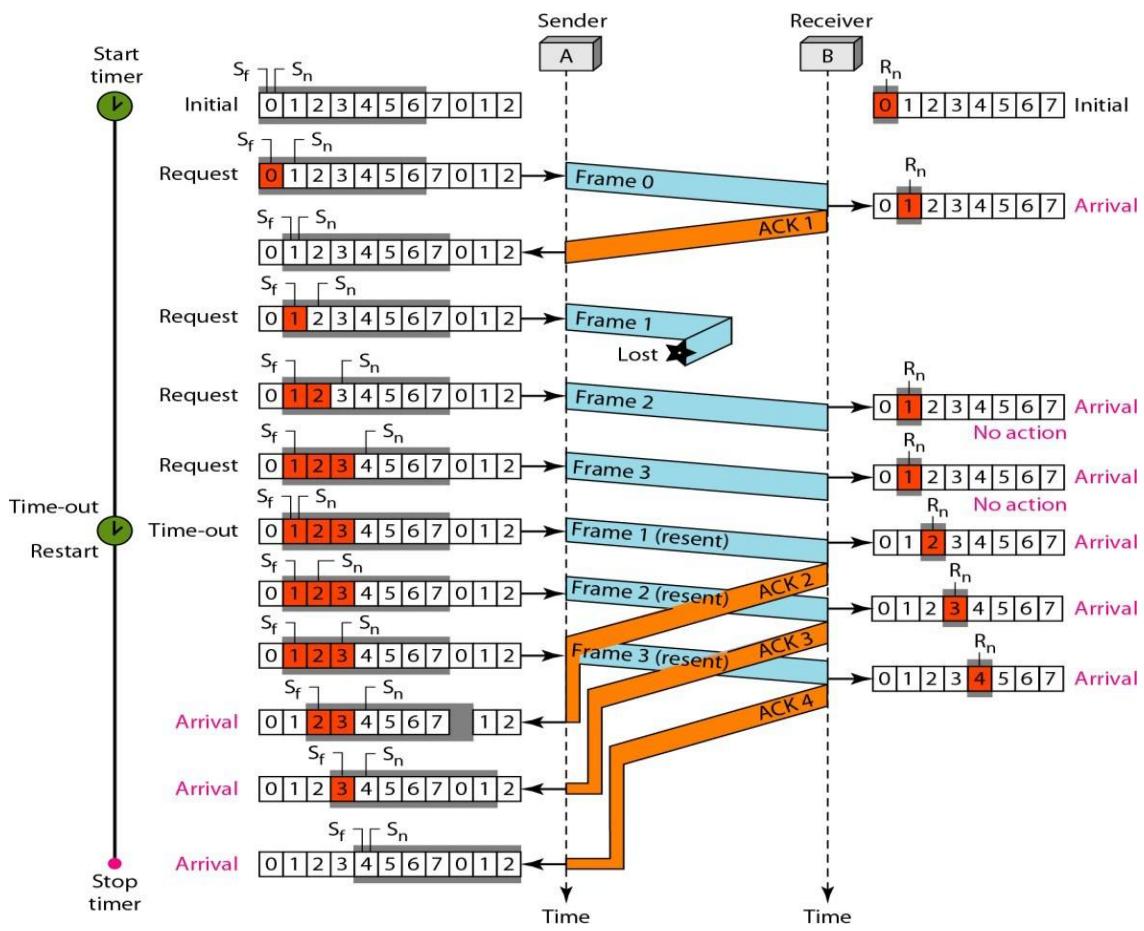
## Resending a Frame

When the timer expires, the sender resends all outstanding frames. For example, suppose the sender has already sent frame 6, but the timer for frame 3 expires. This means that frame 3 has not been acknowledged; the sender goes back and sends frames 3,4,5, and 6 again. That is why the protocol is called *Go-Back-N* ARQ.

Below figure is an example(if ack lost) of a case where the forward channel is reliable, but the reverse is not. No data frames are lost, but some ACKs are delayed and one is lost. The example also shows how cumulative acknowledgments can help if acknowledgments are delayed or lost



Below figure is an example(if frame lost)



Stop-and-Wait ARQ is a special case of Go-Back-N ARQ in which the size of the send window is 1.

### **3 Selective Repeat Automatic Repeat Request**

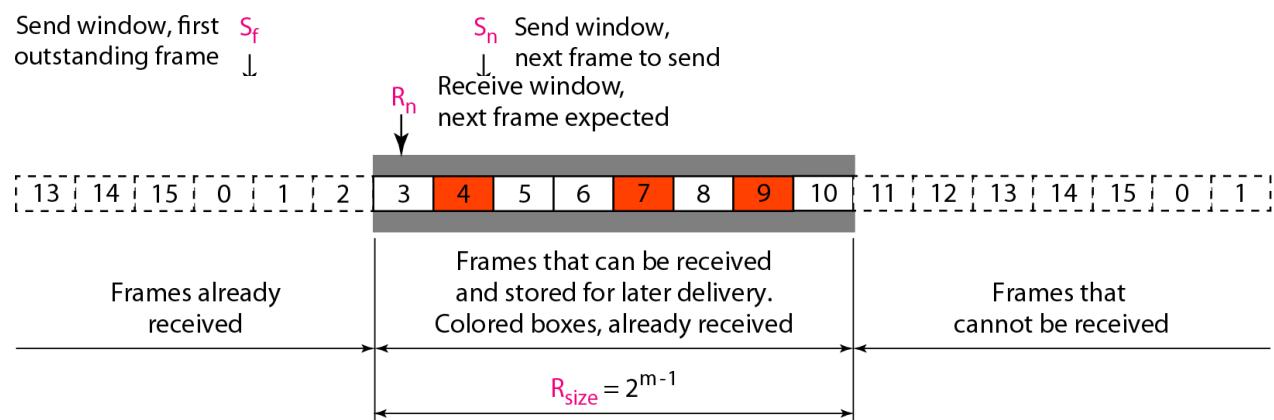
In Go-Back-N ARQ, The receiver keeps track of only one variable, and there is no need to buffer out-of-order frames; they are simply discarded. However, this protocol is very inefficient for a noisy link.

In a noisy link a frame has a higher probability of damage, which means the resending of multiple frames. This resending uses up the bandwidth and slows down the transmission.

For noisy links, there is another mechanism that does not resend  $N$  frames when just one frame is damaged; only the damaged frame is resent. This mechanism is called Selective Repeat ARQ.

It is more efficient for noisy links, but the processing at the receiver is more complex.

**Sender Window** (explain go-back N sender window concept (before & after sliding.) The only difference in sender window between Go-back N and Selective Repeat is Window size)



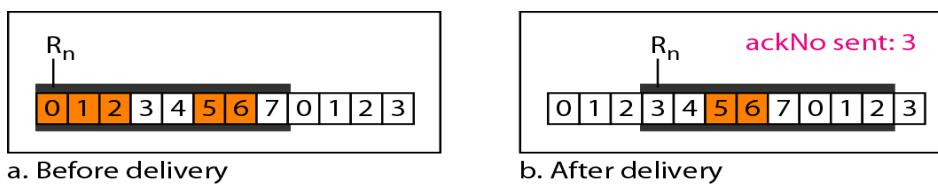
### Receiver window

The receiver window in Selective Repeat is totally different from the one in Go Back-N. First, the size of the receive window is the same as the size of the send window ( $2^{m-1}$ ).

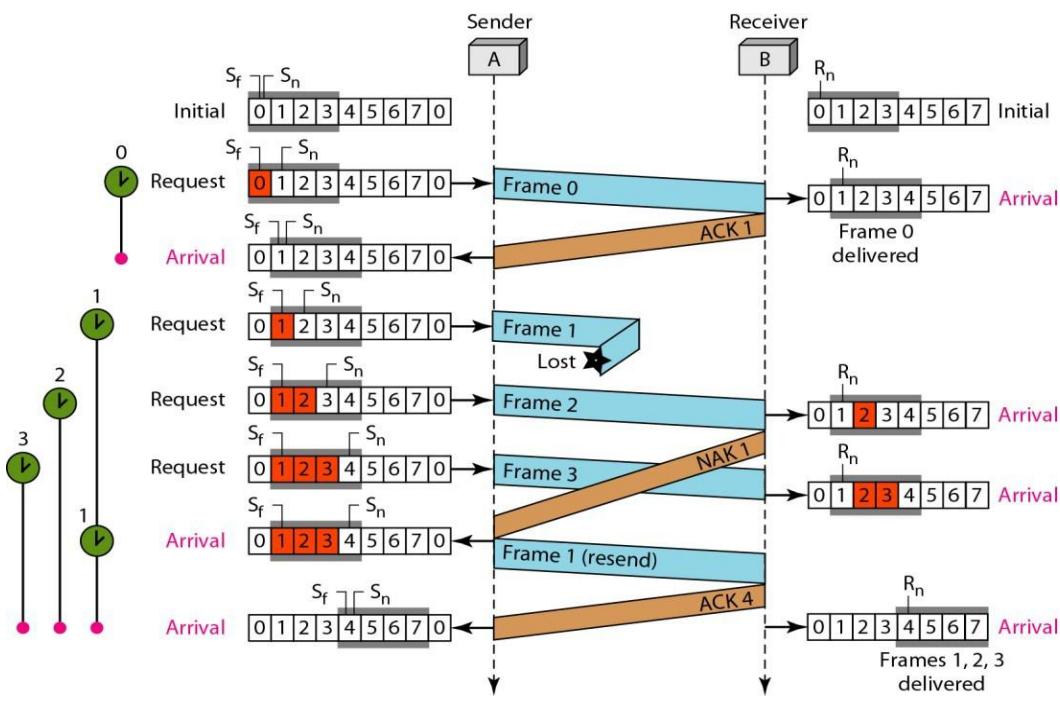
The Selective Repeat Protocol allows as many frames as the size of the receiver window to arrive out of order and be kept until there is a set of in-order frames to be delivered to the network layer. Because the sizes of the send window and receive window are the same, all the frames in the send frame can arrive out of order and be stored until they can be delivered. However the receiver never delivers packets out of order to the network layer. Above Figure shows the receive window. Those slots inside the window that are colored define frames that have arrived out of order and are waiting for their neighbors to arrive before delivery to the network layer.

In Selective Repeat ARQ, the size of the sender and receiver window must be at most one-half of  $2^m$

### **Delivery of Data in Selective Repeat ARQ:**



### **Flow Diagram**



### Differences between Go-Back N & Selective Repeat

One main difference is the number of timers. Here, each frame sent or resent needs a timer, which means that the timers need to be numbered (0, 1, 2, and 3). The timer for frame 0 starts at the first request, but stops when the ACK for this frame arrives.

There are two conditions for the delivery of frames to the network layer: First, a set of consecutive frames must have arrived. Second, the set starts from the beginning of the window. After the first arrival, there was only one frame and it started from the beginning of the window. After the last arrival, there are three frames and the first one starts from the beginning of the window.

Another important point is that a NAK is sent.

The next point is about the ACKs. Notice that only two ACKs are sent here. The first one acknowledges only the first frame; the second one acknowledges three frames. In Selective Repeat, ACKs are sent when data are delivered to the network layer. If the data belonging to  $n$  frames are delivered in one shot, only one ACK is sent for all of them.

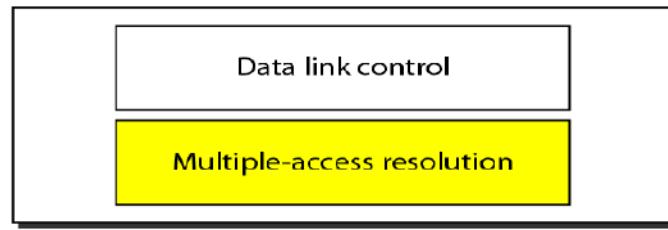
### **Piggybacking**

A technique called **piggybacking** is used to improve the efficiency of the bidirectional protocols. When a frame is carrying data from A to B, it can also carry control information about arrived (or lost) frames from B; when a frame is carrying data from B to A, it can also carry control information about the arrived (or lost) frames from A.

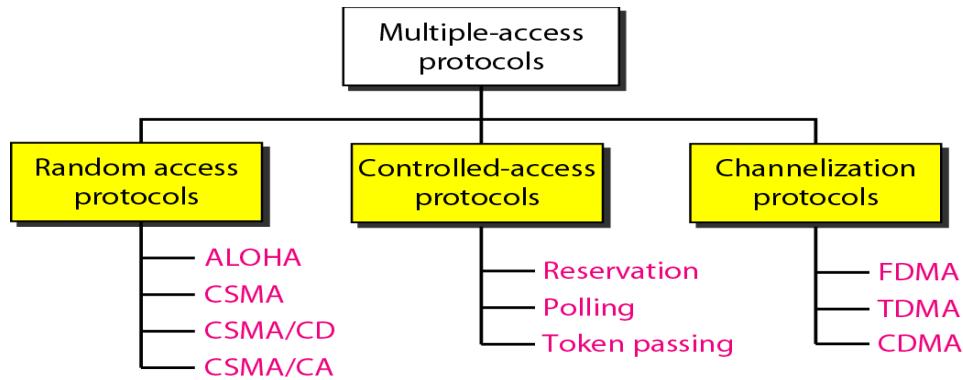
## **RANDOM ACCESS PROTOCOLS**

We can consider the data link layer as two sub layers. The upper sub layer is responsible for data link control, and the lower sub layer is responsible for resolving access to the shared media

## Data link layer



The upper sub layer that is responsible for flow and error control is called the logical link control (LLC) layer; the lower sub layer that is mostly responsible for multiple access resolution is called the media access control (MAC) layer. When nodes or stations are connected and use a common link, called a multipoint or broadcast link, we need a multiple-access protocol to coordinate access to the link.



## Taxonomy of multiple-access protocols

## RANDOM ACCESS

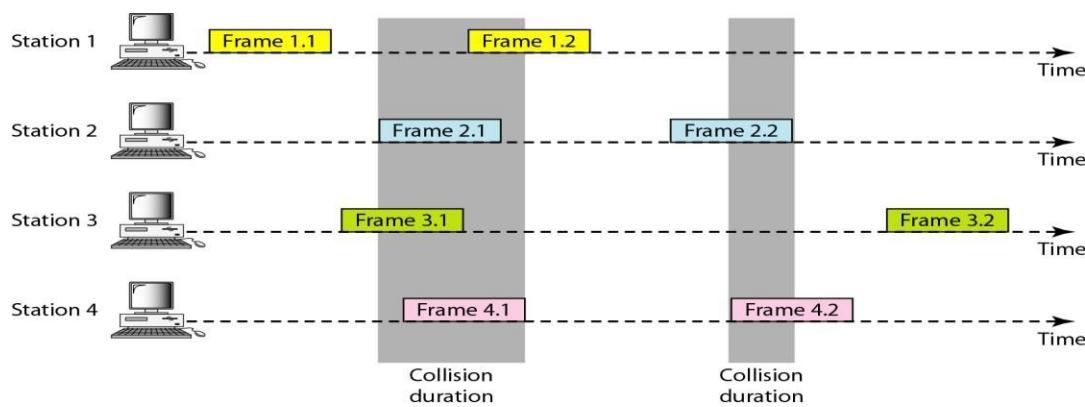
In random access or contention methods, no station is superior to another station and none is assigned the control over another.

Two features give this method its name. First, there is no scheduled time for a station to transmit. Transmission is random among the stations. That is why these methods are called *random access*. Second, no rules specify which station should send next. Stations compete with one another to access the medium. That is why these methods are also called *contention* methods.

## ALOHA

### 1 Pure ALOHA

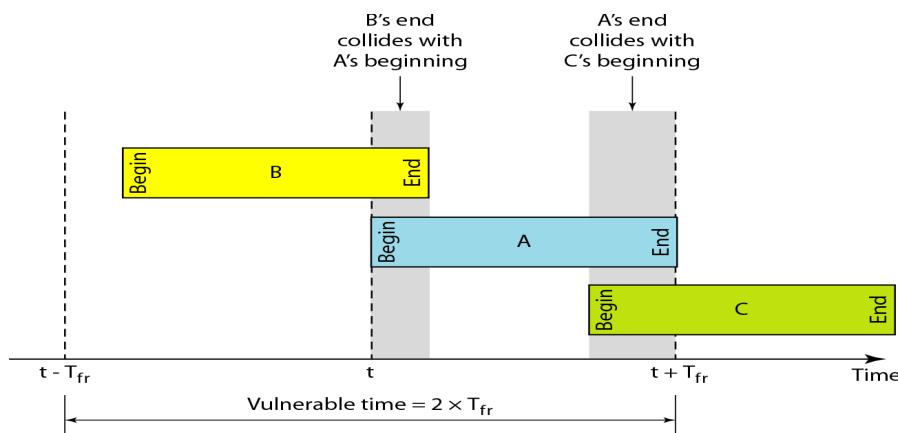
The original ALOHA protocol is called pure ALOHA. This is a simple, but elegant protocol. The idea is that each station sends a frame whenever it has a frame to send. However, since there is only one channel to share, there is the possibility of collision between frames from different stations. Below Figure shows an example of frame collisions in pure ALOHA.



### Frames in a pure ALOHA network

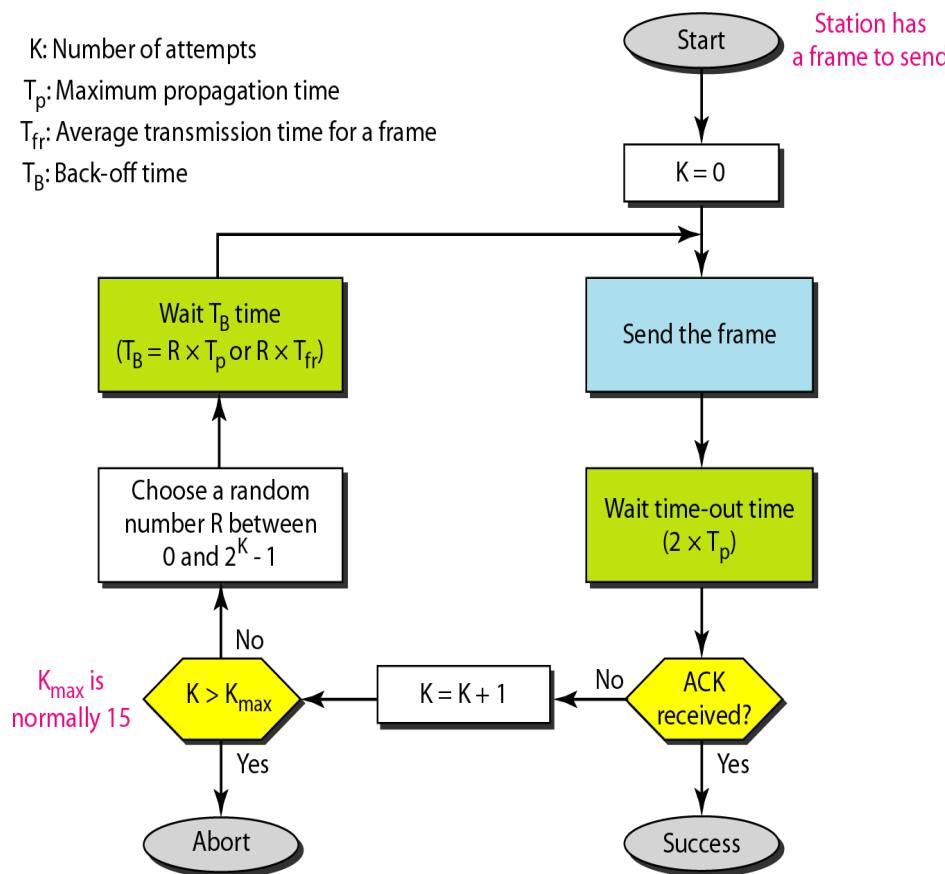
- In pure ALOHA, the stations transmit frames whenever they have data to send.
- When two or more stations transmit simultaneously, there is collision and the frames are destroyed.
- In pure ALOHA, whenever any station transmits a frame, it expects the acknowledgement from the receiver.
- If acknowledgement is not received within specified time, the station assumes that the frame (or acknowledgement) has been destroyed.
- If the frame is destroyed because of collision the station waits for a random amount of time and sends it again. This waiting time must be random otherwise same frames will collide again and again.
- Therefore pure ALOHA dictates that when time-out period passes, each station must wait for a random amount of time before resending its frame. This randomness will help avoid more collisions.

**Vulnerable time** Let us find the length of time, the **vulnerable time**, in which there is a possibility of collision. We assume that the stations send fixed-length frames with each frame taking  $T_{fr}$  S to send. Below Figure shows the vulnerable time for station A.



Station A sends a frame at time  $t$ . Now imagine station B has already sent a frame between  $t - T_{fr}$  and  $t$ . This leads to a collision between the frames from station A and station B. The end of B's frame collides with the beginning of A's frame. On the other hand, suppose that station C sends a frame between  $t$  and  $t + T_{fr}$ . Here, there is a collision between frames from station A and station C. The beginning of C's frame collides with the end of A's frame.

Looking at Figure, we see that the vulnerable time, during which a collision may occur in pure ALOHA, is 2 times the frame transmission time. Pure ALOHA vulnerable time =  $2 \times T_{fr}$



### Procedure for pure ALOHA protocol

#### Example

A pure ALOHA network transmits 200-bit frames on a shared channel of 200 kbps. What is the requirement to make this frame collision-free?

#### Solution

Average frame transmission time  $T_{fr}$  is 200 bits/200 kbps or 1 ms. The vulnerable time is  $2 \times 1 \text{ ms} = 2 \text{ ms}$ . This means no station should send later than 1 ms before this station starts transmission and no station should start sending during the one 1-ms period that this station is sending.

**The throughput for pure ALOHA is  $S = G \times e^{-2G}$ . The maximum throughput  $S_{max} = 0.184$  when  $G = (1/2)$ .**

#### PROBLEM

A pure ALOHA network transmits 200-bit frames on a shared channel of 200 kbps. What is the throughput if the system (all stations together) produces a. 1000 frames per second b. 500 frames per second c. 250 frames per second. The frame transmission time is 200/200 kbps or 1 ms.

- If the system creates 1000 frames per second, this is 1 frame per

millisecond. The load is 1. In this case  $S = G \times e^{-2G}$  or  $S = 0.135$  (13.5 percent). This means that the throughput is  $1000 \times 0.135 = 135$  frames. Only 135 frames out of 1000 will probably survive.

- b. If the system creates 500 frames per second, this is  $(1/2)$  frame per millisecond. The load is  $(1/2)$ . In this case  $S = G \times e^{-2G}$  or  $S = 0.184$  (18.4 percent). This means that the throughput is  $500 \times 0.184 = 92$  and that only 92 frames out of 500 will probably survive. Note that this is the maximum throughput case, percentage wise.
- c. If the system creates 250 frames per second, this is  $(1/4)$  frame per millisecond. The load is  $(1/4)$ . In this case  $S = G \times e^{-2G}$  or  $S = 0.152$  (15.2 percent). This means that the throughput is  $250 \times 0.152 = 38$ . Only 38 frames out of 250 will probably survive.

## **2 Slotted ALOHA**

Pure ALOHA has a vulnerable time of  $2 \times T_{fr}$ . This is so because there is no rule that defines when the station can send. A station may send soon after another station has started or soon before another station has finished. Slotted ALOHA was invented to improve the efficiency of pure ALOHA.

In slotted ALOHA we divide the time into slots of  $T_{fr}$ s and force the station to send only at the beginning of the time slot. Figure 3 shows an example of frame collisions in slotted ALOHA

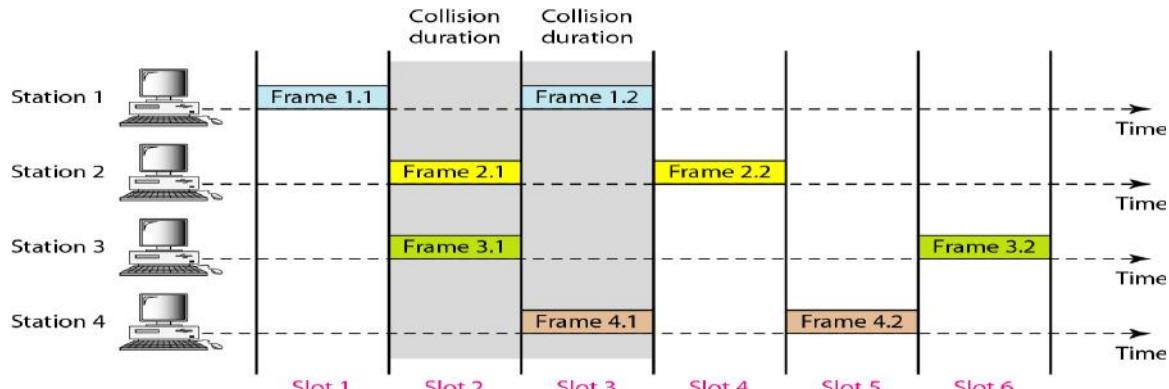
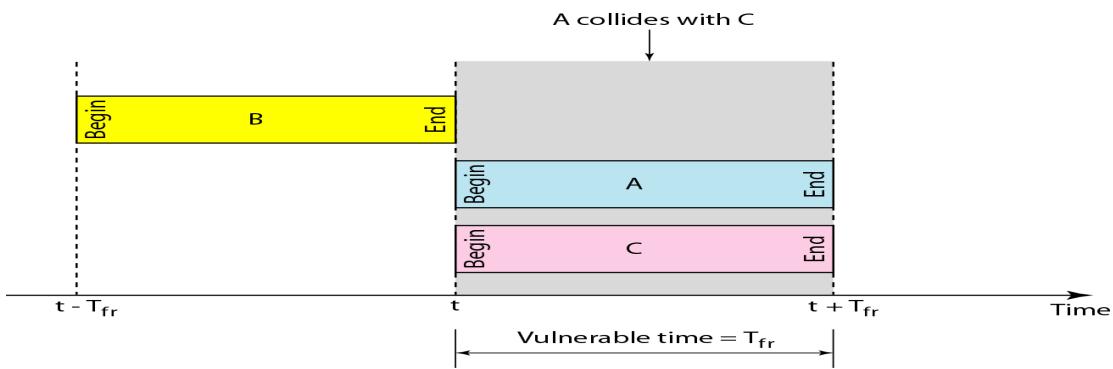


FIG:3

Because a station is allowed to send only at the beginning of the synchronized time slot, if a station misses this moment, it must wait until the beginning of the next time slot. This means that the station which started at the beginning of this slot has already finished sending its frame. Of course, there is still the possibility of collision if two stations try to send at the beginning of the same time slot. However, the vulnerable time is now reduced to one-half, equal to  $T_{fr}$ . Figure 4 shows the situation.

Below fig shows that the vulnerable time for slotted ALOHA is one-half that of pure ALOHA. Slotted ALOHA vulnerable time =  $T_{fr}$



**The throughput for slotted ALOHA is  $S = G \times e^{-G}$ . The maximum throughput  $S_{max} = 0.368$  when  $G = 1$ .**

A slotted ALOHA network transmits 200-bit frames using a shared channel with a 200- Kbps bandwidth. Find the throughput if the system (all stations together) produces

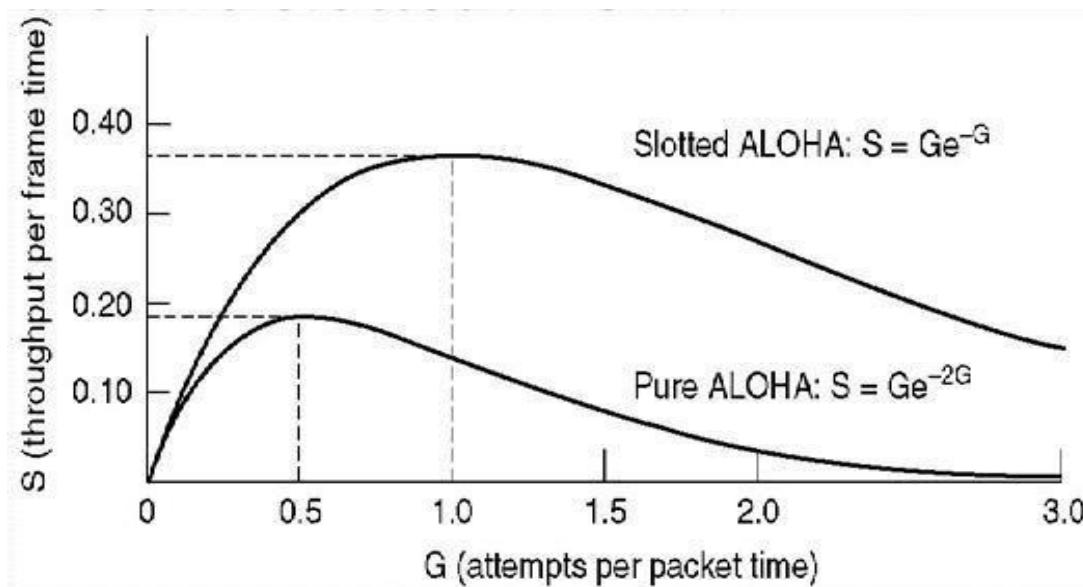
- a. 1000 frames per second b. 500 frames per second c. 250 frames per second

### Solution

This situation is similar to the previous exercise except that the network is using slotted ALOHA instead of pure ALOHA. The frame transmission time is  $200/200$  kbps or 1 ms.

- a. In this case  $G$  is 1. So  $S = G \times e^{-G}$  or  $S = 0.368$  (36.8 percent). This means that the throughput is  $1000 \times 0.0368 = 368$  frames. Only 368 out of 1000 frames will probably survive. Note that this is the maximum throughput case, percentagewise.
- b. Here  $G$  is  $1/2$  In this case  $S = G \times e^{-G}$  or  $S = 0.303$  (30.3 percent). This means that the throughput is  $500 \times 0.0303 = 151$ . Only 151 frames out of 500 will probably survive.
- c. Now  $G$  is  $1/4$ . In this case  $S = G \times e^{-G}$  or  $S = 0.195$  (19.5 percent). This means that the throughput is  $250 \times 0.195 = 49$ . Only 49 frames out of 250 will probably survive

### Comparison between Pure Aloha & Slotted Aloha

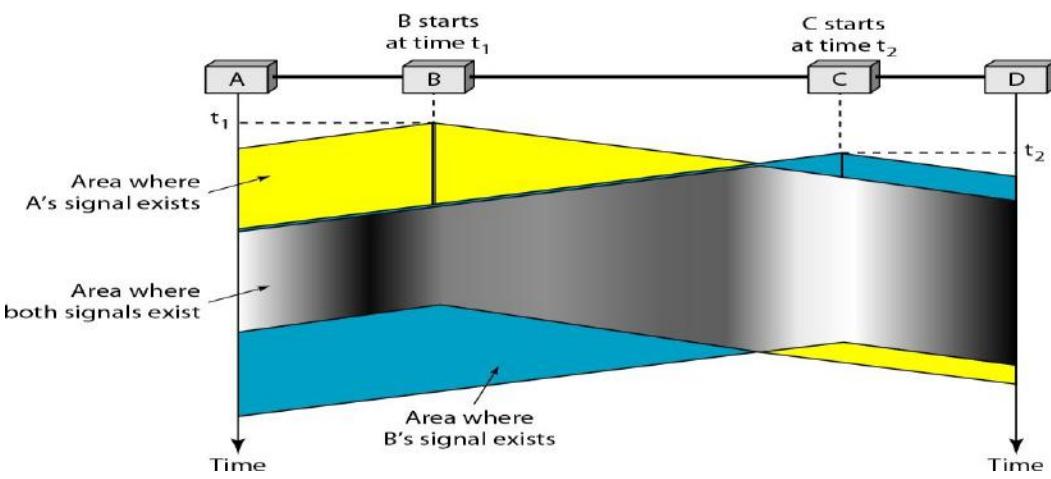


## **Carrier Sense Multiple Access (CSMA)**

To minimize the chance of collision and, therefore, increase the performance, the CSMA method was developed. The chance of collision can be reduced if a station senses the medium before trying to use it. Carrier sense multiple access (CSMA) requires that each station first listen to the medium (or check the state of the medium) before sending. In other words, CSMA is based on the principle "sense before transmit" or "listen before talk."

CSMA can reduce the possibility of collision, but it cannot eliminate it. The reason for this is shown in below Figure. Stations are connected to a shared channel (usually a dedicated medium). The possibility of collision still exists because of propagation delay; station may sense the medium and find it idle, only because the first bits sent by another station has not yet been received.

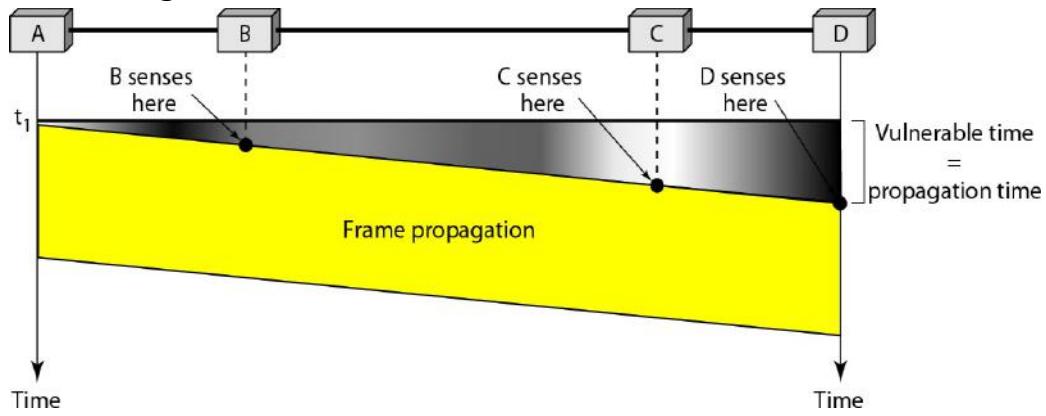
At time  $t_1$  station B senses the medium and finds it idle, so it sends a frame. At time  $t_2$  ( $t_2 > t_1$ ) station C senses the medium and finds it idle because, at this time, the first bits from station B have not reached station C. Station C also sends a frame. The two signals collide and both frames are destroyed.



Space/time model of the collision in CSMA

### Vulnerable Time

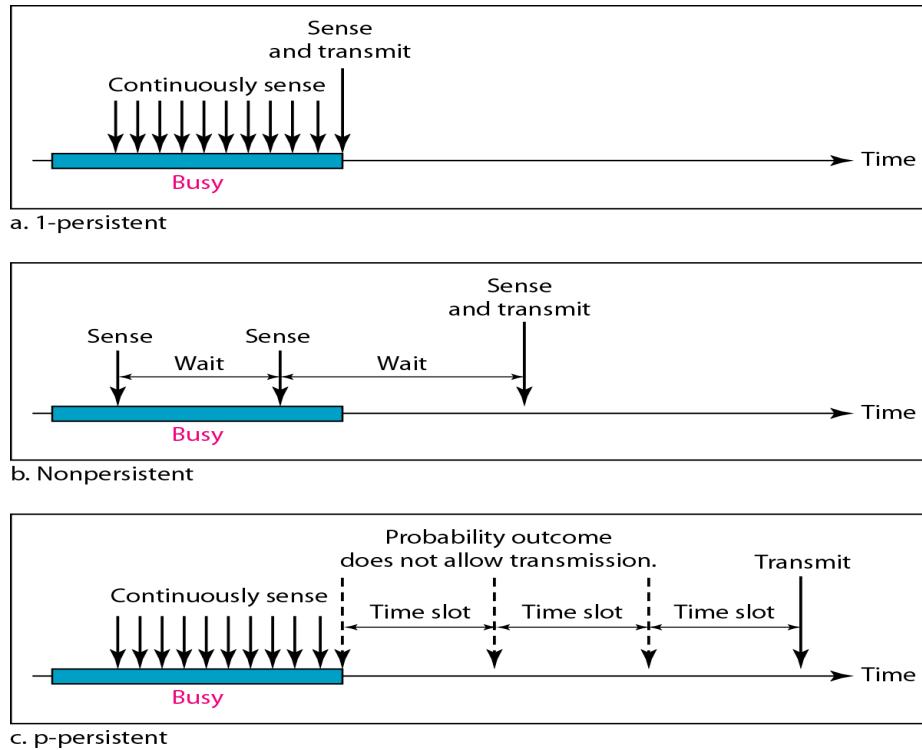
The vulnerable time for CSMA is the propagation time  $T_p$ . This is the time needed for a signal to propagate from one end of the medium to the other. When a station sends a frame, and any other station tries to send a frame during this time, a collision will result. But if the first bit of the frame reaches the end of the medium, every station will already have heard the bit and will refrain from sending.



Vulnerable time in CSMA

### Persistence Methods

What should a station do if the channel is busy? What should a station do if the channel is idle? Three methods have been devised to answer these questions: the 1-persistent method, the non-persistent method, and the p-persistent method.



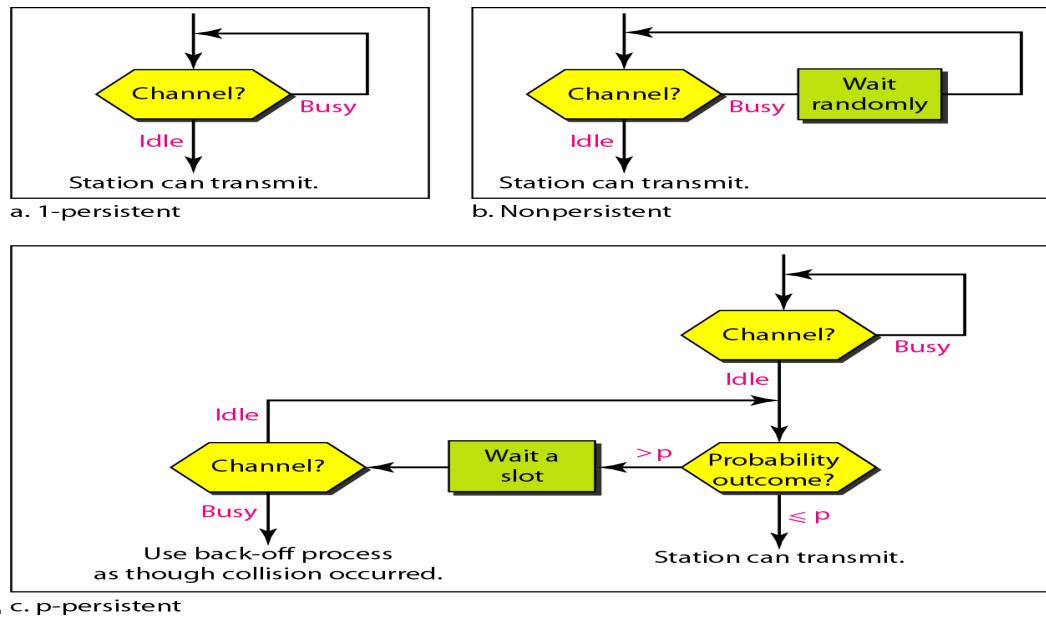
**1-Persistent:** In this method, after the station finds the line idle, it sends its frame immediately (with probability 1). This method has the highest chance of collision because two or more stations may find the line idle and send their frames immediately.

**Non-persistent:** a station that has a frame to send senses the line. If the line is idle, it sends immediately. If the line is not idle, it waits a random amount of time and then senses the line again. This approach reduces the chance of collision because it is unlikely that two or more stations will wait the same amount of time and retry to send simultaneously. However, this method reduces the efficiency of the network because the medium remains idle when there may be stations with frames to send.

**p-Persistent:** This is used if the channel has time slots with a slot duration equal to or greater than the maximum propagation time. The p-persistent approach combines the advantages of the other two strategies. It reduces the chance of collision and improves efficiency.

In this method, after the station finds the line idle it follows these steps:

1. With probability  $p$ , the station sends its frame.
2. With probability  $q = 1 - p$ , the station waits for the beginning of the next time slot and checks the line again.
  - a. If the line is idle, it goes to step 1.
  - b. If the line is busy, it acts as though a collision has occurred and uses the backoff procedure.

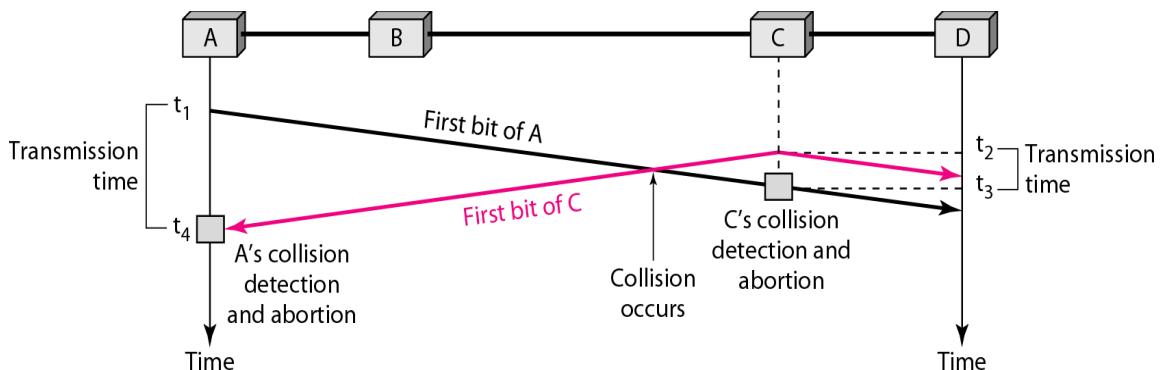


## Carrier Sense Multiple Access with Collision Detection (CSMA/CD)

The CSMA method does not specify the procedure following a collision. Carrier sense multiple access with collision detection (CSMA/CD) augments the algorithm to handle the collision.

In this method, a station monitors the medium after it sends a frame to see if the transmission was successful. If so, the station is finished. If, however, there is a collision, the frame is sent again.

To better understand CSMA/CD, let us look at the first bits transmitted by the two stations involved in the collision. Although each station continues to send bits in the frame until it detects the collision, we show what happens as the first bits collide. In below Figure, stations A and C are involved in the collision.



### Collision of the first bit in CSMA/CD

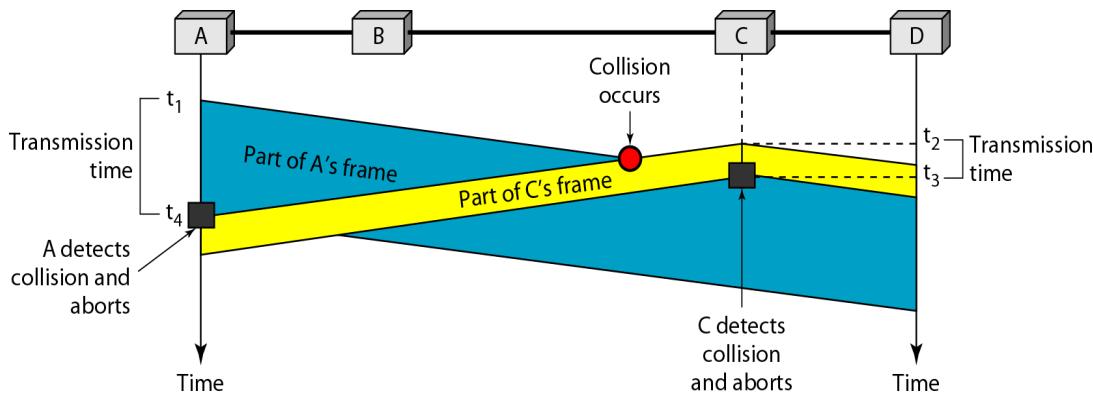
At time  $t_1$ , station A has executed its persistence procedure and starts sending the bits of its frame. At time  $t_2$ , station C has not yet sensed the first bit sent by A. Station C executes its persistence procedure and starts sending the bits in its frame, which propagate both to the left and to the right. The collision occurs sometime after time  $t_2$ . Station C detects a collision at time  $t_3$  when it receives the first bit of A's frame. Station C immediately (or after a short time, but we assume immediately) aborts transmission.

Station A detects collision at time  $t_4$  when it receives the first bit of C's frame; it also immediately aborts transmission. Looking at the figure, we see that A

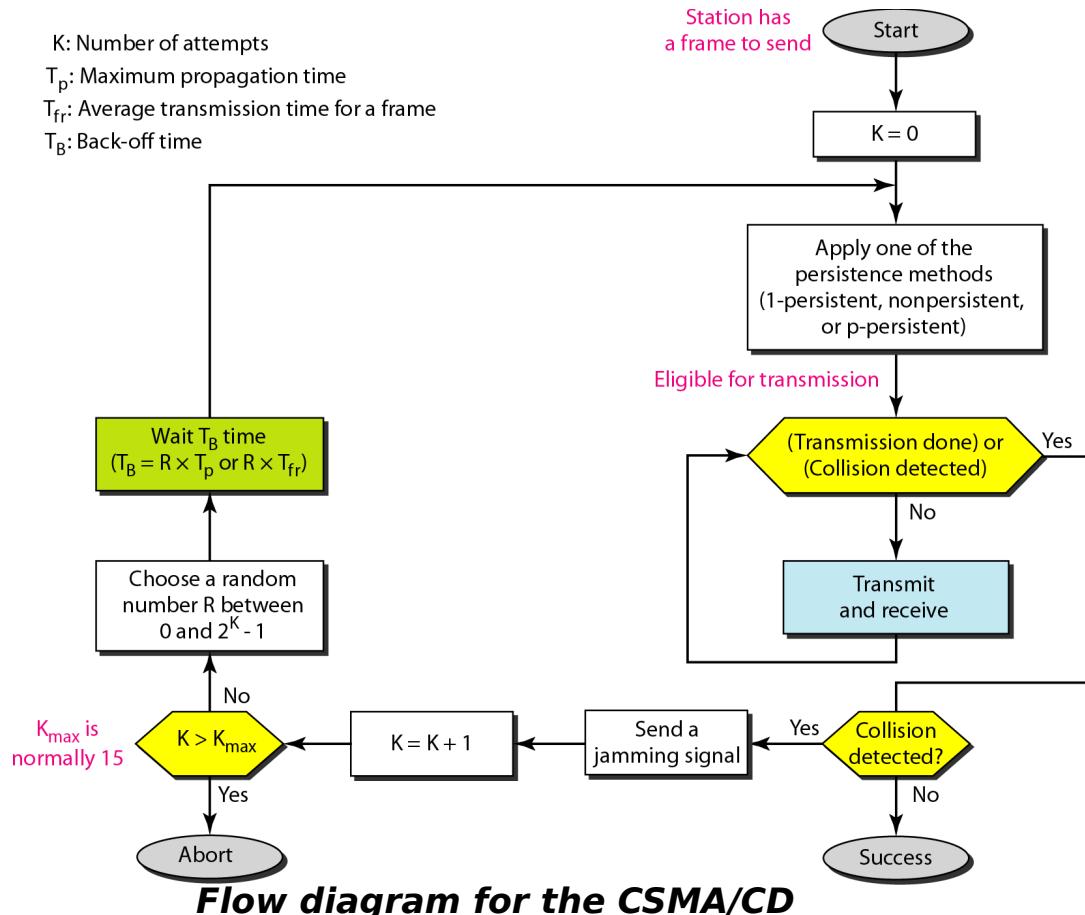
transmits for the duration  $t_4 - t_1$ ; C transmits for the duration  $t_3 - t_2$ .

### Minimum Frame Size

For CSMA/CD to work, we need a restriction on the frame size. Before sending the last bit of the frame, the sending station must detect a collision, if any, and abort the transmission. This is so because the station, once the entire frame is sent, does not keep a copy of the frame and does not monitor the line for collision detection. Therefore, the frame transmission time  $T_{fr}$  must be at least two times the maximum propagation time  $T_p$ . To understand the reason, let us think about the worst-case scenario. If the two stations involved in a collision are the maximum distance apart, the signal from the first takes time  $T_p$  to reach the second, and the effect of the collision takes another time  $T_p$  to reach the first. So the requirement is that the first station must still be transmitting after  $2T_p$ .



## Collision and abortion in CSMA/CD



### PROBLEM

A network using CSMA/CD has a bandwidth of 10 Mbps. If the maximum propagation time (including the delays in the devices and ignoring the time needed to send a jamming signal, as we see later) is 25.6  $\mu$ s, what is the minimum size of the frame?

### SOL

The frame transmission time is  $T_{fr} = 2 \times T_p = 51.2 \mu$ s. This means, in the worst case, a station needs to transmit for a period of 51.2  $\mu$ s to detect the collision. The minimum size of the frame is  $10 \text{ Mbps} \times 51.2 \mu\text{s} = 512 \text{ bits}$  or 64 bytes. This is actually the minimum size of the frame for Standard Ethernet.

### DIFFERENCES BETWEEN ALOHA & CSMA/CD

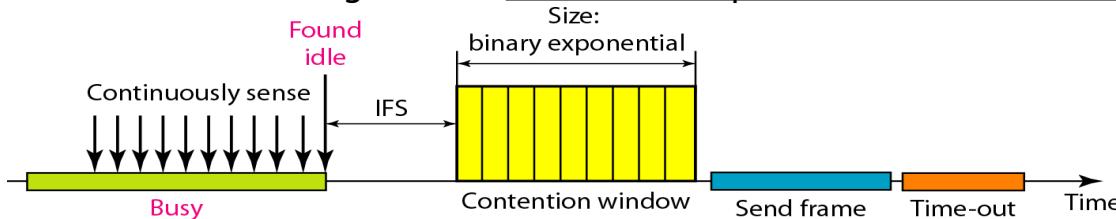
The first difference is the addition of the persistence process. We need to sense the channel before we start sending the frame by using one of the persistence processes

The second difference is the frame transmission. In ALOHA, we first transmit the entire frame and then wait for an acknowledgment. In CSMA/CD, transmission and collision detection is a continuous process. We do not send the entire frame and then look for a collision. The station transmits and receives continuously and simultaneously

The third difference is the sending of a short jamming signal that enforces the collision in case other stations have not yet sensed the collision.

## Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA)

We need to avoid collisions on wireless networks because they cannot be detected. Carrier sense multiple access with collision avoidance (CSMA/CA) was invented for wireless network. Collisions are avoided through the use of CSMA/CA's three strategies: the inter frame space, the contention window, and



acknowledgments, as shown in Figure

### Timing in CSMA/CA

#### Inter frame Space (IFS)

First, collisions are avoided by deferring transmission even if the channel is found idle. When an idle channel is found, the station does not send immediately. It waits for a period of time called the inter frame space or IFS.

Even though the channel may appear idle when it is sensed, a distant station may have already started transmitting. The distant station's signal has not yet reached this station. The IFS time allows the front of the transmitted signal by the distant station to reach this station. If after the IFS time the channel is still idle, the station can send, but it still needs to wait a time equal to the contention time. The IFS variable can also be used to prioritize stations or frame types. For example, a station that is assigned shorter IFS has a higher priority.

In CSMA/CA, the IFS can also be used to define the priority of a station or a frame.

#### Contention Window

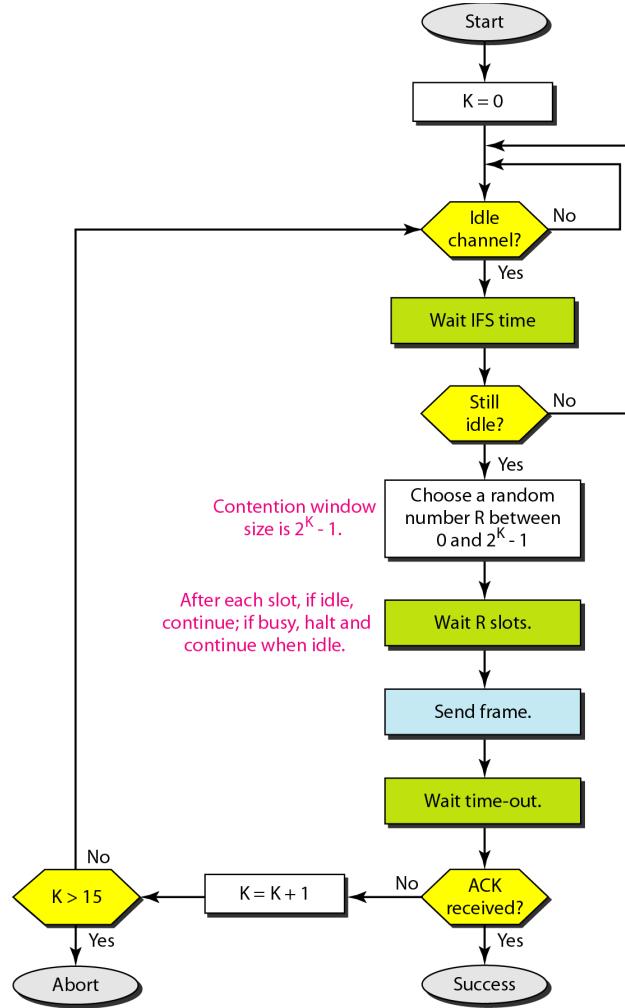
The contention window is an amount of time divided into slots. A station that is ready to send chooses a random number of slots as its wait time. The number of slots in the window changes according to the binary exponential back-off strategy. This means that it is set to one slot the first time and then doubles each time the station cannot detect an idle channel after the IFS time. This is very similar to the p-persistent method except that a random outcome defines the number of slots taken by the waiting station.

One interesting point about the contention window is that the station needs to sense the channel after each time slot. However, if the station finds the channel busy, it does not restart the process; it just stops the timer and restarts it when the channel is sensed as idle. This gives priority to the station with the longest waiting time.

In CSMA/CA, if the station finds the channel busy, it does not restart the timer of the contention window; it stops the timer and restarts it when the channel becomes idle.

### Acknowledgment

With all these precautions, there still may be a collision resulting in destroyed data. In addition, the data may be corrupted during the transmission. The positive acknowledgment and the time-out timer can help guarantee that the receiver has received the frame.



This is the CSMA protocol with collision avoidance.

- The station ready to transmit, senses the line by using one of the persistent strategies.
- As soon as it finds the line to be idle, the station waits for an IFS (Inter frame space) amount of time.
- If then waits for some random time and sends the frame.
- After sending the frame, it sets a timer and waits for the acknowledgement from the receiver.
- If the acknowledgement is received before expiry of the timer, then the transmission is successful.
- But if the transmitting station does not receive the expected acknowledgement before the timer expiry then it increments the back off

parameter, waits for the back off time and re senses the line

## Controlled Access Protocols

In controlled access, the stations seek information from one another to find which station has the right to send. It allows only one node to send at a time, to avoid collision of messages on shared medium.

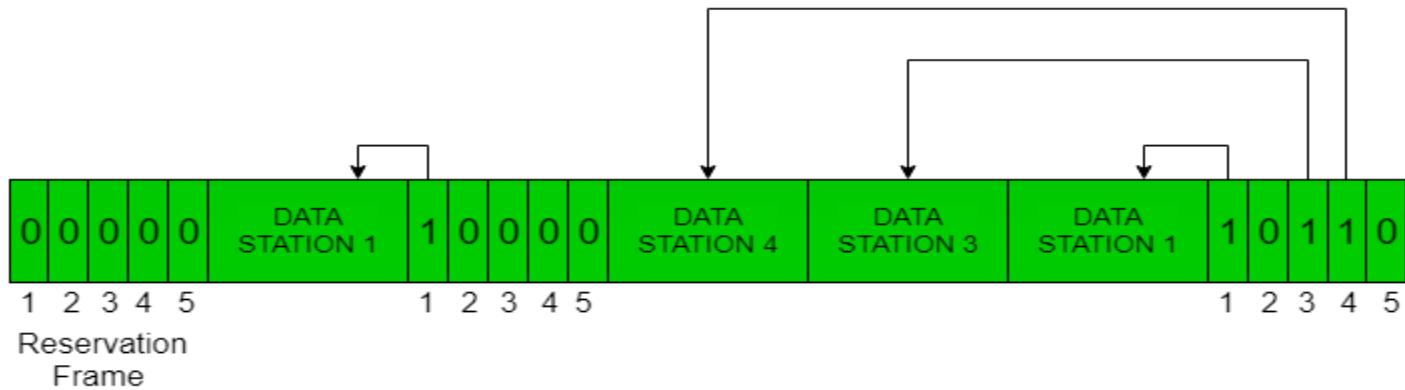
The three controlled-access methods are:

1 Reservation 2 Polling 3 Token Passing

### Reservation

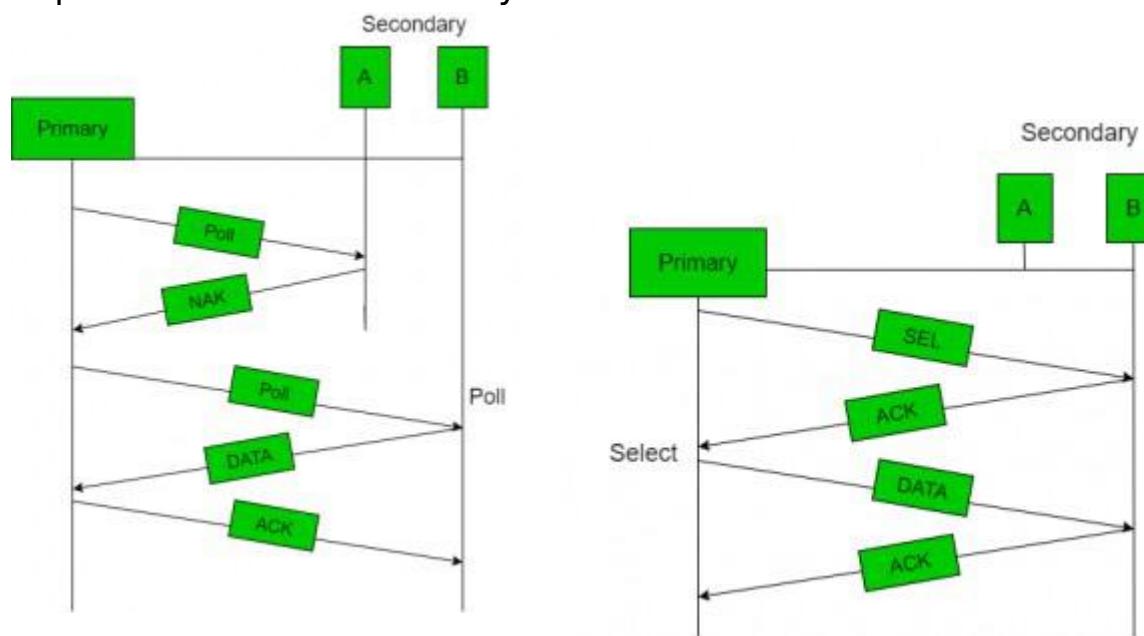
- In the reservation method, a station needs to make a reservation before sending data.
- The time line has two kinds of periods:
  1. Reservation interval of fixed time length
  2. Data transmission period of variable frames.
- If there are M stations, the reservation interval is divided into M slots, and each station has one slot.
- Suppose if station 1 has a frame to send, it transmits 1 bit during the slot 1. No other station is allowed to transmit during this slot.
- In general,  $i^{\text{th}}$  station may announce that it has a frame to send by inserting a 1 bit into  $i^{\text{th}}$  slot. After all N slots have been checked, each station knows which stations wish to transmit.
- The stations which have reserved their slots transfer their frames in that order.
- After data transmission period, next reservation interval begins.
- Since everyone agrees on who goes next, there will never be any collisions.

The following figure shows a situation with five stations and a five slot reservation frame. In the first interval, only stations 1, 3, and 4 have made reservations. In the second interval, only station 1 has made a reservation.



## Polling

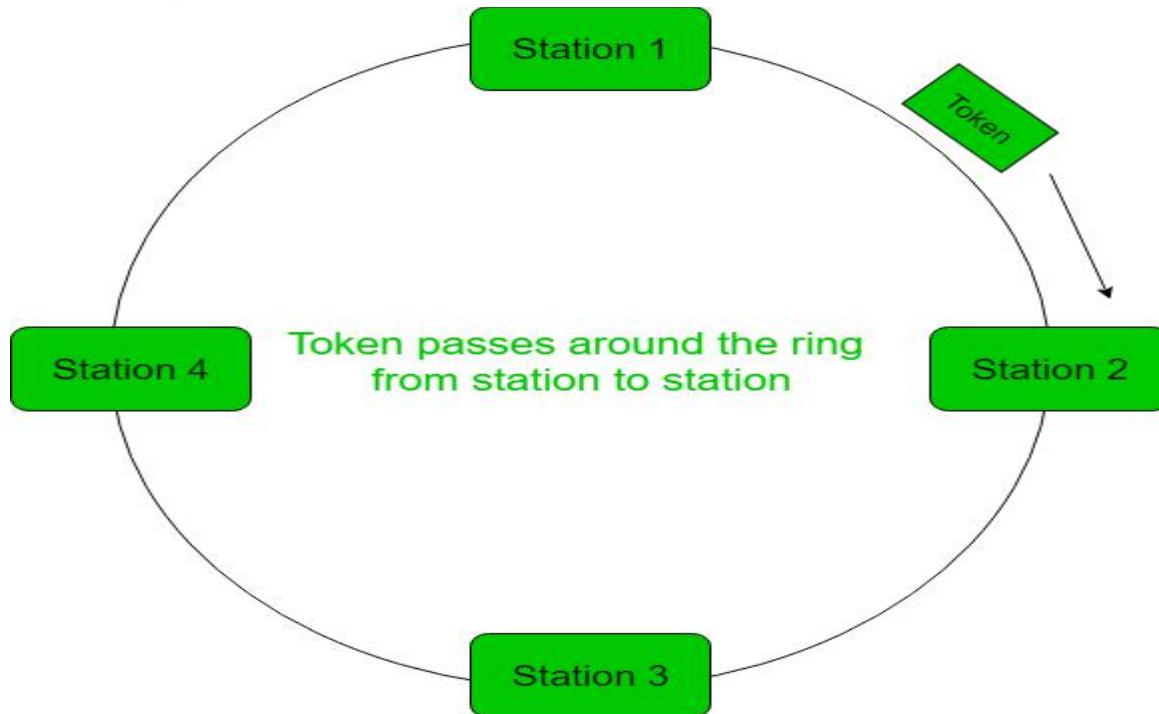
- Polling process is similar to the roll-call performed in class. Just like the teacher, a controller sends a message to each node in turn.
- In this, one acts as a primary station(controller) and the others are secondary stations. All data exchanges must be made through the controller.
- The message sent by the controller contains the address of the node being selected for granting access.
- Although all nodes receive the message but the addressed one responds to it and sends data, if any. If there is no data, usually a “poll reject”(NAK) message is sent back.
- Problems include high overhead of the polling messages and high dependence on the reliability of the controller.



## Token Passing

- In token passing scheme, the stations are connected logically to each other in form of ring and access of stations is governed by tokens.
- A token is a special bit pattern or a small message, which circulate from one station to the next in the some predefined order.

- In Token ring, token is passed from one station to another adjacent station in the ring whereas incase of Token bus, each station uses the bus to send the token to the next station in some predefined order.
- In both cases, token represents permission to send. If a station has a frame queued for transmission when it receives the token, it can send that frame before it passes the token to the next station. If it has no queued frame, it passes the token simply.
- After sending a frame, each station must wait for all N stations (including itself) to send the token to their neighbors and the other N - 1 stations to send a frame, if they have one.
- There exists problems like duplication of token or token is lost or insertion of new station, removal of a station, which need be tackled for correct and reliable operation of this scheme.



## Error Detection

### Error

A condition when the receiver's information does not matches with the sender's information. During transmission, digital signals suffer from noise that can introduce errors in the binary bits travelling from sender to receiver. That means a 0 bit may change to 1 or a 1 bit may change to 0.

### Error Detecting Codes (Implemented either at Data link layer or Transport Layer of OSI Model)

Whenever a message is transmitted, it may get scrambled by noise or data may get corrupted. To avoid this, we use error-detecting codes which are additional data added to a given digital message to help us detect if any error has occurred during transmission of the message.

Basic approach used for error detection is the use of redundancy bits, where

additional bits are added to facilitate detection of errors. Some popular techniques for error detection are:

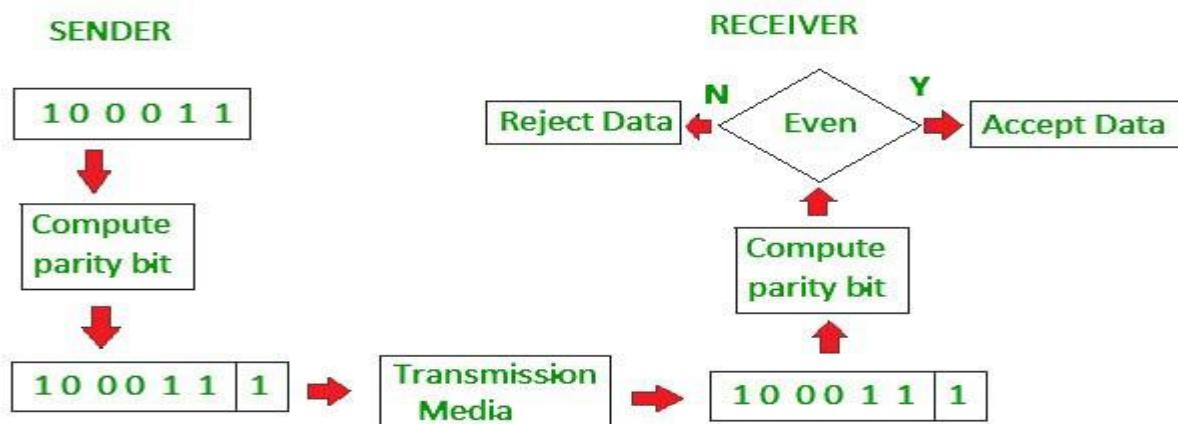
1. Simple Parity check
2. Two-dimensional Parity check
3. Checksum
4. Cyclic redundancy check

### **Simple Parity check**

Blocks of data from the source are subjected to a check bit or parity bit generator form, where a parity of : 1 is added to the block if it contains odd number of 1's, and

0 is added if it contains even number of 1's

This scheme makes the total number of 1's even, that is why it is called even parity checking.



### **Two-dimensional Parity check**

Parity check bits are calculated for each row, which is equivalent to a simple parity check bit. Parity check bits are also calculated for all columns, then both are sent along with the data. At the receiving end these are compared with the parity bits calculated on the received data.

### Original Data

10011001	11100010	00100100	10000100
----------	----------	----------	----------

### Row parities

10011001	0
11100010	0
00100100	0
10000100	0
11011011	0

Column  
parities

### Data to be sent

100110010	111000100	001001000	100001000	110110110
-----------	-----------	-----------	-----------	-----------

## Checksum

- In checksum error detection scheme, the data is divided into  $k$  segments each of  $m$  bits.
- In the sender's end the segments are added using 1's complement arithmetic to get the sum. The sum is complemented to get the checksum.
- The checksum segment is sent along with the data segments.
- At the receiver's end, all received segments are added using 1's complement arithmetic to get the sum. The sum is complemented.
- If the result is zero, the received data is accepted; otherwise discarded.

### Original Data

10011001	11100010	00100100	10000100
----------	----------	----------	----------

$k=4, m=8$

### Receiver

#### Sender

1	10011001
2	11100010
	<u>101111011</u>
	1
	01111100
3	00100100
	<u>10100000</u>
4	10000100
	<u>100100100</u>
	1
Sum:	<u>00100101</u>
CheckSum:	11011010

1 10011001

2 11100010

101111011 1

01111100

00100100

10100000

10000100

100100100 1

00100101

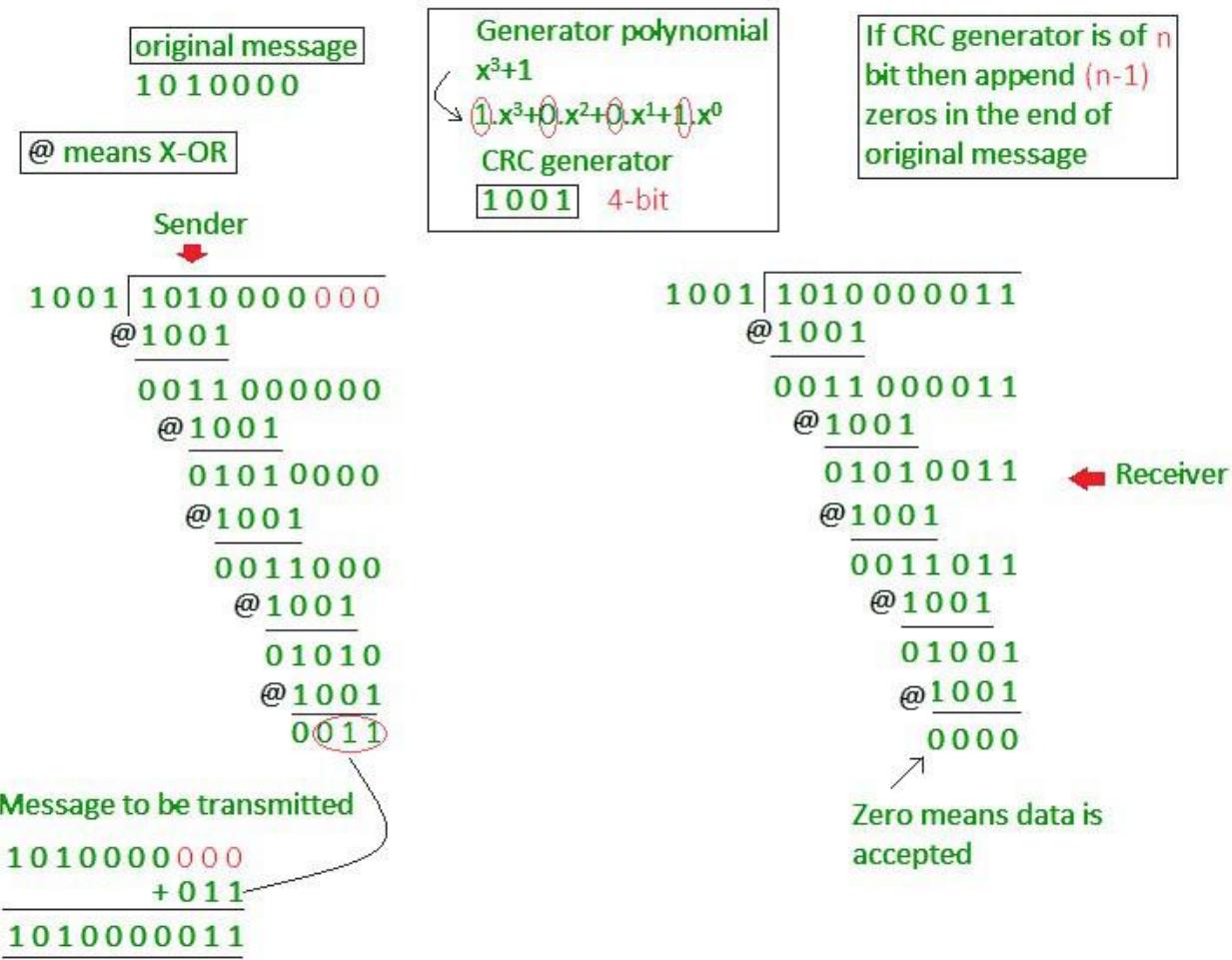
11011010

Sum: 11111111

Complement: 00000000

Conclusion: Accept Data

## Cyclic redundancy check (CRC)



- Unlike checksum scheme, which is based on addition, CRC is based on binary division.
- In CRC, a sequence of redundant bits, called cyclic redundancy check bits, are appended to the end of data unit so that the resulting data unit becomes exactly divisible by a second, predetermined binary number.
- At the destination, the incoming data unit is divided by the same number. If at this step there is no remainder, the data unit is assumed to be correct and is therefore accepted.
- A remainder indicates that the data unit has been damaged in transit and therefore must be rejected.

## Error Correction

Error Correction codes are used to detect and correct the errors when data is transmitted from the sender to the receiver.

Error Correction can be handled in two ways:

**Backward error correction:** Once the error is discovered, the receiver requests the sender to retransmit the entire data unit.

**Forward error correction:** In this case, the receiver uses the error-correcting code which automatically corrects the errors.

A single additional bit can detect the error, but cannot correct it.

For correcting the errors, one has to know the exact position of the error. For example, If we want to calculate a single-bit error, the error correction code will determine which one of seven bits is in error. To achieve this, we have to add some additional redundant bits.

Suppose  $r$  is the number of redundant bits and  $d$  is the total number of the data bits. The number of redundant bits  $r$  can be calculated by using the formula:

$$2^r \geq d+r+1$$

The value of  $r$  is calculated by using the above formula. For example, if the value of  $d$  is 4, then the possible smallest value that satisfies the above relation would be 3.

To determine the position of the bit which is in error, a technique developed by R.W Hamming is Hamming code which can be applied to any length of the data unit and uses the relationship between data units and redundant units.

### **Hamming Code**

**Parity bits:** The bit which is appended to the original data of binary bits so that the total number of 1s is even or odd.

**Even parity:** To check for even parity, if the total number of 1s is even, then the value of the parity bit is 0. If the total number of 1s occurrences is odd, then the value of the parity bit is 1.

**Odd Parity:** To check for odd parity, if the total number of 1s is even, then the value of parity bit is 1. If the total number of 1s is odd, then the value of parity bit is 0.

#### **Algorithm of Hamming code:**

An information of ' $d$ ' bits are added to the redundant bits ' $r$ ' to form  $d+r$ .

The location of each of the  $(d+r)$  digits is assigned a decimal value.

The ' $r$ ' bits are placed in the positions  $1, 2, \dots, 2^{k-1}$

At the receiving end, the parity bits are recalculated. The decimal value of the parity bits determines the position of an error.

#### **Relationship b/w Error position & binary number.**

Error Position	Binary Number
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Let's understand the concept of Hamming code through an example:

Suppose the original data is 1010 which is to be sent.

Total number of data bits ' $d$ ' = 4

Number of redundant bits  $r$  :  $2^r \geq d+r+1$

$$2^r \geq 4+r+1$$

Therefore, the value of r is 3 that satisfies the above relation.

Total number of bits = d+r = 4+3 = 7;

### Determining the position of the redundant bits

The number of redundant bits is 3. The three bits are represented by r1, r2, r4.

The position of the redundant bits is calculated with corresponds to the raised power of 2. Therefore, their corresponding positions are 1,  $2^1$ ,  $2^2$ .

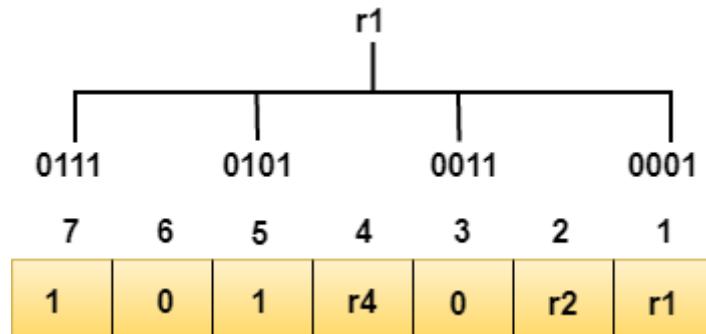
The position of r1 = 1, The position of r2 = 2 , The position of r4 = 4

Representation of Data on the addition of parity bits:

7	6	5	4	3	2	1
1	0	1	r4	0	r2	r1

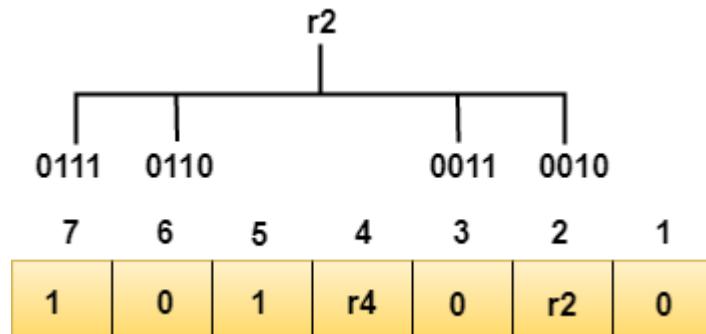
### Determining the Parity bits

**Determining the r1 bit:** The r1 bit is calculated by performing a parity check on the bit positions whose binary representation includes 1 in the first position.



We observe from the above figure that the bit position that includes 1 in the first position are 1, 3, 5, 7. Now, we perform the even-parity check at these bit positions. The total number of 1 at these bit positions corresponding to r1 is even, therefore, the value of the r1 bit is 0.

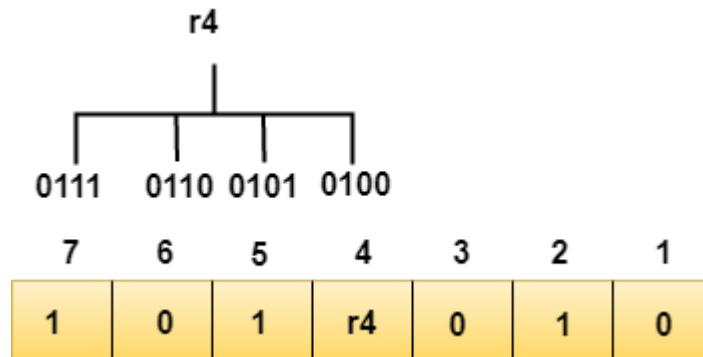
**Determining r2 bit:** The r2 bit is calculated by performing a parity check on the bit positions whose binary representation includes 1 in the second position



We observe from the above figure that the bit positions that includes 1 in the second position are 2, 3, 6, 7. Now, we perform the even-parity check at these

bit positions. The total number of 1 at these bit positions corresponding to r2 is odd, therefore, the value of the r2 bit is 1.

**Determining r4 bit:** The r4 bit is calculated by performing a parity check on the bit positions whose binary representation includes 1 in the third position.



We observe from the above figure that the bit positions that includes 1 in the third position are 4, 5, 6, 7. Now, we perform the even-parity check at these bit positions. The total number of 1 at these bit positions corresponding to r4 is even, therefore, the value of the r4 bit is 0.

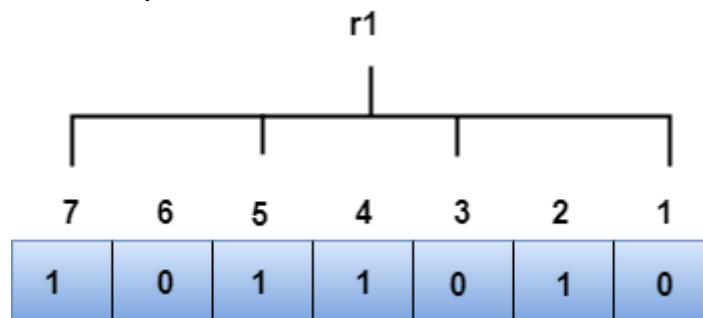
**Data transferred is given below:**

7	6	5	4	3	2	1
1	0	1	0	0	1	0

Suppose the 4th bit is changed from 0 to 1 at the receiving end, then parity bits are recalculated.

### R1 bit

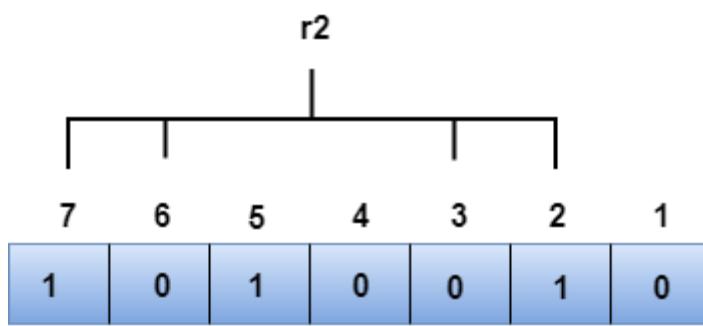
The bit positions of the r1 bit are 1,3,5,7



We observe from the above figure that the binary representation of r1 is 1100. Now, we perform the even-parity check, the total number of 1s appearing in the r1 bit is an even number. Therefore, the value of r1 is 0.

### R2 bit

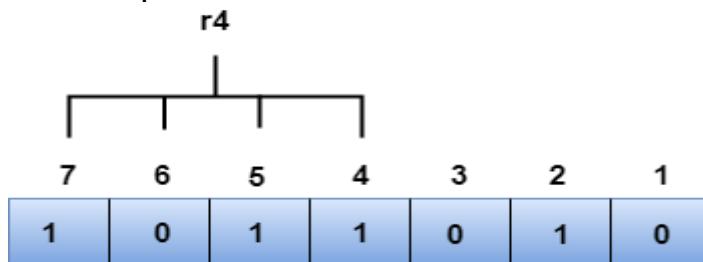
The bit positions of r2 bit are 2,3,6,7.



We observe from the above figure that the binary representation of  $r_2$  is 1001. Now, we perform the even-parity check, the total number of 1s appearing in the  $r_2$  bit is an even number. Therefore, the value of  $r_2$  is 0.

### R4 bit

The bit positions of  $r_4$  bit are 4,5,6,7.



We observe from the above figure that the binary representation of  $r_4$  is 1011. Now, we perform the even-parity check, the total number of 1s appearing in the  $r_4$  bit is an odd number. Therefore, the value of  $r_4$  is 1.

The binary representation of redundant bits, i.e.,  $r_4r_2r_1$  is 100, and its corresponding decimal value is 4. Therefore, the error occurs in a 4th bit position. The bit value must be changed from 1 to 0 to correct the error.

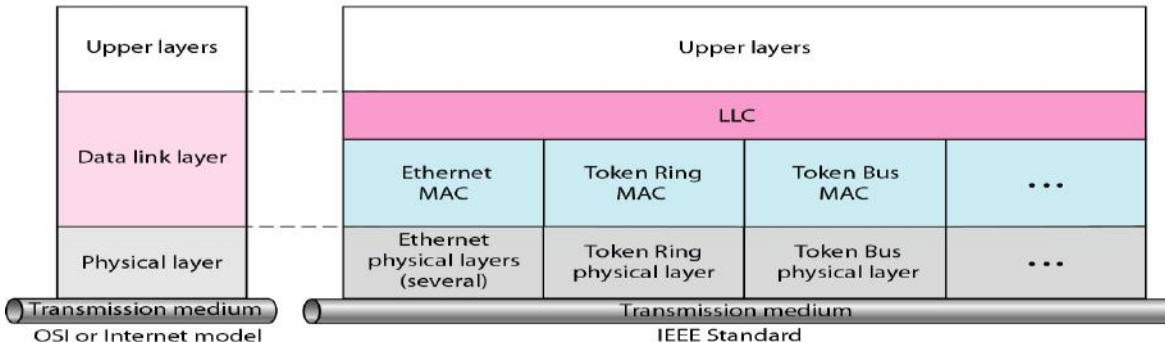
## Wired LANs: Ethernet

In 1985, the Computer Society of the IEEE started a project, called Project 802, to set standards to enable intercommunication among equipment from a variety of manufacturers. Project 802 is a way of specifying functions of the physical layer and the data link layer of major LAN protocols.

The relationship of the 802 Standard to the traditional OSI model is shown in below Figure. The IEEE has subdivided the data link layer into two sub layers: logical link control (LLC) and media access control).

IEEE has also created several physical layer standards for different LAN protocols

LLC: Logical link control  
MAC: Media access control



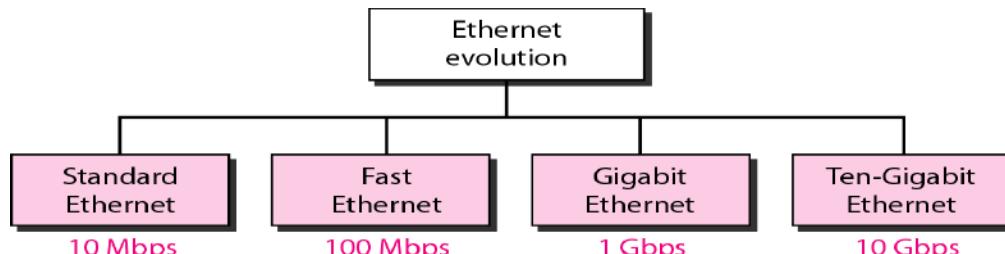
### *IEEE standard for LANs*

## STANDARD ETHERNET

The original Ethernet was created in 1976 at Xerox's Palo Alto Research Center (PARC). Since then, it has gone through four generations.

Standard Ethernet (10 Mbps), Fast Ethernet (100 Mbps), Gigabit Ethernet (1 Gbps), and Ten-Gigabit Ethernet (10 Gbps),

We briefly discuss the Standard (or traditional) Ethernet in this section



*Ethernet evolution through four generations*

## MAC Sublayer

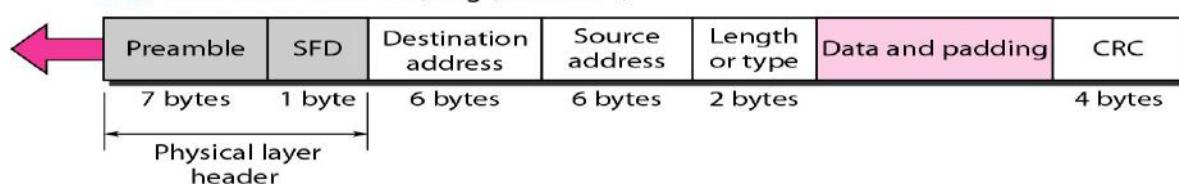
In Standard Ethernet, the MAC sublayer governs the operation of the access method. It also frames data received from the upper layer and passes them to the physical layer.

## Frame Format

The Ethernet frame contains seven fields: preamble, SFD, DA, SA, length or type of protocol data unit (PDU), upper-layer data, and the CRC. Ethernet does not provide any mechanism for acknowledging received frames, making it what is known as an unreliable medium. Acknowledgments must be implemented at the higher layers. The format of the MAC frame is shown in below figure

Preamble: 56 bits of alternating 1s and 0s.

SFD: Start frame delimiter, flag (10101011)



## 802.3 MAC frame

Preamble. The first field of the 802.3 frame contains 7 bytes (56 bits) of alternating 0s and 1s that alerts the receiving system to the coming frame and enables it to synchronize its input timing. The pattern provides only an alert and a timing pulse. The 56-bit pattern allows the stations to miss some bits at the beginning of the frame. The preamble is actually added at the physical layer and is not (formally) part of the frame.

Start frame delimiter (SFD). The second field (1 byte: 10101011) signals the beginning of the frame. The SFD warns the station or stations that this is the last chance for synchronization. The last 2 bits is 11 and alerts the receiver that the next field is the destination address.

Destination address (DA). The DA field is 6 bytes and contains the physical address of the destination station or stations to receive the packet.

Source address (SA). The SA field is also 6 bytes and contains the physical address of the sender of the packet.

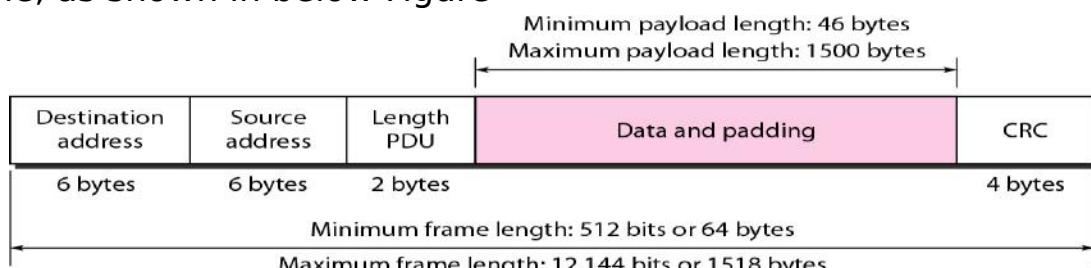
Length or type. This field is defined as a type field or length field. The original Ethernet used this field as the type field to define the upper-layer protocol using the MAC frame. The IEEE standard used it as the length field to define the number of bytes in the data field. Both uses are common today.

Data. This field carries data encapsulated from the upper-layer protocols. It is a minimum of 46 and a maximum of 1500 bytes.

CRC. The last field contains error detection information, in this case a CRC-32

### **Frame Length**

Ethernet has imposed restrictions on both the minimum and maximum lengths of a frame, as shown in below Figure



*Minimum and maximum lengths*

An Ethernet frame needs to have a minimum length of 512 bits or 64 bytes. Part of this length is the header and the trailer. If we count 18 bytes of header and trailer (6 bytes of source address, 6 bytes of destination address, 2 bytes of length or type, and 4 bytes of CRC), then the minimum length of data from the upper layer is  $64 - 18 = 46$  bytes. If the upper-layer packet is less than 46 bytes, padding is added to make up the difference

The standard defines the maximum length of a frame (without preamble and SFD field) as 1518 bytes. If we subtract the 18 bytes of header and trailer,

the maximum length of the payload is 1500 bytes.

The maximum length restriction has two historical reasons.

First, memory was very expensive when Ethernet was designed: a maximum length restriction helped to reduce the size of the buffer.

Second, the maximum length restriction prevents one station from monopolizing the shared medium, blocking other stations that have data to send.

### **Addressing**

The Ethernet address is 6 bytes (48 bits), normally written in hexadecimal notation, with a colon between the bytes.

***Example of an Ethernet address in hexadecimal notation***

**06 : 01 : 02 : 01 : 2C : 4B**

A diagram showing the conversion of an Ethernet address. It consists of six horizontal boxes labeled 'Byte 1' through 'Byte 6'. Above Byte 1, there is a note: 'Unicast: 0; multicast: 1'. A vertical line with a dot at its intersection with Byte 1 indicates the bit position. Below the boxes, a bracket spans all six bytes with the text '6 bytes = 12 hex digits = 48 bits'.

**6 bytes = 12 hex digits = 48 bits**

Unicast, Multicast, and Broadcast Addresses A source address is always a unicast address-the frame comes from only one station. The destination address, however, can be **unicast, multicast, or broadcast**. Below Figure shows how to distinguish a unicast address from a multicast address.

If the least significant bit of the first byte in a destination address is 0, the address is unicast; otherwise, it is multicast.



### *Unicast and multicast addresses*

A unicast destination address defines only one recipient; the relationship between the sender and the receiver is one-to-one.

A multicast destination address defines a group of addresses; the relationship between the sender and the receivers is one-to-many.

The broadcast address is a special case of the multicast address; the recipients are all the stations on the LAN. A broadcast destination address is forty-eight 1s.

### **Access Method: CSMA/CD**

Standard Ethernet uses I-persistent CSMA/CD

### **Slot Time In an Ethernet network.**

Slot time =round-trip time + time required to send the jam sequence

The slot time in Ethernet is defined in bits. It is the time required for a station

to send 512 bits. This means that the actual slot time depends on the data rate; for traditional 10-Mbps Ethernet it is 51.2 micro sec.

Slot Time and Maximum Network Length There is a relationship between the slot time and the maximum length of the network (collision domain). It is dependent on the propagation speed of the signal in the particular medium.

In most transmission media, the signal propagates at  $2 \times 10^8$  m/s (two-thirds of the rate for propagation in air).

For traditional Ethernet, we calculate

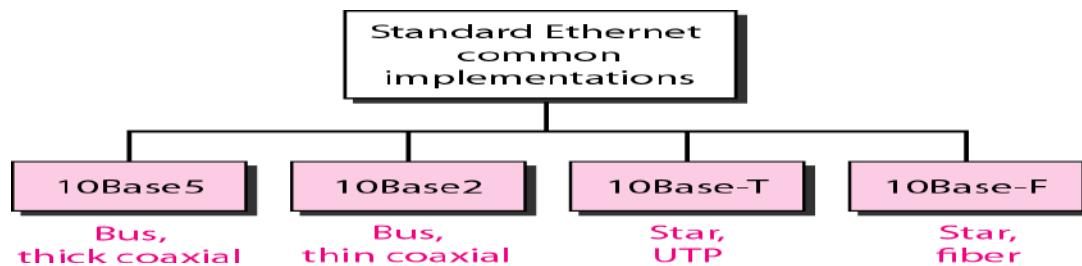
$$\text{MaxLength} = \text{PropagationSpeed} \times (\text{SlotTime}/2)$$

$$\text{MaxLength} = (2 \times 10^8) \times (51.2 \times 10^{-6})/2 = 5120\text{m}$$

Of course, we need to consider the delay times in repeaters and interfaces, and the time required to send the jam sequence. These reduce the maximum-length of a traditional Ethernet network to 2500 m, just 48 percent of the theoretical calculation.  $\text{MaxLength}=2500\text{m}$

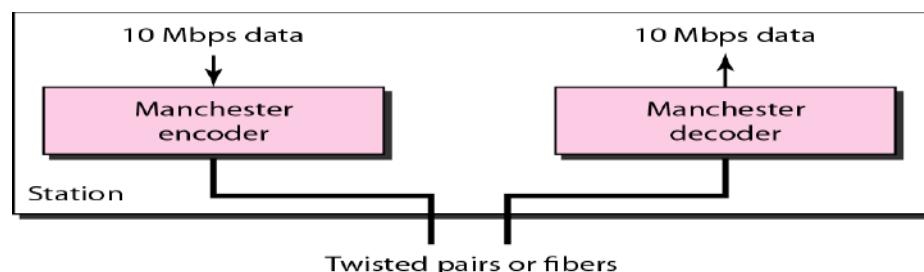
## **Physical Layer**

The Standard Ethernet defines several physical layer implementations; four of the most common, are shown in Figure

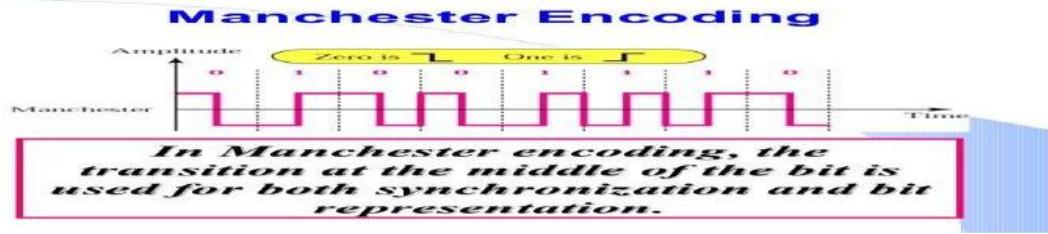


## **Encoding and Decoding**

All standard implementations use digital signaling (baseband) at 10 Mbps. At the sender, data are converted to a digital signal using the Manchester scheme; at the receiver, the received signal is interpreted as Manchester and decoded into data. Manchester encoding is self-synchronous, providing a transition at each bit interval. Figure shows the encoding scheme for Standard Ethernet

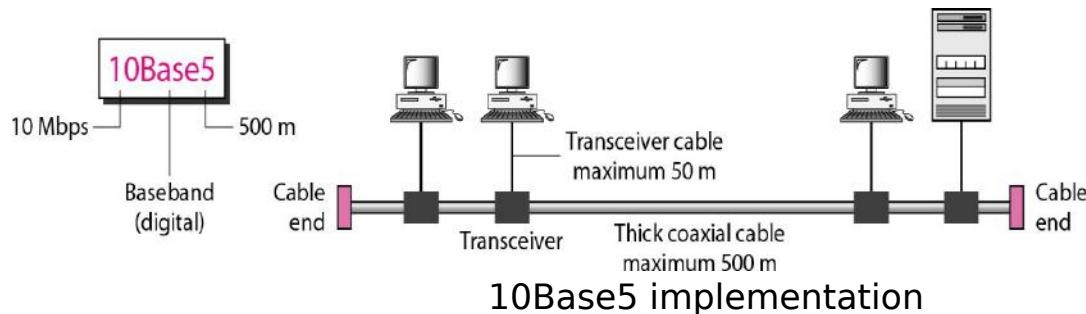


In Manchester encoding, the transition at the middle of the bit is used for synchronization



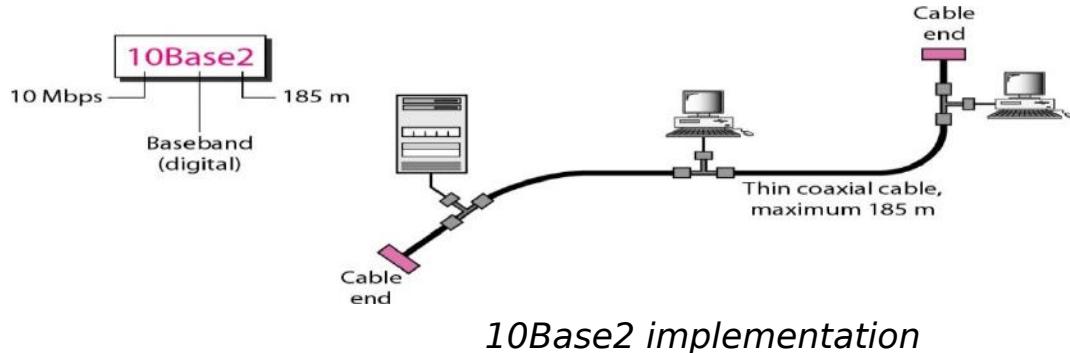
### IOBase5: Thick Ethernet

The first implementation is called **10Base5, thick Ethernet, or Thicknet**. IOBase5 was the first Ethernet specification to use a bus topology with an external **transceiver** (transmitter/receiver) connected via a tap to a thick coaxial cable. Figure shows a schematic diagram of a IOBase5 implementation



### 10Base2: Thin Ethernet

The second implementation is called 10 Base2, **thin Ethernet**, or Cheapernet. 10Base2 also uses a bus topology, but the cable is much thinner and more flexible. Figure shows the schematic diagram of a 10Base2 implementation.



thin coaxial cable is less expensive than thick coaxial.

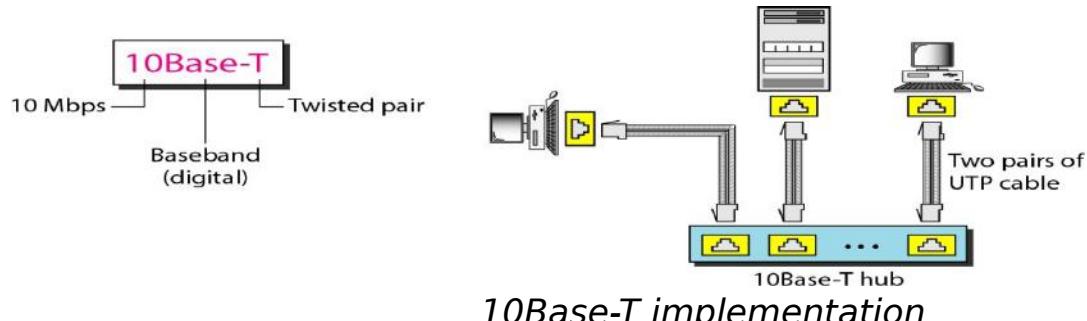
Installation is simpler because the thin coaxial cable is very flexible.

However, the length of each segment cannot exceed 185 m (close to 200 m) due to the high level of attenuation in thin coaxial cable.

### 10Base-T: Twisted-Pair Ethernet

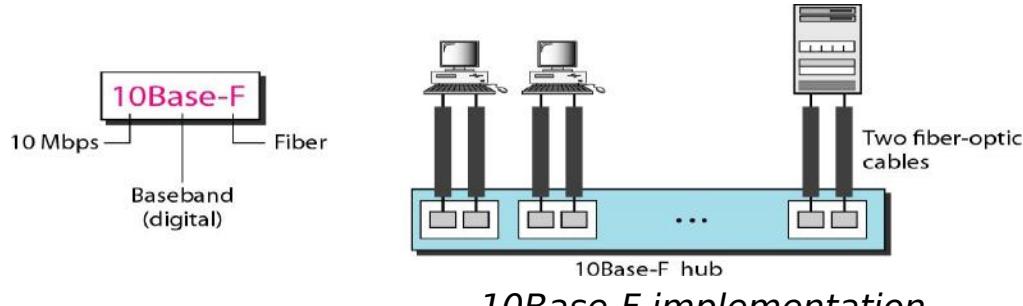
The third implementation is called 10Base-T or twisted-pair Ethernet. It uses a physical star topology. The stations are connected to a hub via two pairs of twisted cable, as shown in Figure

The maximum length of the twisted cable here is defined as 100 m, to minimize the effect of attenuation in the twisted cable



*10Base-T implementation*

Although there are several types of optical fiber 10-Mbps Ethernet, the most common is called 10Base-F. 10Base-F uses a star topology to connect stations to a hub. The stations are connected to the hub using two fiber-optic cables, as shown in Figure



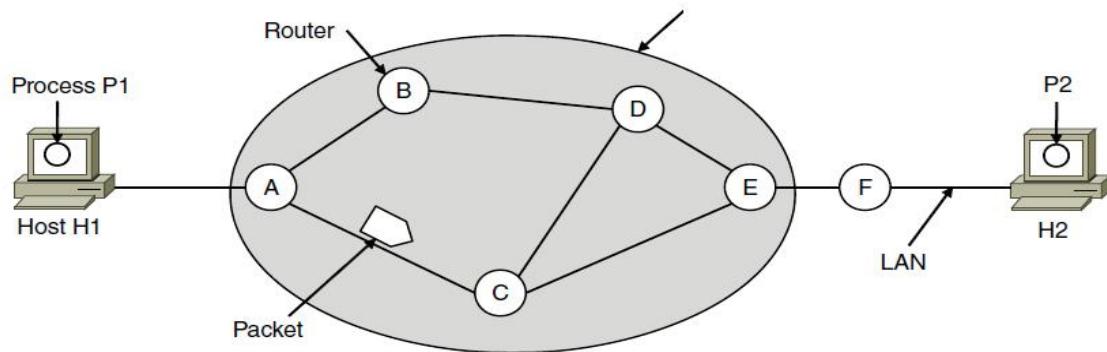
*10Base-F implementation*

## UNIT-III

### Network Layer Design Issues

1. Store-and-forward packet switching
2. Services provided to transport layer
3. Implementation of connectionless service
4. Implementation of connection-oriented service
5. Comparison of virtual-circuit and datagram networks

### 1 Store-and-forward packet switching



A host with a packet to send transmits it to the nearest router, either on its own LAN or over a point-to-point link to the ISP. The packet is stored there until it has fully arrived and the link has finished its processing by verifying the checksum. Then it is forwarded to the next router along the path until it reaches the destination host, where it is delivered. This mechanism is store-and-forward packet switching.

### 2 Services provided to transport layer

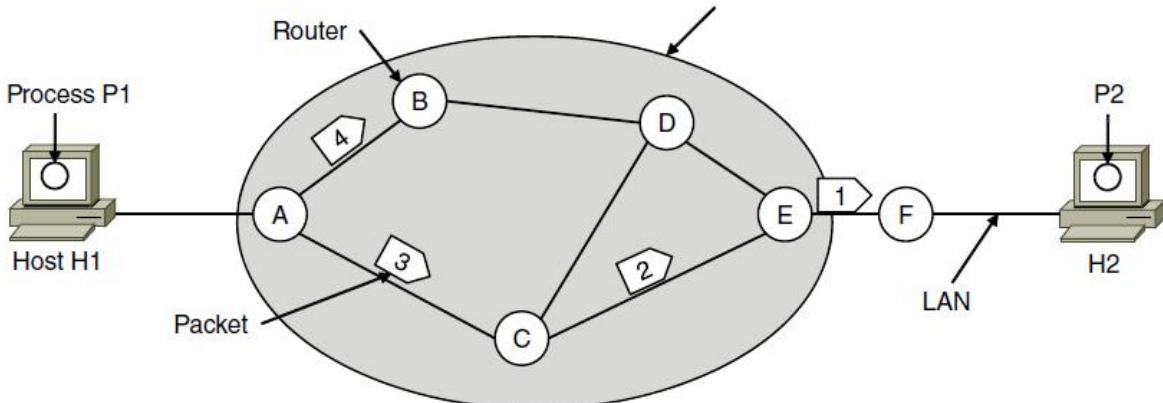
The network layer provides services to the transport layer at the network layer/transport layer interface. The services need to be carefully designed with the following goals in mind:

1. Services independent of router technology.
2. Transport layer shielded from number, type, topology of routers.
3. Network addresses available to transport layer use uniform numbering plan
  - even across LANs and WANs

### 3 Implementation of connectionless service

If connectionless service is offered, packets are injected into the network individually and routed independently of each other. No advance setup is needed. In this context, the packets

are frequently called **datagrams** (in analogy with telegrams) and the network is called a **datagram network**.



A's table (initially)

A	X
B	B
C	C
D	B
E	C
F	C

Dest. Line

A's table (later)

A	X
B	B
C	C
D	B
E	D
F	D

C's Table

A	A
B	A
C	X
D	E
E	E
F	E

E's Table

A	C
B	D
C	C
D	D
E	X
F	F

Let us assume for this example that the message is four times longer than the maximum packet size, so the network layer has to break it into four packets, 1, 2, 3, and 4, and send each of them in turn to router A.

Every router has an internal table telling it where to send packets for each of the possible destinations. Each table entry is a pair(destination and the outgoing line). Only directly connected lines can be used.

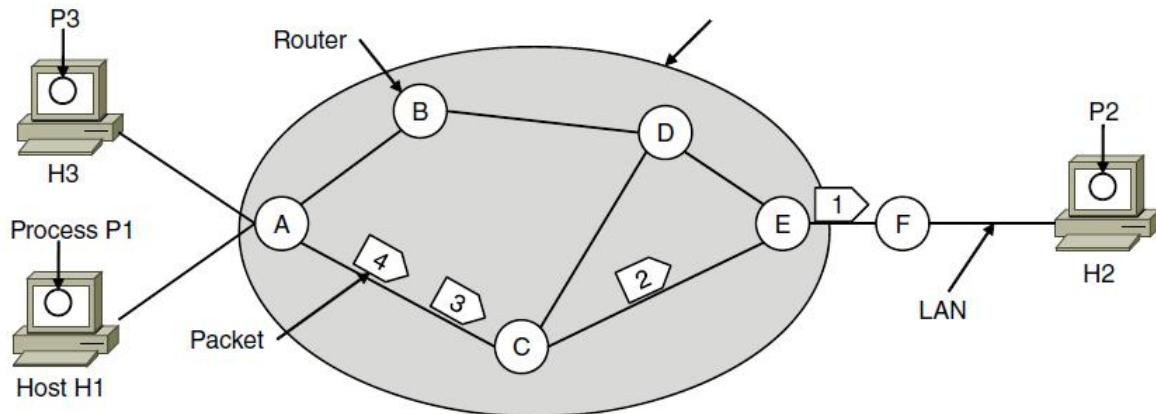
A's initial routing table is shown in the figure under the label "initially."

At A, packets 1, 2, and 3 are stored briefly, having arrived on the incoming link. Then each packet is forwarded according to A's table, onto the outgoing link to C within a new frame. Packet 1 is then forwarded to E and then to F.

However, something different happens to packet 4. When it gets to A it is sent to router B, even though it is also destined for F. For some reason (traffic jam along ACE path), A decided to send packet 4 via a different route than that of the first three packets. Router A updated its routing table, as shown under the label "later."

The algorithm that manages the tables and makes the routing decisions is called the **routing algorithm**.

## 4 Implementation of connection-oriented service



A's table	C's Table	E's Table
H1   1 H3   1	A   1 C   2	C   1 E   1
In      Out	A   2 E   2	C   2 F   1

If connection-oriented service is used, a path from the source router all the way to the destination router must be established before any data packets can be sent. This connection is called a **VC (virtual circuit)**, and the network is called a **virtual-circuit network**

When a connection is established, a route from the source machine to the destination machine is chosen as part of the connection setup and stored in tables inside the routers. That route is used for all traffic flowing over the connection, exactly the same way that the telephone system works. When the connection is released, the virtual circuit is also terminated. With connection-oriented service, each packet carries an identifier telling which virtual circuit it belongs to.

As an example, consider the situation shown in Figure. Here, host *H1* has established connection 1 with host *H2*. This connection is remembered as the first entry in each of the routing tables. The first line of *A*'s table says that if a packet bearing connection identifier 1 comes in from *H1*, it is to be sent to router *C* and given connection identifier 1. Similarly, the first entry at *C* routes the packet to *E*, also with connection identifier 1.

Now let us consider what happens if *H3* also wants to establish a connection to *H2*. It chooses connection identifier 1 (because it is initiating the connection and this is its only connection) and tells the network to establish the virtual circuit.

This leads to the second row in the tables. Note that we have a conflict here because although A can easily distinguish connection 1 packets from  $H_1$  from connection 1 packets from  $H_3$ , C cannot do this. For this reason, A assigns a different connection identifier to the outgoing traffic for the second connection. Avoiding conflicts of this kind is why routers need the ability to replace connection identifiers in outgoing packets.

In some contexts, this process is called **label switching**. An example of a connection-oriented network service is **MPLS (Multi Protocol Label Switching)**.

## 5 Comparison of virtual-circuit and datagram networks

Issue	Datagram network	Virtual-circuit network
Circuit setup	Not needed	Required
Addressing	Each packet contains the full source and destination address	Each packet contains a short VC number
State information	Routers do not hold state information about connections	Each VC requires router table space per connection
Routing	Each packet is routed independently	Route chosen when VC is set up; all packets follow it
Effect of router failures	None, except for packets lost during the crash	All VCs that passed through the failed router are terminated
Quality of service	Difficult	Easy if enough resources can be allocated in advance for each VC
Congestion control	Difficult	Easy if enough resources can be allocated in advance for each VC

---

## Routing Algorithms

The main function of NL (Network Layer) is routing packets from the source machine to the destination machine.

There are two processes inside router:

- a) One of them handles each packet as it arrives, looking up the outgoing line to use for it in the routing table. This process is **forwarding**.
- b) The other process is responsible for filling in and updating the routing tables. That is where the routing algorithm comes into play. This process is **routing**.

Regardless of whether routes are chosen independently for each packet or only when new connections are established, certain properties are desirable in a routing algorithm **correctness, simplicity, robustness, stability, fairness, optimality**

Routing algorithms can be grouped into two major classes:

- 1) nonadaptive (Static Routing)
- 2) adaptive. (Dynamic Routing)

Nonadaptive algorithm do not base their routing decisions on measurements or estimates of the current traffic and topology. Instead, the choice of the route to use to get from I to J is computed in advance, off line, and downloaded to the routers when the network is booted. This procedure is sometimes called static routing.

Adaptive algorithm, in contrast, change their routing decisions to reflect changes in the topology, and usually the traffic as well.

Adaptive algorithms differ in

- 1) Where they get their information (e.g., locally, from adjacent routers, or from all routers),
- 2) When they change the routes (e.g., every  $\Delta T$  sec, when the load changes or when the topology changes), and
- 3) What metric is used for optimization (e.g., distance, number of hops, or estimated transit time).

This procedure is called dynamic routing

### Different Routing Algorithms

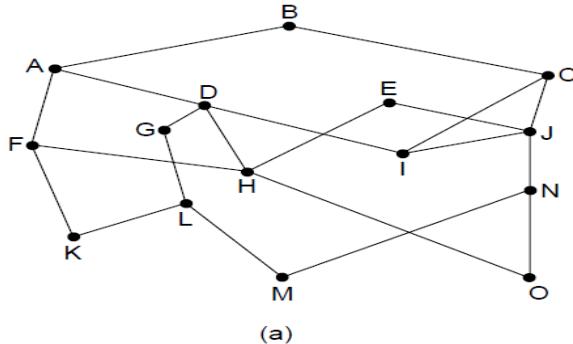
- Optimality principle
- Shortest path algorithm
- Flooding
- Distance vector routing
- Link state routing
- Hierarchical Routing

### The Optimality Principle

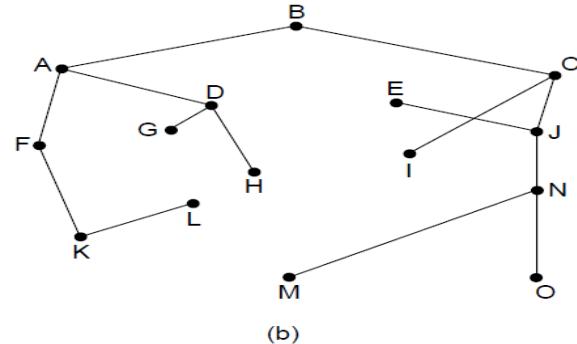
One can make a general statement about optimal routes without regard to network topology or traffic. This statement is known as the optimality principle.

It states that if router J is on the optimal path from router I to router K, then the optimal path from J to K also falls along the same

As a direct consequence of the optimality principle, we can see that the set of optimal routes from all sources to a given destination form a tree rooted at the destination. Such a tree is called a **sink tree**. The goal of all routing algorithms is to discover and use the sink trees for all routers



(a) A network.



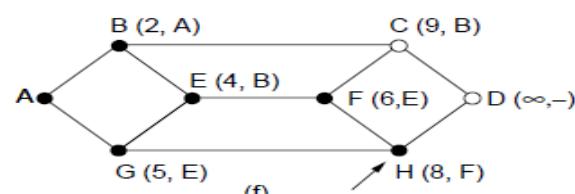
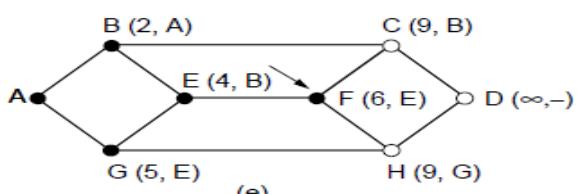
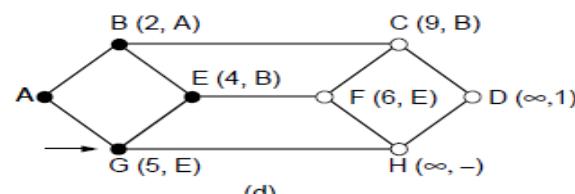
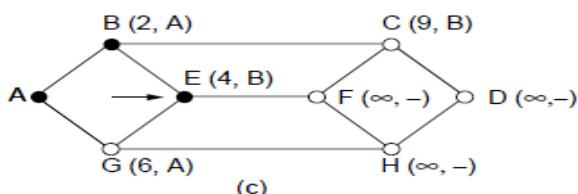
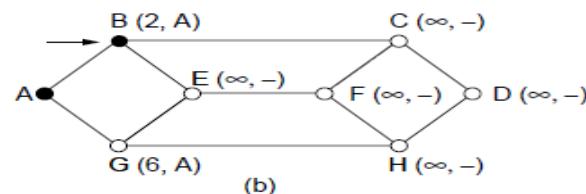
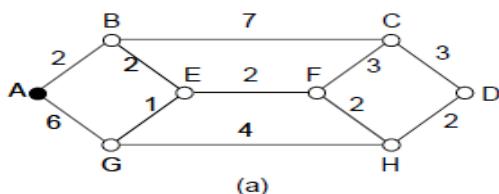
(b) A sink tree for router  $B$ .

### Shortest Path Routing (Dijkstra's)

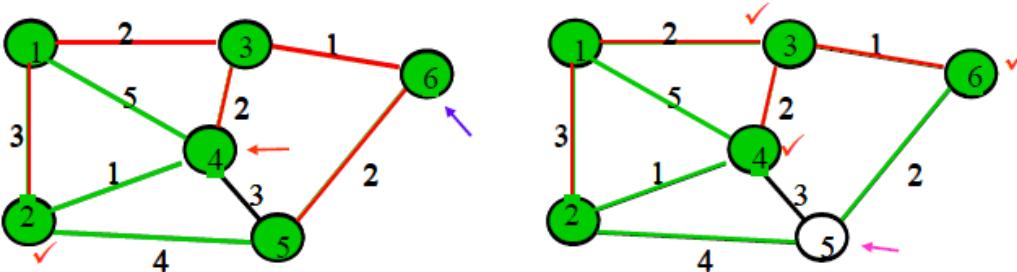
The idea is to build a graph of the subnet, with each node of the graph representing a router and each arc of the graph representing a communication line or link.

To choose a route between a given pair of routers, the algorithm just finds the shortest path between them on the graph

1. Start with the local node (router) as the root of the tree. Assign a cost of 0 to this node and make it the first permanent node.
2. Examine each neighbor of the node that was the last permanent node.
3. Assign a cumulative cost to each node and make it tentative
4. Among the list of tentative nodes
  - a. Find the node with the smallest cost and make it Permanent
  - b. If a node can be reached from more than one route then select the route with the shortest cumulative cost.
5. Repeat steps 2 to 4 until every node becomes permanent



# Execution of Dijkstra's algorithm



Iteration	Permanent	tentative	D <sub>2</sub>	D <sub>3</sub>	D <sub>4</sub>	D <sub>5</sub>	D <sub>6</sub>
Initial	{1}	{2,3,4}	3	2 ✓	5	∞	∞
1	{1,3}	{2,4,6}	3 ✓	2	4	∞	3
2	{1,2,3}	{4,6,5}	3	2	4	7	3 ✓
3	{1,2,3,6}	{4,5}	3	2	4 ✓	5	3
4	{1,2,3,4,6}	{5}	3	2	4	5 ✓	3
5	{1,2,3,4,5,6}	{}	3	2	4	5	3

## Flooding

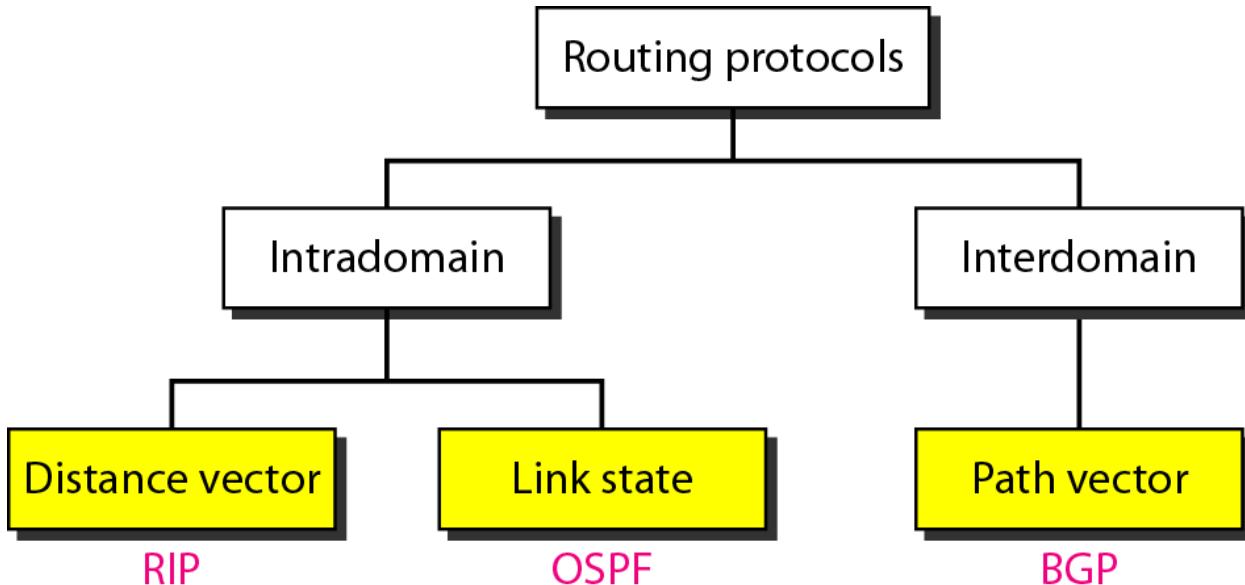
- Another static algorithm is flooding, in which every incoming packet is sent out on every outgoing line except the one it arrived on.
- Flooding obviously generates vast numbers of duplicate packets, in fact, an infinite number unless some measures are taken to damp the process.
- One such measure is to have a hop counter contained in the header of each packet, which is decremented at each hop, with the packet being discarded when the counter reaches zero. Ideally, the hop counter should be initialized to the length of the path from source to destination.
- A variation of flooding that is slightly more practical is selective flooding. In this algorithm the routers do not send every incoming packet out on every line, only on those lines that are going approximately in the right direction.
- Flooding is not practical in most applications.

## Intra- and Inter domain Routing

An autonomous system (AS) is a group of networks and routers under the authority of a single administration.

Routing inside an autonomous system is referred to as intra domain routing. (DISTANCE VECTOR, LINK STATE)

Routing between autonomous systems is referred to as inter domain routing. (PATH VECTOR)  
 Each autonomous system can choose one or more intra domain routing protocols to handle routing inside the autonomous system. However, only one inter domain routing protocol handles routing between autonomous systems.



### Distance Vector Routing

In distance vector routing, the least-cost route between any two nodes is the route with minimum distance. In this protocol, as the name implies, each node maintains a vector (table) of minimum distances to every node.

Mainly 3 things in this

*Initialization*

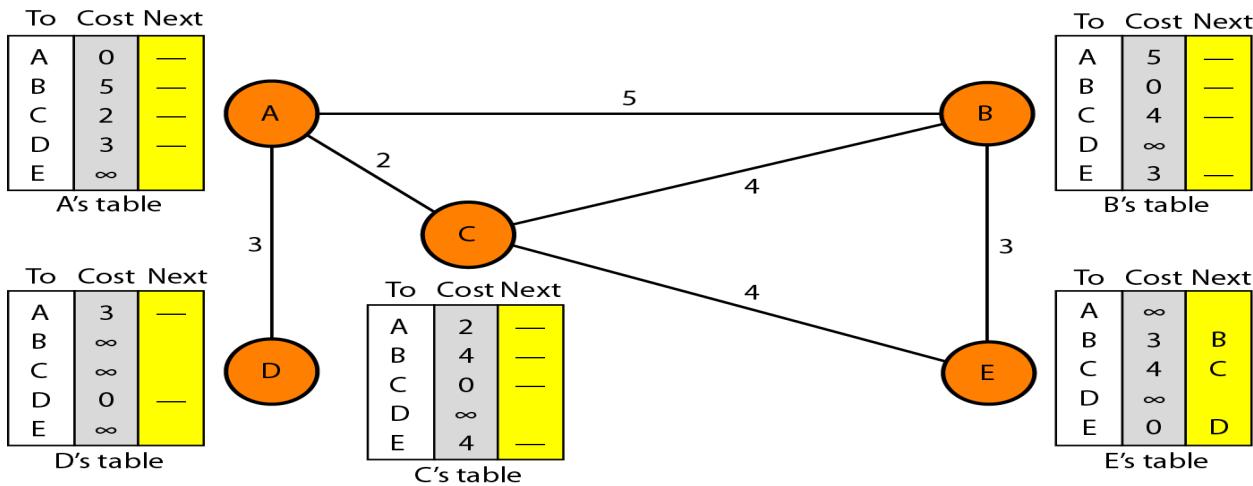
*Sharing*

*Updating*

### ***Initialization***

Each node can know only the distance between itself and its immediate neighbors, those directly connected to it. So for the moment, we assume that each node can send a message to the immediate neighbors and find the distance between itself and these neighbors. Below fig shows the initial tables for each node. The distance for any entry that is not a neighbor is marked as infinite (unreachable).

## Initialization of tables in distance vector routing



## Sharing

The whole idea of distance vector routing is the sharing of information between neighbors. Although node A does not know about node E, node C does. So if node C shares its routing table with A, node A can also know how to reach node E. On the other hand, node C does not know how to reach node D, but node A does. If node A shares its routing table with node C, node C also knows how to reach node D. In other words, nodes A and C, as immediate neighbors, can improve their routing tables if they help each other.

NOTE: In distance vector routing, each node shares its routing table with its immediate neighbors periodically and when there is a change

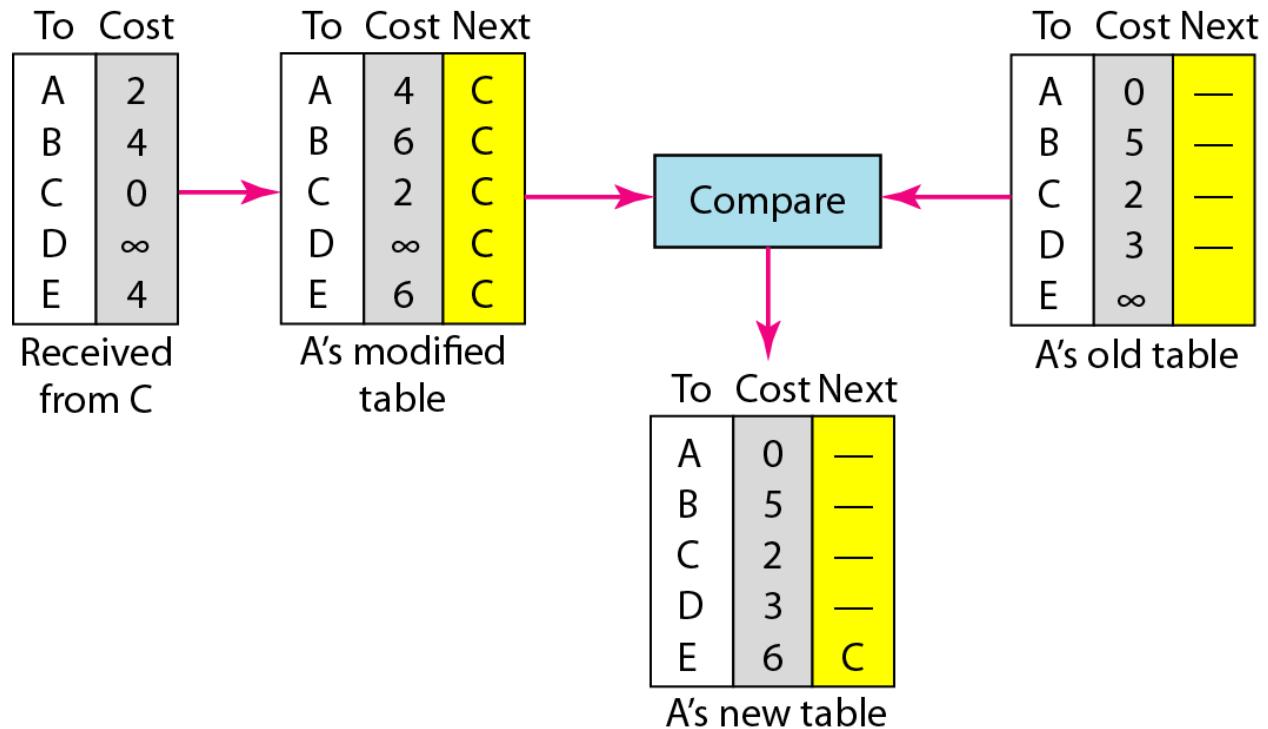
## Updating

When a node receives a two-column table from a neighbor, it needs to update its routing table. Updating takes three steps:

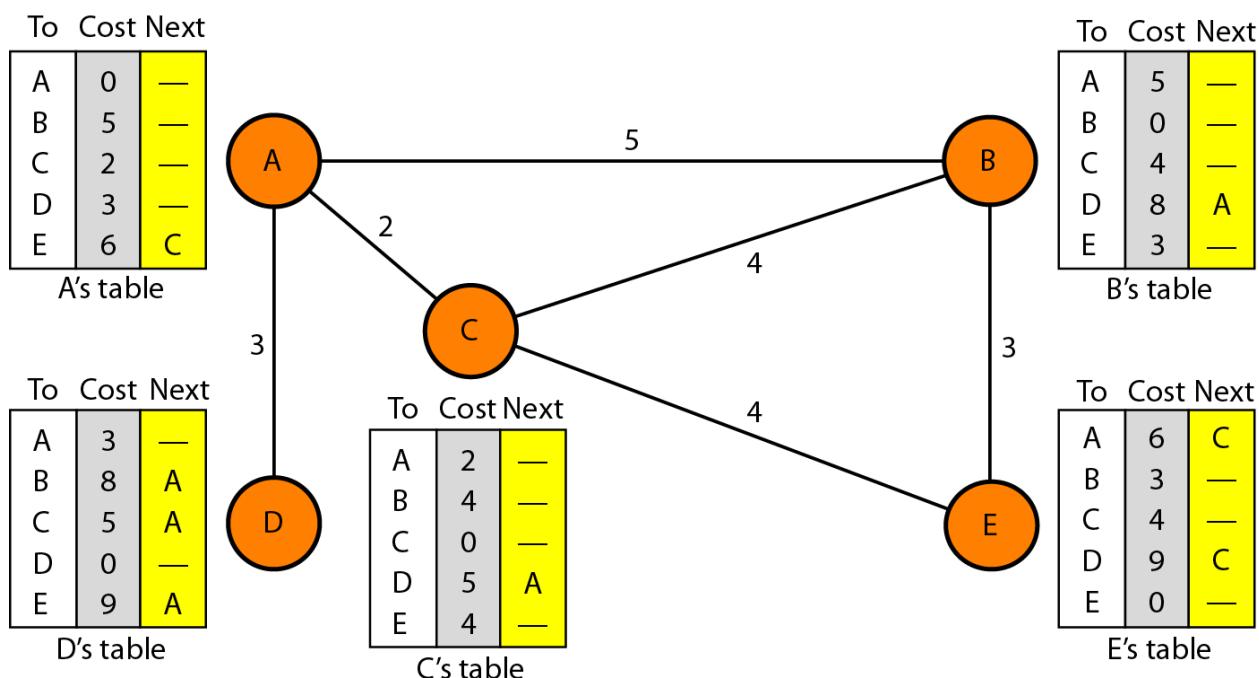
1. The receiving node needs to add the cost between itself and the sending node to each value in the second column. ( $x+y$ )
2. If the receiving node uses information from any row. The sending node is the next node in the route.
3. The receiving node needs to compare each row of its old table with the corresponding row of the modified version of the received table.
  - a. If the next-node entry is different, the receiving node chooses the row with the smaller cost. If there is a tie, the old one is kept.
  - b. If the next-node entry is the same, the receiving node chooses the new row.

For example, suppose node C has previously advertised a route to node X with distance 3. Suppose that now there is no path between C and X; node C now advertises this route with a distance of infinity. Node A must not ignore this value even though its old entry is smaller. The old route does not exist anymore. The new route has a distance of infinity.

### ***Updating in distance vector routing***



### Final Diagram



## When to Share

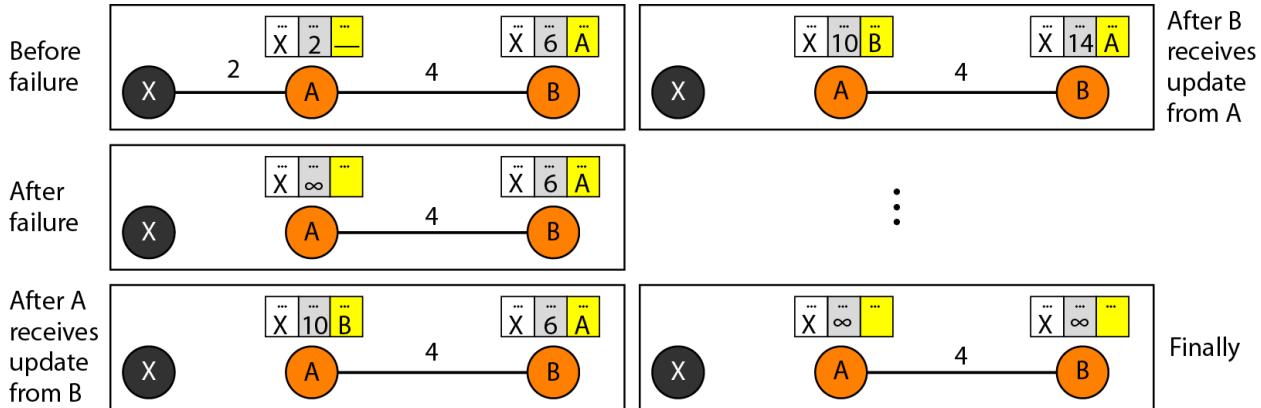
The question now is, When does a node send its partial routing table (only two columns) to all its immediate neighbors? The table is sent both periodically and when there is a change in the table.

Periodic Update A node sends its routing table, normally every 30 s, in a periodic update. The period depends on the protocol that is using distance vector routing.

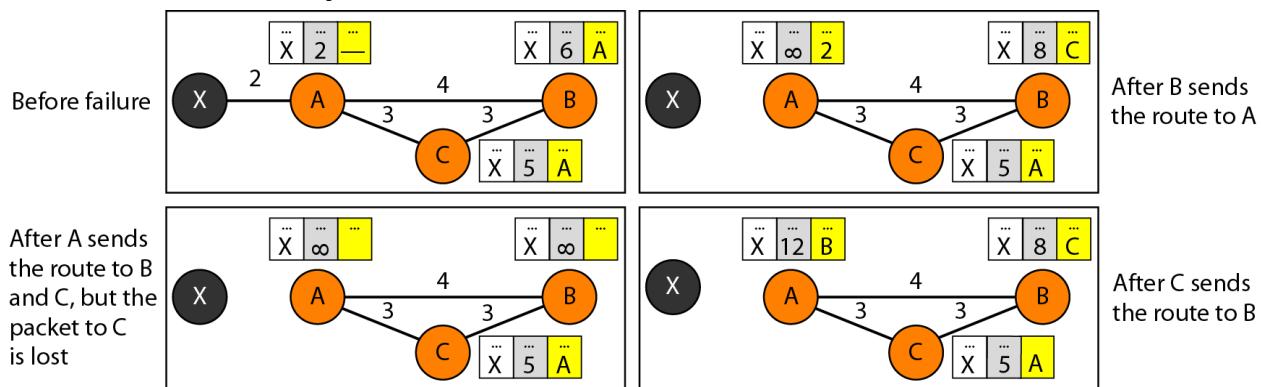
Triggered Update A node sends its two-column routing table to its neighbors anytime there is a change in its routing table. This is called a triggered update. The change can result from the following.

1. A node receives a table from a neighbor, resulting in changes in its own table after updating.
2. A node detects some failure in the neighboring links which results in a distance change to infinity.

## **Two-node instability**



## **Three-node instability**



## **SOLUTIONS FOR INSTABILITY**

1. **Defining Infinity:** redefine infinity to a smaller number, such as 100. For our previous scenario, the system will be stable in less than 20 updates. As a matter of fact, most implementations of the distance vector protocol define the distance between each node to

be 1 and define 16 as infinity. However, this means that the distance vector routing cannot be used in large systems. The size of the network, in each direction, cannot exceed 15 hops.

2. **Split Horizon:** In this strategy, instead of flooding the table through each interface, each node sends **only part of its table** through each interface. If, according to its table, node B thinks that the optimum route to reach X is via A, it does not need to advertise this piece of information to A; the information has come from A (A already knows). Taking information from node A, modifying it, and sending it back to node A creates the confusion. In our scenario, node B eliminates the last line of its routing table before it sends it to A. In this case, node A keeps the value of infinity as the distance to X. Later when node A sends its routing table to B, node B also corrects its routing table. The system becomes stable after the first update: both node A and B know that X is not reachable.
3. **Split Horizon and Poison Reverse** Using the split horizon strategy has one drawback. Normally, the distance vector protocol uses a timer, and if there is no news about a route, the node deletes the route from its table. When node B in the previous scenario eliminates the route to X from its advertisement to A, node A cannot guess that this is due to the split horizon strategy (the source of information was A) or because B has not received any news about X recently. The split horizon strategy can be combined with the poison reverse strategy. Node B can still advertise the value for X, but if the source of information is A, it can replace the distance with infinity as a warning: "Do not use this value; what I know about this route comes from you."

### The Count-to-Infinity Problem

A	B	C	D	E	
•	•	•	•	•	Initially
1	•	•	•	•	After 1 exchange
1	2	•	•	•	After 2 exchanges
1	2	3	•	•	After 3 exchanges
1	2	3	4	•	After 4 exchanges

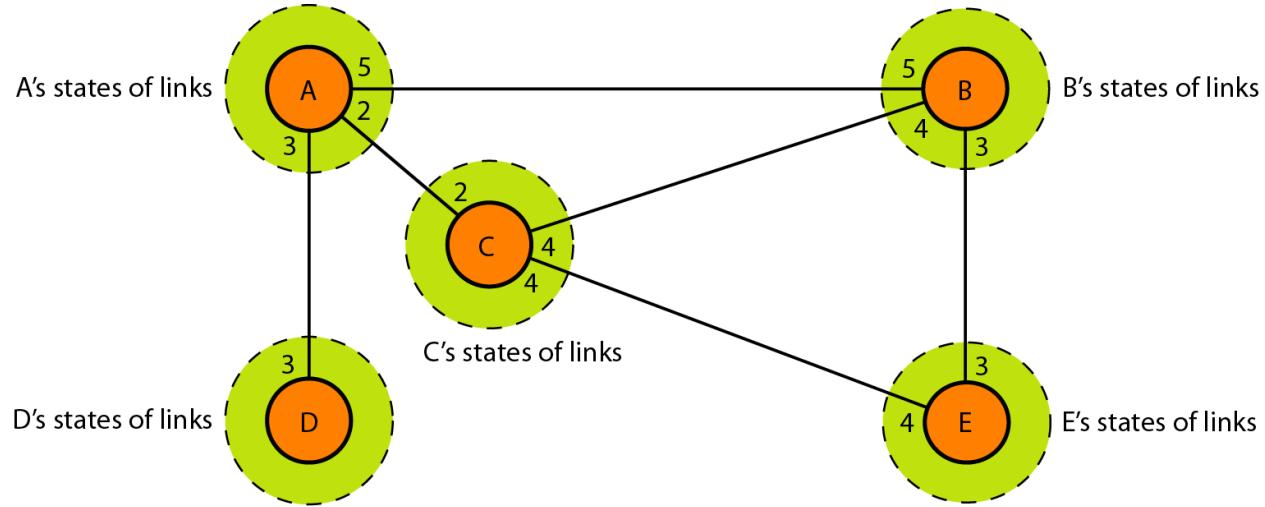
(a)

A	B	C	D	E	
•	1	2	3	4	Initially
3	2	3	4	•	After 1 exchange
3	4	3	4	•	After 2 exchanges
5	4	5	4	•	After 3 exchanges
5	6	5	6	•	After 4 exchanges
7	6	7	6	•	After 5 exchanges
7	8	7	8	•	After 6 exchanges
⋮					
•	•	•	•	•	

(b)

## Link State Routing

Link state routing is based on the assumption that, although the global knowledge about the topology is not clear, each node has partial knowledge: it knows the state (type, condition, and cost) of its links. **In other words, the whole topology can be compiled from the partial knowledge of each node**



## Building Routing Tables

1. Creation of the states of the links by each node, called the link state packet (LSP).
2. Dissemination of LSPs to every other router, called **flooding, in an efficient and reliable way**.
3. Formation of a shortest path tree for each node.
4. Calculation of a routing table based on the shortest path tree

- I. **Creation of Link State Packet (LSP)** A link state packet can carry a large amount of information. For the moment, we assume that it carries a minimum amount of data: the node identity, the list of links, a sequence number, and age. The first two, node identity and the list of links, are needed to make the topology. The third, sequence number, facilitates flooding and distinguishes new LSPs from old ones. The fourth, age, prevents old LSPs from remaining in the domain for a long time.

LSPs are generated on two occasions:

1. When there is a change in the topology of the domain
2. on a periodic basis: The period in this case is much longer compared to distance vector. The timer set for periodic dissemination is normally in the range of **60 min or 2 h** based on the implementation. A longer period ensures that flooding does not create too much traffic on the network.

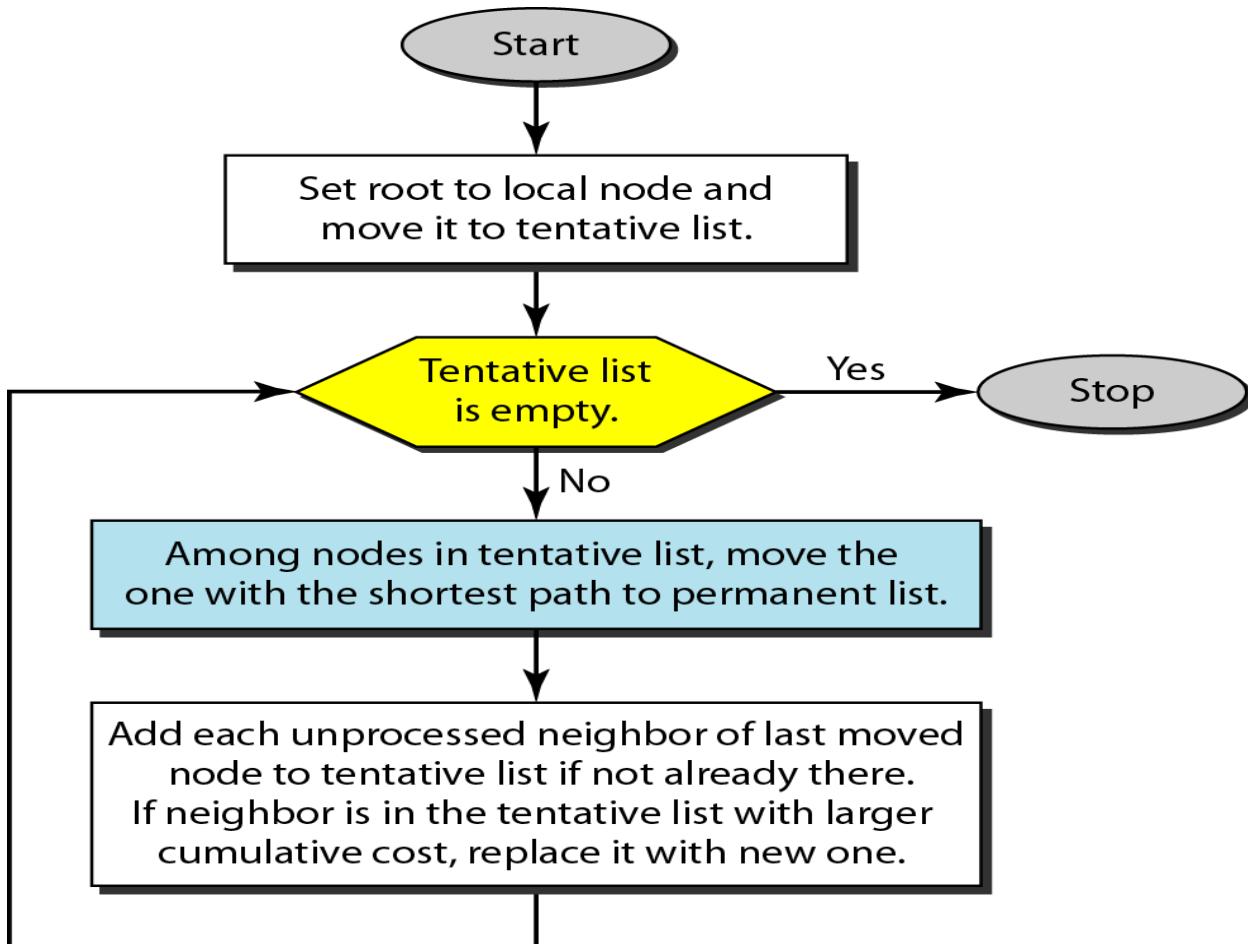
- II. **Flooding of LSPs:** After a node has prepared an LSP, it must be disseminated to all other nodes, not only to its neighbors. The process is called flooding and based on the following
  1. The creating node sends a copy of the LSP out of each interface

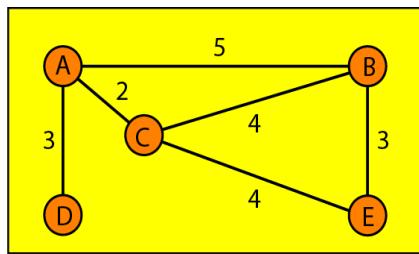
2. A node that receives an LSP compares it with the copy it may already have. If the newly arrived LSP is older than the one it has (found by checking the sequence number), it discards the LSP. If it is newer, the node does the following:
- It discards the old LSP and keeps the new one.
  - It sends a copy of it out of each interface except the one from which the packet arrived. This guarantees that flooding stops somewhere in the domain (where a node has only one interface).

### III. Formation of Shortest Path Tree: Dijkstra Algorithm

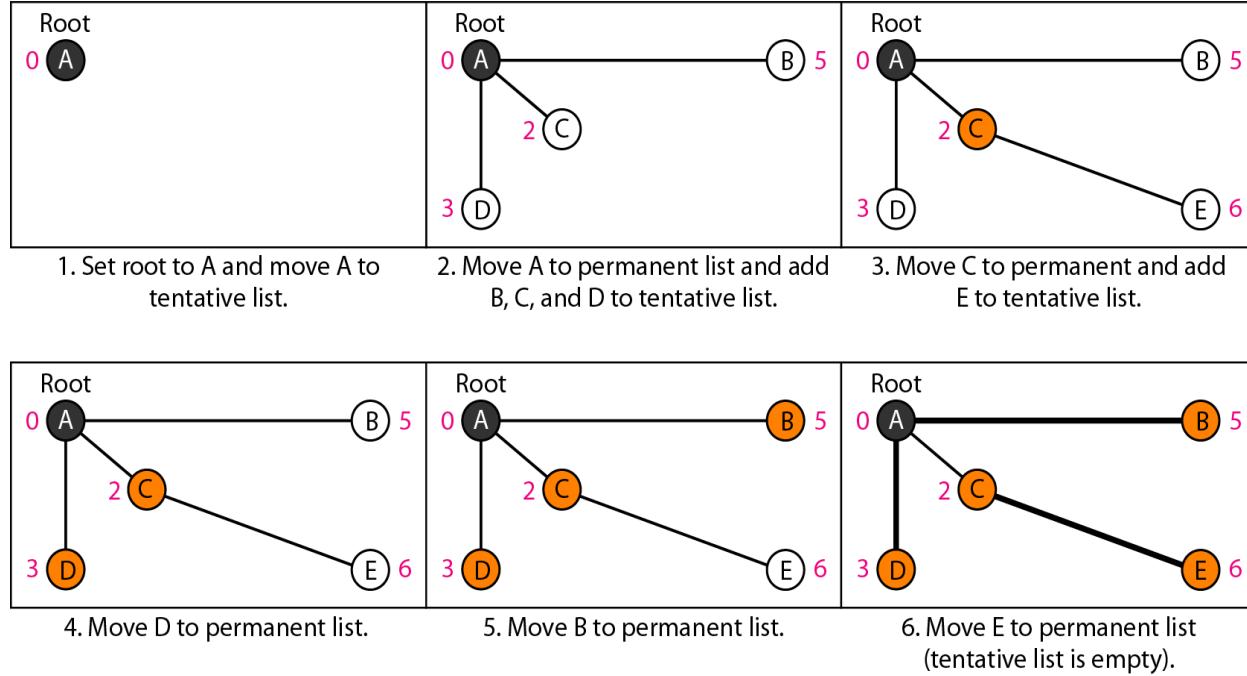
A shortest path tree is a tree in which the path between the root and every other node is the shortest.

The Dijkstra algorithm creates a shortest path tree from a graph. The algorithm divides the nodes into two sets: **tentative and permanent**. It finds the neighbors of a current node, makes them tentative, examines them, and if they pass the criteria, makes them permanent.





Topology



#### IV. Calculation of a routing table

routing table for node A

<i>Node</i>	<i>Cost</i>	<i>Next Router</i>
A	0	—
B	5	—
C	2	—
D	3	—
E	6	C

#### Path Vector Routing

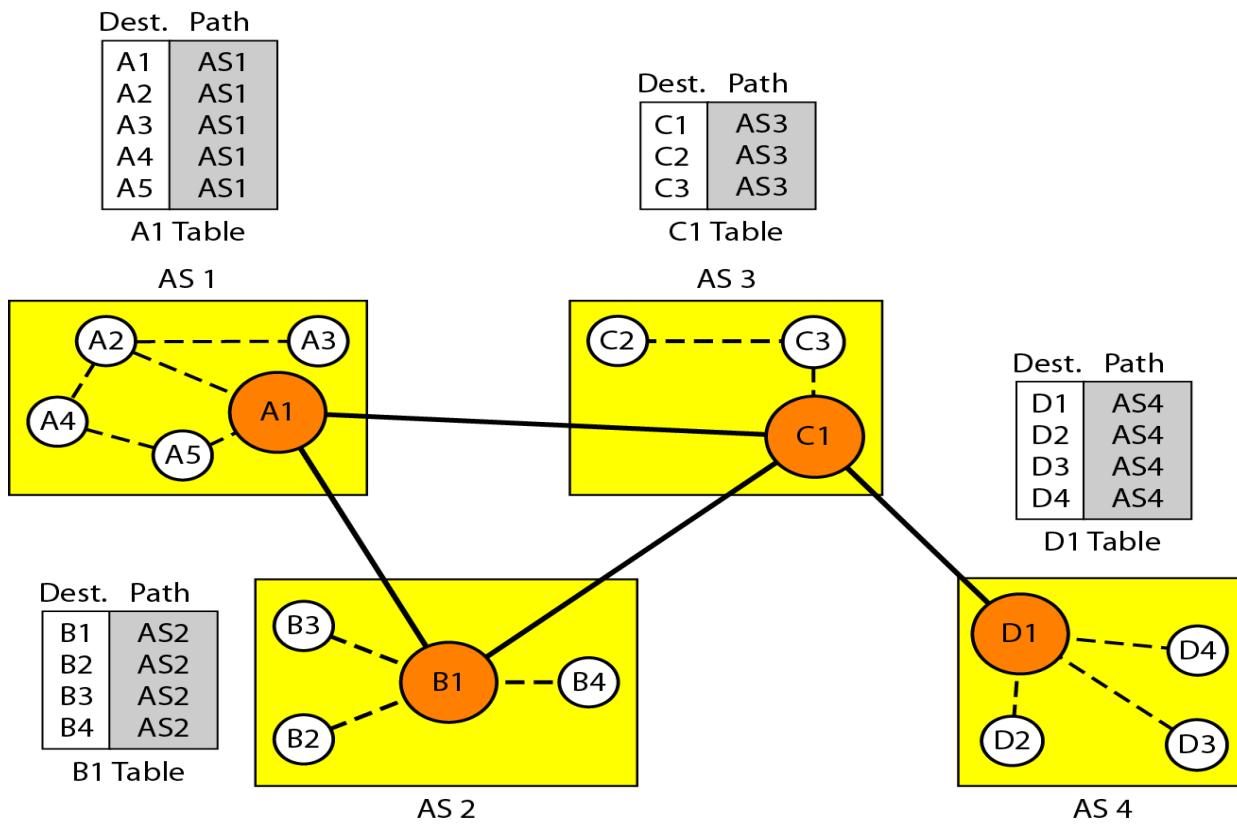
Distance vector and link state routing are both intra domain routing protocols. They can be used inside an autonomous system, but not between autonomous systems. These two protocols are not suitable for inter domain routing mostly because of scalability. Both of these routing protocols become intractable when the domain of operation becomes large. **Distance vector routing is subject to instability** in the domain of operation. **Link state routing needs a**

**huge amount of resources** to calculate routing tables. It also creates heavy traffic because of flooding. There is a need for a third routing protocol which we call path vector routing.

Path vector routing proved to be useful for inter domain routing. The principle of path vector routing is similar to that of distance vector routing. **In path vector routing, we assume that there is one node** (there can be more, but one is enough for our conceptual discussion) **in each AS** that acts on behalf of the entire AS. Let us call it the **speaker node**. The speaker node in an AS creates a routing table and advertises it to speaker nodes in the neighboring ASs. The idea is the same as for distance vector routing except that only speaker nodes in each AS can communicate with each other. However, what is advertised is different. A speaker node advertises the path, not the metric of the nodes, in its autonomous system or other autonomous systems

## Initialization

*Initial routing tables in path vector routing*



## Sharing

Just as in distance vector routing, in path vector routing, a speaker in an autonomous system shares its table with immediate neighbors. In Figure, node A1 shares its table with nodes B1

and C1. Node C1 shares its table with nodes D1, B1, and A1. Node B1 shares its table with C1 and A1. Node D1 shares its table with C1.

Dest.	Path	Dest.	Path	Dest.	Path	Dest.	Path
A1	AS1	A1	AS2-AS1	A1	AS3-AS1	A1	AS4-AS3-AS1
...		...		...		...	
A5	AS1	A5	AS2-AS1	A5	AS3-AS1	A5	AS4-AS3-AS1
B1	AS1-AS2	B1	AS2	B1	AS3-AS2	B1	AS4-AS3-AS2
...	...	...		...		...	
B4	AS1-AS2	B4	AS2	B4	AS3-AS2	B4	AS4-AS3-AS2
C1	AS1-AS3	C1	AS2-AS3	C1	AS3	C1	AS4-AS3
...	...	...		...		...	
C3	AS1-AS3	C3	AS2-AS3	C3	AS3	C3	AS4-AS3
D1	AS1-AS2-AS4	D1	AS2-AS3-AS4	D1	AS3-AS4	D1	AS4
...	...	...		...		...	
D4	AS1-AS2-AS4	D4	AS2-AS3-AS4	D4	AS3-AS4	D4	AS4

A1 Table

B1 Table

C1 Table

D1 Table

**Updating** When a speaker node receives a two-column table from a neighbor, it updates its own table by adding the nodes that are not in its routing table and adding its own autonomous system and the autonomous system that sent the table. After a while each speaker has a table and knows how to reach each node in other Ass

- Loop prevention.** The instability of distance vector routing and the creation of loops can be avoided in path vector routing. When a router receives a message, it checks to see if its AS is in the path list to the destination. If it is, looping is involved and the message is ignored.
- Policy routing.** Policy routing can be easily implemented through path vector routing. When a router receives a message, it can check the path. If one of the AS listed in the path is against its policy, it can ignore that path and that destination. It does not update its routing table with this path, and it does not send this message to its neighbors.
- Optimum path.** What is the optimum path in path vector routing? We are looking for a path to a destination that is the best for the organization that runs the AS. One system may use RIP, which defines hop count as the metric; another may use OSPF with minimum delay defined as the metric. In our previous figure, each AS may have more than one path to a destination. For example, a path from AS4 to AS1 can be AS4-AS3-AS2-AS1, or it can be AS4-AS3-AS1. For the tables, **we chose the one that had the smaller number of ASs**, but this is not always the case. Other criteria, such as security, safety, and reliability, can also be applied

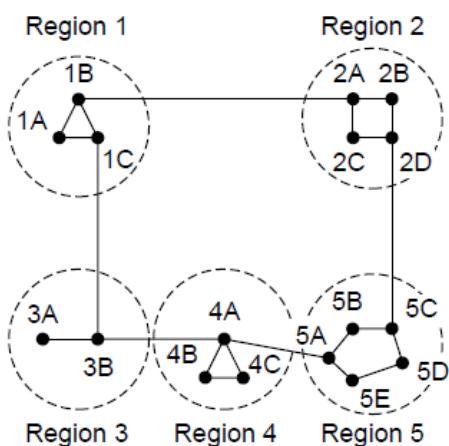
## Hierarchical Routing

As networks grow in size, the router routing tables grow proportionally. Not only is router memory consumed by ever-increasing tables, but more CPU time is needed to scan them and more bandwidth is needed to send status reports about them.

At a certain point, the network may grow to the point where it is no longer feasible for every router to have an entry for every other router, so the routing will have to be done hierarchically, as it is in the telephone network.

When hierarchical routing is used, the routers are divided into what we will call regions. Each router knows all the details about how to route packets to destinations within its own region but knows nothing about the internal structure of other regions.

For huge networks, a two-level hierarchy may be insufficient; it may be necessary to group the regions into clusters, the clusters into zones, the zones into groups, and so on, until we run out of names for aggregations



(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)

When a single network becomes very large, an interesting question is “how many levels should the hierarchy have?”

For example, consider a network with 720 routers. If there is no hierarchy, each router needs 720 routing table entries.

If the network is partitioned into 24 regions of 30 routers each, each router needs 30 local entries plus 23 remote entries for a total of 53 entries.

If a three-level hierarchy is chosen, with 8 clusters each containing 9 regions of 10 routers, each router needs 10 entries for local routers, 8 entries for routing to other regions within its own cluster, and 7 entries for distant clusters, for a total of 25 entries

Kamoun and Kleinrock (1979) discovered that the optimal number of levels for an  $N$  router network is  $\ln N$ , requiring a total of  $e \ln N$  entries per router

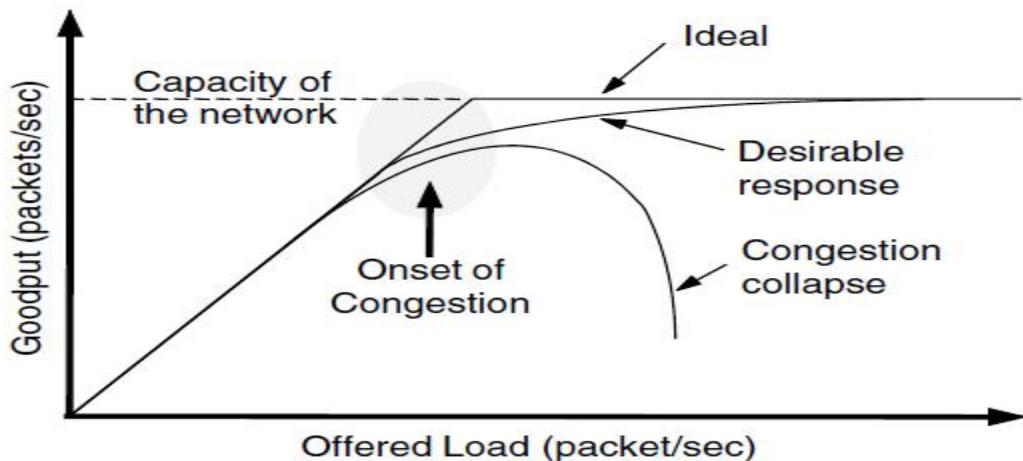
---

## CONGESTION CONTROL ALGORITHMS

Too many packets present in (a part of) the network causes packet delay and loss that degrades performance. This situation is called **congestion**.

The network and transport layers share the responsibility for handling congestion. Since congestion occurs within the network, it is the network layer that directly experiences it and must ultimately determine what to do with the excess packets.

However, the most effective way to control congestion is to reduce the load that the transport layer is placing on the network. This requires the network and transport layers to work together. In this chapter we will look at the network aspects of congestion.



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

Above Figure depicts the onset of congestion. When the number of packets hosts send into the network is well within its carrying capacity, the number delivered is proportional to the number sent. If twice as many are sent, twice as many are delivered. However, as the offered load approaches the carrying capacity, bursts of traffic occasionally fill up the buffers inside routers and some packets are lost. These lost packets consume some of the capacity, so the number of delivered packets falls below the ideal curve. The network is now congested. Unless the network is well designed, it may experience a **congestion collapse**

## **difference between congestion control and flow control.**

Congestion control has to do with making sure the network is able to carry the offered traffic. It is a global issue, involving the behavior of all the hosts and routers.

Flow control, in contrast, relates to the traffic between a particular sender and a particular receiver. Its job is to make sure that a fast sender cannot continually transmit data faster than the receiver is able to absorb it.

To see the difference between these two concepts, consider a network made up of 100-Gbps fiber optic links on which a supercomputer is trying to force feed a large file to a personal computer that is capable of handling only 1 Gbps. Although there is no congestion (the network itself is not in trouble), flow control is needed to force the supercomputer to stop frequently to give the personal computer a chance to breathe.

At the other extreme, consider a network with 1-Mbps lines and 1000 large computers, half of which are trying to transfer files at 100 kbps to the other half. Here, the problem is not that of fast senders overpowering slow receivers, but that the total offered traffic exceeds what the network can handle.

The reason congestion control and flow control are often confused is that the best way to handle both problems is to get the host to slow down. Thus, a host can get a “slow down” message either because the receiver cannot handle the load or because the network cannot handle it.

Several techniques can be employed. These include:

1. Warning bit
2. Choke packets
3. Load shedding
4. Random early discard
5. Traffic shaping

The first 3 deal with congestion detection and recovery. The last 2 deal with congestion avoidance

### **Warning Bit**

1. A special bit in the packet header is set by the router to warn the source when congestion is detected.
2. The bit is copied and piggy-backed on the ACK and sent to the sender.
3. The sender monitors the number of ACK packets it receives with the warning bit set and adjusts its transmission rate accordingly.

## **Choke Packets**

1. A more direct way of telling the source to slow down.
2. A choke packet is a control packet generated at a congested node and transmitted to restrict traffic flow.
3. The source, on receiving the choke packet must reduce its transmission rate by a certain percentage.
4. An example of a choke packet is the ICMP Source Quench Packet.

### Hop-by-Hop Choke Packets

1. Over long distances or at high speeds choke packets are not very effective.
2. A more efficient method is to send choke packets hop-by-hop.
3. This requires each hop to reduce its transmission even before the choke packet arrive at the source

## **Load Shedding**

1. When buffers become full, routers simply discard packets.
2. Which packet is chosen to be the victim depends on the application and on the error strategy used in the data link layer.
3. For a file transfer, for, e.g. cannot discard older packets since this will cause a gap in the received data.
4. For real-time voice or video it is probably better to throw away old data and keep new packets.
5. Get the application to mark packets with discard priority.

## **Random Early Discard (RED)**

1. This is a proactive approach in which the router discards one or more packets *before* the buffer becomes completely full.
2. Each time a packet arrives, the RED algorithm computes the average queue length, *avg*.
3. If *avg* is lower than some lower threshold, congestion is assumed to be minimal or non-existent and the packet is queued.
4. If *avg* is greater than some upper threshold, congestion is assumed to be serious and the packet is discarded.
5. If *avg* is between the two thresholds, this might indicate the onset of congestion. The probability of congestion is then calculated.

## Traffic Shaping

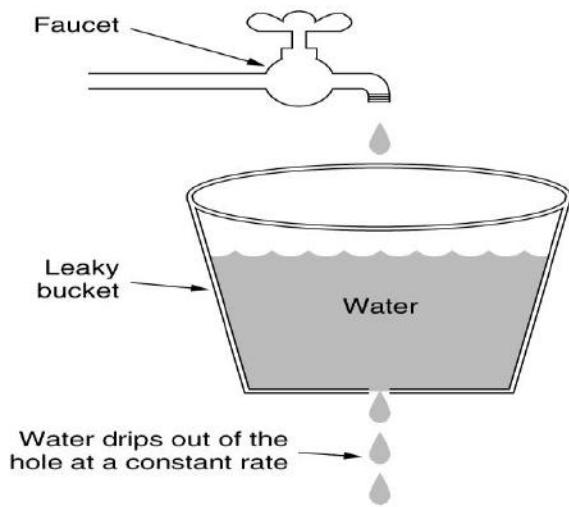
1. Another method of congestion control is to “shape” the traffic before it enters the network.
2. Traffic shaping controls the *rate* at which packets are sent (not just how many). Used in ATM and Integrated Services networks.
3. At connection set-up time, the sender and carrier negotiate a traffic pattern (shape).

Two traffic shaping algorithms are:

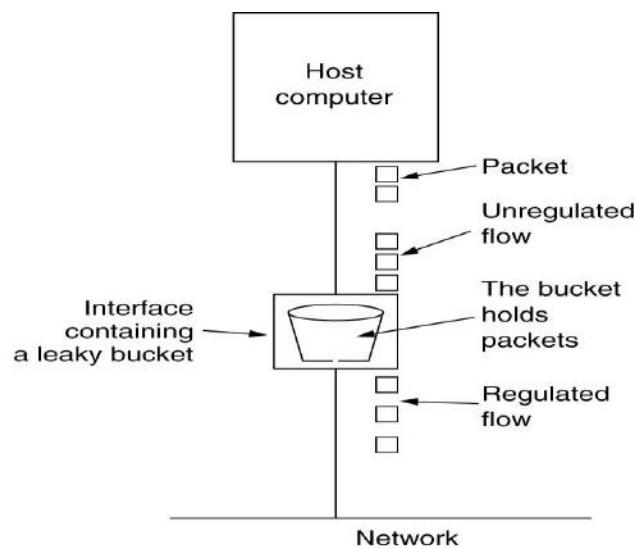
Leaky Bucket

Token Bucket

The **Leaky Bucket Algorithm** used to control rate in a network. It is implemented as a single-server queue with constant service time. If the bucket (buffer) overflows then packets are discarded.



(a)



(b)

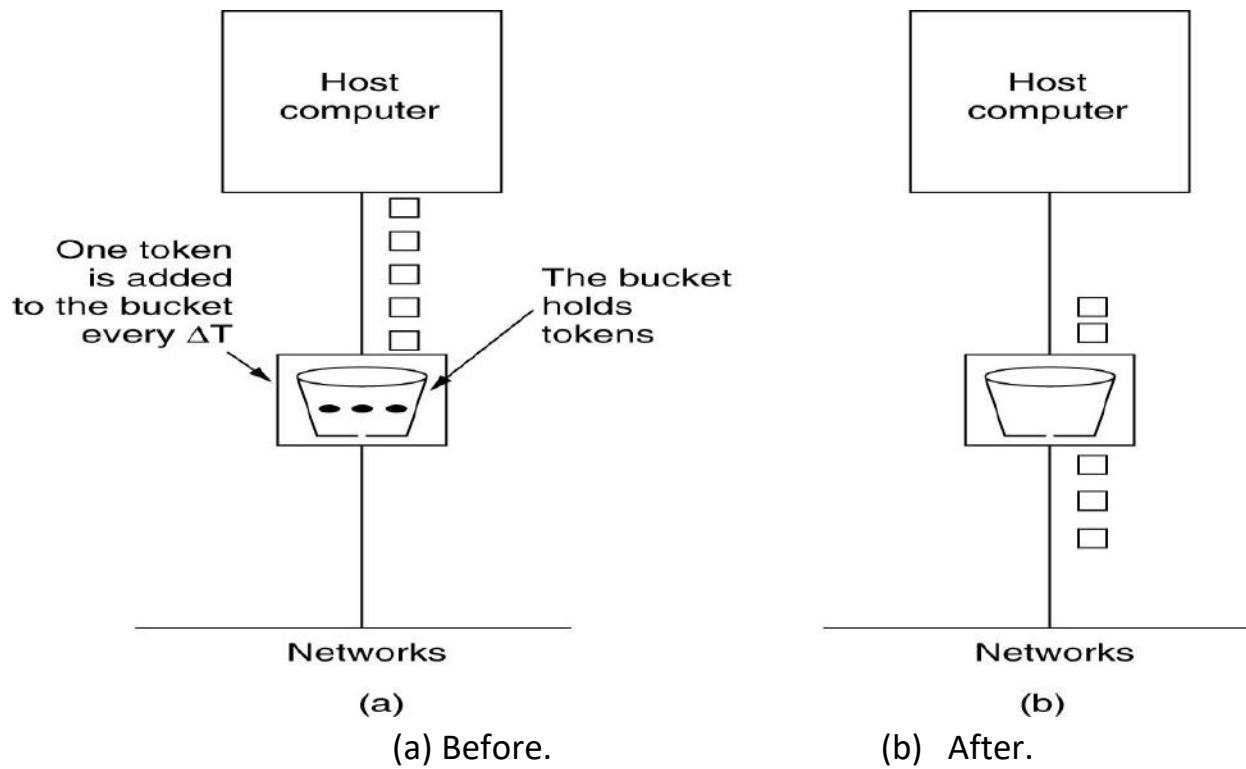
(a) A leaky bucket with water.

(b) a leaky bucket with packets.

1. The leaky bucket enforces a constant output rate (average rate) regardless of the burstiness of the input. Does nothing when input is idle.
2. The host injects one packet per clock tick onto the network. This results in a uniform flow of packets, smoothing out bursts and reducing congestion.
3. When packets are the same size (as in ATM cells), the one packet per tick is okay. For variable length packets though, it is better to allow a 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

## Token Bucket Algorithm

1. In contrast to the LB, the Token Bucket Algorithm, allows the output rate to vary, depending on the size of the burst.
2. In the TB algorithm, the bucket holds tokens. To transmit a packet, the host must capture and destroy one token.
3. Tokens are generated by a clock at the rate of one token every  $\Delta t$  sec.
4. Idle hosts can capture and save up tokens (up to the max. size of the bucket) in order to send larger bursts later.



## Leaky Bucket vs. Token Bucket

1. LB discards packets; TB does not. TB discards tokens.
2. With TB, a packet can only be transmitted if there are enough tokens to cover its length in bytes.
3. LB sends packets at an average rate. TB allows for large bursts to be sent faster by speeding up the output.
4. TB allows saving up tokens (permissions) to send large bursts. LB does not allow saving.

## TRANSPORT LAYER

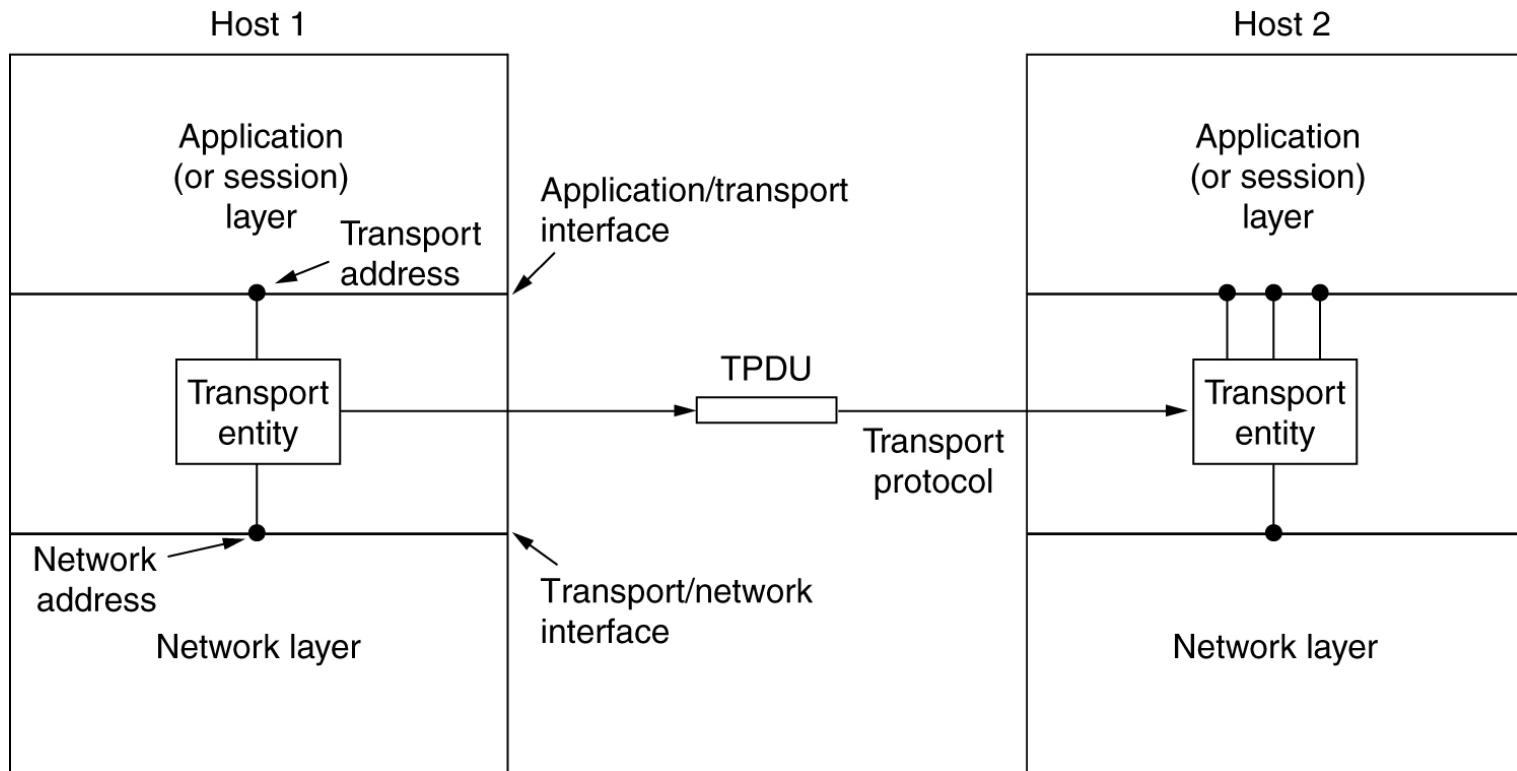
The network layer provides end-to-end packet delivery using datagrams or virtual circuits.

The transport layer builds on the network layer to provide data transport from a process on a source machine to a process on a destination machine with a desired level of reliability that is independent of the physical networks currently in use.

## Services Provided to the Upper Layers

The ultimate goal of the transport layer is to provide efficient, reliable, and cost-effective data transmission service to its users, normally processes in the application layer.

To achieve this, the transport layer makes use of the services provided by the network layer. The software and/or hardware within the transport layer that does the work is called the **transport entity**.

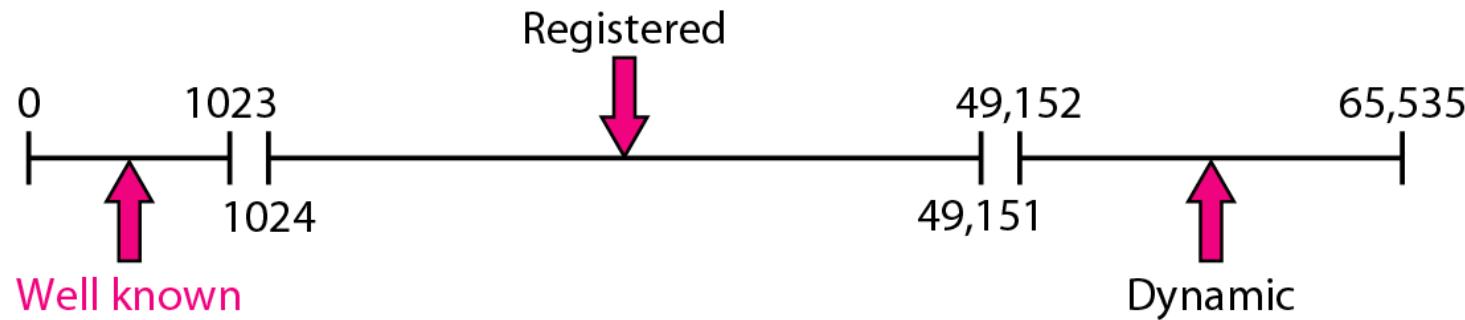


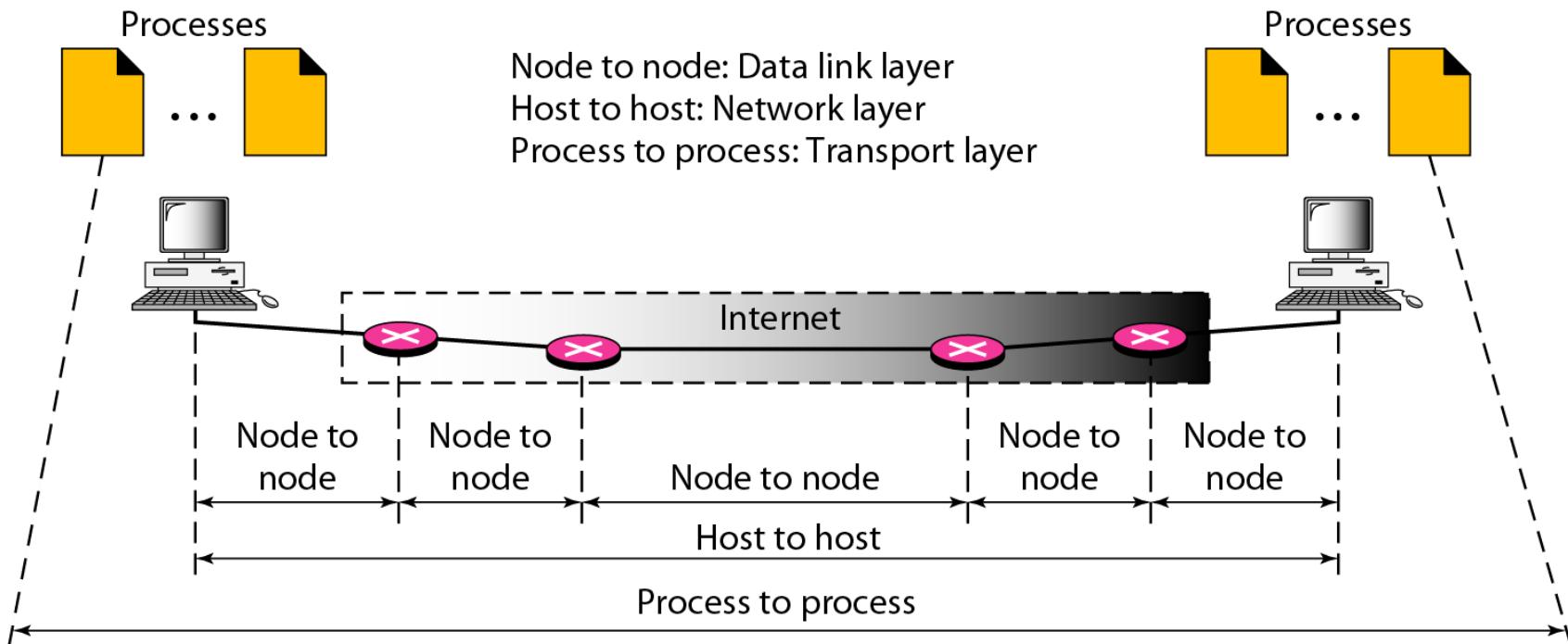
The network, transport, and application layers.

The connection-oriented transport service. connections have three phases: establishment, data transfer, and release.

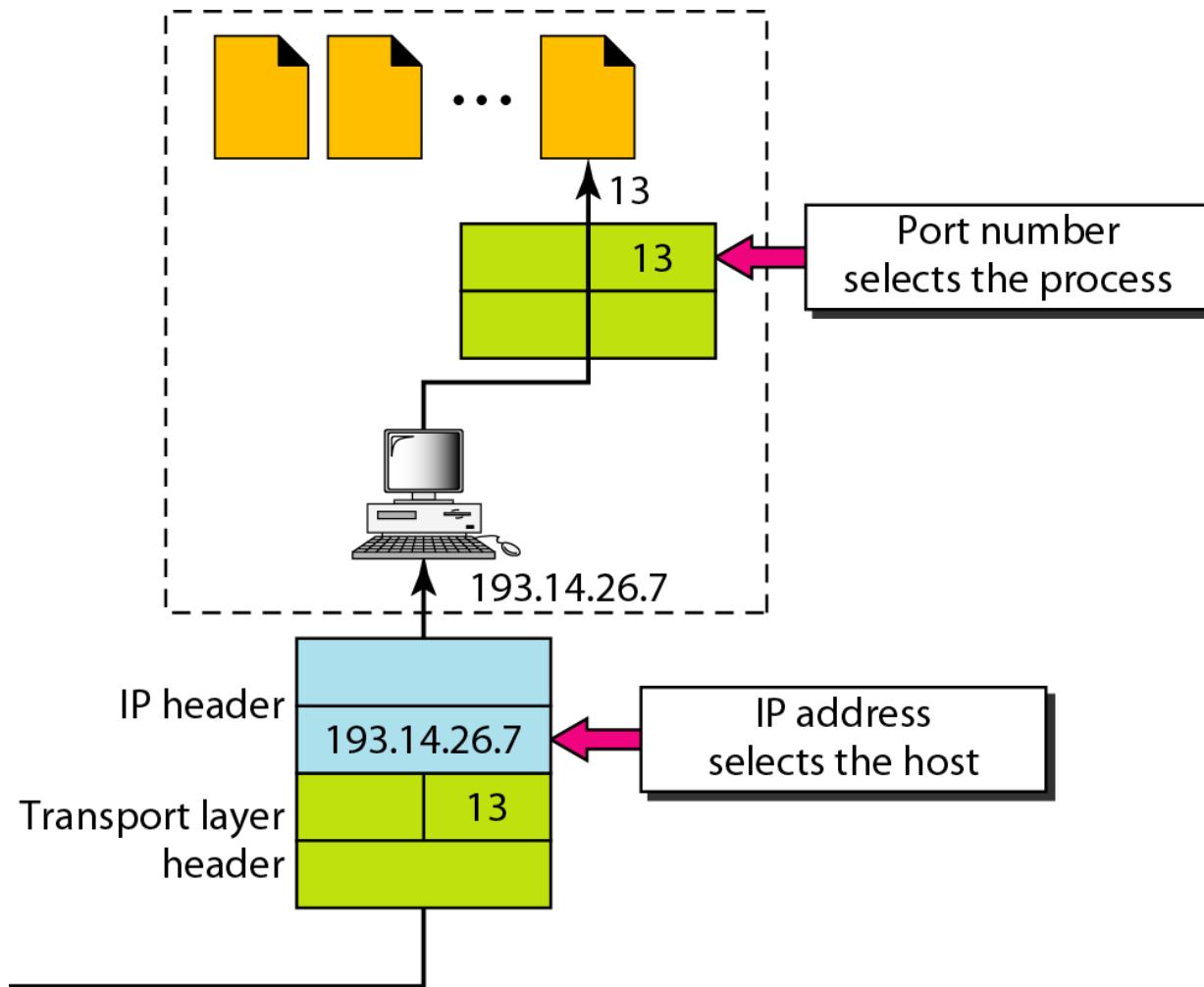
Addressing and flow control

The connectionless transport service .





## *IP addresses versus port numbers*



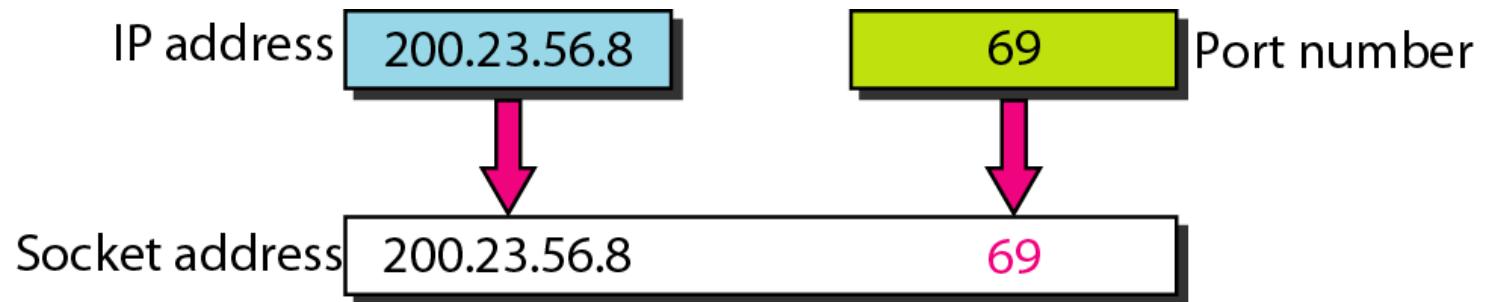
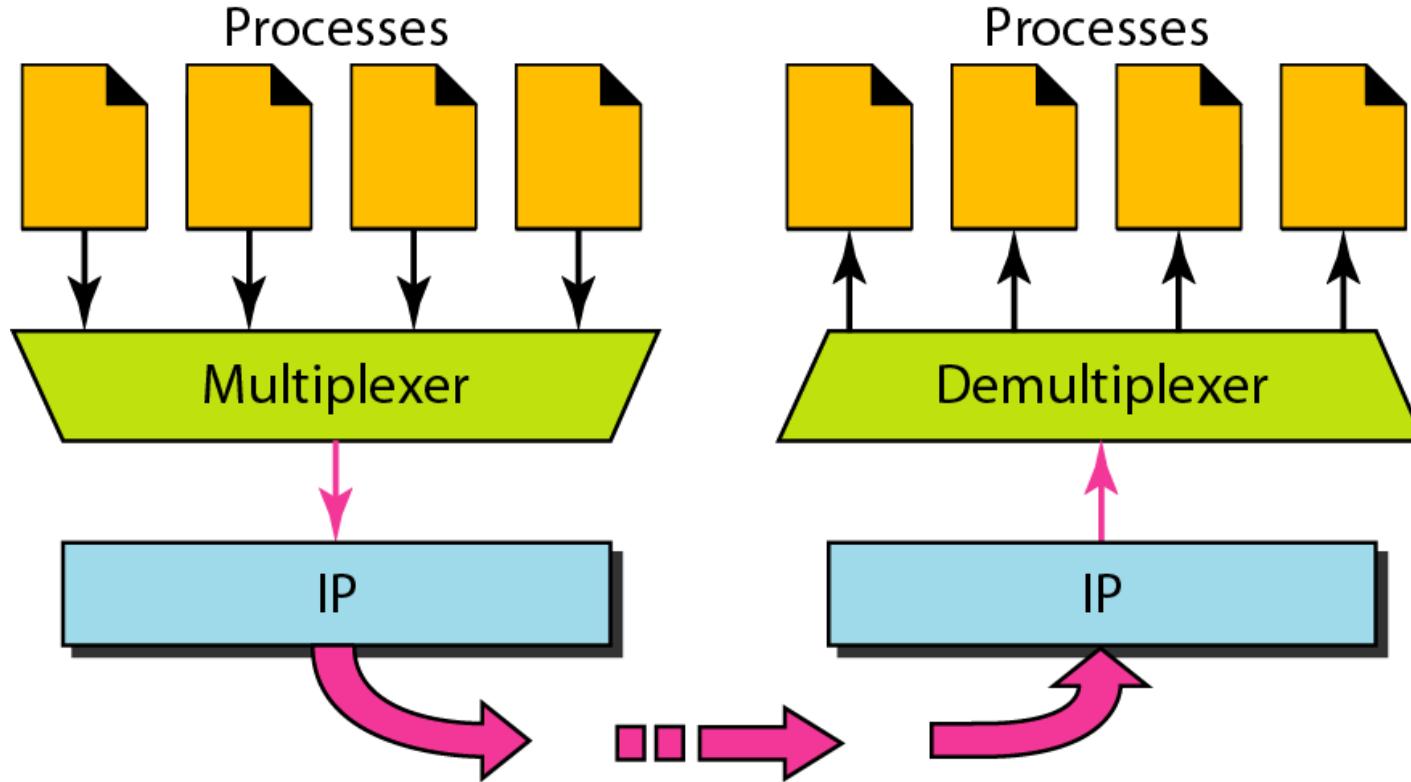
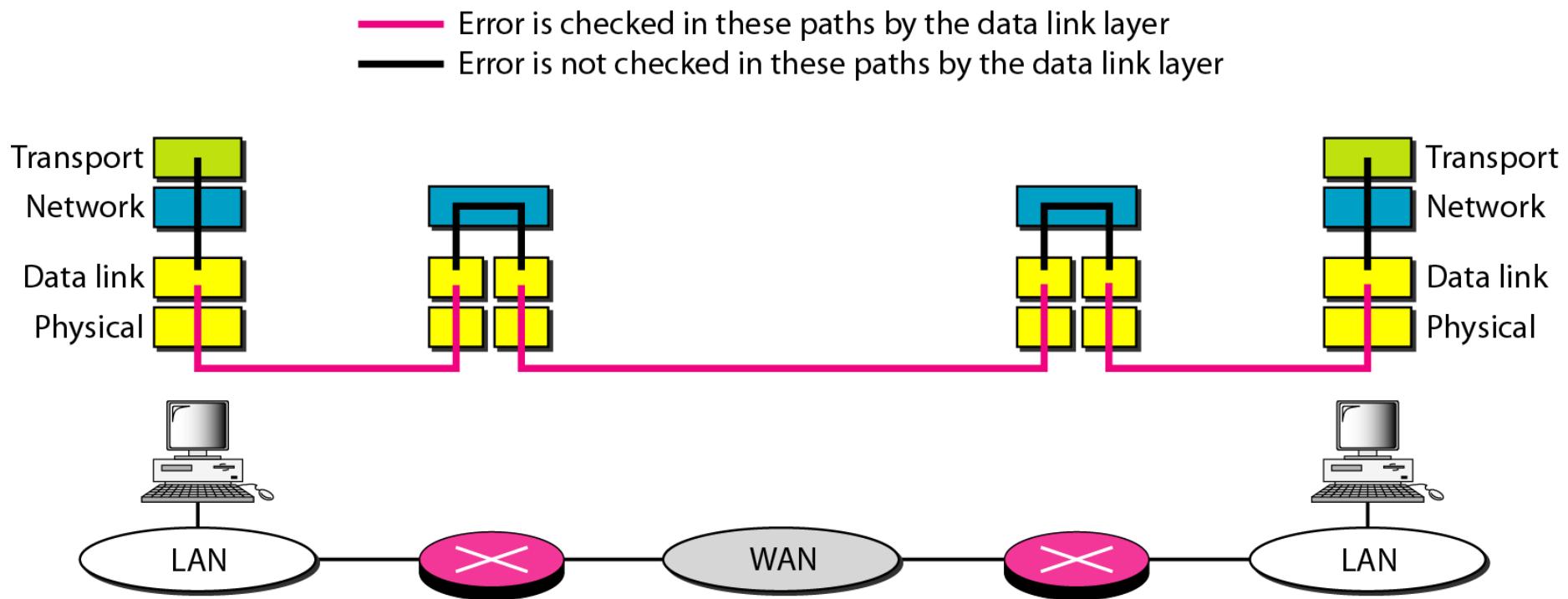


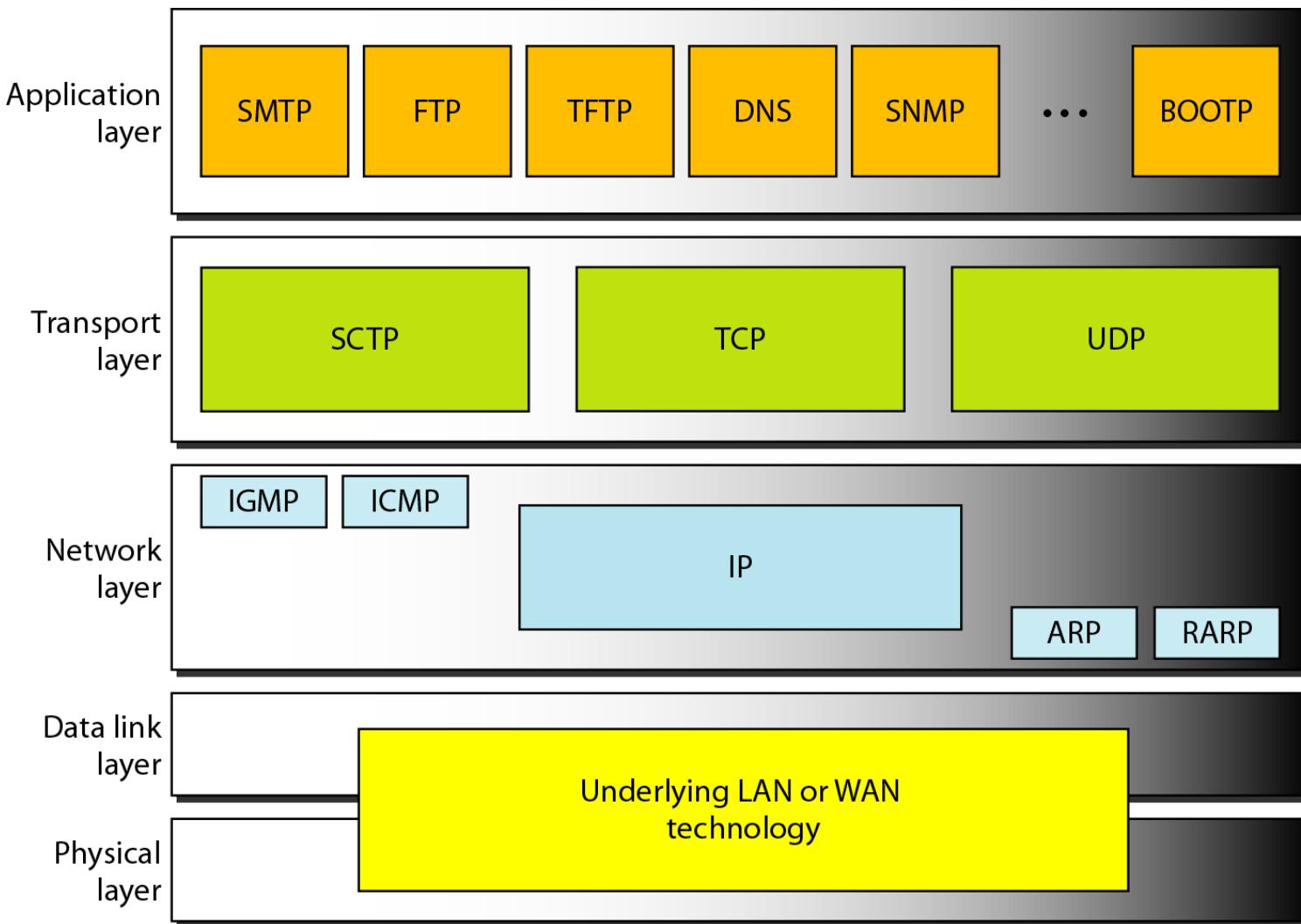
Figure 23.6 Multiplexing and demultiplexing



**Figure 23.7** Error control



**Figure 23.8 Position of UDP, TCP, and SCTP in TCP/IP suite**



# Transport Service Primitives

Primitive	Packet sent	Meaning
LISTEN	(none)	Block until some process tries to connect
CONNECT	CONNECTION REQ.	Actively attempt to establish a connection
SEND	DATA	Send information
RECEIVE	(none)	Block until a DATA packet arrives
DISCONNECT	DISCONNECTION REQ.	This side wants to release the connection

The primitives for a simple transport service.

To start with, the server executes a LISTEN primitive, typically by calling a library procedure that makes a system call that blocks the server until a client turns up.

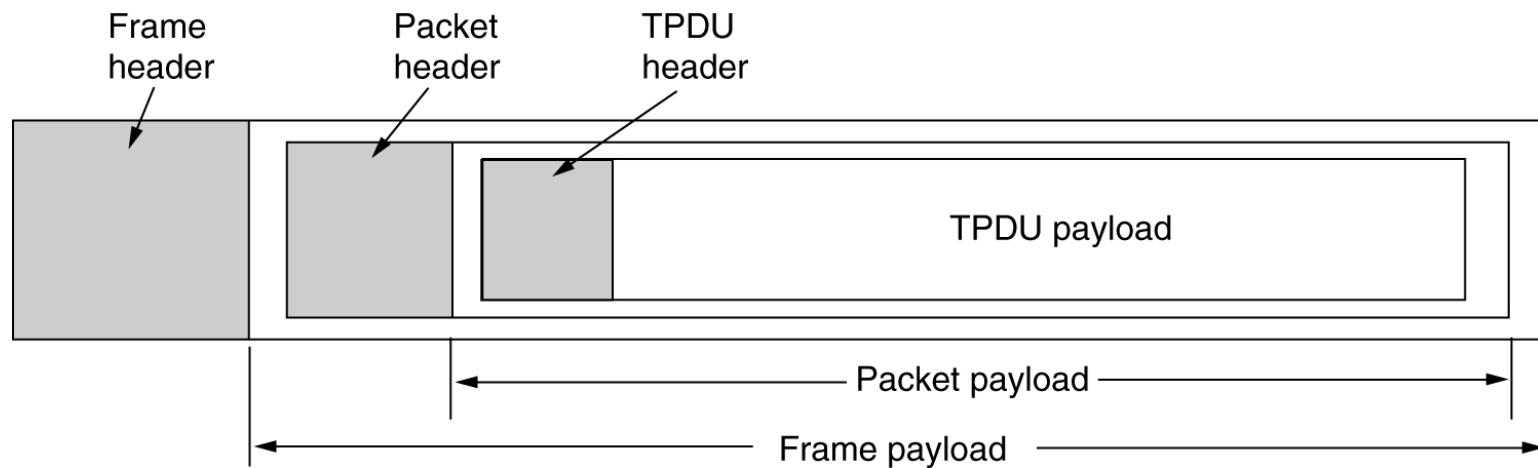
When a client wants to talk to the server, it executes a CONNECT primitive. The transport entity carries out this primitive by blocking the caller and sending a packet to the server. The client's CONNECT call causes a CONNECTION REQUEST segment to be sent to the server. When it arrives, the transport entity checks to see that the server is blocked on a LISTEN (i.e., is interested in handling requests). If so, it then unblocks the server and sends a CONNECTION ACCEPTED segment back to the client. When this segment arrives, the client is unblocked and the connection is established.

Data can now be exchanged using the SEND and RECEIVE primitives. In the simplest form, either party can do a (blocking) RECEIVE to wait for the other party to do a SEND. When the segment arrives, the receiver is unblocked. It can then process the segment and send a reply. As long as both sides can keep track of whose turn it is to send, this scheme works fine.

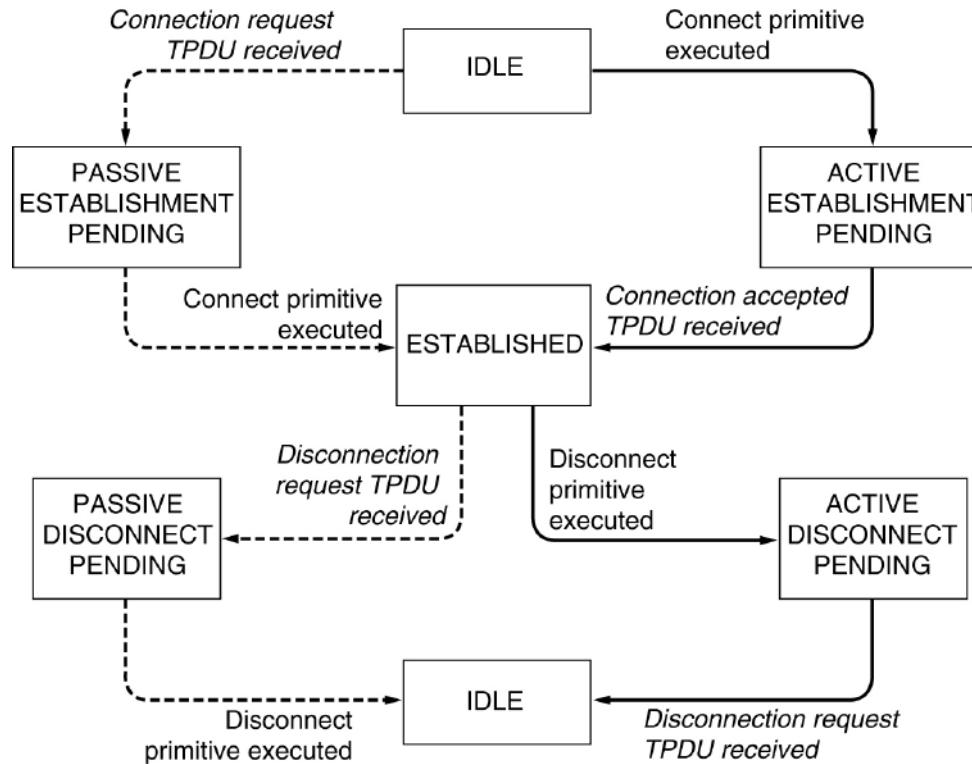
When a connection is no longer needed, it must be released to free up table space within the two transport entities. Disconnection has two variants: asymmetric and symmetric.

In the asymmetric variant, either transport user can issue a DISCONNECT primitive, which results in a DISCONNECT segment being sent to the remote transport entity. Upon its arrival, the connection is released.

In the symmetric variant, each direction is closed separately, independently of the other one. When one side does a DISCONNECT, that means it has no more data to send but it is still willing to accept data from its partner. In this model, a connection is released when both sides have done a DISCONNECT



# Transport Service Primitives

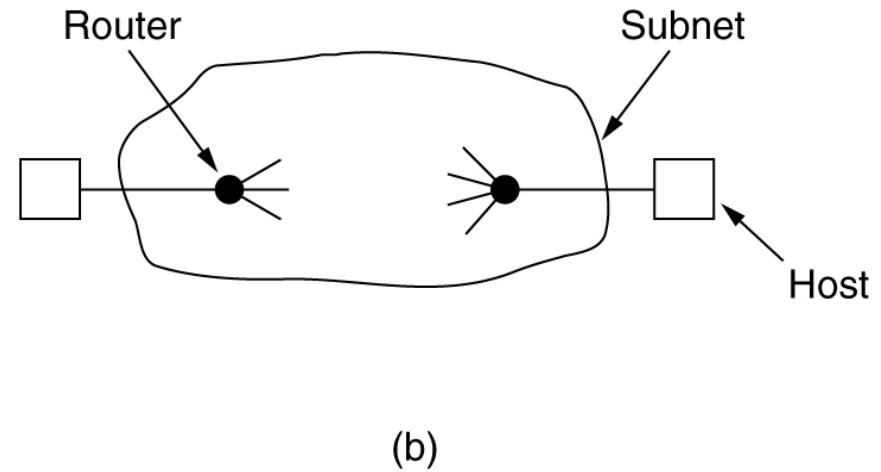
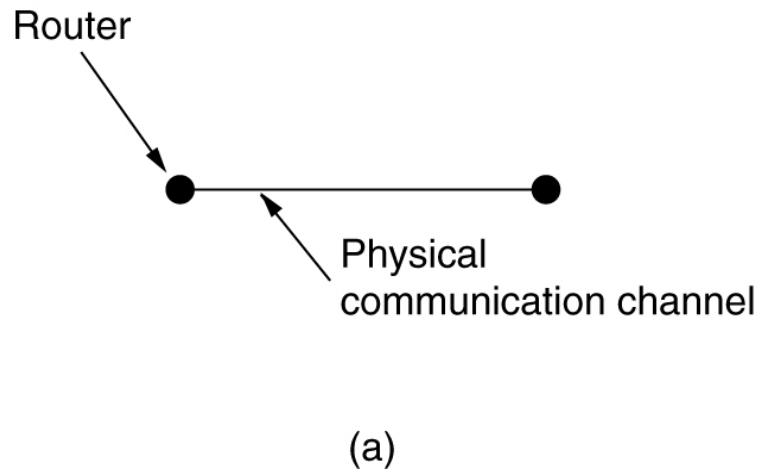


A state diagram for a simple connection management scheme. Transitions labeled in italics are caused by packet arrivals. The solid lines show the client's state sequence. The dashed lines show the server's state sequence.

# Elements of Transport Protocols

- Addressing
- Connection Establishment
- Connection Release
- Flow Control and Buffering
- Multiplexing
- Crash Recovery

# Transport Protocol



- (a) Environment of the data link layer.  
(b) Environment of the transport layer.

1 Over point-to-point links such as wires or optical fiber, it is usually not necessary for a router to specify which router it wants to talk to—each outgoing line leads directly to a particular router. In the transport layer, explicit addressing of destinations is required.

2 The process of establishing a connection over the wire of Fig(a) is simple: the other end is always there (unless it has crashed, in which case it is not there). Either way, there is not much to do. Even on wireless links the process is not much different. Just sending a message is sufficient to have it reach all other destinations. If the message is not acknowledged due to an error, it can be resent. In the transport layer, initial connection establishment is complicated, as we will see.

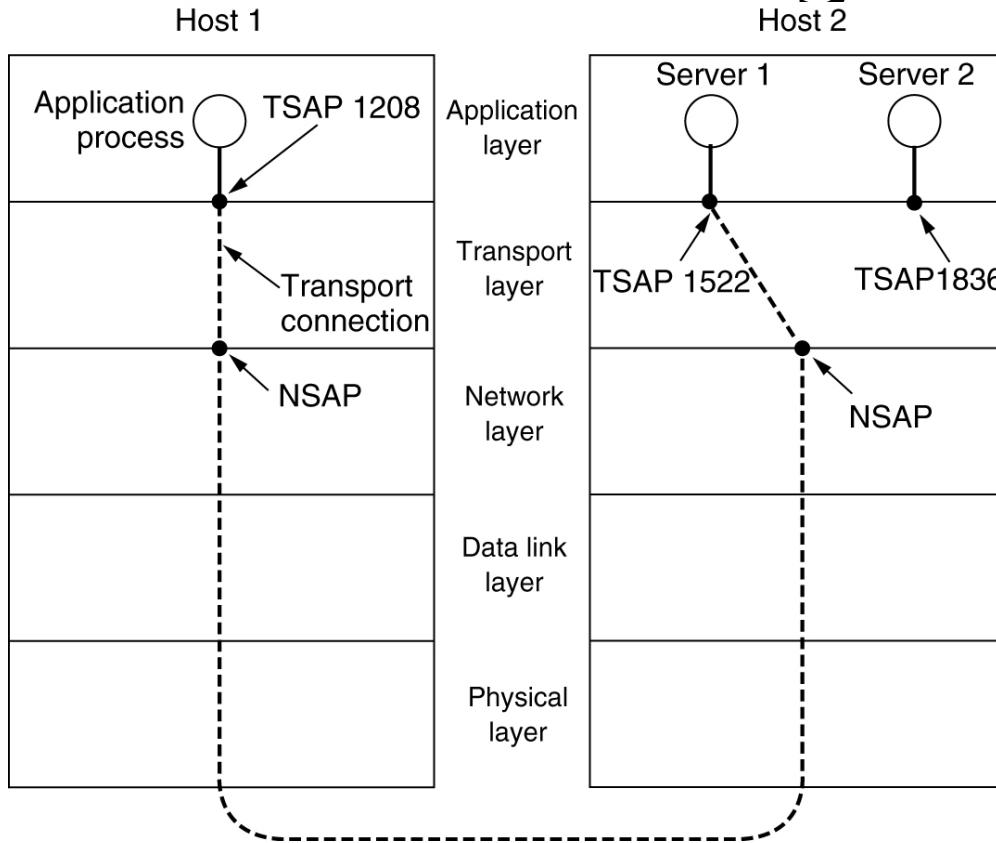
3 Another (exceedingly annoying) difference between the data link layer and the transport layer is the potential existence of storage capacity in the network. The consequences of the network's ability to delay and duplicate packets can sometimes be disastrous and can require the use of special protocols to correctly transport information.

4. Buffering and flow control are needed in both layers, but the presence in the transport layer of a large and varying number of connections with bandwidth that fluctuates as the connections compete with each other may require a different approach than we used in the data link layer.

# Addressing

When an application (e.g., a user) process wishes to set up a connection to a remote application process, it must specify which one to connect to. (Connectionless transport has the same problem: to whom should each message be sent?) The method normally used is to define transport addresses to which processes can listen for connection requests. In the Internet, these endpoints are called **ports**. We will use the generic term **TSAP (Transport Service Access Point)** to mean a specific endpoint in the transport layer. The analogous endpoints in the network layer (i.e., network layer addresses) are not-surprisingly called **NSAPs (Network Service Access Points)**. IP addresses are examples of NSAPs.

# Addressing



TSAPs, NSAPs and transport  
connections.

A possible scenario for a transport connection is as follows:

1. A mail server process attaches itself to TSAP 1522 on host 2 to wait for an incoming call. A call such as our LISTEN might be used, for example.
2. An application process on host 1 wants to send an email message, so it attaches itself to TSAP 1208 and issues a CONNECT request. The request specifies TSAP 1208 on host 1 as the source and TSAP 1522 on host 2 as the destination. This action ultimately results in a transport connection being established between the application process and the server.
3. The application process sends over the mail message.
4. The mail server responds to say that it will deliver the message.
5. The transport connection is released.

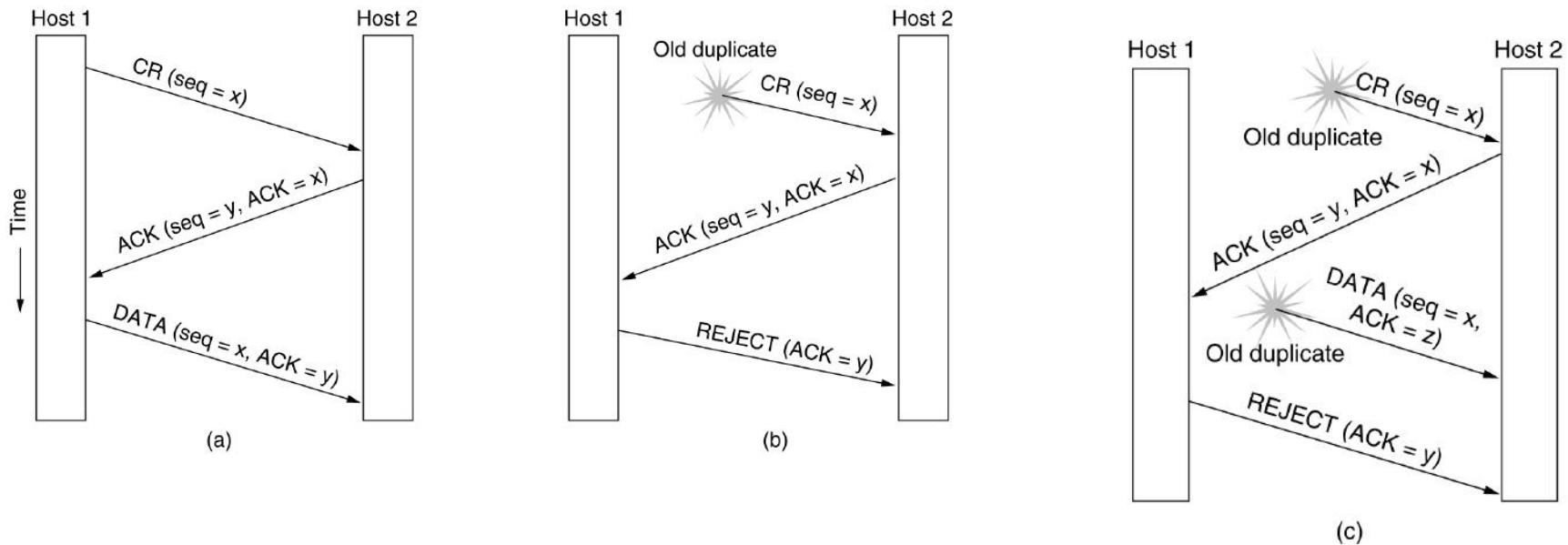
special process called a **portmapper**

## CONNECTION ESTABLISHMENT

Establishing a connection sounds easy, but it is actually surprisingly tricky. At first glance, it would seem sufficient for one transport entity to just send a CONNECTION REQUEST segment to the destination and wait for a CONNECTION ACCEPTED reply. The problem occurs when the network can lose, delay, corrupt, and duplicate packets. This behavior causes serious complications

To solve this specific problem,(DELAYED DUPLICATES) Tomlinson (1975) introduced the **three-way handshake**. This establishment protocol involves one peer checking with the other that the connection request is indeed current. The normal setup procedure when host 1 initiates is shown in Fig. (a). Host 1 chooses a sequence number,  $x$ , and sends a CONNECTION REQUEST segment containing it to host 2. Host 2 replies with an ACK segment acknowledging  $x$  and announcing its own initial sequence number,  $y$ . Finally, host 1 acknowledges host 2's choice of an initial sequence number in the first data segment that it sends.

# Connection Establishment



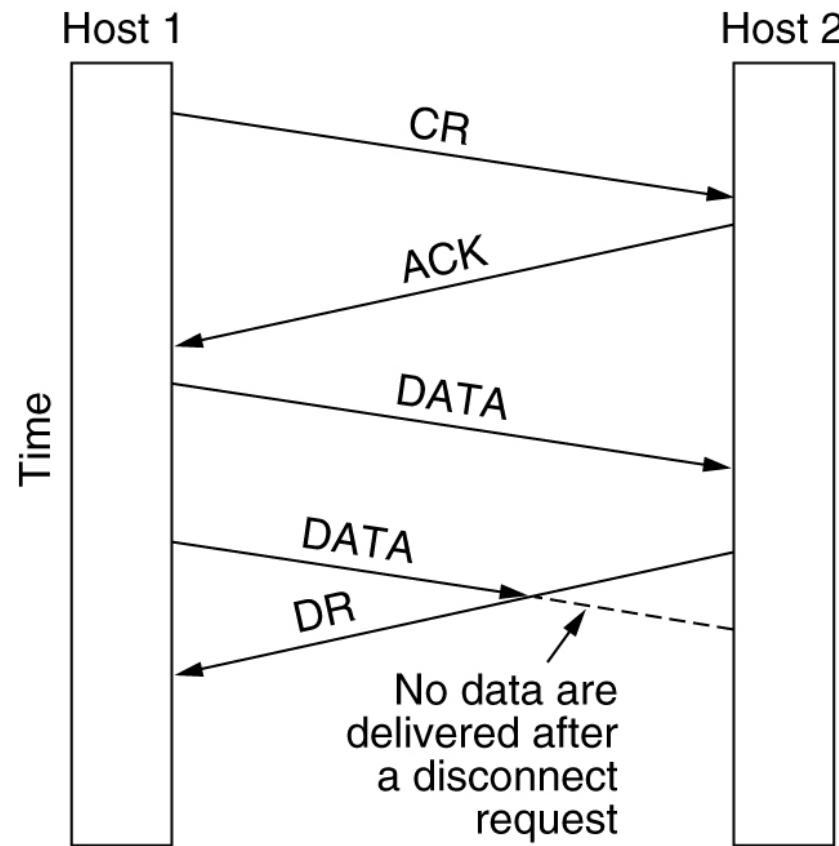
Three protocol scenarios for establishing a connection using a three-way handshake. CR denotes CONNECTION REQUEST.

- (a)** Normal operation,
- (b)** Old CONNECTION REQUEST appearing out of nowhere.
- (c)** Duplicate CONNECTION REQUEST and duplicate ACK.

In Fig.(b), the first segment is a delayed duplicate CONNECTION REQUEST from an old connection. This segment arrives at host 2 without host 1's knowledge. Host 2 reacts to this segment by sending host 1 an ACK segment, in effect asking for verification that host 1 was indeed trying to set up a new connection. When host 1 rejects host 2's attempt to establish a connection, host 2 realizes that it was tricked by a delayed duplicate and abandons the connection. In this way, a delayed duplicate does no damage

The worst case is when both a delayed CONNECTION REQUEST and an ACK are floating around in the subnet. This case is shown in Fig. (c). As in the previous example, host 2 gets a delayed CONNECTION REQUEST and replies to it. At this point, it is crucial to realize that host 2 has proposed using  $y$  as *the initial sequence number* for host 2 to host 1 traffic, knowing full well that no segments containing sequence number  $y$  or acknowledgements to  $y$  are *still in existence*. When the second delayed segment arrives at host 2, the fact that  $z$  *has been acknowledged rather than  $y$*  tells host 2 that this, too, is an old duplicate. *The important thing* to realize here is that there is no combination of old segments that can cause the protocol to fail and have a connection set up by accident when no one wants it.

# Connection Release



Abrupt disconnection with loss of data.

there are two styles of terminating a connection: asymmetric release and symmetric release. Asymmetric release is the way the telephone system works: when one party hangs up, the connection is broken. Symmetric release treats the connection as two separate unidirectional connections and requires each one to be released separately.

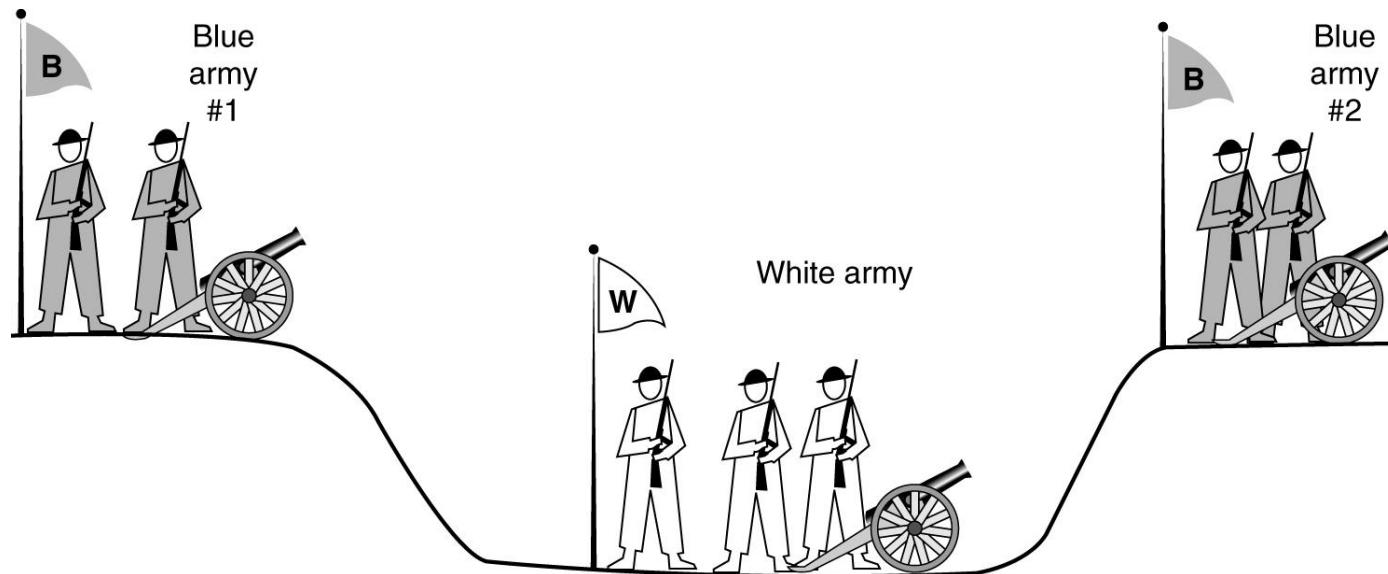
Asymmetric release is abrupt and may result in data loss. Consider the scenario of Fig. After the connection is established, host 1 sends a segment that arrives properly at host 2. Then host 1 sends another segment. Unfortunately, host 2 issues a DISCONNECT before the second segment arrives. The result is that the connection is released and data are lost.

Clearly, a more sophisticated release protocol is needed to avoid data loss. One way is to use symmetric release, in which each direction is released independently of the other one. Here, a host can continue to receive data even after it has sent a DISCONNECT segment.

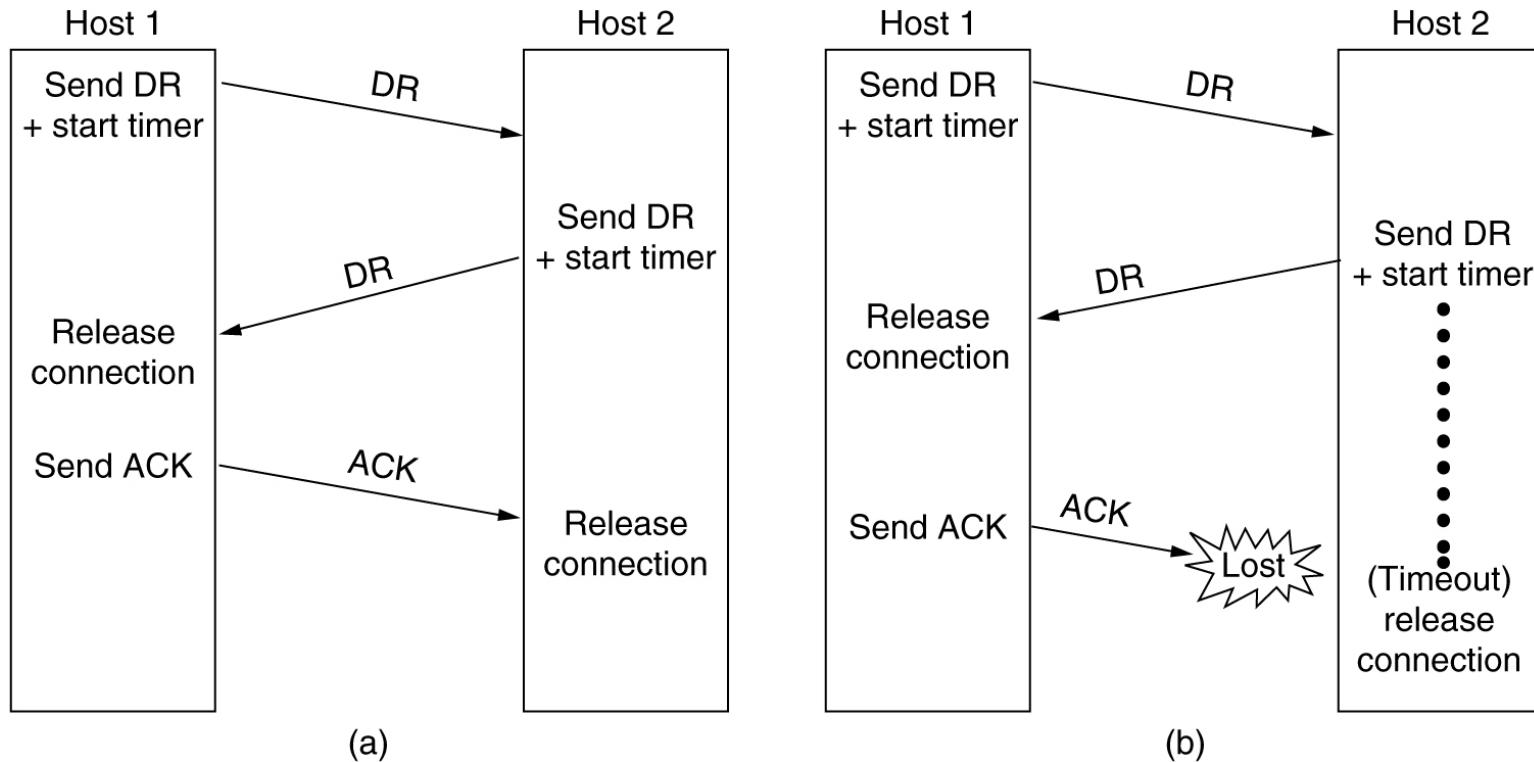
Symmetric release does the job when each process has a fixed amount of data to send and clearly knows when it has sent it. One can envision a protocol in which host 1 says “I am done. Are you done too?” If host 2 responds: “I am done too. Goodbye, the connection can be safely released.”

# Connection Release

The two-army problem.

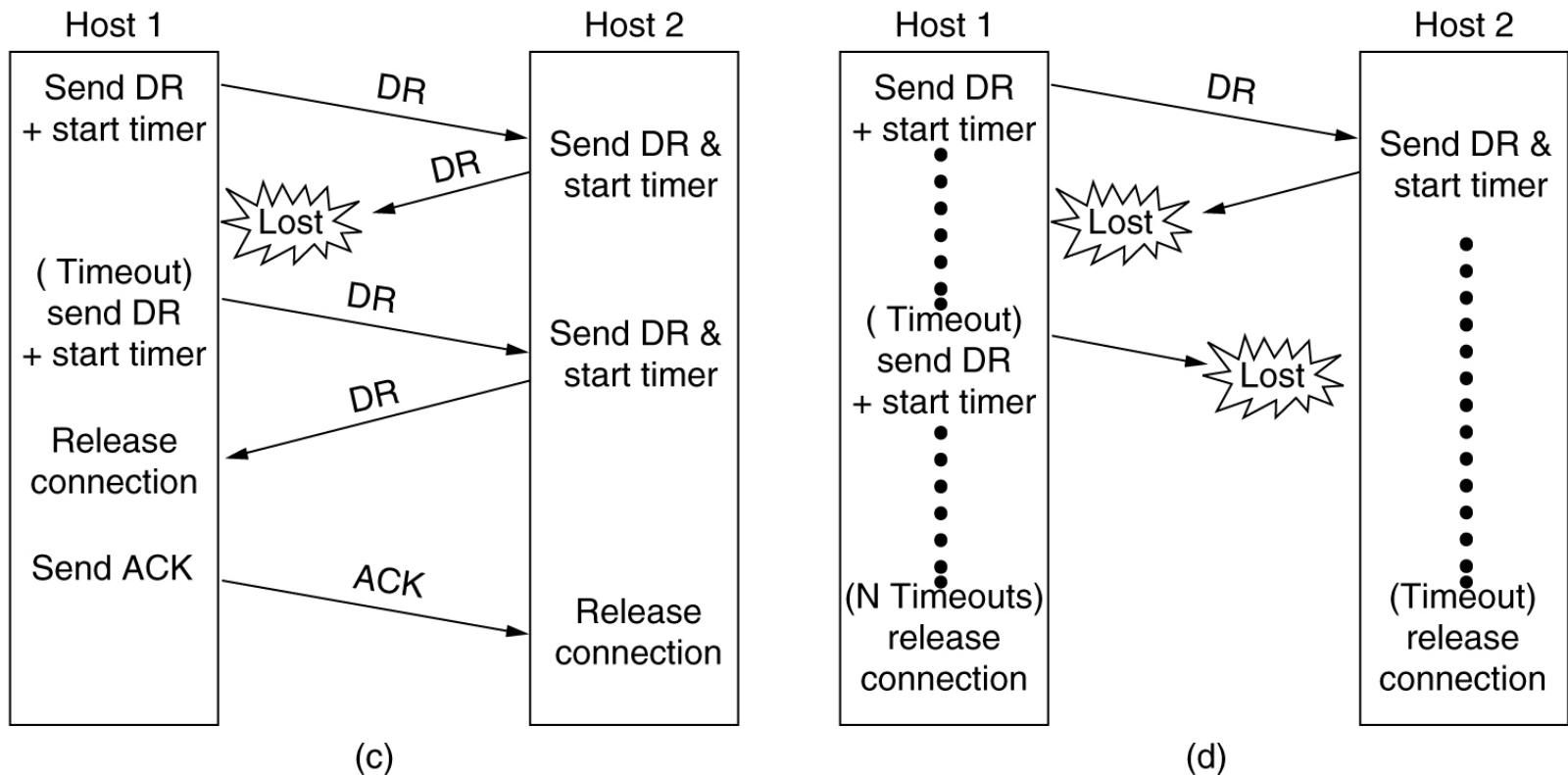


# Connection Release



Four protocol scenarios for releasing a connection. **(a)** Normal case of a three-way handshake. **(b)** final ACK lost.

# Connection Release



**(c)** Response lost. **(d)** Response lost and subsequent DRs lost.

In Fig. (a), we see the normal case in which one of the users sends a DR (DISCONNECTION REQUEST) segment to initiate the connection release. When it arrives, the recipient sends back a DR segment and starts a timer, just in case its DR is lost. When this DR arrives, the original sender sends back an ACK segment and releases the connection. Finally, when the ACK segment arrives, the receiver also releases the connection.

If the final ACK segment is lost, as shown in Fig.(b), the situation is saved by the timer. When the timer expires, the connection is released anyway. Now consider the case of the second DR being lost. The user initiating the disconnection will not receive the expected response, will time out, and will start all over again.

In Fig.(c), we see how this works, assuming that the second time no segments are lost and all segments are delivered correctly and on time.

Last scenario, Fig.(d), is the same as Fig. (c) except that now we assume all the repeated attempts to retransmit the DR also fail due to lost segments. After  $N$  retries, the sender just gives up and releases the connection. Meanwhile, the receiver times out and

# TCP

TCP is a connection oriented protocol; it creates a virtual connection between two TCPs to send data. In addition, TCP uses flow and error control mechanisms at the transport level. In brief, TCP is called a *connection-oriented, reliable transport protocol. It adds* connection-oriented and reliability features to the services of IP.

## Topics discussed in this section:

TCP Services

TCP Features

Segment

A TCP Connection

Flow Control

Error Control

# TCP Services

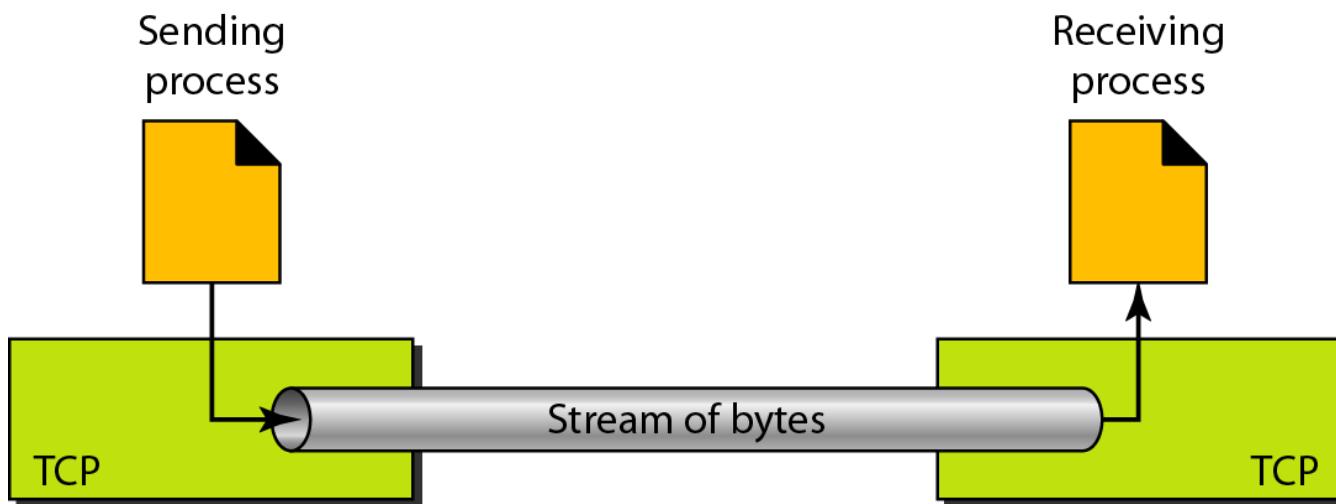
## **1 Process-to-Process Communication**

TCP provides process-to-process communication using port numbers. Below Table lists some well-known port numbers used by TCP.

<i>Port</i>	<i>Protocol</i>	<i>Description</i>
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
20	FTP, Data	File Transfer Protocol (data connection)
21	FTP, Control	File Transfer Protocol (control connection)
23	TELNET	Terminal Network
25	SMTP	Simple Mail Transfer Protocol
53	DNS	Domain Name Server
67	BOOTP	Bootstrap Protocol
79	Finger	Finger
80	HTTP	Hypertext Transfer Protocol
111	RPC	Remote Procedure Call

## **2 Stream Delivery Service**

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



**3 Sending and Receiving Buffers** Because the sending and the receiving processes may not write or read data at the same speed, TCP needs buffers for storage. There are two buffers, the sending buffer and the receiving buffer, one for each direction. One way to implement a buffer is to use a circular array of 1-byte locations as shown in Figure. For simplicity, we have shown two buffers of 20 bytes each. Normally the buffers are hundreds or thousands of bytes, depending on the implementation. We also show the buffers as the same size, which is not always the case.

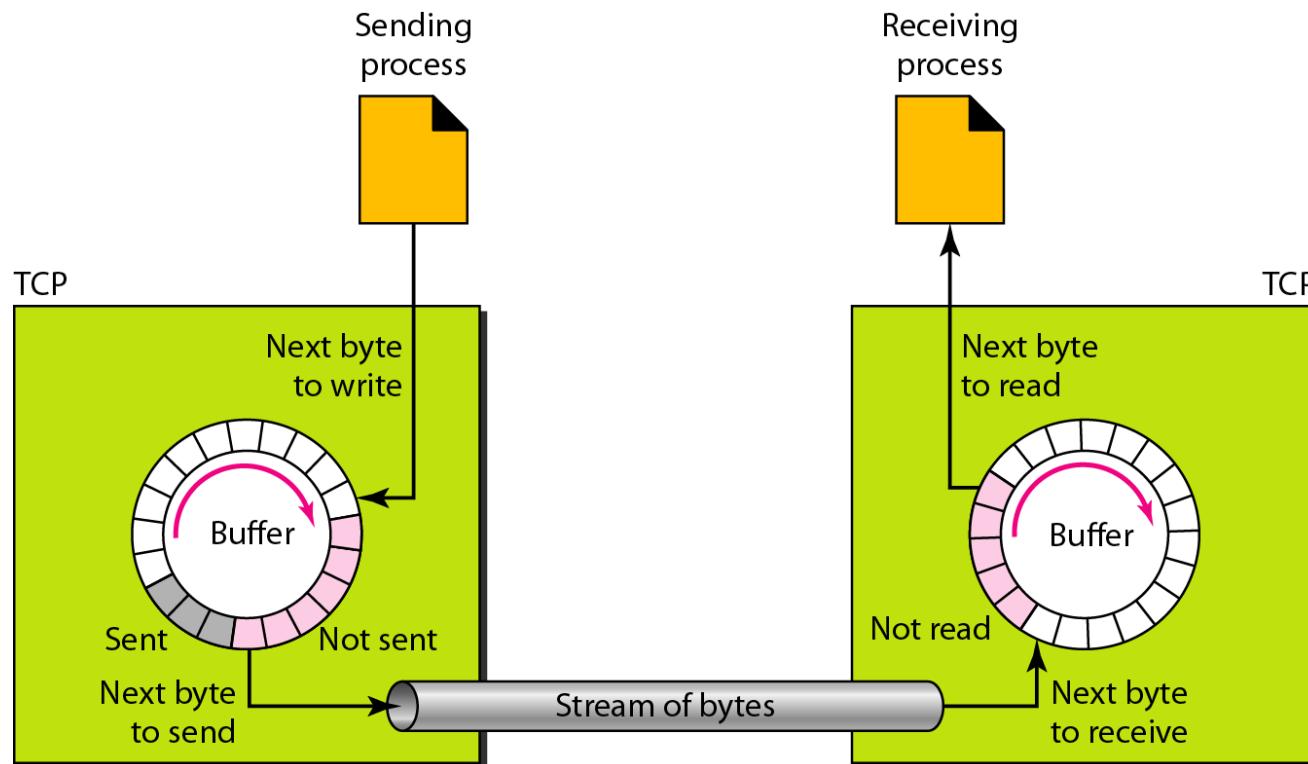


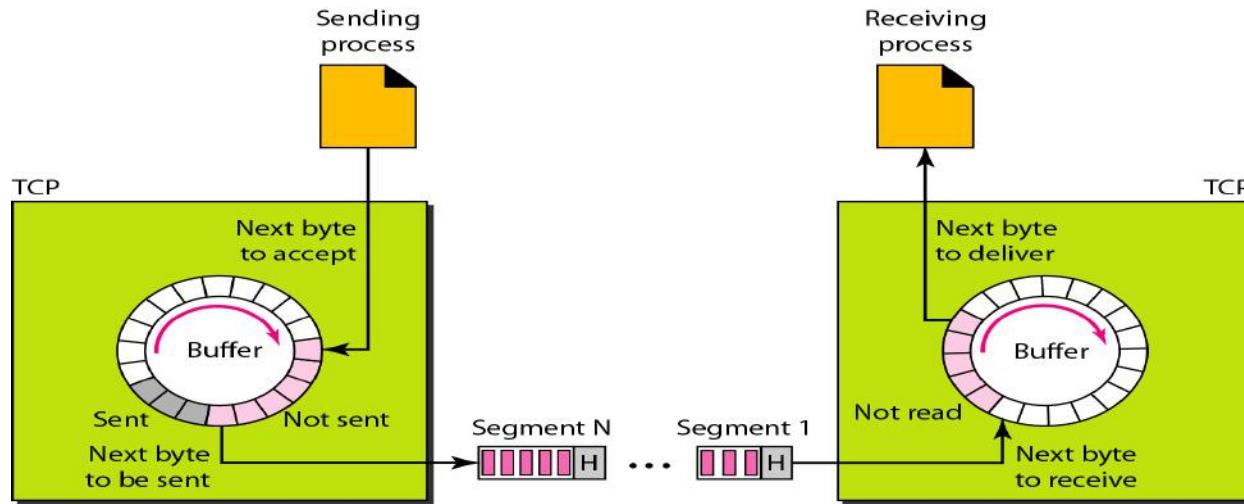
Figure shows the movement of the data in one direction. At the sending site, the buffer has three types of chambers. The white section contains empty chambers that can be filled by the sending process (producer). The gray area holds bytes that have been sent but not yet acknowledged. TCP keeps these bytes in the buffer until it receives an acknowledgment. The colored area contains bytes to be sent by the sending TCP.

However, as we will see later in this chapter, TCP may be able to send only part of this colored section. This could be due to the slowness of the receiving process or perhaps to congestion in the network. Also note that after the bytes in the gray chambers are acknowledged, the chambers are recycled and available for use by the sending process.

This is why we show a circular buffer.

The operation of the buffer at the receiver site is simpler. The circular buffer is divided into two areas (shown as white and colored). The white area contains empty chambers to be filled by bytes received from the network. The colored sections contain received bytes that can be read by the receiving process. When a byte is read by the receiving process, the chamber is recycled and added to the pool of empty chambers.

## 4 TCP segments



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. The segments are encapsulated in IP datagrams and transmitted.

This entire operation is transparent to the receiving process. Later we will see that segments may be received out of order, lost, or corrupted and resent. All these are handled by TCP with the receiving process unaware of any activities. Above fig shows how segments are created from the bytes in the buffers

## **5 Full-Duplex Communication**

TCP offers full-duplex service, in which data can flow in both directions at the same time. Each TCP then has a sending and receiving buffer, and segments move in both directions

## **6 Connection-Oriented Service**

TCP is a connection-oriented protocol. When a process at site A wants to send and receive data from another process at site B, the following occurs:

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

## **7 Reliable Service**

TCP is a reliable transport protocol. It uses an acknowledgment mechanism to check the safe and sound arrival of data. We will discuss this feature further in the section on error control.

# TCP Features

## **1 Numbering System**

There are two fields called the sequence number and the acknowledgment number. These two fields refer to the byte number and not the segment number.

**Byte Number** The bytes of data being transferred in each connection are numbered by TCP. The numbering starts with a randomly generated number. For example, if the random number happens to be 1057 and the total data to be sent are 6000 bytes, the bytes are numbered from 1057 to 7056. We will see that byte numbering is used for flow and error control.

**Sequence Number** After the bytes have been numbered, TCP assigns a sequence number to each segment that is being sent. The sequence number for each segment is the number of the first byte carried in that segment.

**Acknowledgment Number** The value of the acknowledgment field in a segment defines the number of the next byte a party expects to receive. The acknowledgment number is cumulative.

## **2 Flow Control**

TCP, provides *flow control*. *The receiver of the data controls the amount of data that are to be sent by the sender.* This is done to prevent the receiver from being overwhelmed with data. The numbering system allows TCP to use a byte-oriented flow control.

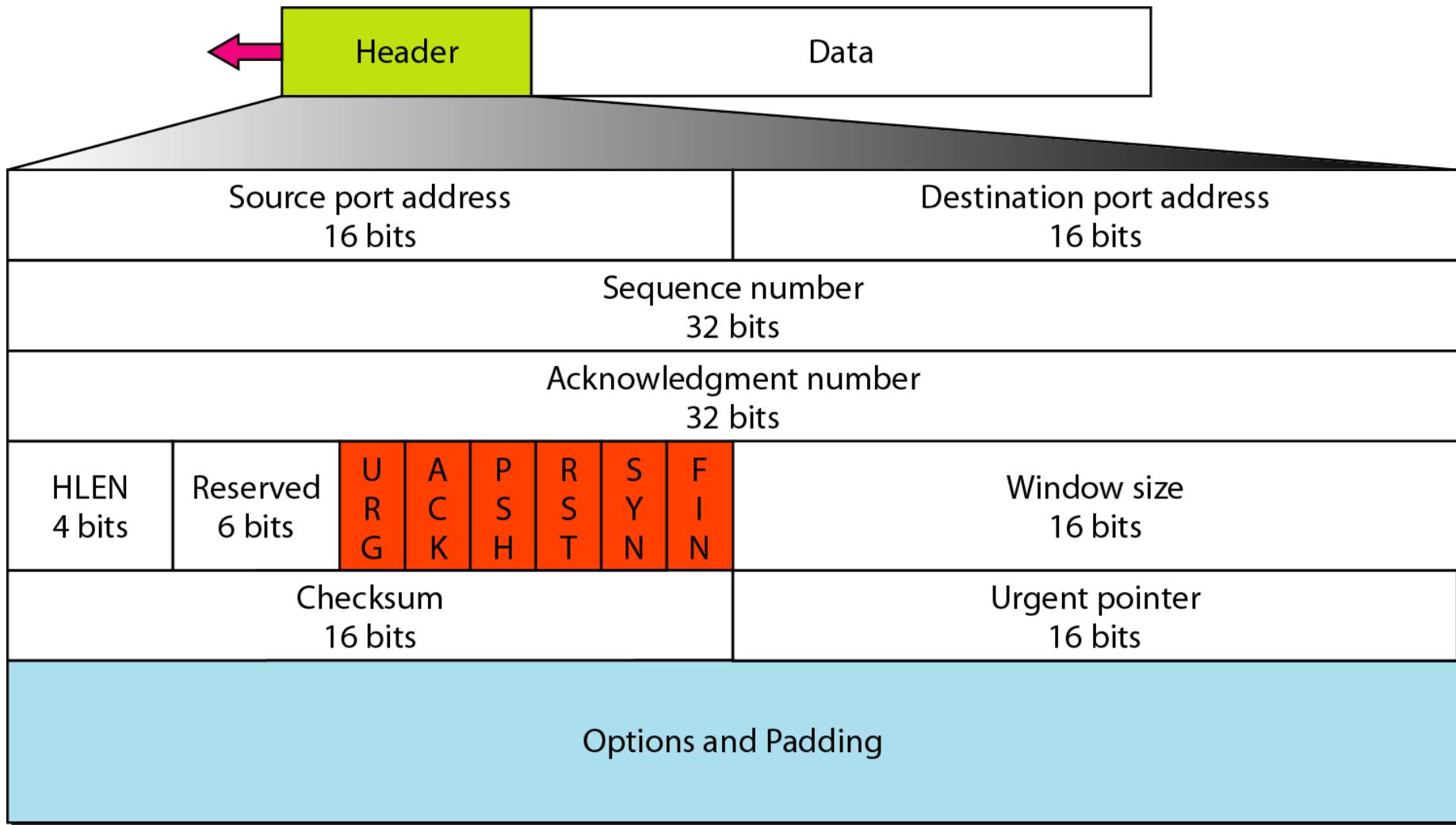
## **3 Error Control**

To provide reliable service, TCP implements an error control mechanism. Although error control considers a segment as the unit of data for error detection (loss or corrupted segments), error control is byte-oriented, as we will see later.

## **4 Congestion Control**

TCP takes into account congestion in the network. The amount of data sent by a sender is not only controlled by the receiver (flow control), but is also determined by the level of congestion in the network

## TCP segment format



The segment consists of a 20- to 60-byte header.,.

**Source port address.** This is a 16-bit field that defines the port number of the application program in the host that is sending the segment.

**Destination port address.** This is a 16-bit field that defines the port number of the application program in the host that is receiving the segment.

**Sequence number.** This 32-bit field defines the number assigned to the first byte of data contained in this segment. As we said before, TCP is a stream transport protocol. To ensure connectivity, each byte to be transmitted is numbered. The sequence number tells the destination which byte in this sequence comprises the first byte in the segment. During connection establishment, each party uses a random number generator to create an initial sequence number (ISN), which is usually different in each direction.

**Acknowledgment number.** This 32-bit field defines the byte number that the receiver of the segment is expecting to receive from the other party. If the receiver of the segment has successfully received byte number  $x$  *from the other party*, it defines  $x + 1$  as the acknowledgment number. *Acknowledgment and data can be piggybacked together.*

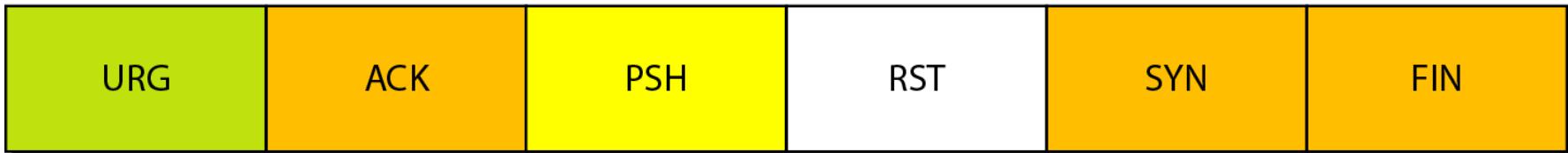
**Header length.** This 4-bit field indicates the number of 4-byte words in the TCP header. The length of the header can be between 20 and 60 bytes. Therefore, the value of this field can be between 5 ( $5 \times 4 = 20$ ) and 15 ( $15 \times 4 = 60$ ).

**Reserved.** This is a 6-bit field reserved for future use.

**Control.** This field defines 6 different control bits or flags as shown in Figure. One or more of these bits can be set at a time.

URG: Urgent pointer is valid  
ACK: Acknowledgment is valid  
PSH: Request for push

RST: Reset the connection  
SYN: Synchronize sequence numbers  
FIN: Terminate the connection



These bits enable flow control, connection establishment and termination, connection abortion, and the mode of data transfer in TCP.

**Window size.** This field defines the size of the window, in bytes, that the other party must maintain. Note that the length of this field is 16 bits, which means that the maximum size of the window is 65,535 bytes. This value is normally referred to as the receiving window (rwnd) and is determined by the receiver. The sender must obey the dictation of the receiver in this case.

**Checksum.** This 16-bit field contains the checksum. The calculation of the checksum for TCP follows the same procedure as the one described for UDP. However, the inclusion of the checksum in the UDP datagram is optional, whereas the inclusion of the checksum for TCP is mandatory. The same pseudoheader, serving the same purpose, is added to the segment. For the TCP pseudoheader, the value for the protocol field is 6.

[Urgent pointer](#). This 16-bit field, which is valid only if the urgent flag is set, is used when the segment contains urgent data. It defines the number that must be added to the sequence number to obtain the number of the last urgent byte in the data section of the segment. This will be discussed later in this chapter.

[Options](#). There can be up to 40 bytes of optional information in the TCP header. We will not discuss these options here; please refer to the reference list for more information.

## **A TCP Connection**

TCP is connection-oriented. A connection-oriented transport protocol establishes a virtual path between the source and destination. All the segments belonging to a message are then sent over this virtual path. Using a single virtual pathway for the entire message facilitates the acknowledgment process as well as retransmission of damaged or lost frames.

In TCP, connection-oriented transmission requires three phases:

1. connection establishment,
2. data transfer,
3. connection termination.

# TCP connection establishment(3 way handshaking)

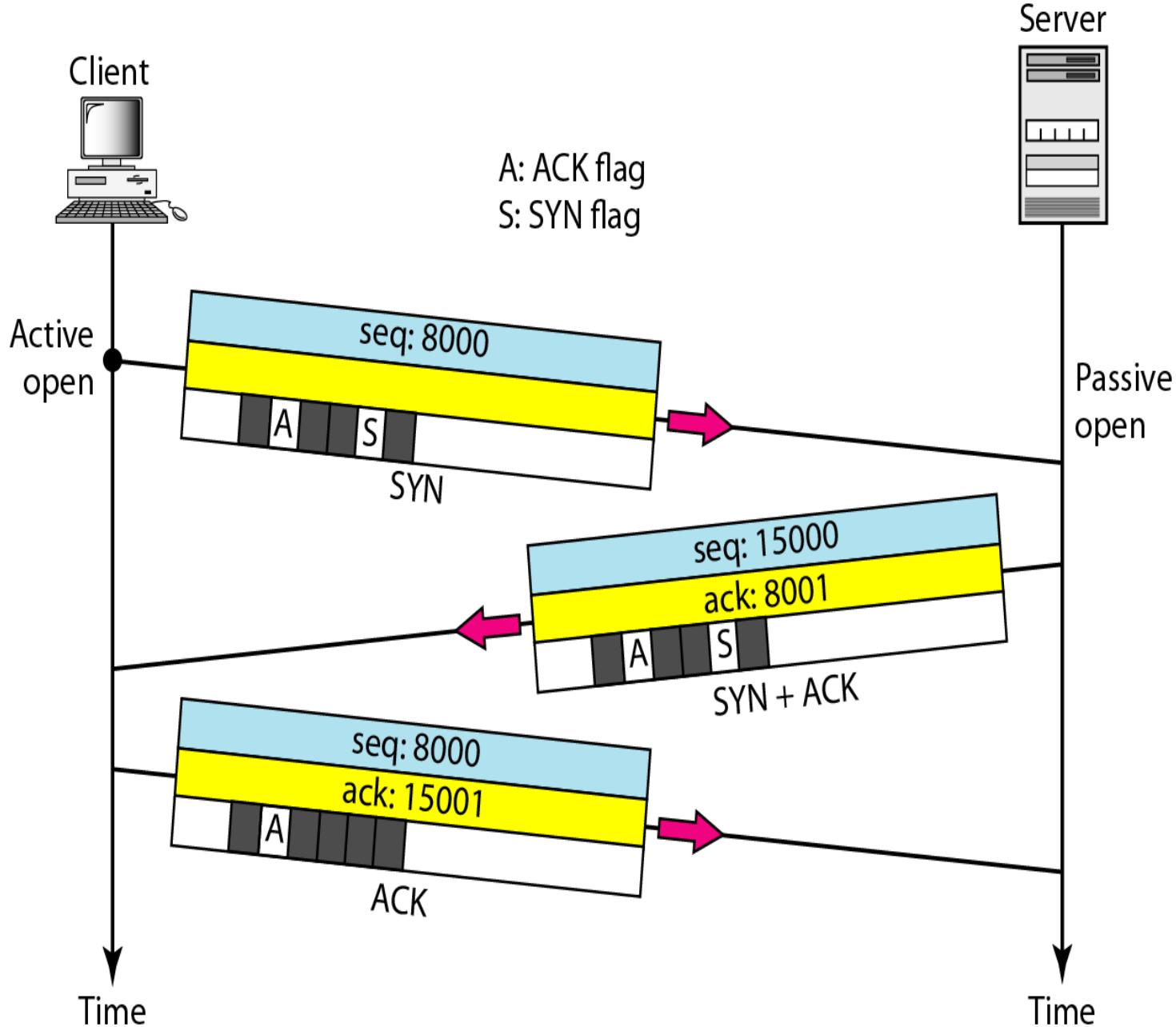
1 The client sends the first segment, a SYN segment, in which only the SYN flag is set.

NOTE:A SYN segment cannot carry data, but it consumes one sequence number.

2. The server sends the second segment, a SYN +ACK segment, with 2 flag bits set: SYN and ACK. This segment has a dual purpose. It is a SYN segment for communication in the other direction and serves as the acknowledgment for the SYN segment. It consumes one sequence number.  
NOTE:A SYN+ACK segment cannot carry data, but does consume one sequence number

3. The client sends the third segment. This is just an ACK segment. It acknowledges the receipt of the second segment with the ACK flag and acknowledgment number field. Note that the sequence number in this segment is the same as the one in the SYN segment; the ACK segment does not consume any sequence numbers.

NOTE: An ACK segment, if carrying no data, consumes no sequence number



## SYN Flooding Attack

This happens when a malicious attacker sends a large number of SYN segments to a server, pretending that each of them is coming from a different client by faking the source IP addresses in the datagram's.

The server, assuming that the clients are issuing an active open, allocates the necessary resources, such as creating communication tables and setting timers. The TCP server then sends the SYN +ACK segments to the fake clients, which are lost. During this time, however, a lot of resources are occupied without being used. If, during this short time, the number of SYN segments is large, the server eventually runs out of resources and may crash. This SYN flooding attack belongs to a type of security attack known as a denial-of-service attack, in which an attacker monopolizes a system with so many service requests that the system collapses and denies service to every request.

### SOLUTIONS:

- 1 Some have imposed a limit on connection requests during a specified period of time.
- 2 Others filter out datagrams coming from unwanted source addresses.
- 3 One recent strategy is to postpone resource allocation until the entire connection is set up using what is called a cookie.

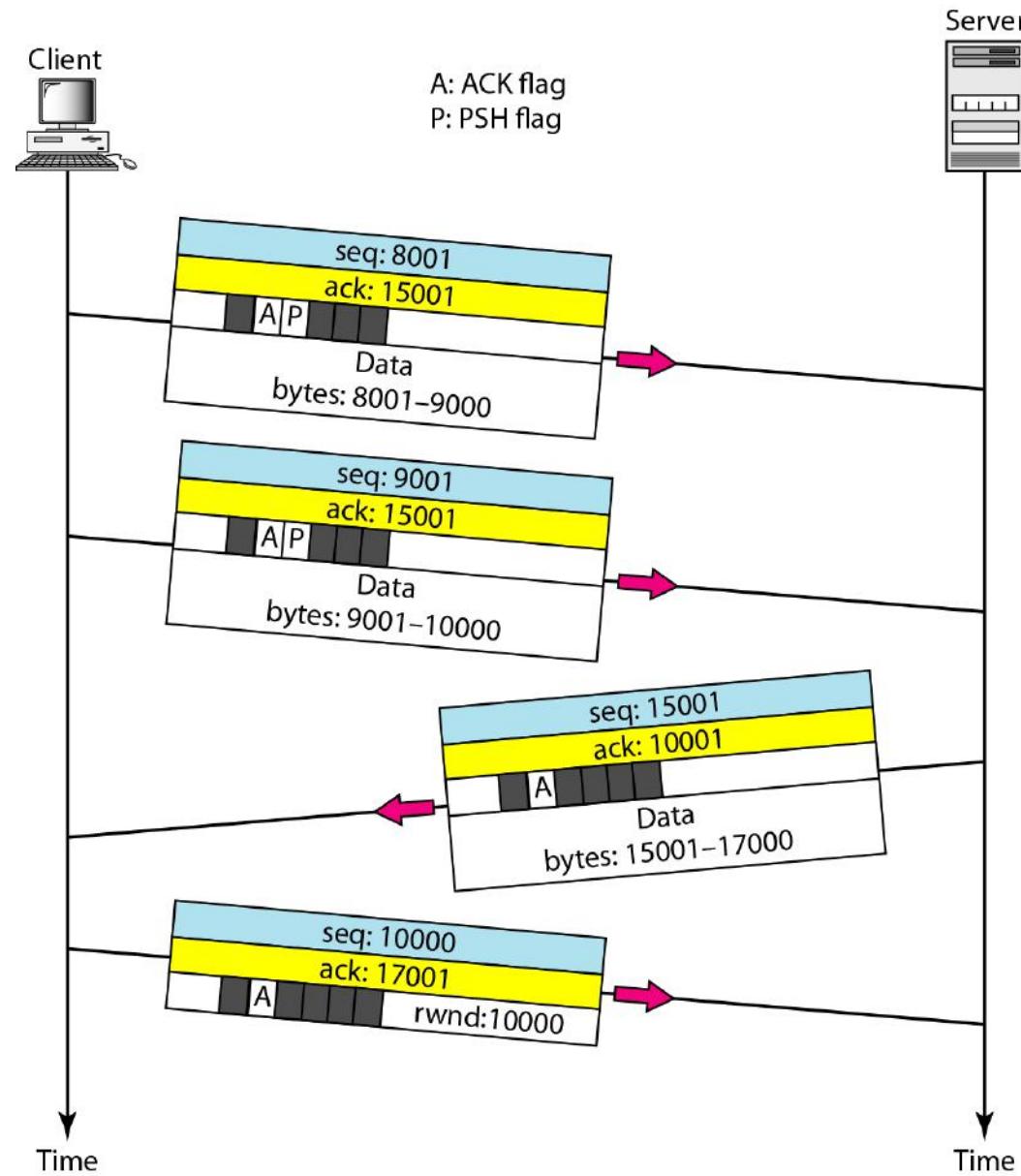
## Data Transfer

After connection is established, bidirectional data transfer can take place. The client and server can both send data and acknowledgments. Data traveling in the same direction as an acknowledgment are carried on the same segment. The acknowledgment is piggybacked with the data

In this example, after connection is established (not shown in the figure), the client sends 2000 bytes of data in two segments. The server then sends 2000 bytes in one segment. The client sends one more segment. The first three segments carry both data and acknowledgment, but the last segment carries only an acknowledgment because there are no more data to be sent.

Note the values of the sequence and acknowledgment numbers. The data segments sent by the client have the PSH (push) flag set so that the server TCP knows to deliver data to the server process as soon as they are received.

## Data transfer



**PUSHING DATA:** Delayed transmission and delayed delivery of data may not be acceptable by the application program.

TCP can handle such a situation. The application program at the sending site can request a *push operation*. *This means that the sending TCP must not wait for the window to be filled.* It must create a segment and send it immediately. The sending TCP must also set the push bit (PSH) to let the receiving TCP know that the segment includes data that must be delivered to the receiving application program as soon as possible and not to wait for more data to come.

**Urgent Data :** TCP is a stream-oriented protocol. This means that the data are presented from the application program to TCP as a stream of bytes. Each byte of data has a position in the stream. However, sending application program wants a piece of data to be read out of order by the receiving application program.

**Connection Termination** (three-way handshaking and four-way handshaking with a half-close option.)

1. In a normal situation, the client TCP, after receiving a close command from the client process, sends the first segment, a FIN segment in which the FIN flag is set.

Note that a FIN segment can include the last chunk of data sent by the client, or it can be just a control segment as shown in Figure. If it is only a control segment, it consumes only one sequence number.

NOTE: The FIN segment consumes one sequence number if it does not carry data.

2 The server TCP, after receiving the FIN segment, informs its process of the situation and sends the second segment, a FIN +ACK segment, to confirm the receipt of the FIN segment from the client and at the same time to announce the closing of the connection in the other direction. This segment can also contain the last chunk of data from the server. If it does not carry data, it consumes only one sequence number.

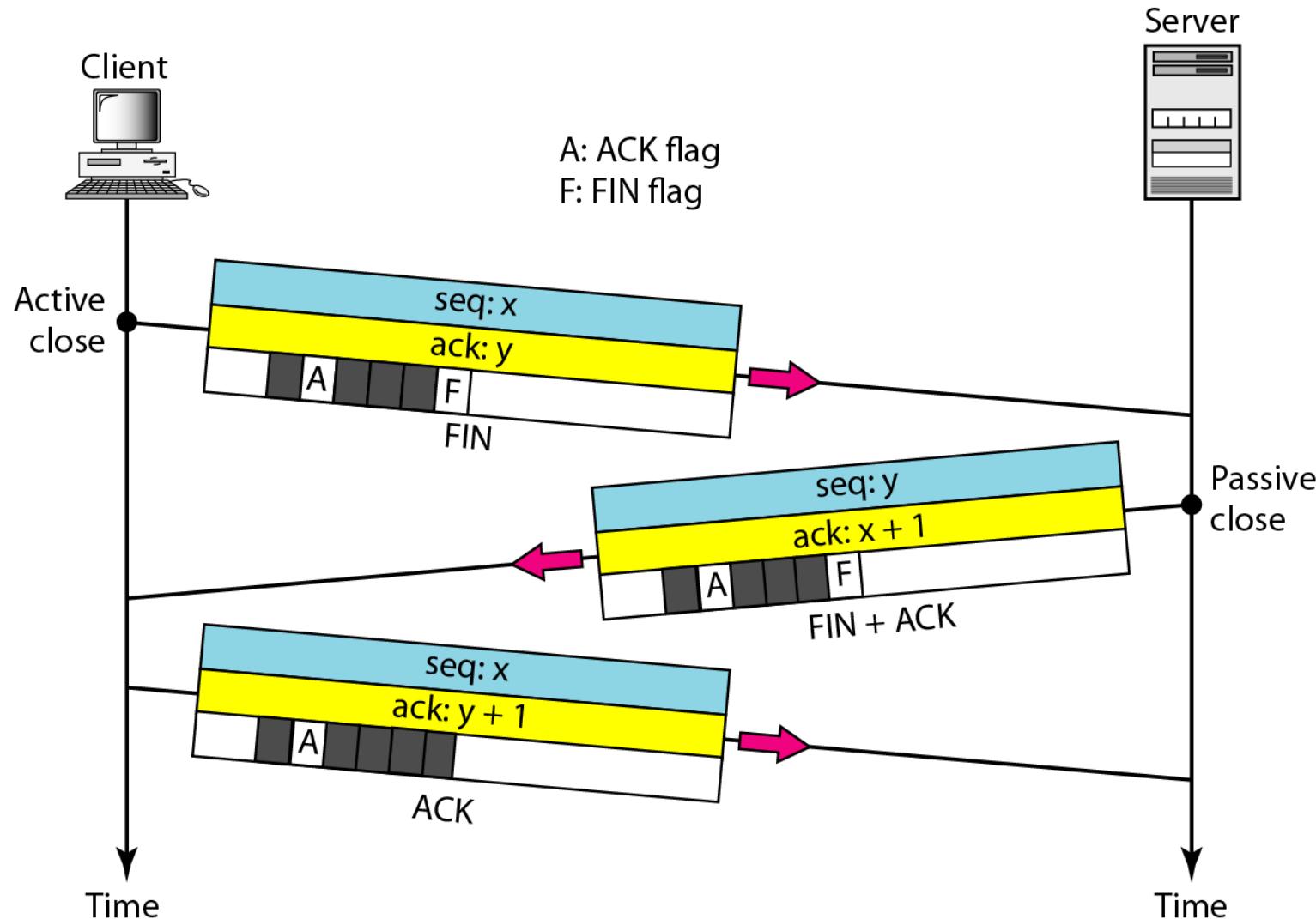
NOTE: The FIN +ACK segment consumes one sequence number if it does not carry data.

3. The client TCP sends the last segment, an ACK segment, to confirm the receipt of the FIN segment from the TCP server. This segment contains the acknowledgment number, which is 1 plus the sequence number received in the FIN segment from the server. This segment cannot carry data and consumes no sequence numbers.

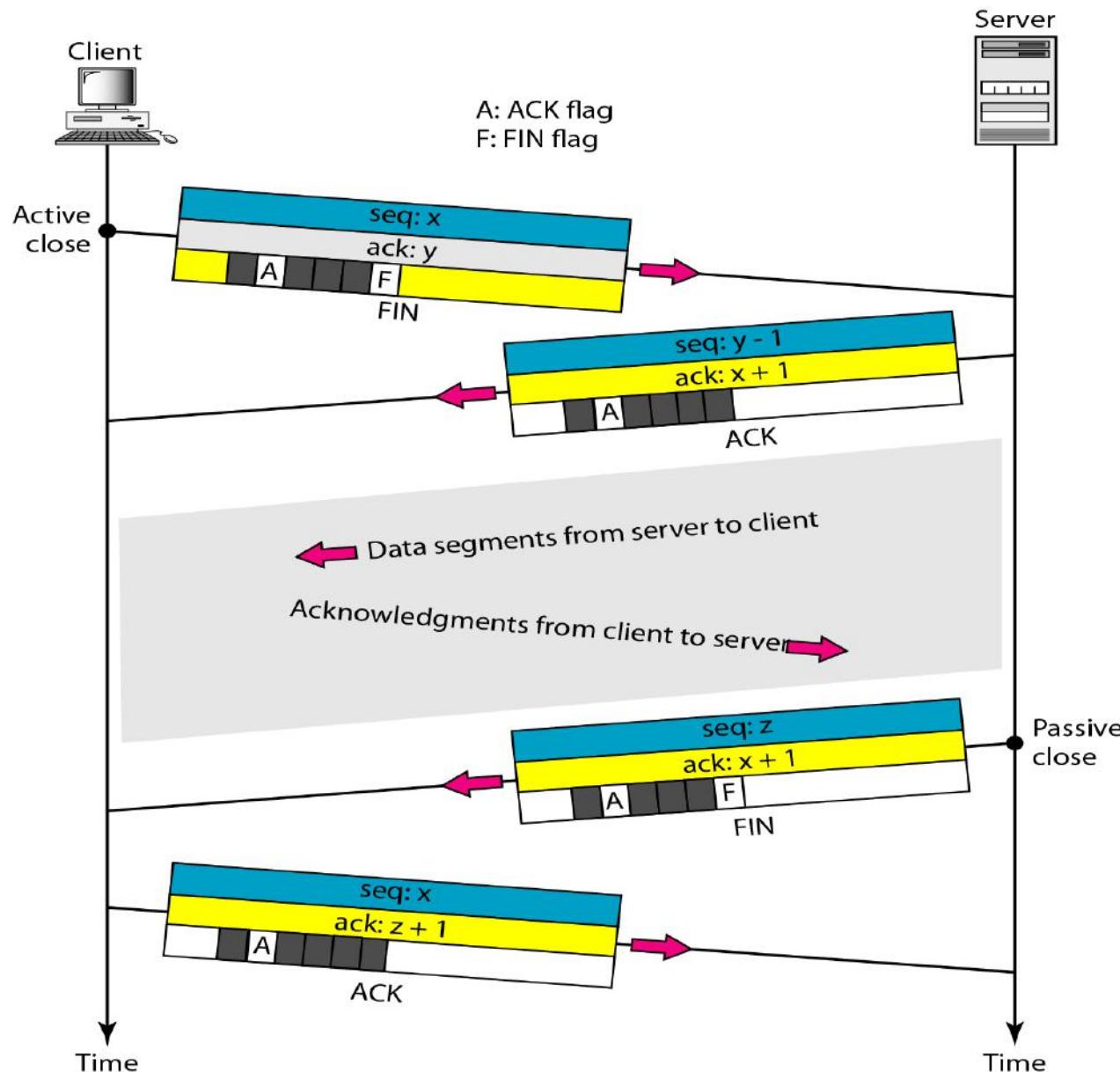
**Half-Close** In TCP, one end can stop sending data while still receiving data. This is called a half-close. Although either end can issue a half-close, it is normally initiated by the client. It can occur when the server needs all the data before processing can begin.

A good example is sorting. When the client sends data to the server to be sorted, the server needs to receive all the data before sorting can start. This means the client, after sending all the data, can close the connection in the outbound direction. However, the inbound direction must remain open to receive the sorted data. The server, after receiving the data, still needs time for sorting; its outbound direction must remain open

## Connection termination using three-way handshaking



## Figure 23.21 Half-close



## Flow Control or TCP Sliding Window

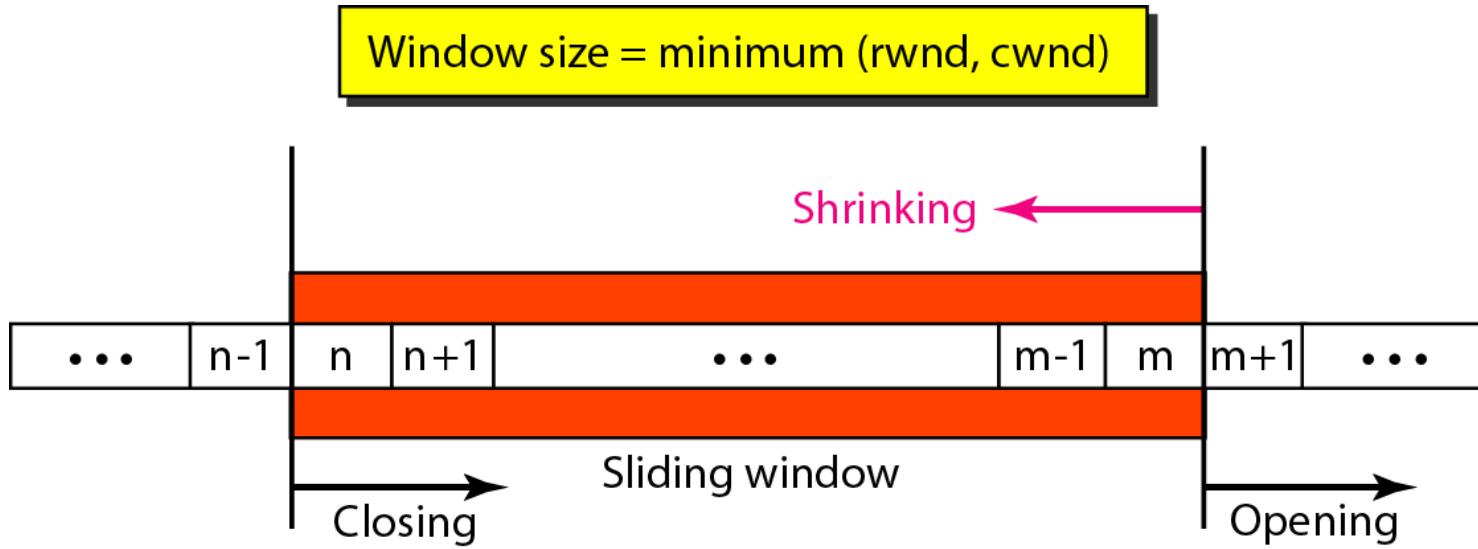
TCP uses a sliding window, to handle flow control. The sliding window protocol used by TCP, however, is something between the *Go-Back-N* and Selective Repeat sliding window.

The sliding window protocol in TCP looks like the *Go-Back-N* protocol because it does not use NAKs;  
it looks like Selective Repeat because the receiver holds the out-of-order segments until the missing ones arrive.

There are two big differences between this sliding window and the one we used at the data link layer.

- 1 the sliding window of TCP is byte-oriented; the one we discussed in the data link layer is frame-oriented.
- 2 the TCP's sliding window is of variable size; the one we discussed in the data link layer was of fixed size

## *Sliding window*



The window is opened, closed, or shrunk. These three activities, as we will see, are in the control of the receiver (and depend on congestion in the network), not the sender.

The sender must obey the commands of the receiver in this matter.

Opening a window means moving the right wall to the right. This allows more new bytes in the buffer that are eligible for sending.

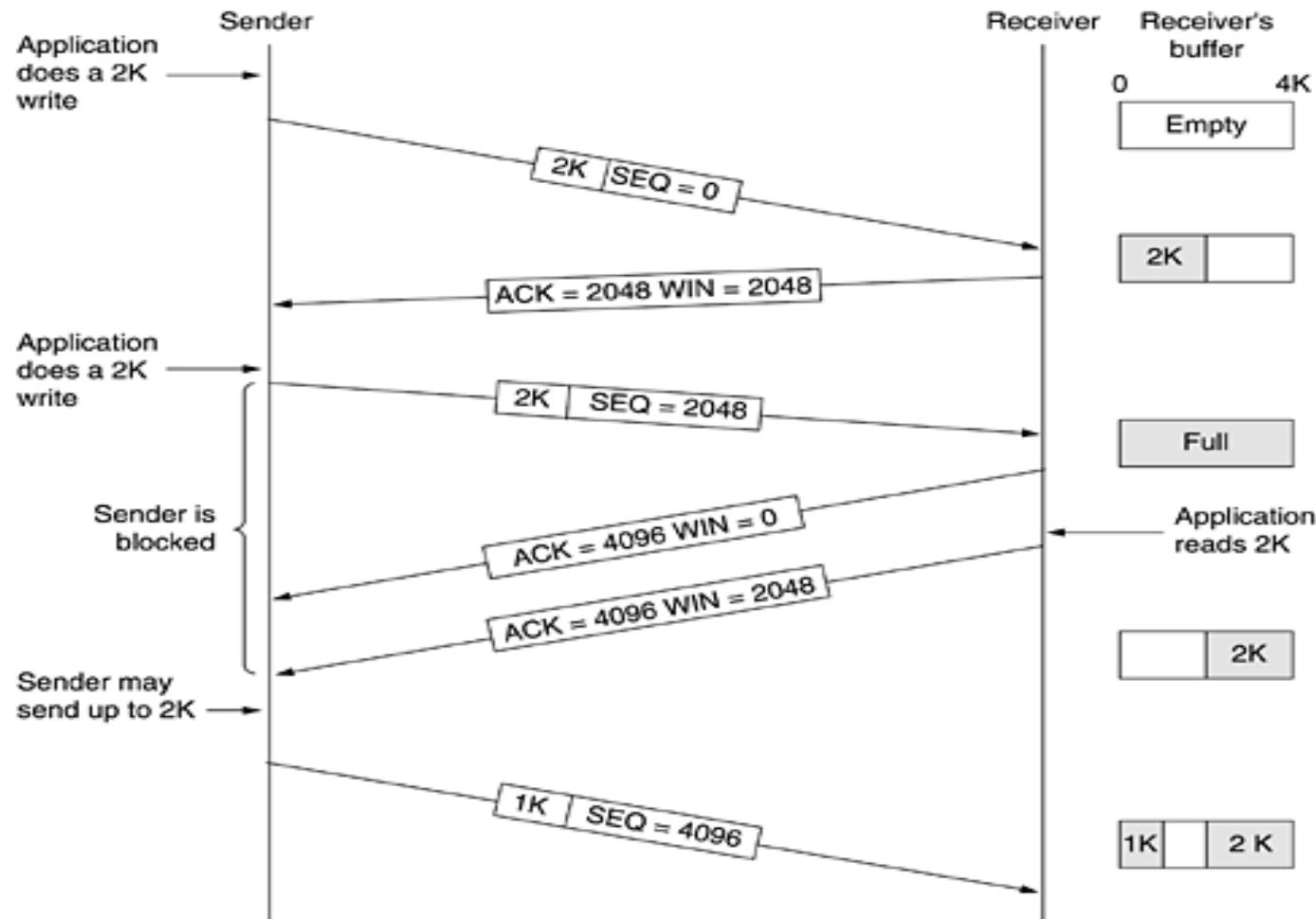
Closing the window means moving the left wall to the right. This means that some bytes have been acknowledged and the sender need not worry about them anymore.

Shrinking the window means moving the right wall to the left.

The size of the window at one end is determined by the lesser of two values: receiver window (rwnd) or congestion window (cwnd).

The receiver window is the value advertised by the opposite end in a segment containing acknowledgment. It is the number of bytes the other end can accept before its buffer overflows and data are discarded.

The congestion window is a value determined by the network to avoid congestion



Window management in TCP

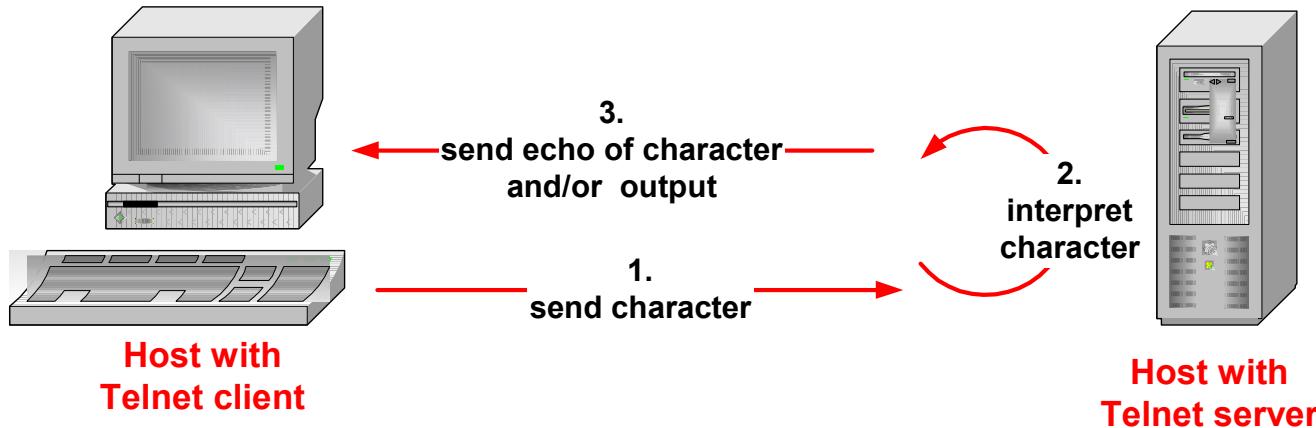
When the window is 0, the sender may not normally send segments, with two exceptions.

- 1) urgent data may be sent, for example, to allow the user to kill the process running on the remote machine.
- 2) the sender may send a 1-byte segment to force the receiver to reannounce the next byte expected and the window size. This packet is called a **window probe**.

The TCP standard explicitly provides this option to prevent deadlock if a window update ever gets lost.

Senders are not required to transmit data as soon as they come in from the application. Neither are receivers required to send acknowledgements as soon as possible.

For example, in Fig. when the first 2 KB of data came in, TCP, knowing that it had a 4-KB window, would have been completely correct in just buffering the data until another 2 KB came in, to be able to transmit a segment with a 4-KB payload. This freedom can be used to improve performance



Remote terminal applications (e.g., Telnet) send characters to a server. The server interprets the character and sends the output at the server to the client.

For each character typed, you see three packets:

**Client ☐ Server:** Send typed character

**Server ☐ Client:** Echo of character (or user output) and acknowledgement for first packet

**Client ☐ Server:** Acknowledgement for second packet

# Delayed Acknowledgement

- TCP delays transmission of ACKs for up to 500ms
- Avoid to send ACK packets that do not carry data.
  - The hope is that, within the delay, the receiver will have data ready to be sent to the receiver. Then, the ACK can be piggybacked with a data segment

## Exceptions:

- ACK should be sent for every full sized segment
- Delayed ACK is not used when packets arrive out of order

Although delayed acknowledgements reduce the load placed on the network by the receiver, a sender that sends multiple short packets (e.g., 41-byte packets containing 1 byte of data) is still operating inefficiently. A way to reduce this usage is known as **Nagle's algorithm (Nagle, 1984)**.

## Nagel's Rule

Send one byte and buffer all subsequent bytes until acknowledgement is received. Then send all buffered bytes in a single TCP segment and start buffering again until the sent segment is acknowledged.

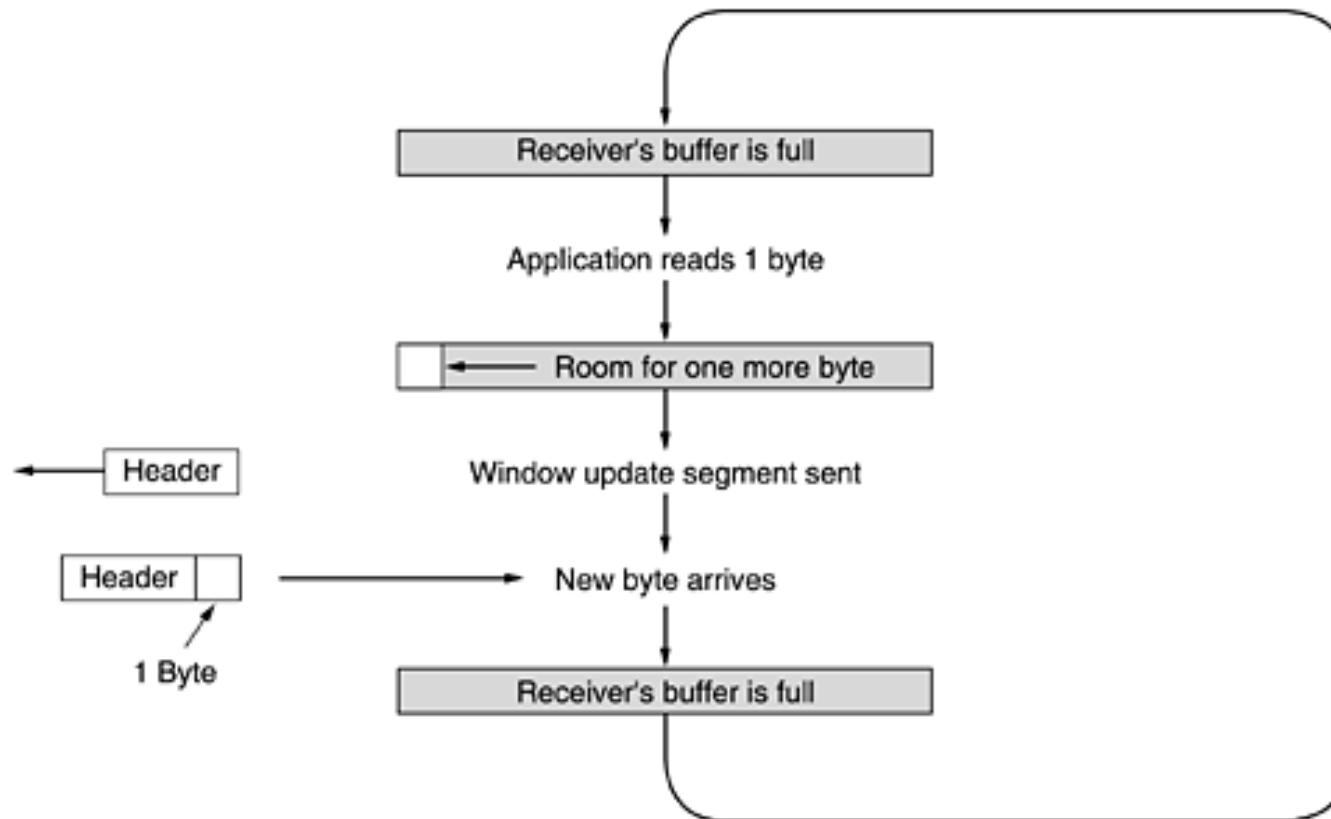
Nagle's algorithm will put the many pieces in one segment, greatly reducing the bandwidth used

Nagle's algorithm is widely used by TCP implementations, but there are times when it is better to disable it. In particular, in interactive games that are run over the Internet.

A more subtle problem is that Nagle's algorithm can sometimes interact with delayed acknowledgements to cause a temporary deadlock: the receiver waits for data on which to piggyback an acknowledgement, and the sender waits on the acknowledgement to send more data.

Because of these problems, Nagle's algorithm can be disabled (which is called the *TCP NODELAY option*).

Another problem that can degrade TCP performance is the **silly window syndrome** (Clark, 1982).



Clark's solution is to prevent the receiver from sending a window update for 1 byte. Instead, it is forced to wait until it has a decent amount of space available and advertise that instead. Specifically, the receiver should not send a window update until it can handle the maximum segment size it advertised when the connection was established or until its buffer is half empty, whichever is smaller.

Furthermore, the sender can also help by not sending tiny segments. Instead, it should wait until it can send a full segment, or at least one containing half of the receiver's buffer size.

The goal is for the sender not to send small segments and the receiver not to ask for them. (Nagel + Clark). Both are used to improve TCP performance

The receiver will buffer the data until it can be passed up to the application in order (handling out of order segments)

## **Cumulative acknowledgements**

## Error Control

TCP is a reliable transport layer protocol. This means that an application program that delivers a stream of data to TCP relies on TCP to deliver the entire stream to the application program on the other end in order, without error, and without any part lost or duplicated.

TCP provides reliability using error control. Error control includes mechanisms for detecting corrupted segments, lost segments, out-of-order segments, and duplicated segments. Error control also includes a mechanism for correcting errors after they are detected. Error detection and correction in TCP is achieved through the use of three simple tools: **checksum, acknowledgment, and time-out.**

### Checksum

Each segment includes a checksum field which is used to check for a corrupted segment. If the segment is corrupted, it is discarded by the destination TCP and is considered as lost. TCP uses a 16-bit checksum that is mandatory in every segment

**Figure 23.11** *Checksum calculation of a simple UDP user datagram*

153.18.8.105			
171.2.14.10			
All 0s	17	15	
1087		13	
15		All 0s	
T	E	S	T
I	N	G	All 0s

10011001 00010010	→	153.18
00001000 01101001	→	8.105
10101011 00000010	→	171.2
00001110 00001010	→	14.10
00000000 00010001	→	0 and 17
00000000 00001111	→	15
00000100 00111111	→	1087
00000000 00001101	→	13
00000000 00001111	→	15
00000000 00000000	→	0 (checksum)
01010100 01000101	→	T and E
01010011 01010100	→	S and T
01001001 01001110	→	I and N
01000111 00000000	→	G and 0 (padding)
 10010110 11101011		→ Sum
 01101001 00010100		→ Checksum

## **Acknowledgment**

TCP uses acknowledgments to confirm the receipt of data segments. Control segments that carry no data but consume a sequence number are also acknowledged. ACK segments are never acknowledged. ACK segments do not consume sequence numbers and are not acknowledged.

## **Retransmission**

The heart of the error control mechanism is the retransmission of segments. When a segment is corrupted, lost, or delayed, it is retransmitted.

In modern implementations, a retransmission occurs if the retransmission timer expires or three duplicate ACK segments have arrived.

Retransmission After RTO (retransmission time out)

Retransmission After Three Duplicate ACK Segments (also called fast retransmission)

## **Out-of-Order Segments**

Data may arrive out of order and be temporarily stored by the receiving TCP, but yet guarantees that no out-of-order segment is delivered to the process

# TCP Congestion Control

When the load offered to any network is more than it can handle, congestion builds up.

The network layer detects congestion when queues grow large at routers and tries to manage it, if only by dropping packets. It is up to the transport layer to receive congestion feedback from the network layer and slow down the rate of traffic that it is sending into the network.

For Congestion control, transport protocol uses an AIMD (Additive Increase Multiplicative Decrease) control law.

TCP congestion control is based on implementing this approach using a window called **congestion window**. TCP adjusts the size of the window according to the AIMD rule.

The window size at the sender is set as follows:

**Send Window = MIN (flow control window,  
congestion window)**

where

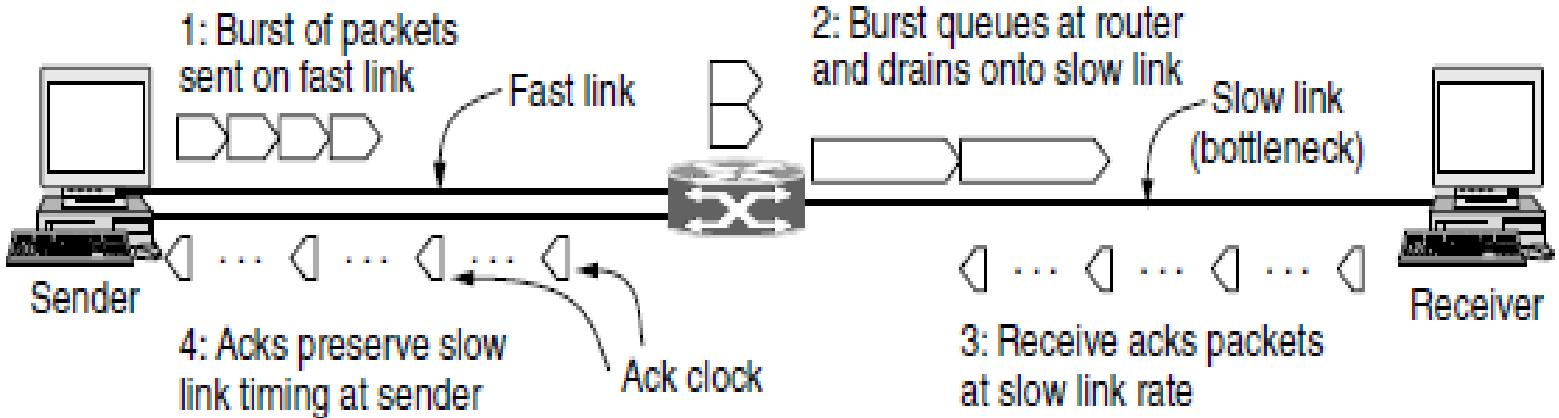
**flow control window** is advertised by the receiver (rwnd)

**congestion window** is adjusted based on feedback from the

Modern congestion control was added to TCP largely through the efforts of Van Jacobson (1988). It is a fascinating story. Starting in 1986, the growing popularity of the early Internet led to the first occurrence of what became known as a **congestion collapse**, a prolonged period during which good put dropped suddenly (i.e., by more than a factor of 100) due to congestion in the network. Jacobson (and many others) set out to understand what was happening and remedy the situation.

To start, he observed that packet loss is a suitable signal of congestion. This signal comes a little late (as the network is already congested) but it is quite dependable

At the beginning how sender knows at what speed receiver can receive the packets?

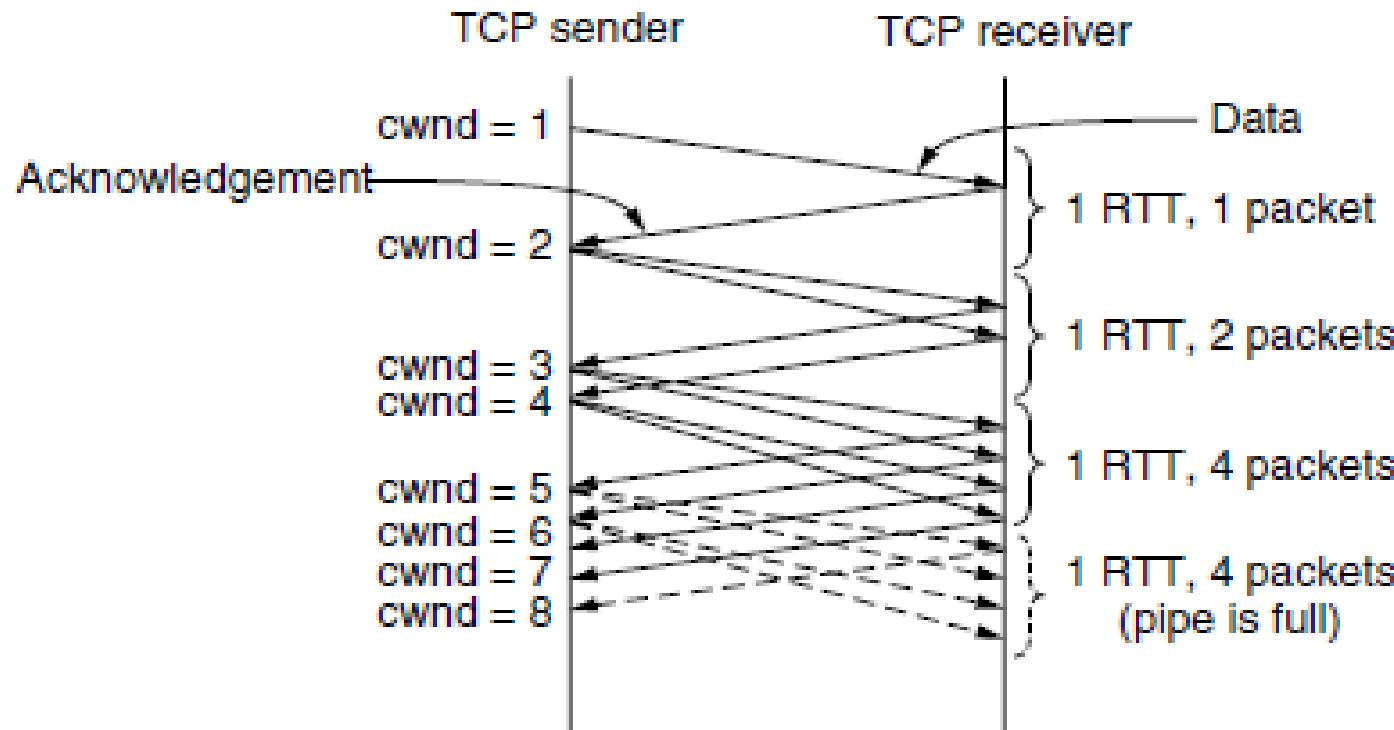


The key observation is this: the acknowledgements return to the sender at about the rate that packets can be sent over the slowest link in the path. This is precisely the rate that the sender wants to use. If it injects new packets into the network at this rate, they will be sent as fast as the slow link permits, but they will not queue up and congest any router along the path. This timing is known as an ack clock. It is an essential part of TCP. By using an ack clock, TCP smoothes out traffic and avoids unnecessary queues at routers. This is first consideration

A second consideration is that the AIMD rule will take a very long time to reach a good operating point on fast networks if the congestion window is started from a small size

Instead, the solution Jacobson chose to handle both of these considerations is a mix of linear and multiplicative increase.

## **SLOW-START**



# TCP Congestion Control

## Slow Start

- Additive Increase / Multiplicative Decrease is only suitable for source, that is operating close to the available capacity of the network, but it takes too long to ramp up a connection when it is starting from scratch.
- slow start, that is used to increase the congestion window rapidly from a cold start.
- Slow start effectively **increases the congestion window exponentially**, rather than linearly.
  - the source starts out by setting CongestionWindow to one packet.
  - When the ACK for this packet arrives, TCP adds 1 to CongestionWindow and then sends two packets.
  - Upon receiving the corresponding two ACKs, TCP increments CongestionWindow by 2—one for each ACK—and next sends four packets.
  - The end result is that TCP effectively doubles the number of packets it has in transit every RTT.

Whenever a packet loss is detected, for example, by a timeout, the slow start threshold is set to be half of the congestion window and the entire process is restarted.

Congestion avoidance phase is started if cwnd has reached the slow start threshold value

Whenever the slow start threshold is crossed, TCP switches from slow start to additive increase. In this mode, the congestion window is increased by one segment every round-trip time.

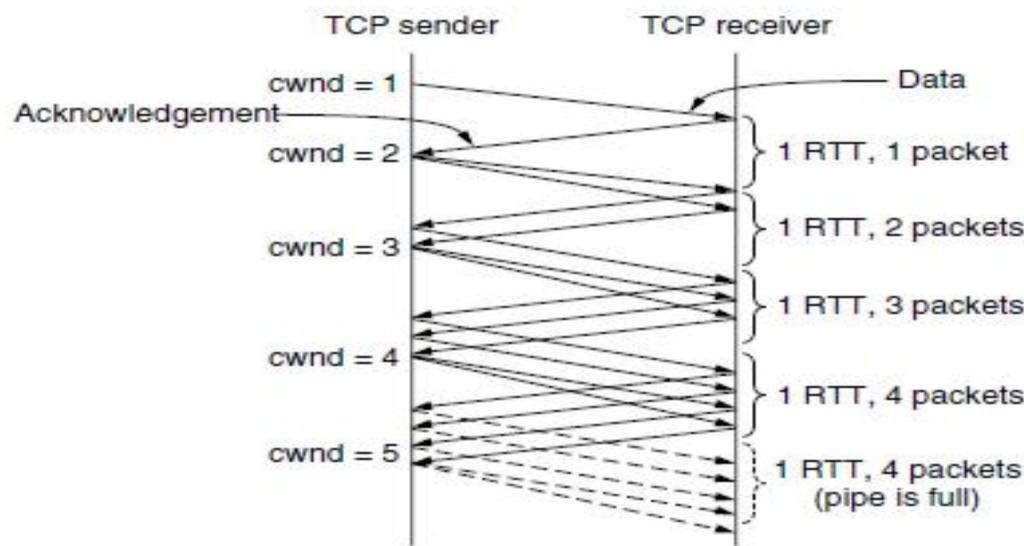


Figure 6-45. Additive increase from an initial congestion window of one segment.

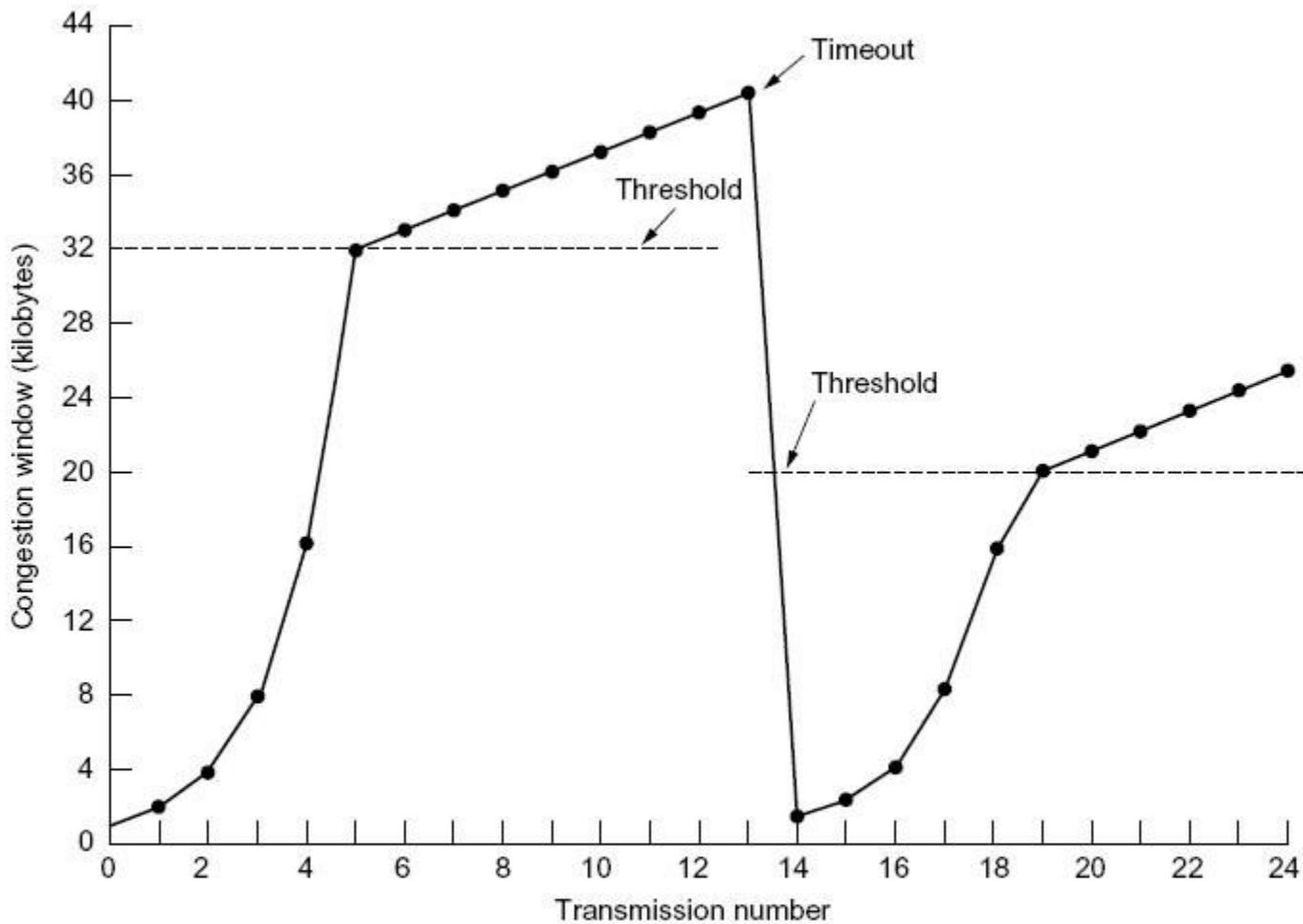


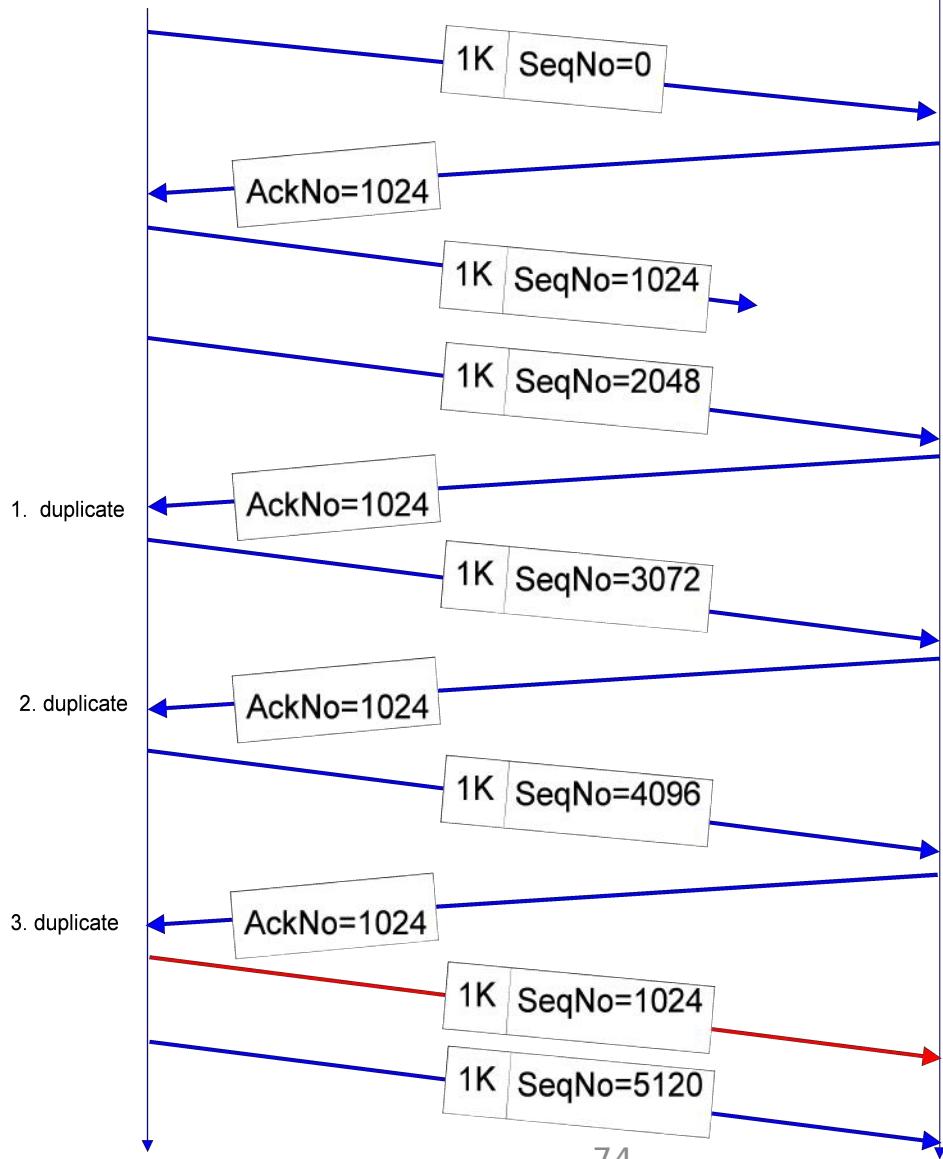
Fig. 6-37. An example of the Internet congestion algorithm.

# Responses to Congestion

- So, TCP assumes there is congestion if it detects a packet loss
- A TCP sender can detect lost packets via:
  - Timeout of a retransmission timer
  - Receipt of a duplicate ACK
- TCP interprets a Timeout as a binary congestion signal. When a timeout occurs, the sender performs:
  - cwnd is reset to one:  
 $cwnd = 1$
  - ssthresh is set to half the current size of the congestion window:  
 $ssthresh = cwnd / 2$
  - and slow-start is entered

# Fast Retransmit

- If three or more duplicate ACKs are received in a row, the TCP sender believes that a segment has been lost.
- Then TCP performs a retransmission of what seems to be the missing segment, without waiting for a timeout to happen.
- Enter slow start:  
 $\text{ssthresh} = \text{cwnd}/2$   
 $\text{cwnd} = 1$



# Flavors of TCP Congestion Control

- **TCP Tahoe** (1988)
  - Slow Start
  - Congestion Avoidance
  - Fast Retransmit
- **TCP Reno** (1990) (TCP Tahoe+FR)
  - Fast Recovery
- **New Reno** (1996)
- **SACK** (1996) (**SACK (Selective ACKnowledgements)**)
- **RED** (Floyd and Jacobson 1993)

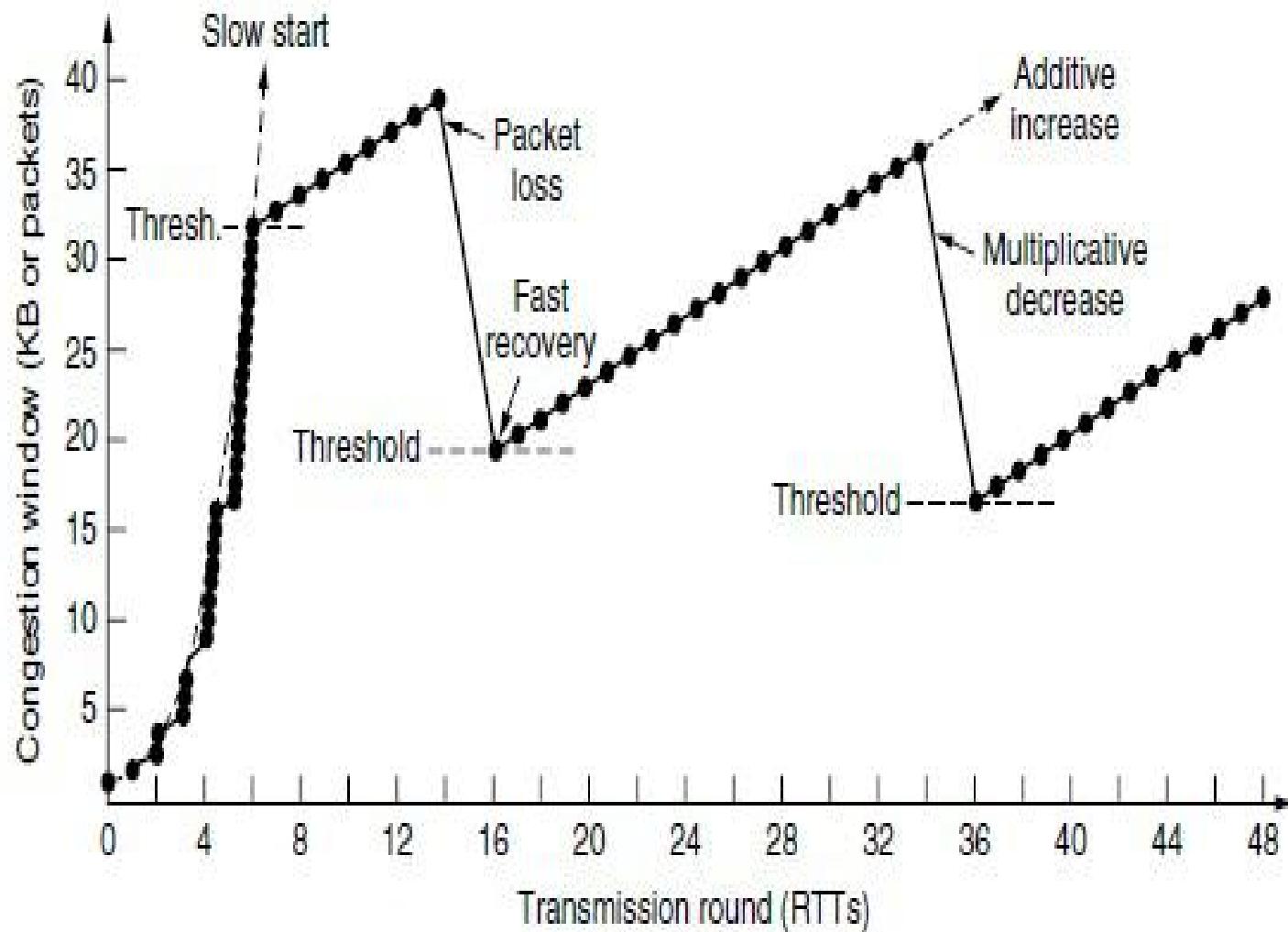


Figure 6-47. Fast recovery and the sawtooth pattern of TCP Reno.

The use of ECN (Explicit Congestion Notification) in addition to packet loss as a congestion signal. ECN is an IP layer mechanism to notify hosts of congestion.

The sender tells the receiver that it has heard the signal by using the CWR (*Congestion Window Reduced*) flag.

# USER DATAGRAM PROTOCOL (UDP)

*The User Datagram Protocol (UDP) is called a connectionless, unreliable transport protocol. It does not add anything to the services of IP except to provide process-to-process communication instead of host-to-host communication.*

## Topics discussed in this section:

Well-Known Ports for UDP

User Datagram

Checksum

UDP Operation

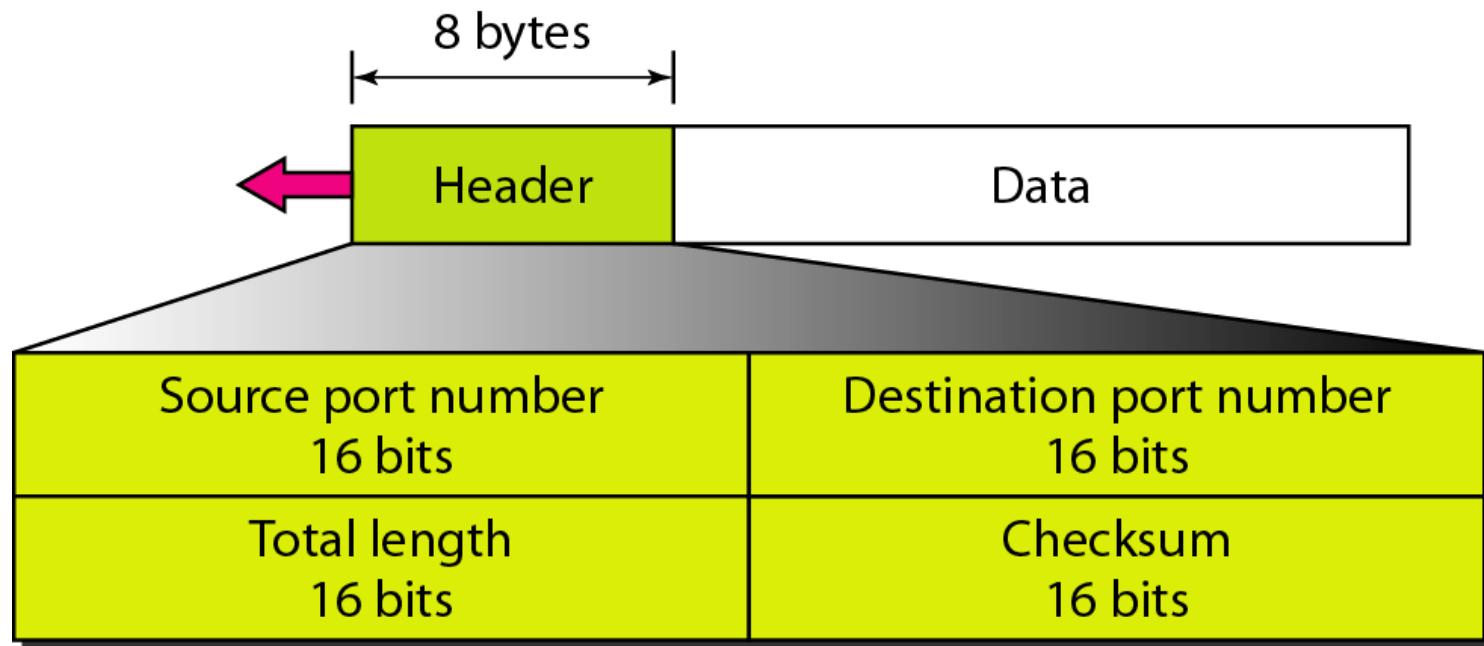
Use of UDP

**Table 23.1** *Well-known ports used with UDP*

<i>Port</i>	<i>Protocol</i>	<i>Description</i>
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	Nameserver	Domain Name Service
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)

Figure 23.9 *User datagram format*

---



## **Checksum** (OPTIONAL, IF NOT USED SET ALL 1'S DEFAULT)

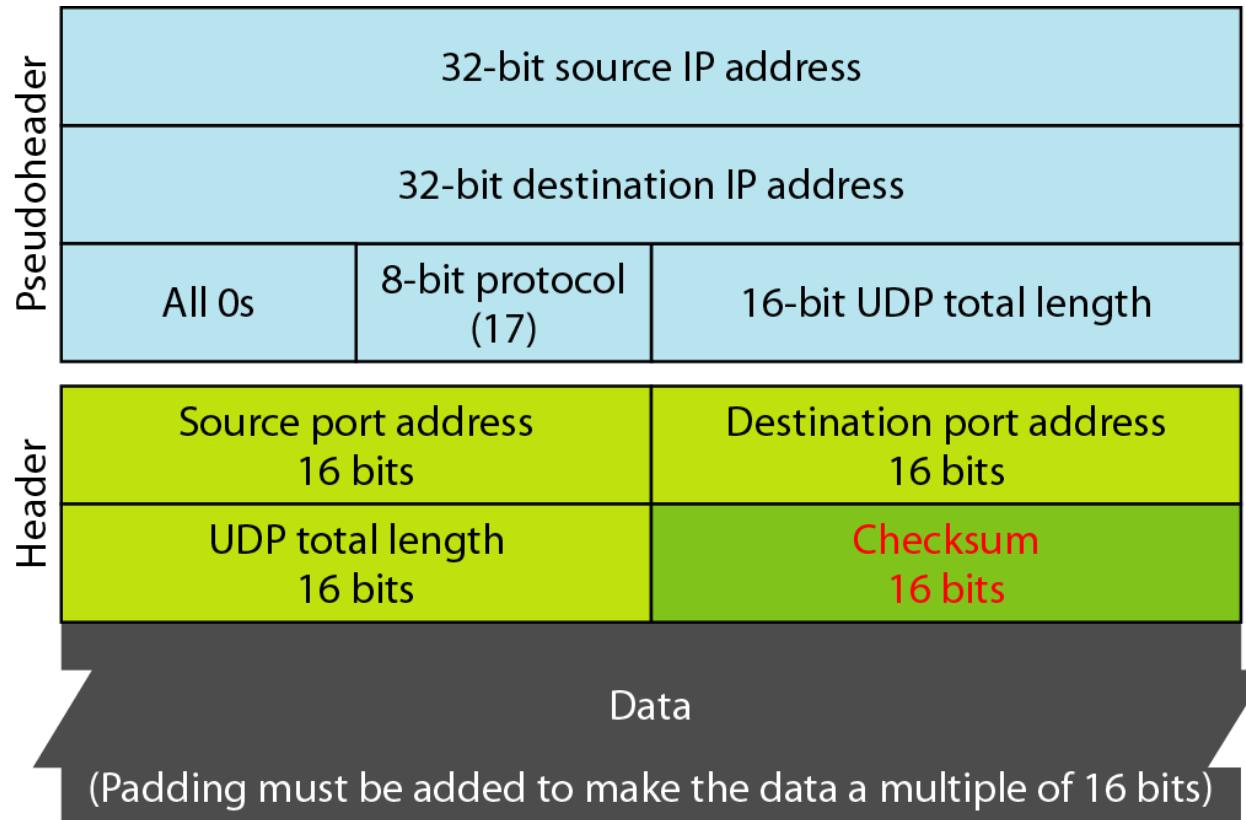
The UDP checksum calculation is different from the one for IP and ICMP. Here the checksum includes three sections: **a pseudo header, the UDP header, and the data** coming from the application layer.

The pseudo header is the part of the header of the IP packet in which the user datagram is to be encapsulated with some fields filled with Os

If the checksum does not include the pseudo header, a user datagram may arrive safe and sound. However, if the IP header is corrupted, it may be delivered to the wrong host.

The protocol field is added to ensure that the packet belongs to UDP, and not to other transport-layer protocols.

**Figure 23.10 Pseudoheader for checksum calculation**



# **UDP Operation**

## **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 datagrams are not numbered. Also, there is no connection establishment and no connection termination, as is the case for TCP. This means that each user datagram can travel on a different path.

## **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. There is no error control mechanism in UDP except for the checksum. This means that the sender does not know if a message has been lost or duplicated. When the receiver detects an error through the checksum, the user datagram is silently discarded. The lack of flow control and error control

## **Encapsulation and Decapsulation**

To send a message from one process to another, the UDP protocol encapsulates and decapsulates messages in an IP datagram.

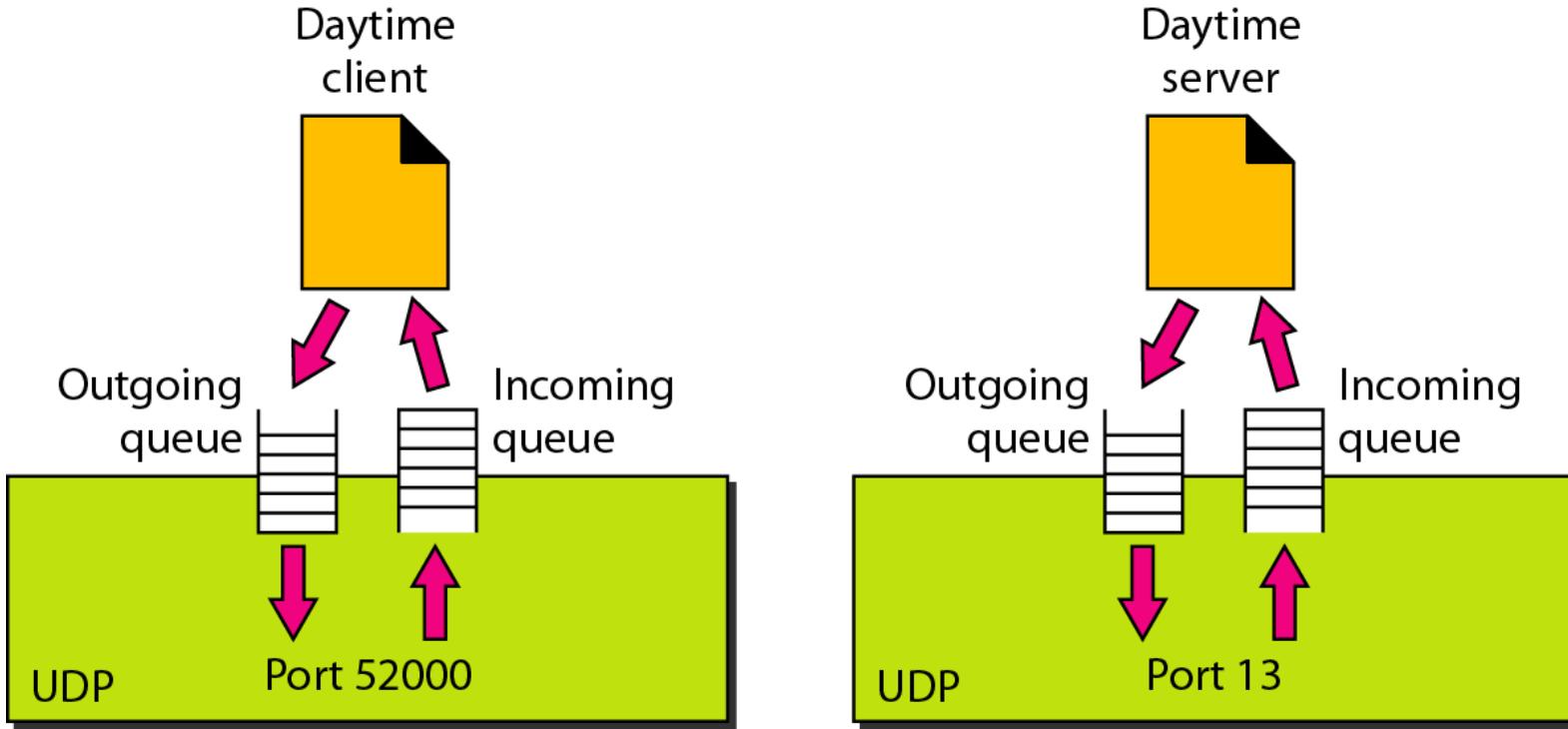
*Figure 23.11 shows the checksum calculation for a very small user datagram with only 7 bytes of data. Because the number of bytes of data is odd, padding is added for checksum calculation. The pseudoheader as well as the padding will be dropped when the user datagram is delivered to IP.*

**Figure 23.11** *Checksum calculation of a simple UDP user datagram*

153.18.8.105			
171.2.14.10			
All 0s	17	15	
1087		13	
15		All 0s	
T	E	S	T
I	N	G	All 0s

10011001 00010010	→	153.18
00001000 01101001	→	8.105
10101011 00000010	→	171.2
00001110 00001010	→	14.10
00000000 00010001	→	0 and 17
00000000 00001111	→	15
00000100 00111111	→	1087
00000000 00001101	→	13
00000000 00001111	→	15
00000000 00000000	→	0 (checksum)
01010100 01000101	→	T and E
01010011 01010100	→	S and T
01001001 01001110	→	I and N
01000111 00000000	→	G and 0 (padding)
 10010110 11101011		→ Sum
 01101001 00010100		→ Checksum

Figure 23.12 *Queues in UDP*



# Remote Procedure Call

The key work was done by Birrell and Nelson (1984). In a nutshell, what Birrell and Nelson suggested was allowing programs to call procedures located on remote hosts. When a process on machine 1 calls a procedure on machine 2, the calling process on 1 is suspended and execution of the called procedure takes place on 2. Information can be transported from the caller to the callee in the parameters and can come back in the procedure result. No message passing is visible to the application programmer. This technique is known as **RPC (Remote Procedure Call)**. Traditionally, the calling procedure is known as the client and the called procedure is known as the server, and we will use those names here too.

to call a remote procedure, the client program must be bound with a small library procedure, called the **client stub**, that represents the server procedure in the client's address space. Similarly, the server is bound with a procedure called the **server stub**. These procedures hide the fact that the procedure call from the client to the server is not local

Step 1 is the client calling the client stub. This call is a local procedure call, with the parameters pushed onto the stack in the normal way.

Step 2 is the client stub packing the parameters into a message and making a system call to send the message. Packing the parameters is called **marshaling**.

Step 3 is the operating system sending the message from the client machine to the server machine.

Step 4 is the operating system passing the incoming packet to the server stub.

Finally, step 5 is the server stub calling the server procedure with the unmarshaled parameters.

The reply traces the same path in the other direction.

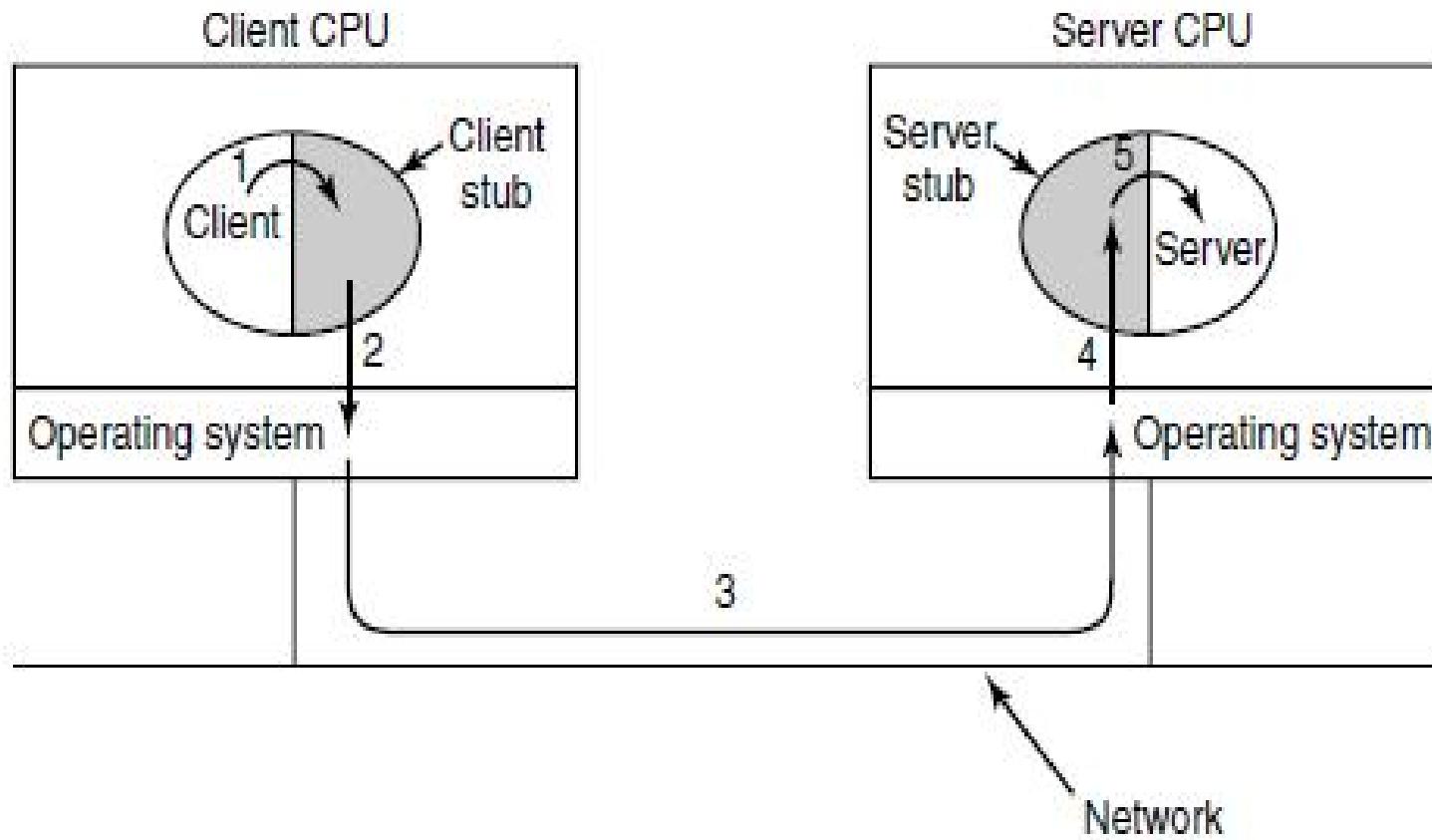


Figure 6-29. Steps in making a remote procedure call. The stubs are shaded.

## Problems with RPC:

- 1 With RPC, passing pointers is impossible because the client and server are in different address spaces.
- 2 It is essentially impossible for the client stub to marshal the parameters: it has no way of determining how large they are.
- 3 A third problem is that it is not always possible to deduce the types of the parameters, not even from a formal specification or the code itself.(exa: printf)
- 4 A fourth problem relates to the use of global variables. Normally, the calling and called procedure can communicate by using global variables, in addition to communicating via parameters. But if the called procedure is moved to a remote machine, the code will fail because the global variables are no longer shared

# TCP

TCP is a connection oriented protocol; it creates a virtual connection between two TCPs to send data. In addition, TCP uses flow and error control mechanisms at the transport level. In brief, TCP is called a *connection-oriented, reliable transport protocol. It adds* connection-oriented and reliability features to the services of IP.

## Topics discussed in this section:

TCP Services

TCP Features

Segment

A TCP Connection

Flow Control

Error Control

# TCP Services

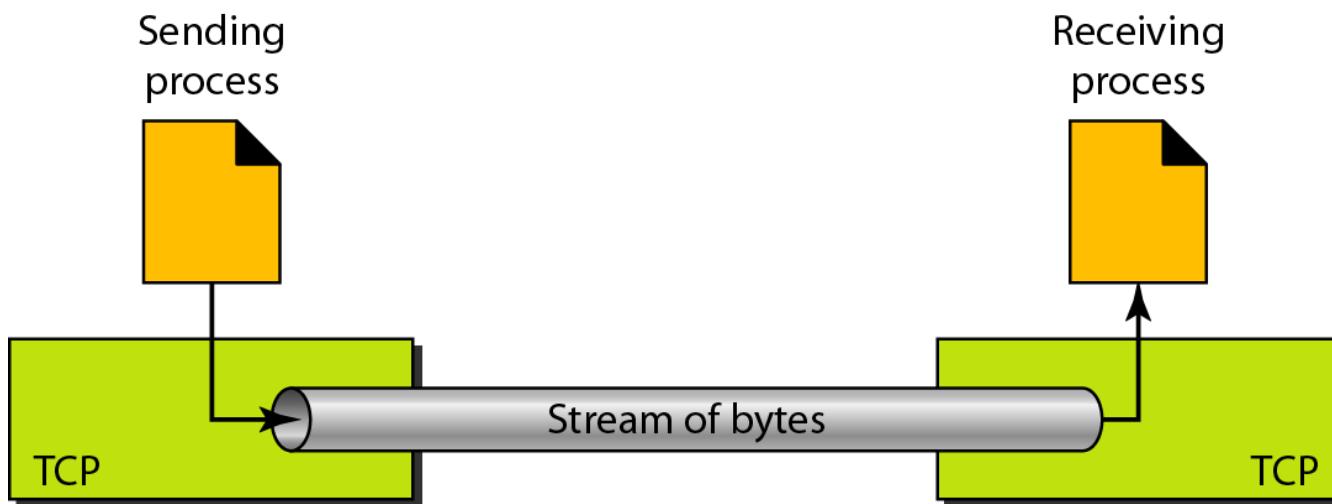
## **1 Process-to-Process Communication**

TCP provides process-to-process communication using port numbers. Below Table lists some well-known port numbers used by TCP.

<i>Port</i>	<i>Protocol</i>	<i>Description</i>
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
20	FTP, Data	File Transfer Protocol (data connection)
21	FTP, Control	File Transfer Protocol (control connection)
23	TELNET	Terminal Network
25	SMTP	Simple Mail Transfer Protocol
53	DNS	Domain Name Server
67	BOOTP	Bootstrap Protocol
79	Finger	Finger
80	HTTP	Hypertext Transfer Protocol
111	RPC	Remote Procedure Call

## **2 Stream Delivery Service**

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



**3 Sending and Receiving Buffers** Because the sending and the receiving processes may not write or read data at the same speed, TCP needs buffers for storage. There are two buffers, the sending buffer and the receiving buffer, one for each direction. One way to implement a buffer is to use a circular array of 1-byte locations as shown in Figure. For simplicity, we have shown two buffers of 20 bytes each. Normally the buffers are hundreds or thousands of bytes, depending on the implementation. We also show the buffers as the same size, which is not always the case.

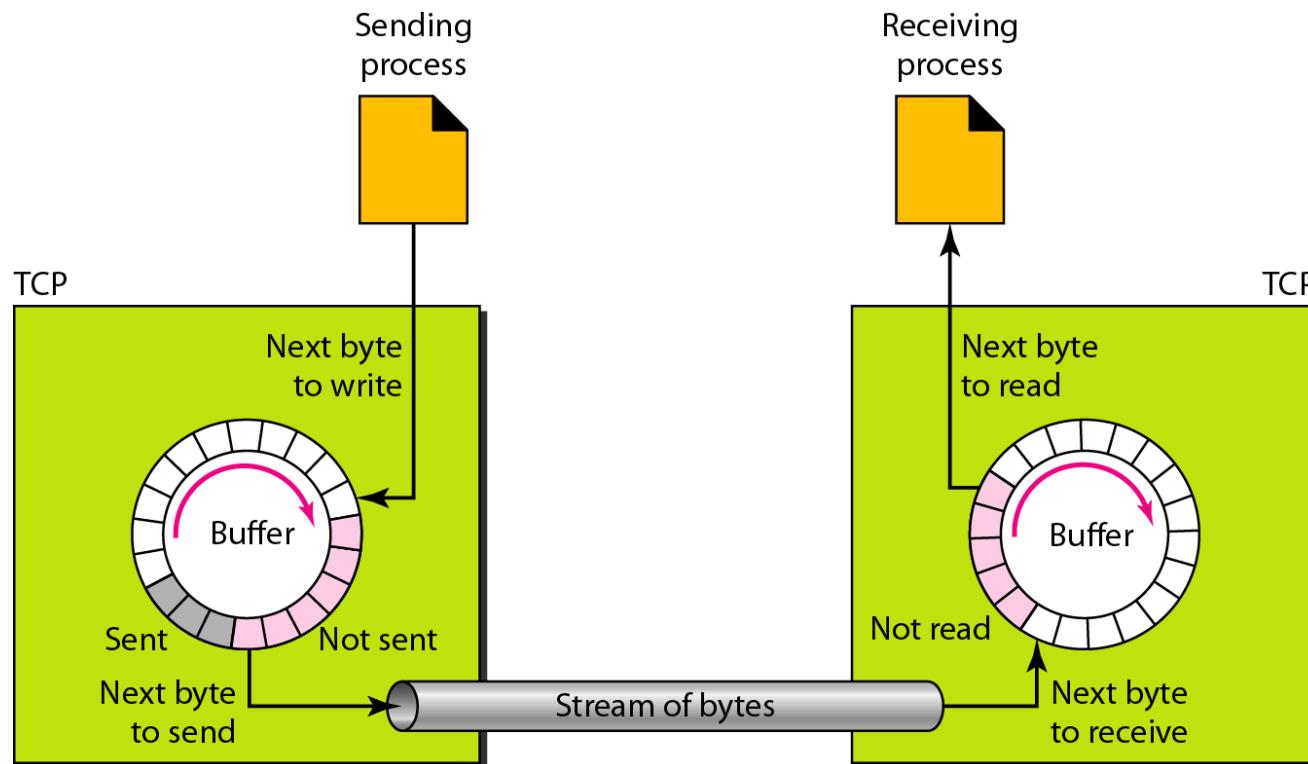


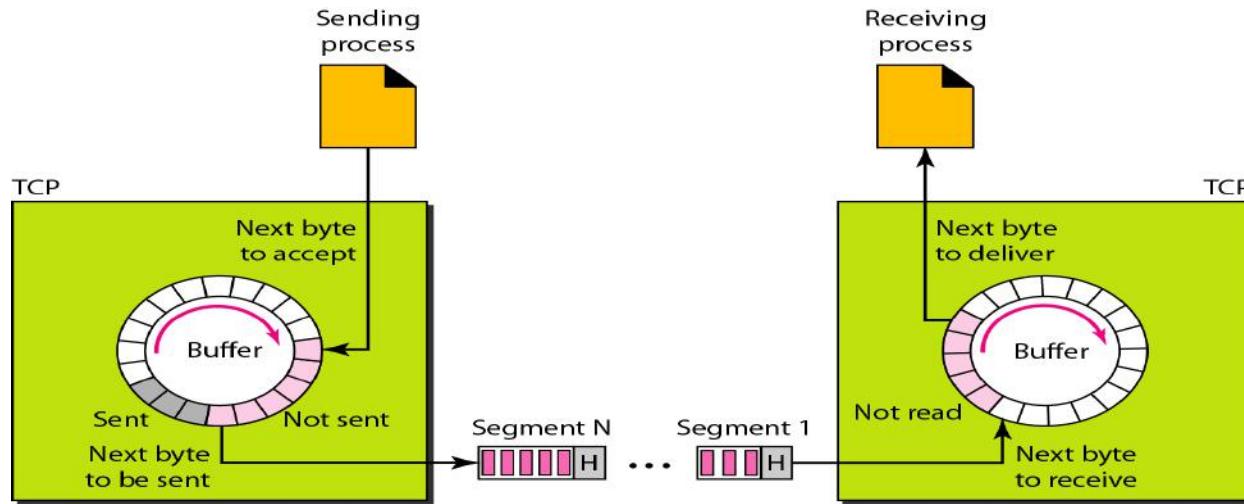
Figure shows the movement of the data in one direction. At the sending site, the buffer has three types of chambers. The white section contains empty chambers that can be filled by the sending process (producer). The gray area holds bytes that have been sent but not yet acknowledged. TCP keeps these bytes in the buffer until it receives an acknowledgment. The colored area contains bytes to be sent by the sending TCP.

However, as we will see later in this chapter, TCP may be able to send only part of this colored section. This could be due to the slowness of the receiving process or perhaps to congestion in the network. Also note that after the bytes in the gray chambers are acknowledged, the chambers are recycled and available for use by the sending process.

This is why we show a circular buffer.

The operation of the buffer at the receiver site is simpler. The circular buffer is divided into two areas (shown as white and colored). The white area contains empty chambers to be filled by bytes received from the network. The colored sections contain received bytes that can be read by the receiving process. When a byte is read by the receiving process, the chamber is recycled and added to the pool of empty chambers.

## 4 TCP segments



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. The segments are encapsulated in IP datagrams and transmitted.

This entire operation is transparent to the receiving process. Later we will see that segments may be received out of order, lost, or corrupted and resent. All these are handled by TCP with the receiving process unaware of any activities. Above fig shows how segments are created from the bytes in the buffers

## **5 Full-Duplex Communication**

TCP offers full-duplex service, in which data can flow in both directions at the same time. Each TCP then has a sending and receiving buffer, and segments move in both directions

## **6 Connection-Oriented Service**

TCP is a connection-oriented protocol. When a process at site A wants to send and receive data from another process at site B, the following occurs:

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

## **7 Reliable Service**

TCP is a reliable transport protocol. It uses an acknowledgment mechanism to check the safe and sound arrival of data. We will discuss this feature further in the section on error control.

# TCP Features

## **1 Numbering System**

There are two fields called the sequence number and the acknowledgment number. These two fields refer to the byte number and not the segment number.

**Byte Number** The bytes of data being transferred in each connection are numbered by TCP. The numbering starts with a randomly generated number. For example, if the random number happens to be 1057 and the total data to be sent are 6000 bytes, the bytes are numbered from 1057 to 7056. We will see that byte numbering is used for flow and error control.

**Sequence Number** After the bytes have been numbered, TCP assigns a sequence number to each segment that is being sent. The sequence number for each segment is the number of the first byte carried in that segment.

**Acknowledgment Number** The value of the acknowledgment field in a segment defines the number of the next byte a party expects to receive. The acknowledgment number is cumulative.

## **2 Flow Control**

TCP, provides *flow control*. *The receiver of the data controls the amount of data that are to be sent by the sender.* This is done to prevent the receiver from being overwhelmed with data. The numbering system allows TCP to use a byte-oriented flow control.

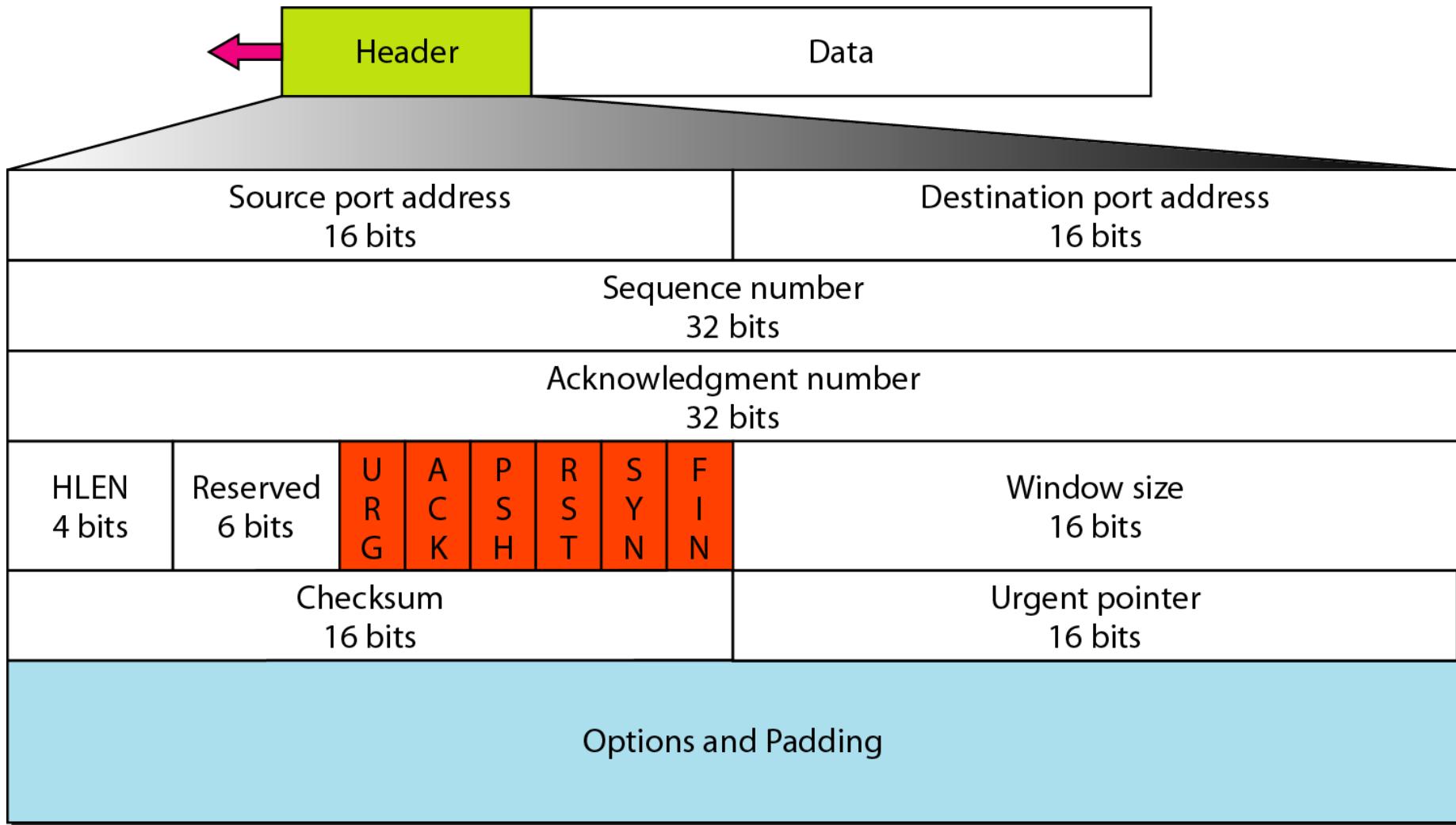
## **3 Error Control**

To provide reliable service, TCP implements an error control mechanism. Although error control considers a segment as the unit of data for error detection (loss or corrupted segments), error control is byte-oriented, as we will see later.

## **4 Congestion Control**

TCP takes into account congestion in the network. The amount of data sent by a sender is not only controlled by the receiver (flow control), but is also determined by the level of congestion in the network

## TCP segment format



The segment consists of a 20- to 60-byte header.,.

**Source port address.** This is a 16-bit field that defines the port number of the application program in the host that is sending the segment.

**Destination port address.** This is a 16-bit field that defines the port number of the application program in the host that is receiving the segment.

**Sequence number.** This 32-bit field defines the number assigned to the first byte of data contained in this segment. As we said before, TCP is a stream transport protocol. To ensure connectivity, each byte to be transmitted is numbered. The sequence number tells the destination which byte in this sequence comprises the first byte in the segment. During connection establishment, each party uses a random number generator to create an initial sequence number (ISN), which is usually different in each direction.

**Acknowledgment number.** This 32-bit field defines the byte number that the receiver of the segment is expecting to receive from the other party. If the receiver of the segment has successfully received byte number  $x$  *from the other party*, it defines  $x + 1$  as the acknowledgment number. *Acknowledgment and data can be piggybacked together.*

**Header length.** This 4-bit field indicates the number of 4-byte words in the TCP header. The length of the header can be between 20 and 60 bytes. Therefore, the value of this field can be between 5 ( $5 \times 4 = 20$ ) and 15 ( $15 \times 4 = 60$ ).

**Reserved.** This is a 6-bit field reserved for future use.

**Control.** This field defines 6 different control bits or flags as shown in Figure. One or more of these bits can be set at a time.

URG: Urgent pointer is valid  
ACK: Acknowledgment is valid  
PSH: Request for push

RST: Reset the connection  
SYN: Synchronize sequence numbers  
FIN: Terminate the connection



These bits enable flow control, connection establishment and termination, connection abortion, and the mode of data transfer in TCP.

**Window size.** This field defines the size of the window, in bytes, that the other party must maintain. Note that the length of this field is 16 bits, which means that the maximum size of the window is 65,535 bytes. This value is normally referred to as the receiving window (rwnd) and is determined by the receiver. The sender must obey the dictation of the receiver in this case.

**Checksum.** This 16-bit field contains the checksum. The calculation of the checksum for TCP follows the same procedure as the one described for UDP. However, the inclusion of the checksum in the UDP datagram is optional, whereas the inclusion of the checksum for TCP is mandatory. The same pseudoheader, serving the same purpose, is added to the segment. For the TCP pseudoheader, the value for the protocol field is 6.

[Urgent pointer](#). This 16-bit field, which is valid only if the urgent flag is set, is used when the segment contains urgent data. It defines the number that must be added to the sequence number to obtain the number of the last urgent byte in the data section of the segment. This will be discussed later in this chapter.

[Options](#). There can be up to 40 bytes of optional information in the TCP header. We will not discuss these options here; please refer to the reference list for more information.

## **A TCP Connection**

TCP is connection-oriented. A connection-oriented transport protocol establishes a virtual path between the source and destination. All the segments belonging to a message are then sent over this virtual path. Using a single virtual pathway for the entire message facilitates the acknowledgment process as well as retransmission of damaged or lost frames.

In TCP, connection-oriented transmission requires three phases:

1. connection establishment,
2. data transfer,
3. connection termination.

# TCP connection establishment(3 way handshaking)

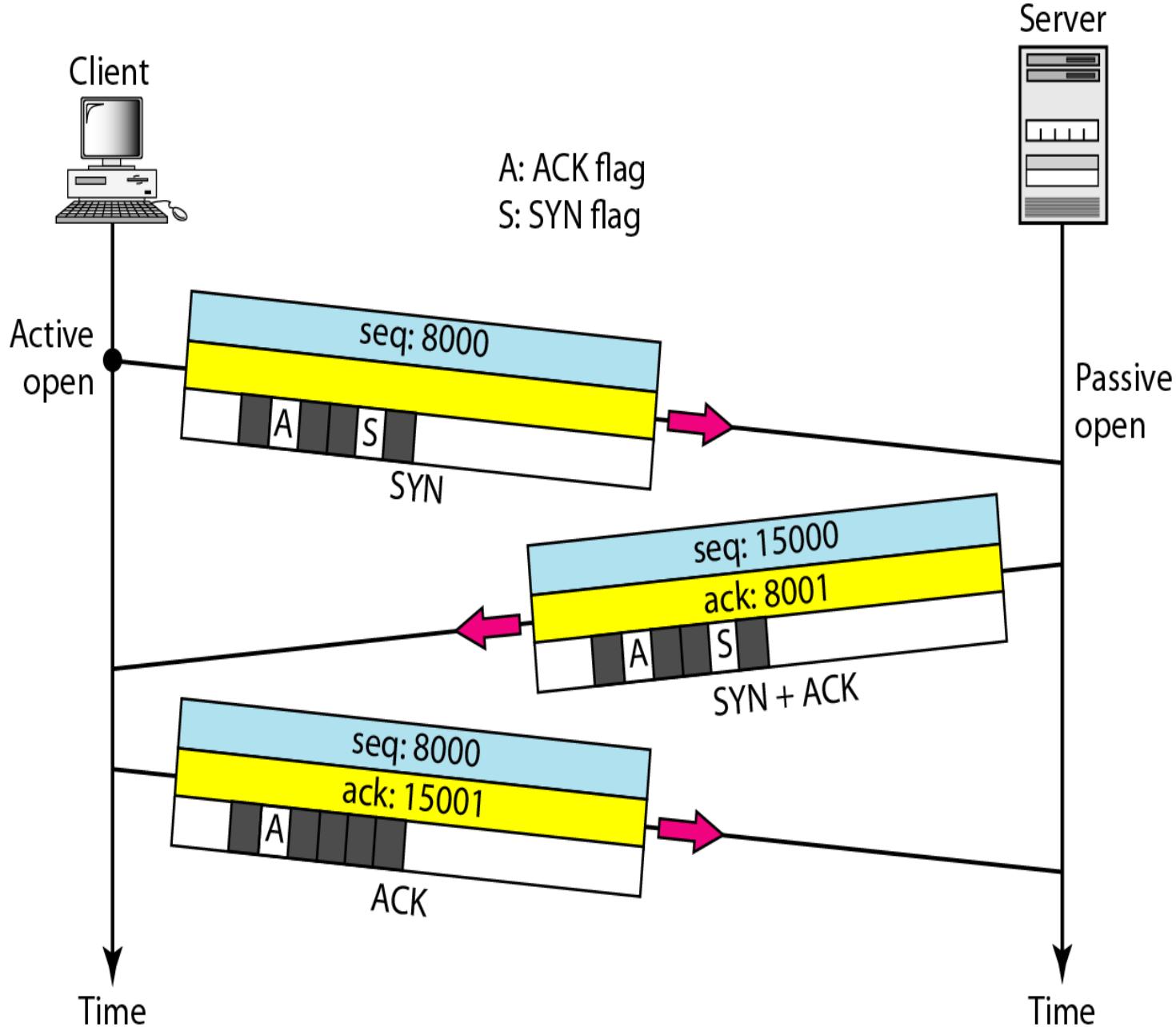
1 The client sends the first segment, a SYN segment, in which only the SYN flag is set.

NOTE:A SYN segment cannot carry data, but it consumes one sequence number.

2. The server sends the second segment, a SYN +ACK segment, with 2 flag bits set: SYN and ACK. This segment has a dual purpose. It is a SYN segment for communication in the other direction and serves as the acknowledgment for the SYN segment. It consumes one sequence number.  
NOTE:A SYN+ACK segment cannot carry data, but does consume one sequence number

3. The client sends the third segment. This is just an ACK segment. It acknowledges the receipt of the second segment with the ACK flag and acknowledgment number field. Note that the sequence number in this segment is the same as the one in the SYN segment; the ACK segment does not consume any sequence numbers.

NOTE: An ACK segment, if carrying no data, consumes no sequence number



## SYN Flooding Attack

This happens when a malicious attacker sends a large number of SYN segments to a server, pretending that each of them is coming from a different client by faking the source IP addresses in the datagram's.

The server, assuming that the clients are issuing an active open, allocates the necessary resources, such as creating communication tables and setting timers. The TCP server then sends the SYN +ACK segments to the fake clients, which are lost. During this time, however, a lot of resources are occupied without being used. If, during this short time, the number of SYN segments is large, the server eventually runs out of resources and may crash. This SYN flooding attack belongs to a type of security attack known as a denial-of-service attack, in which an attacker monopolizes a system with so many service requests that the system collapses and denies service to every request.

### SOLUTIONS:

- 1 Some have imposed a limit on connection requests during a specified period of time.
- 2 Others filter out datagrams coming from unwanted source addresses.
- 3 One recent strategy is to postpone resource allocation until the entire connection is set up using what is called a cookie.

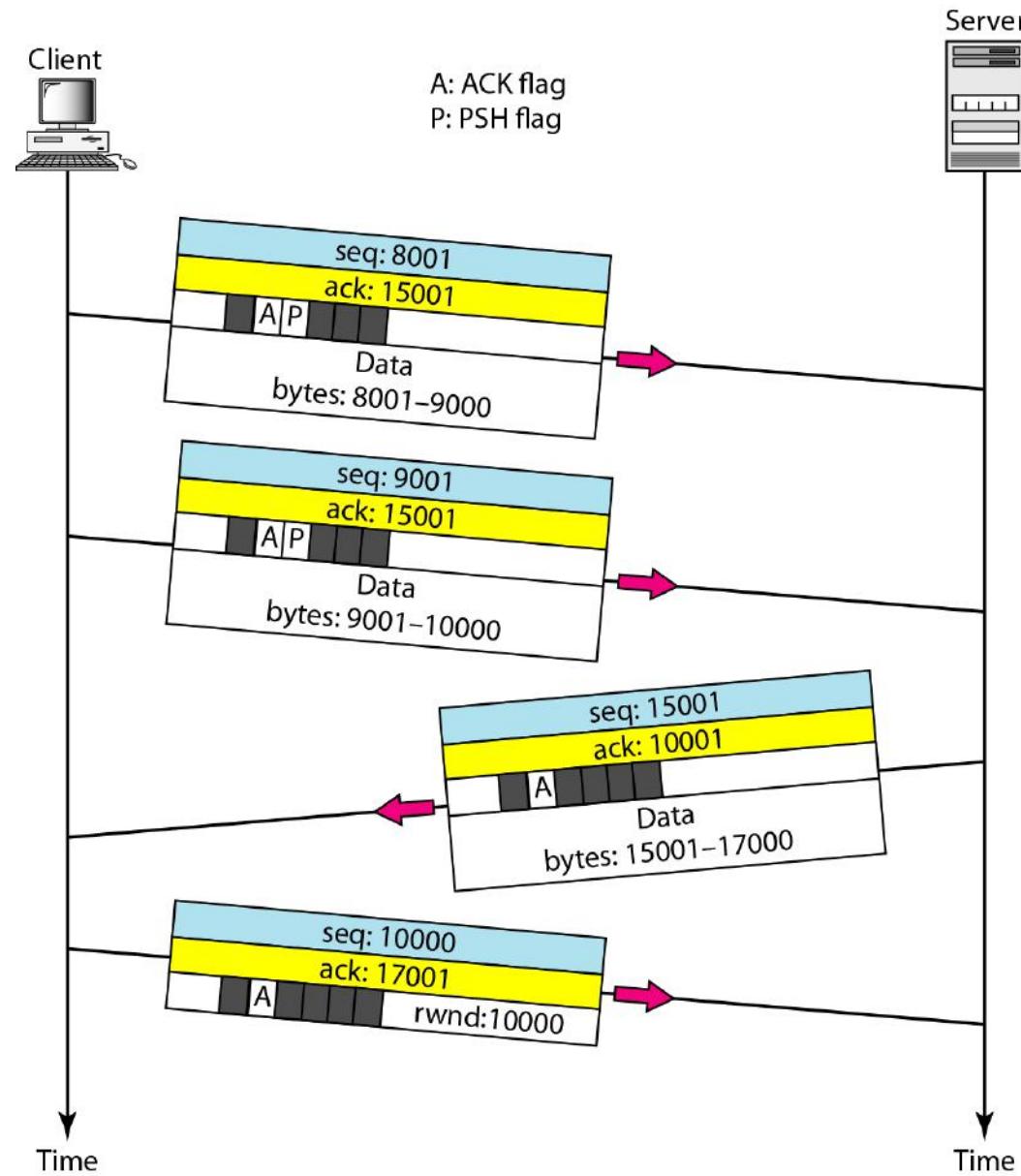
## Data Transfer

After connection is established, bidirectional data transfer can take place. The client and server can both send data and acknowledgments. Data traveling in the same direction as an acknowledgment are carried on the same segment. The acknowledgment is piggybacked with the data

In this example, after connection is established (not shown in the figure), the client sends 2000 bytes of data in two segments. The server then sends 2000 bytes in one segment. The client sends one more segment. The first three segments carry both data and acknowledgment, but the last segment carries only an acknowledgment because there are no more data to be sent.

Note the values of the sequence and acknowledgment numbers. The data segments sent by the client have the PSH (push) flag set so that the server TCP knows to deliver data to the server process as soon as they are received.

## Data transfer



**PUSHING DATA:** Delayed transmission and delayed delivery of data may not be acceptable by the application program.

TCP can handle such a situation. The application program at the sending site can request a *push operation*. *This means that the sending TCP must not wait for the window to be filled.* It must create a segment and send it immediately. The sending TCP must also set the push bit (PSH) to let the receiving TCP know that the segment includes data that must be delivered to the receiving application program as soon as possible and not to wait for more data to come.

**Urgent Data :** TCP is a stream-oriented protocol. This means that the data are presented from the application program to TCP as a stream of bytes. Each byte of data has a position in the stream. However, sending application program wants a piece of data to be read out of order by the receiving application program.

**Connection Termination** (three-way handshaking and four-way handshaking with a half-close option.)

1. In a normal situation, the client TCP, after receiving a close command from the client process, sends the first segment, a FIN segment in which the FIN flag is set.

Note that a FIN segment can include the last chunk of data sent by the client, or it can be just a control segment as shown in Figure. If it is only a control segment, it consumes only one sequence number.

NOTE: The FIN segment consumes one sequence number if it does not carry data.

2 The server TCP, after receiving the FIN segment, informs its process of the situation and sends the second segment, a FIN +ACK segment, to confirm the receipt of the FIN segment from the client and at the same time to announce the closing of the connection in the other direction. This segment can also contain the last chunk of data from the server. If it does not carry data, it consumes only one sequence number.

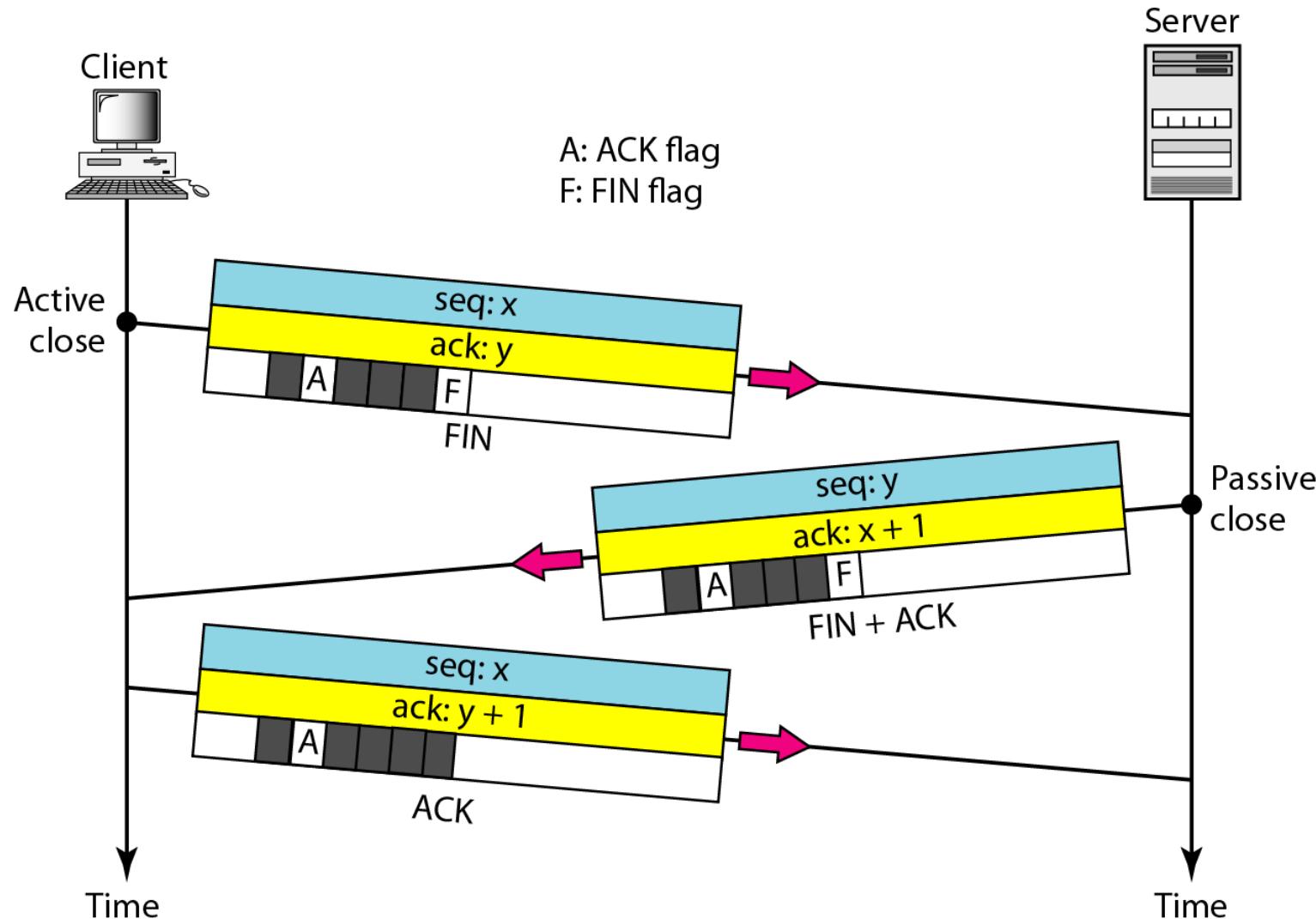
NOTE: The FIN +ACK segment consumes one sequence number if it does not carry data.

3. The client TCP sends the last segment, an ACK segment, to confirm the receipt of the FIN segment from the TCP server. This segment contains the acknowledgment number, which is 1 plus the sequence number received in the FIN segment from the server. This segment cannot carry data and consumes no sequence numbers.

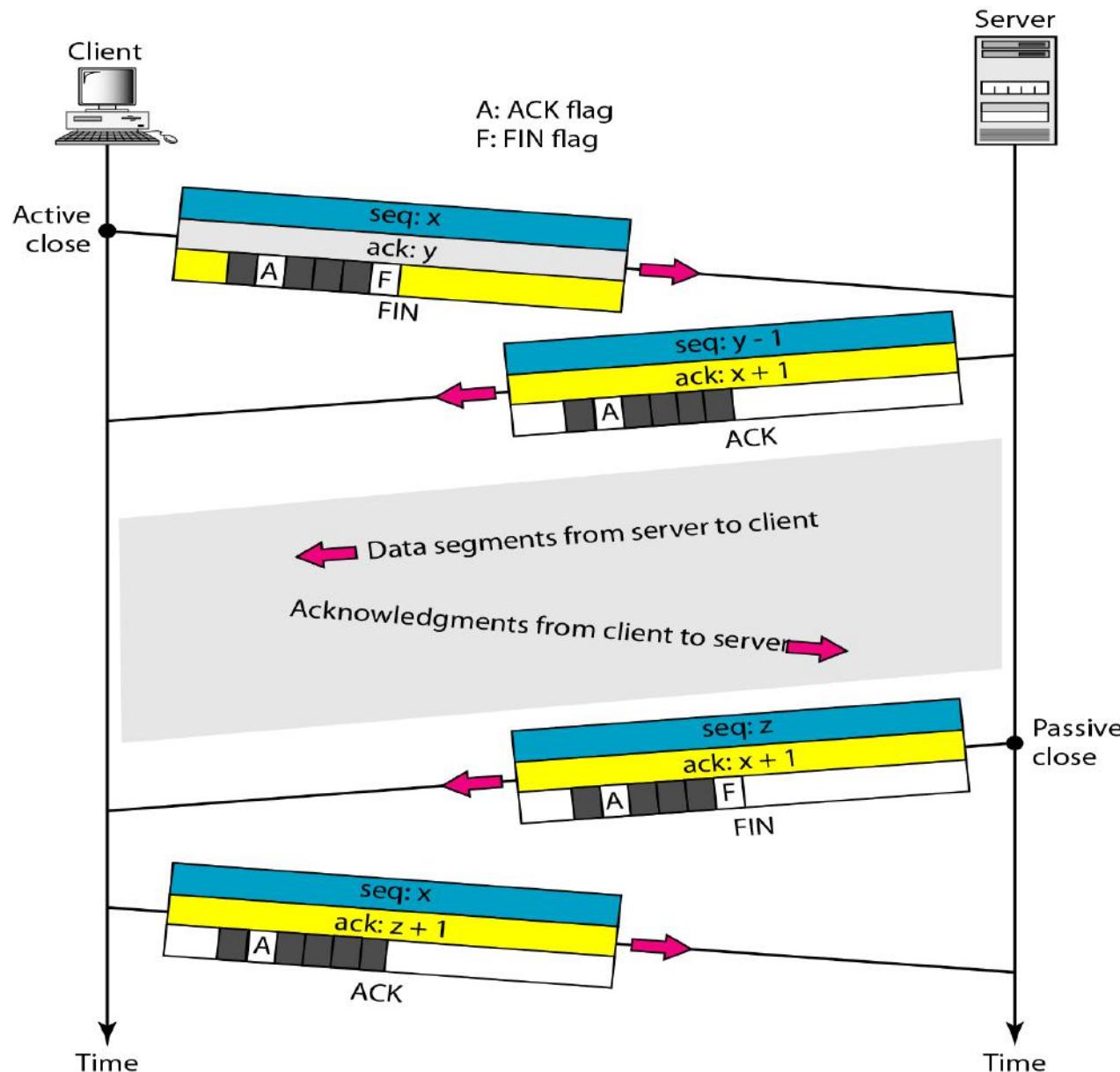
**Half-Close** In TCP, one end can stop sending data while still receiving data. This is called a half-close. Although either end can issue a half-close, it is normally initiated by the client. It can occur when the server needs all the data before processing can begin.

A good example is sorting. When the client sends data to the server to be sorted, the server needs to receive all the data before sorting can start. This means the client, after sending all the data, can close the connection in the outbound direction. However, the inbound direction must remain open to receive the sorted data. The server, after receiving the data, still needs time for sorting; its outbound direction must remain open

## Connection termination using three-way handshaking



## Figure 23.21 Half-close



## Flow Control or TCP Sliding Window

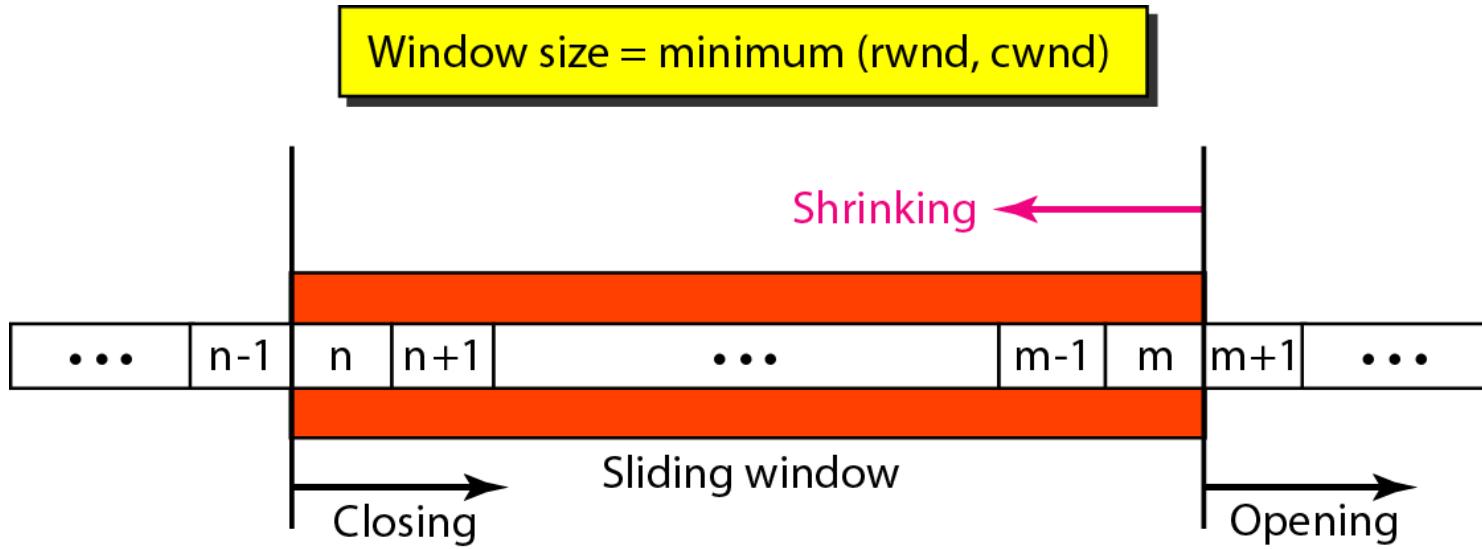
TCP uses a sliding window, to handle flow control. The sliding window protocol used by TCP, however, is something between the *Go-Back-N* and Selective Repeat sliding window.

The sliding window protocol in TCP looks like the *Go-Back-N* protocol because it does not use NAKs;  
it looks like Selective Repeat because the receiver holds the out-of-order segments until the missing ones arrive.

There are two big differences between this sliding window and the one we used at the data link layer.

- 1 the sliding window of TCP is byte-oriented; the one we discussed in the data link layer is frame-oriented.
- 2 the TCP's sliding window is of variable size; the one we discussed in the data link layer was of fixed size

## *Sliding window*



The window is opened, closed, or shrunk. These three activities, as we will see, are in the control of the receiver (and depend on congestion in the network), not the sender.

The sender must obey the commands of the receiver in this matter.

Opening a window means moving the right wall to the right. This allows more new bytes in the buffer that are eligible for sending.

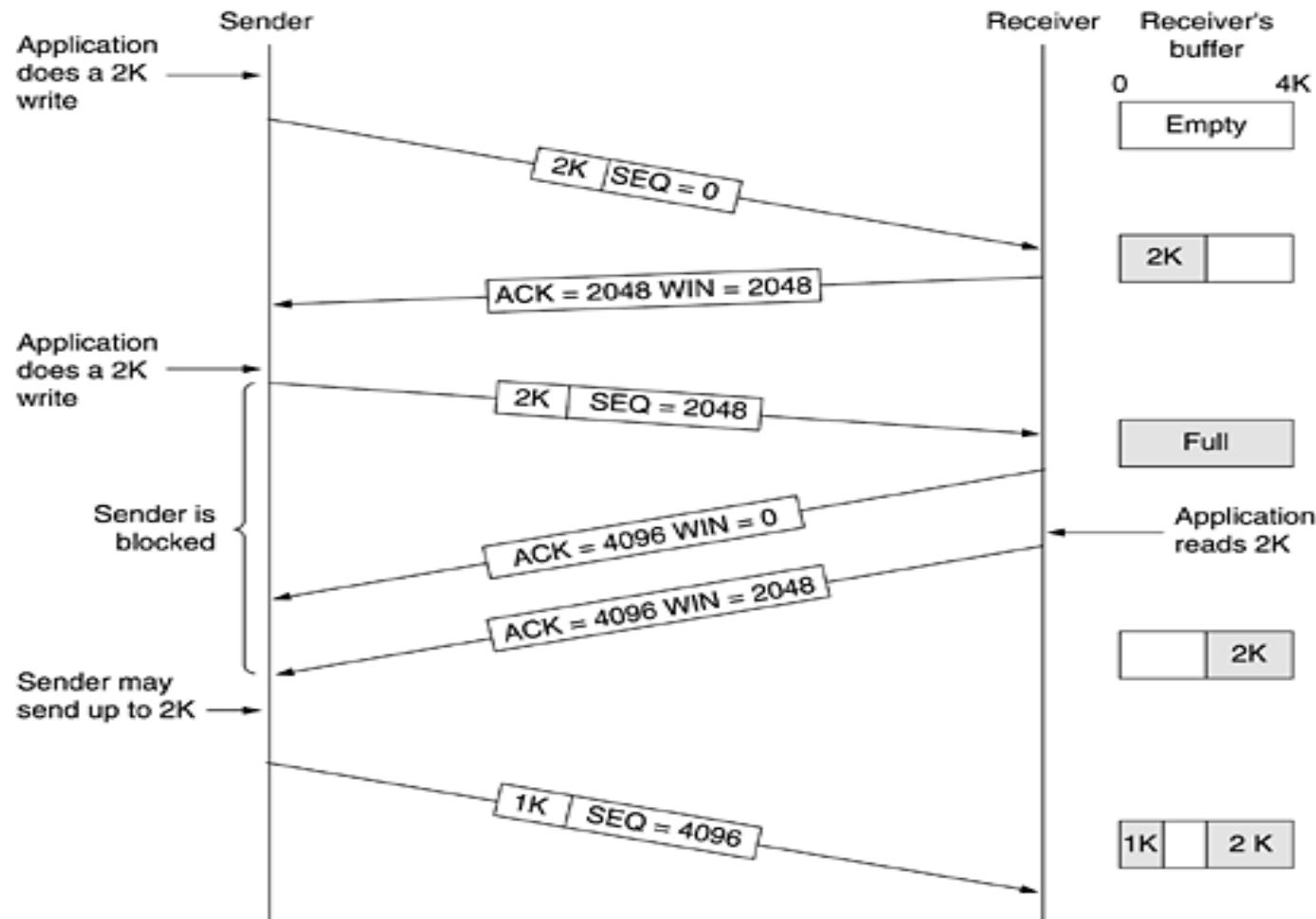
Closing the window means moving the left wall to the right. This means that some bytes have been acknowledged and the sender need not worry about them anymore.

Shrinking the window means moving the right wall to the left.

The size of the window at one end is determined by the lesser of two values: receiver window (rwnd) or congestion window (cwnd).

The receiver window is the value advertised by the opposite end in a segment containing acknowledgment. It is the number of bytes the other end can accept before its buffer overflows and data are discarded.

The congestion window is a value determined by the network to avoid congestion



Window management in TCP

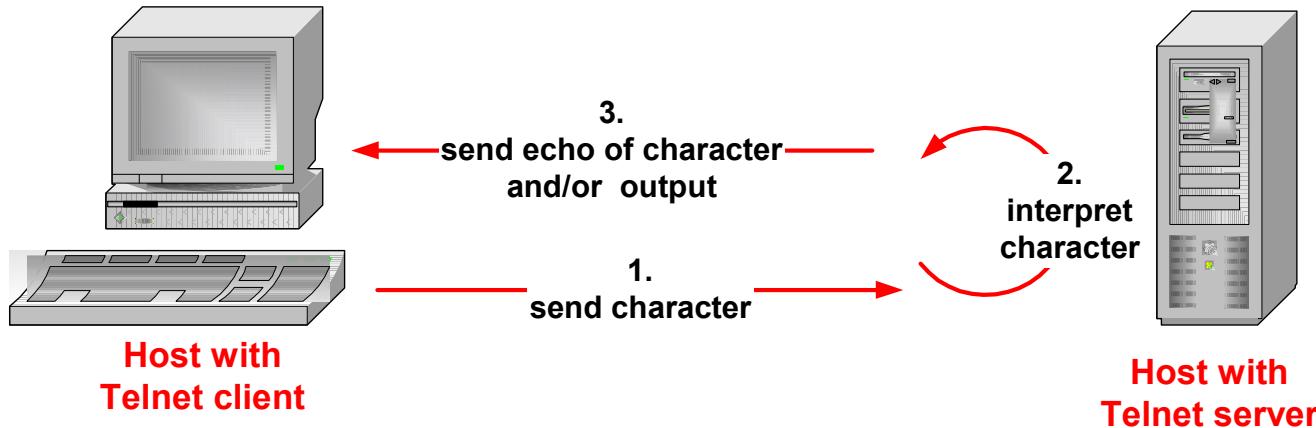
When the window is 0, the sender may not normally send segments, with two exceptions.

- 1) urgent data may be sent, for example, to allow the user to kill the process running on the remote machine.
- 2) the sender may send a 1-byte segment to force the receiver to reannounce the next byte expected and the window size. This packet is called a **window probe**.

The TCP standard explicitly provides this option to prevent deadlock if a window update ever gets lost.

Senders are not required to transmit data as soon as they come in from the application. Neither are receivers required to send acknowledgements as soon as possible.

For example, in Fig. when the first 2 KB of data came in, TCP, knowing that it had a 4-KB window, would have been completely correct in just buffering the data until another 2 KB came in, to be able to transmit a segment with a 4-KB payload. This freedom can be used to improve performance



Remote terminal applications (e.g., Telnet) send characters to a server. The server interprets the character and sends the output at the server to the client.

For each character typed, you see three packets:

**Client ☐ Server:** Send typed character

**Server ☐ Client:** Echo of character (or user output) and acknowledgement for first packet

**Client ☐ Server:** Acknowledgement for second packet

# Delayed Acknowledgement

- TCP delays transmission of ACKs for up to 500ms
- Avoid to send ACK packets that do not carry data.
  - The hope is that, within the delay, the receiver will have data ready to be sent to the receiver. Then, the ACK can be piggybacked with a data segment

## Exceptions:

- ACK should be sent for every full sized segment
- Delayed ACK is not used when packets arrive out of order

Although delayed acknowledgements reduce the load placed on the network by the receiver, a sender that sends multiple short packets (e.g., 41-byte packets containing 1 byte of data) is still operating inefficiently. A way to reduce this usage is known as **Nagle's algorithm (Nagle, 1984)**.

## Nagel's Rule

Send one byte and buffer all subsequent bytes until acknowledgement is received. Then send all buffered bytes in a single TCP segment and start buffering again until the sent segment is acknowledged.

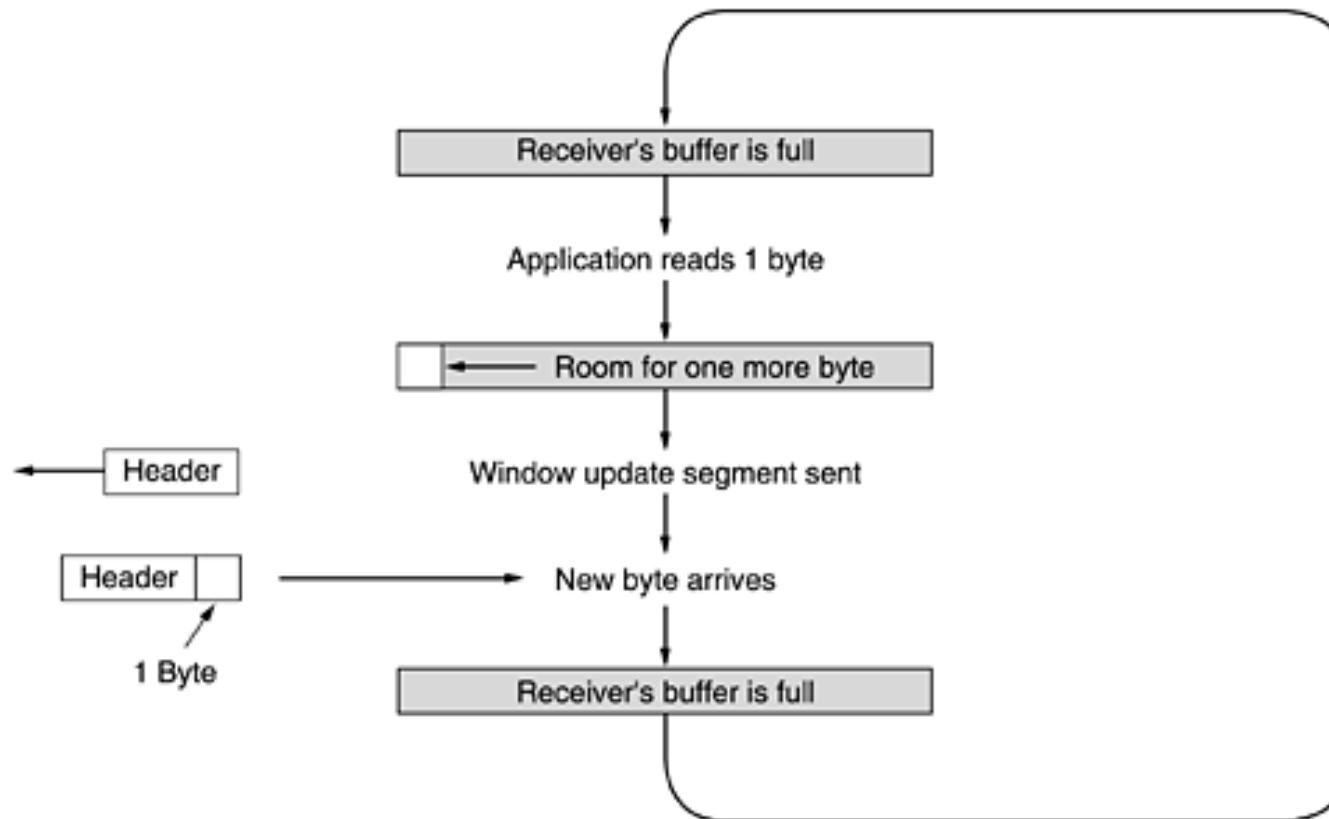
Nagle's algorithm will put the many pieces in one segment, greatly reducing the bandwidth used

Nagle's algorithm is widely used by TCP implementations, but there are times when it is better to disable it. In particular, in interactive games that are run over the Internet.

A more subtle problem is that Nagle's algorithm can sometimes interact with delayed acknowledgements to cause a temporary deadlock: the receiver waits for data on which to piggyback an acknowledgement, and the sender waits on the acknowledgement to send more data.

Because of these problems, Nagle's algorithm can be disabled (which is called the *TCP NODELAY option*).

Another problem that can degrade TCP performance is the **silly window syndrome** (Clark, 1982).



Clark's solution is to prevent the receiver from sending a window update for 1 byte. Instead, it is forced to wait until it has a decent amount of space available and advertise that instead. Specifically, the receiver should not send a window update until it can handle the maximum segment size it advertised when the connection was established or until its buffer is half empty, whichever is smaller.

Furthermore, the sender can also help by not sending tiny segments. Instead, it should wait until it can send a full segment, or at least one containing half of the receiver's buffer size.

The goal is for the sender not to send small segments and the receiver not to ask for them. (Nagel + Clark). Both are used to improve TCP performance

The receiver will buffer the data until it can be passed up to the application in order (handling out of order segments)

## **Cumulative acknowledgements**

## Error Control

TCP is a reliable transport layer protocol. This means that an application program that delivers a stream of data to TCP relies on TCP to deliver the entire stream to the application program on the other end in order, without error, and without any part lost or duplicated.

TCP provides reliability using error control. Error control includes mechanisms for detecting corrupted segments, lost segments, out-of-order segments, and duplicated segments. Error control also includes a mechanism for correcting errors after they are detected. Error detection and correction in TCP is achieved through the use of three simple tools: **checksum, acknowledgment, and time-out.**

### Checksum

Each segment includes a checksum field which is used to check for a corrupted segment. If the segment is corrupted, it is discarded by the destination TCP and is considered as lost. TCP uses a 16-bit checksum that is mandatory in every segment

**Figure 23.11** *Checksum calculation of a simple UDP user datagram*

153.18.8.105			
171.2.14.10			
All 0s	17	15	
1087		13	
15		All 0s	
T	E	S	T
I	N	G	All 0s

10011001 00010010	→	153.18
00001000 01101001	→	8.105
10101011 00000010	→	171.2
00001110 00001010	→	14.10
00000000 00010001	→	0 and 17
00000000 00001111	→	15
00000100 00111111	→	1087
00000000 00001101	→	13
00000000 00001111	→	15
00000000 00000000	→	0 (checksum)
01010100 01000101	→	T and E
01010011 01010100	→	S and T
01001001 01001110	→	I and N
01000111 00000000	→	G and 0 (padding)
 10010110 11101011		→ Sum
 01101001 00010100		→ Checksum

## **Acknowledgment**

TCP uses acknowledgments to confirm the receipt of data segments. Control segments that carry no data but consume a sequence number are also acknowledged. ACK segments are never acknowledged. ACK segments do not consume sequence numbers and are not acknowledged.

## **Retransmission**

The heart of the error control mechanism is the retransmission of segments. When a segment is corrupted, lost, or delayed, it is retransmitted.

In modern implementations, a retransmission occurs if the retransmission timer expires or three duplicate ACK segments have arrived.

Retransmission After RTO (retransmission time out)

Retransmission After Three Duplicate ACK Segments (also called fast retransmission)

## **Out-of-Order Segments**

Data may arrive out of order and be temporarily stored by the receiving TCP, but yet guarantees that no out-of-order segment is delivered to the process

# TCP Congestion Control

When the load offered to any network is more than it can handle, congestion builds up.

The network layer detects congestion when queues grow large at routers and tries to manage it, if only by dropping packets. It is up to the transport layer to receive congestion feedback from the network layer and slow down the rate of traffic that it is sending into the network.

For Congestion control, transport protocol uses an AIMD (Additive Increase Multiplicative Decrease) control law.

TCP congestion control is based on implementing this approach using a window called **congestion window**. TCP adjusts the size of the window according to the AIMD rule.

The window size at the sender is set as follows:

**Send Window = MIN (flow control window, congestion window)**

where

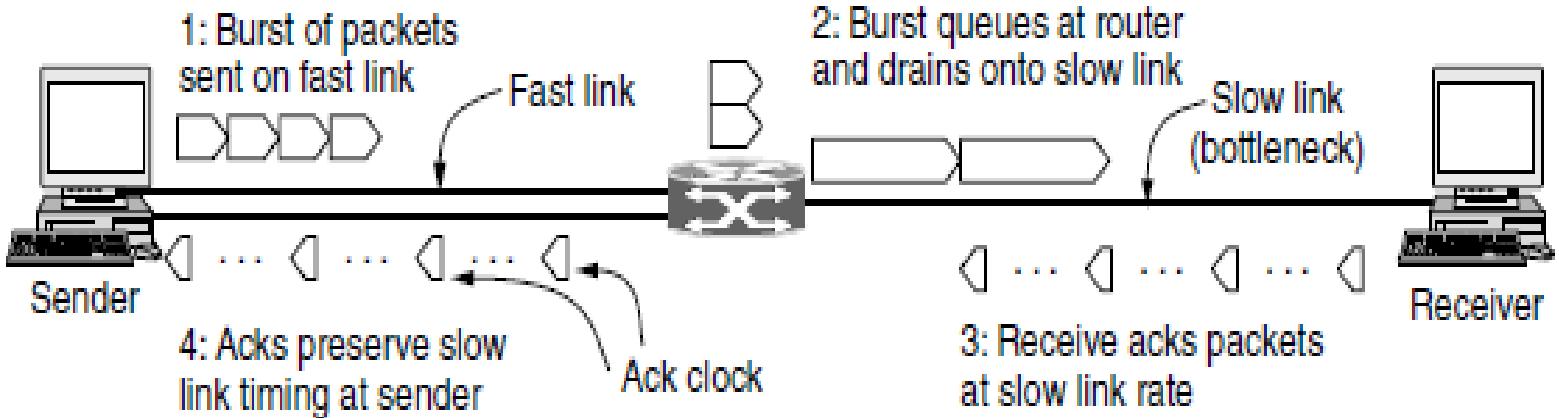
**flow control window** is advertised by the receiver (rwnd)

**congestion window** is adjusted based on feedback from the

Modern congestion control was added to TCP largely through the efforts of Van Jacobson (1988). It is a fascinating story. Starting in 1986, the growing popularity of the early Internet led to the first occurrence of what became known as a **congestion collapse**, a prolonged period during which good put dropped suddenly (i.e., by more than a factor of 100) due to congestion in the network. Jacobson (and many others) set out to understand what was happening and remedy the situation.

To start, he observed that packet loss is a suitable signal of congestion. This signal comes a little late (as the network is already congested) but it is quite dependable

At the beginning how sender knows at what speed receiver can receive the packets?

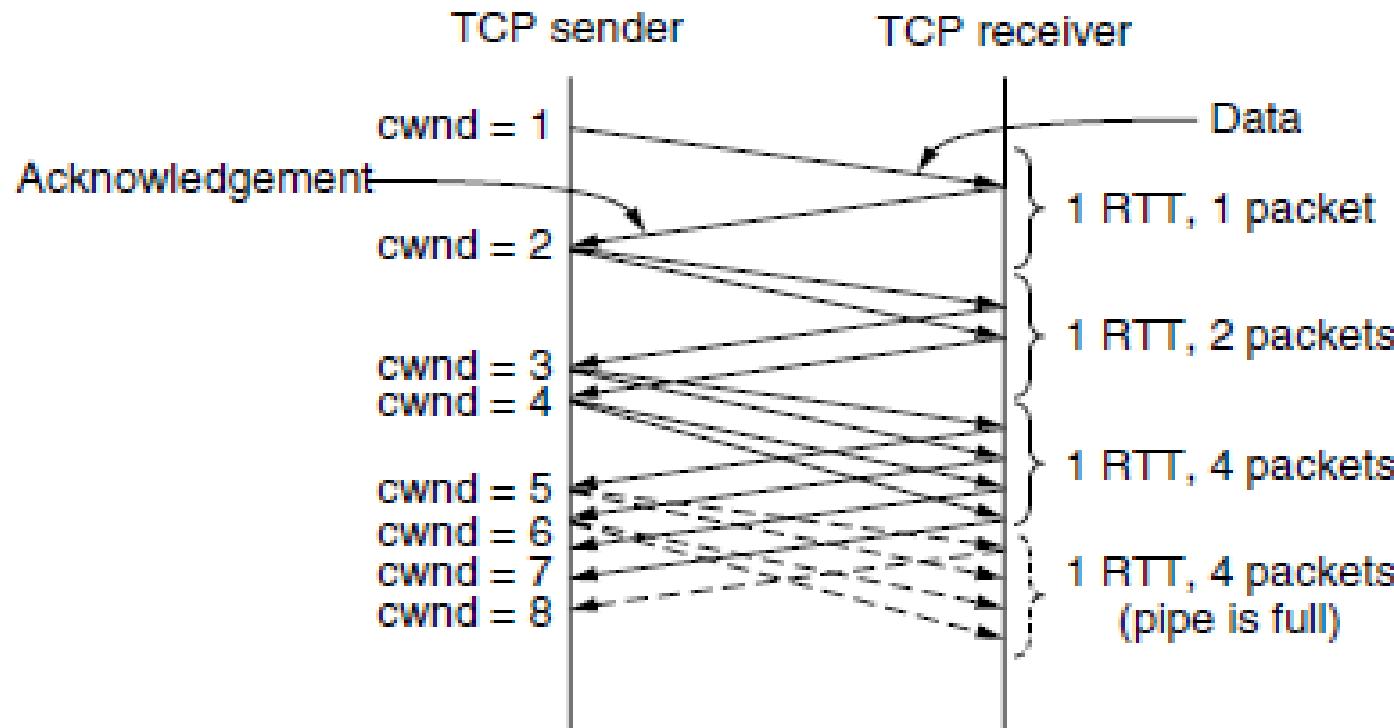


The key observation is this: the acknowledgements return to the sender at about the rate that packets can be sent over the slowest link in the path. This is precisely the rate that the sender wants to use. If it injects new packets into the network at this rate, they will be sent as fast as the slow link permits, but they will not queue up and congest any router along the path. This timing is known as an **ack clock**. It is an essential part of TCP. By using an ack clock, TCP smoothes out traffic and avoids unnecessary queues at routers. This is first consideration

A second consideration is that the AIMD rule will take a very long time to reach a good operating point on fast networks if the congestion window is started from a small size

Instead, the solution Jacobson chose to handle both of these considerations is a mix of linear and multiplicative increase.

## **SLOW-START**



# TCP Congestion Control

## Slow Start

- Additive Increase / Multiplicative Decrease is only suitable for source, that is operating close to the available capacity of the network, but it takes too long to ramp up a connection when it is starting from scratch.
- slow start, that is used to increase the congestion window rapidly from a cold start.
- Slow start effectively **increases the congestion window exponentially**, rather than linearly.
  - the source starts out by setting CongestionWindow to one packet.
  - When the ACK for this packet arrives, TCP adds 1 to CongestionWindow and then sends two packets.
  - Upon receiving the corresponding two ACKs, TCP increments CongestionWindow by 2—one for each ACK—and next sends four packets.
  - The end result is that TCP effectively doubles the number of packets it has in transit every RTT.

Whenever a packet loss is detected, for example, by a timeout, the slow start threshold is set to be half of the congestion window and the entire process is restarted.

Congestion avoidance phase is started if cwnd has reached the slow start threshold value

Whenever the slow start threshold is crossed, TCP switches from slow start to additive increase. In this mode, the congestion window is increased by one segment every round-trip time.

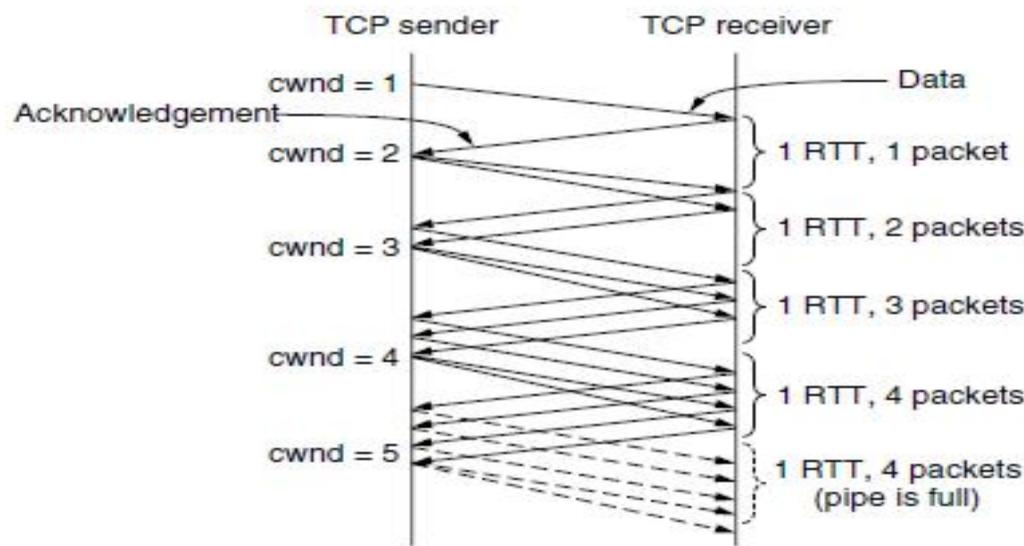


Figure 6-45. Additive increase from an initial congestion window of one segment.

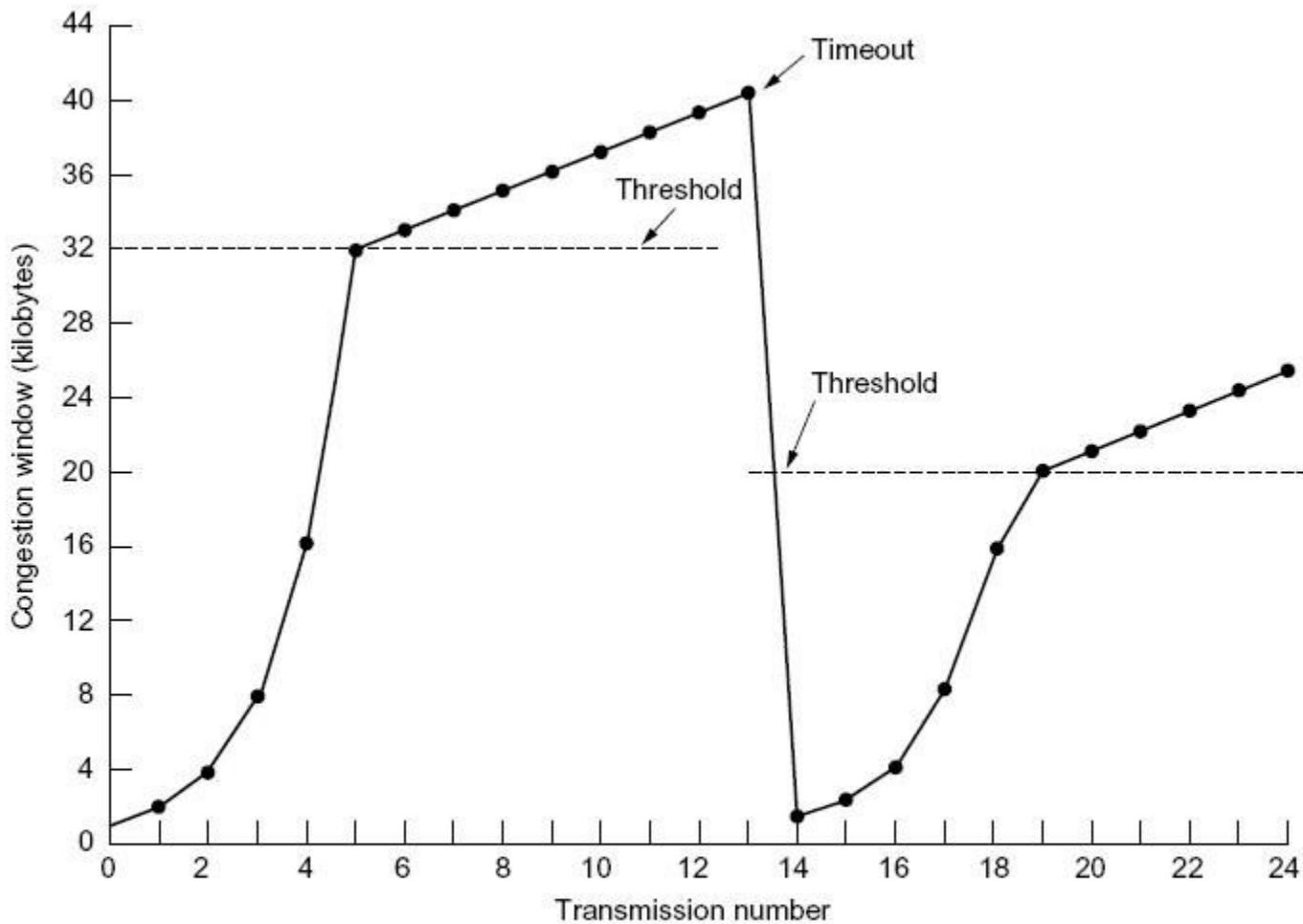


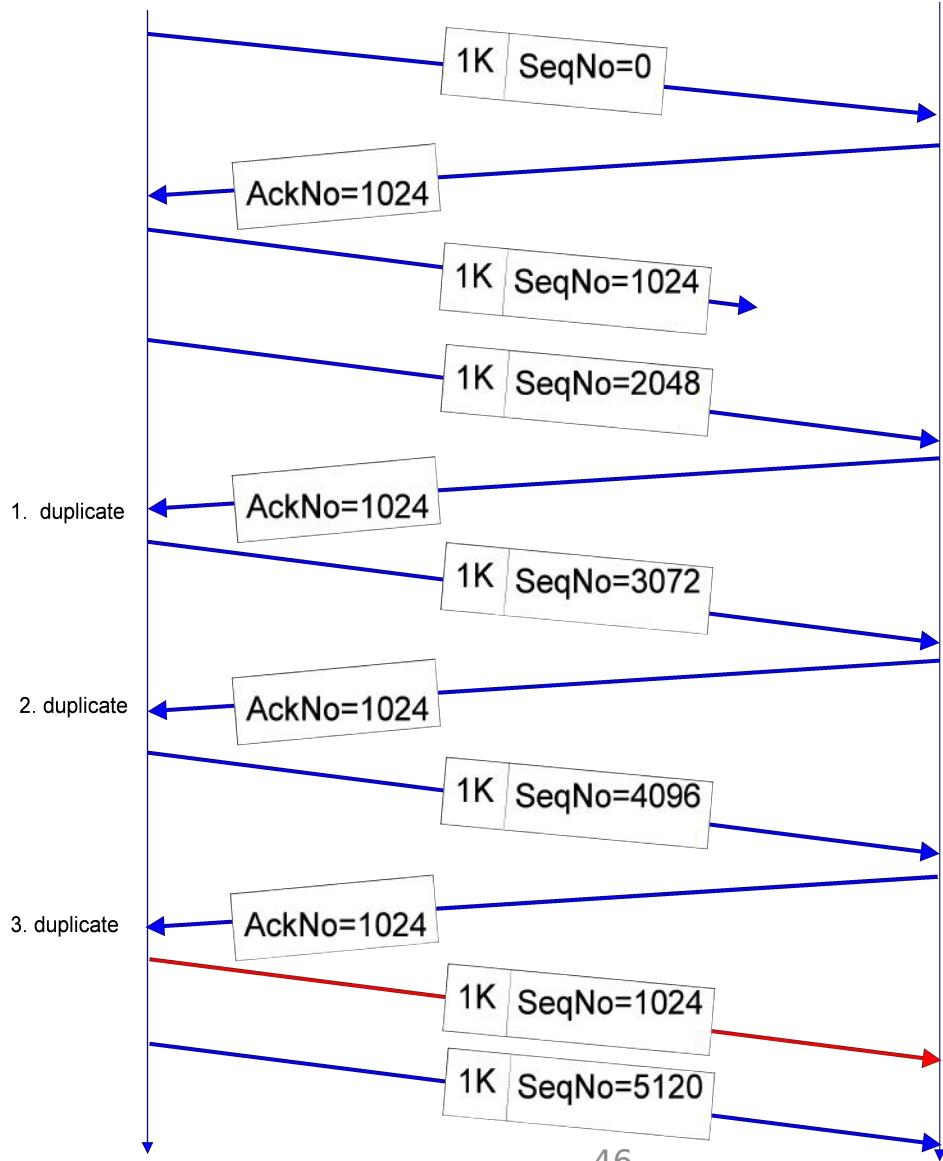
Fig. 6-37. An example of the Internet congestion algorithm.

# Responses to Congestion

- So, TCP assumes there is congestion if it detects a packet loss
- A TCP sender can detect lost packets via:
  - Timeout of a retransmission timer
  - Receipt of a duplicate ACK
- TCP interprets a Timeout as a binary congestion signal. When a timeout occurs, the sender performs:
  - cwnd is reset to one:  
 $cwnd = 1$
  - ssthresh is set to half the current size of the congestion window:  
 $ssthresh = cwnd / 2$
  - and slow-start is entered

# Fast Retransmit

- If three or more duplicate ACKs are received in a row, the TCP sender believes that a segment has been lost.
- Then TCP performs a retransmission of what seems to be the missing segment, without waiting for a timeout to happen.
- Enter slow start:  
 $\text{ssthresh} = \text{cwnd}/2$   
 $\text{cwnd} = 1$



# Flavors of TCP Congestion Control

- **TCP Tahoe** (1988)
  - Slow Start
  - Congestion Avoidance
  - Fast Retransmit
- **TCP Reno** (1990) (TCP Tahoe+FR)
  - Fast Recovery
- **New Reno** (1996)
- **SACK** (1996) (**SACK (Selective ACKnowledgements)**)
- **RED** (Floyd and Jacobson 1993)

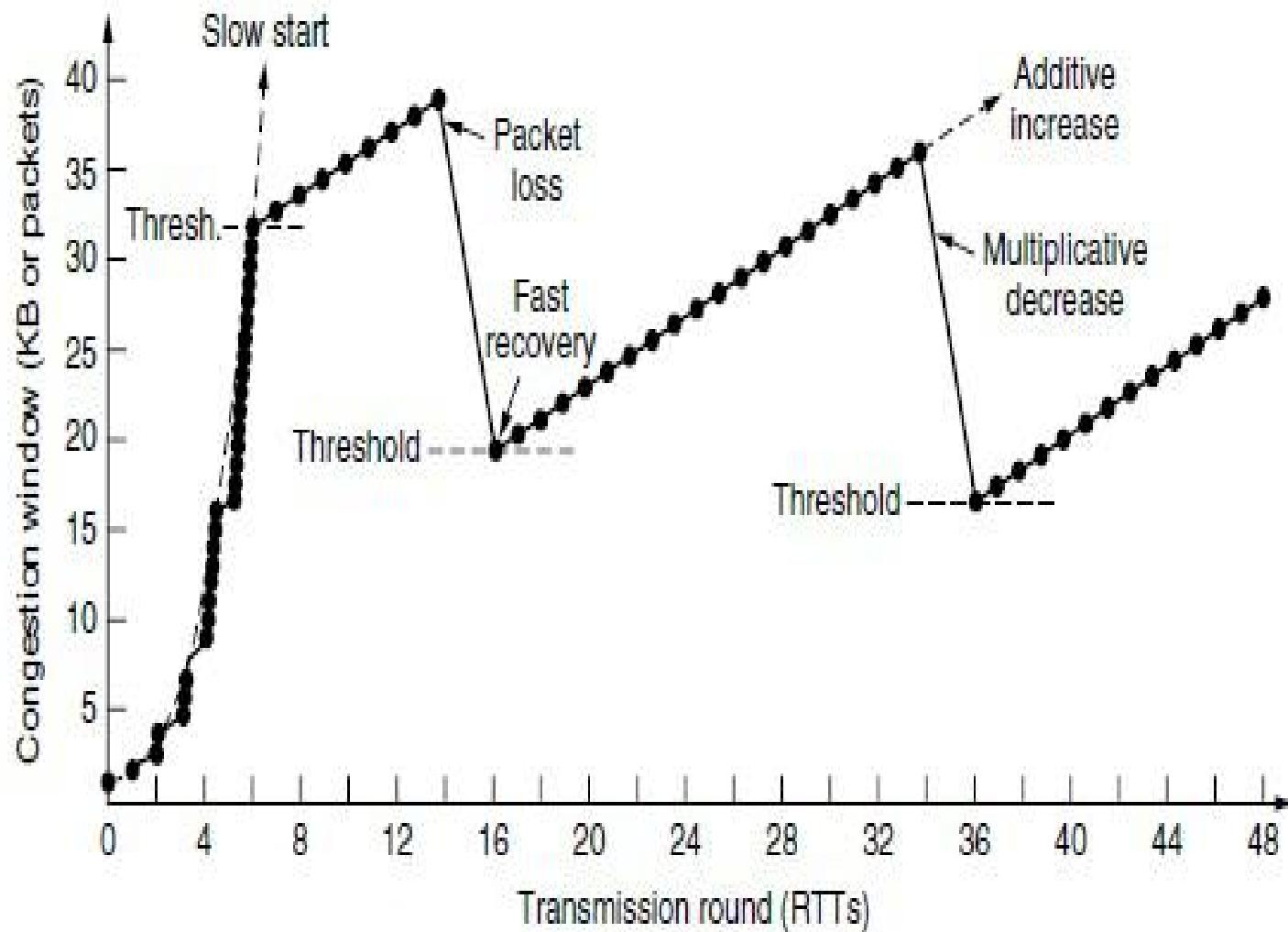


Figure 6-47. Fast recovery and the sawtooth pattern of TCP Reno.

The use of ECN (Explicit Congestion Notification) in addition to packet loss as a congestion signal. ECN is an IP layer mechanism to notify hosts of congestion.

The sender tells the receiver that it has heard the signal by using the CWR (*Congestion Window Reduced*) flag.

# USER DATAGRAM PROTOCOL (UDP)

*The User Datagram Protocol (UDP) is called a connectionless, unreliable transport protocol. It does not add anything to the services of IP except to provide process-to-process communication instead of host-to-host communication.*

## Topics discussed in this section:

Well-Known Ports for UDP

User Datagram

Checksum

UDP Operation

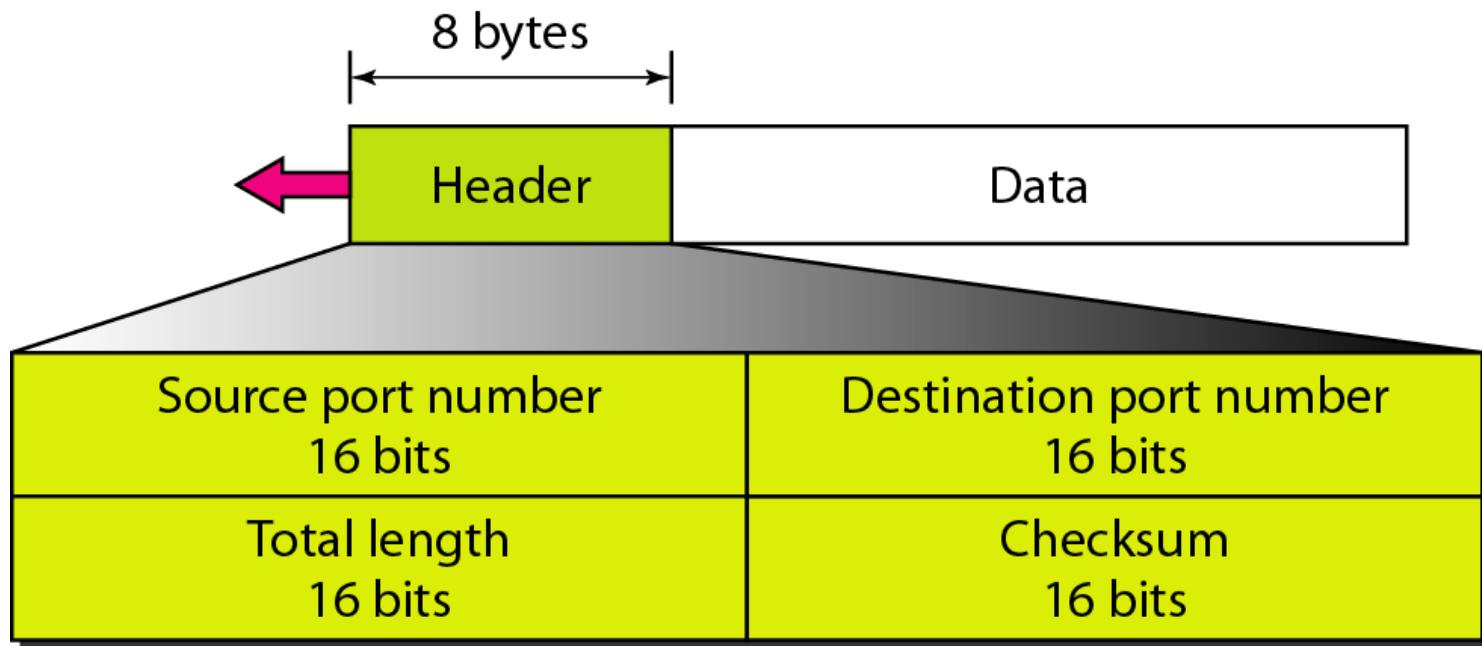
Use of UDP

**Table 23.1** *Well-known ports used with UDP*

<i>Port</i>	<i>Protocol</i>	<i>Description</i>
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	Nameserver	Domain Name Service
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)

Figure 23.9 *User datagram format*

---



## **Checksum** (OPTIONAL, IF NOT USED SET ALL 1'S DEFAULT)

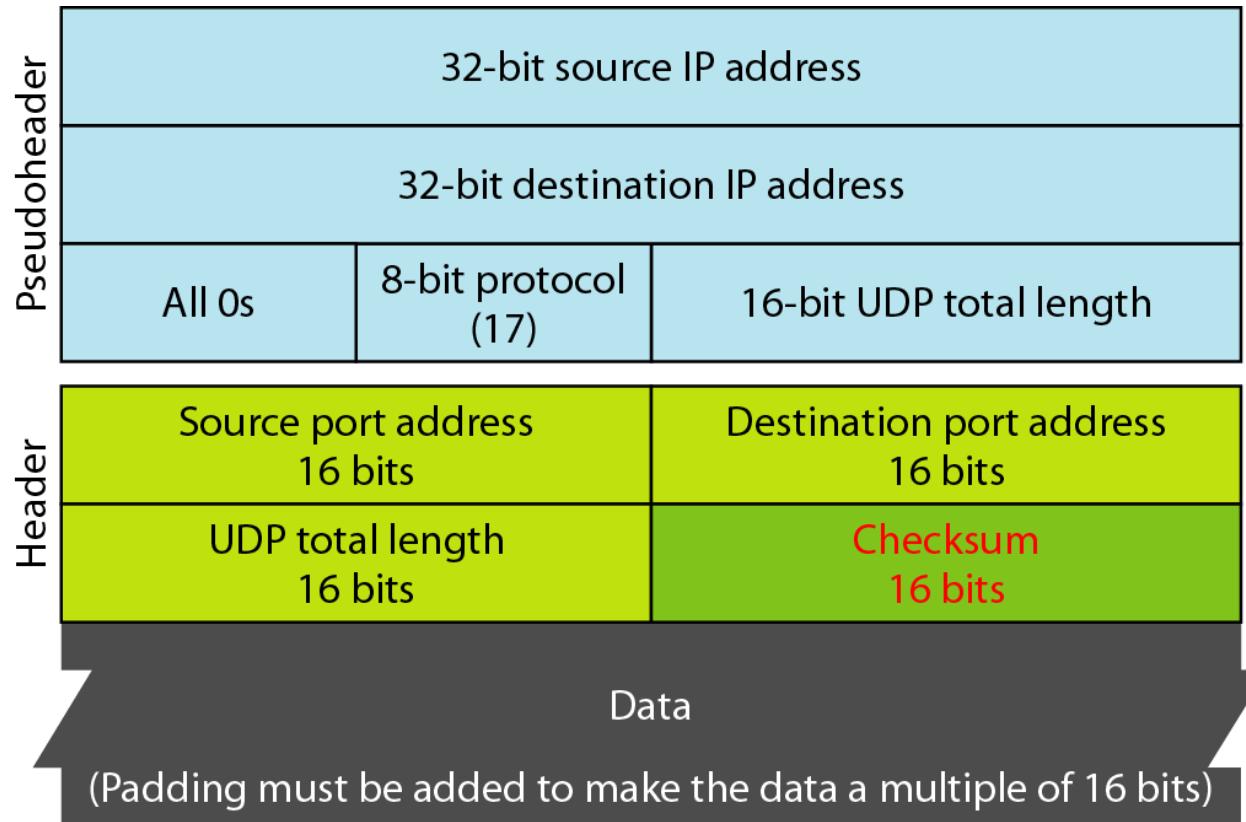
The UDP checksum calculation is different from the one for IP and ICMP. Here the checksum includes three sections: **a pseudo header, the UDP header, and the data** coming from the application layer.

The pseudo header is the part of the header of the IP packet in which the user datagram is to be encapsulated with some fields filled with Os

If the checksum does not include the pseudo header, a user datagram may arrive safe and sound. However, if the IP header is corrupted, it may be delivered to the wrong host.

The protocol field is added to ensure that the packet belongs to UDP, and not to other transport-layer protocols.

**Figure 23.10 Pseudoheader for checksum calculation**



# **UDP Operation**

## **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 datagrams are not numbered. Also, there is no connection establishment and no connection termination, as is the case for TCP. This means that each user datagram can travel on a different path.

## **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. There is no error control mechanism in UDP except for the checksum. This means that the sender does not know if a message has been lost or duplicated. When the receiver detects an error through the checksum, the user datagram is silently discarded. The lack of flow control and error control

## **Encapsulation and Decapsulation**

To send a message from one process to another, the UDP protocol encapsulates and decapsulates messages in an IP datagram.

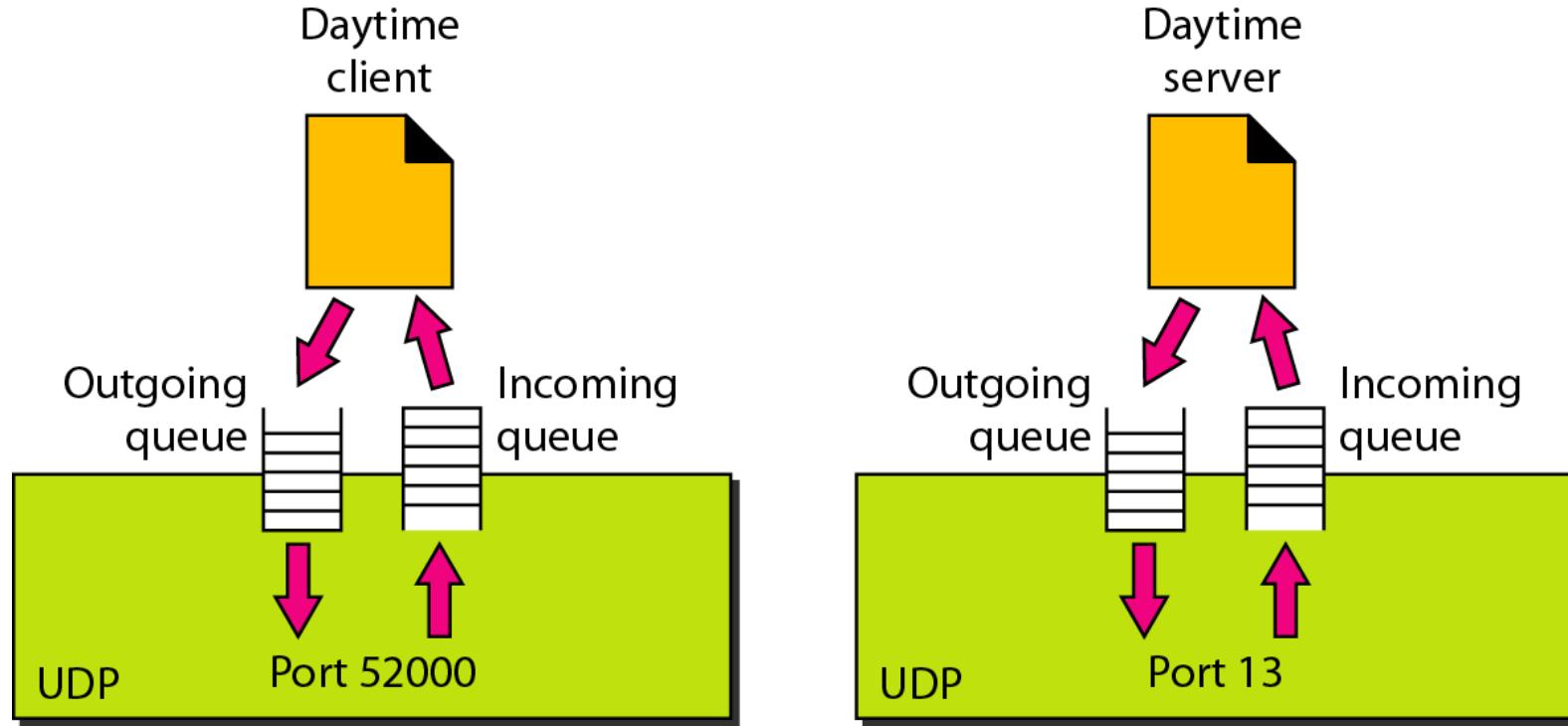
*Figure 23.11 shows the checksum calculation for a very small user datagram with only 7 bytes of data. Because the number of bytes of data is odd, padding is added for checksum calculation. The pseudoheader as well as the padding will be dropped when the user datagram is delivered to IP.*

**Figure 23.11** *Checksum calculation of a simple UDP user datagram*

153.18.8.105			
171.2.14.10			
All 0s	17	15	
1087		13	
15		All 0s	
T	E	S	T
I	N	G	All 0s

10011001 00010010	→	153.18
00001000 01101001	→	8.105
10101011 00000010	→	171.2
00001110 00001010	→	14.10
00000000 00010001	→	0 and 17
00000000 00001111	→	15
00000100 00111111	→	1087
00000000 00001101	→	13
00000000 00001111	→	15
00000000 00000000	→	0 (checksum)
01010100 01000101	→	T and E
01010011 01010100	→	S and T
01001001 01001110	→	I and N
01000111 00000000	→	G and 0 (padding)
10010110 11101011		→ Sum
01101001 00010100		→ Checksum

Figure 23.12 *Queues in UDP*



# Remote Procedure Call

The key work was done by Birrell and Nelson (1984). In a nutshell, what Birrell and Nelson suggested was allowing programs to call procedures located on remote hosts. When a process on machine 1 calls a procedure on machine 2, the calling process on 1 is suspended and execution of the called procedure takes place on 2. Information can be transported from the caller to the callee in the parameters and can come back in the procedure result. No message passing is visible to the application programmer. This technique is known as **RPC (Remote Procedure Call)**. Traditionally, the calling procedure is known as the client and the called procedure is known as the server, and we will use those names here too.

to call a remote procedure, the client program must be bound with a small library procedure, called the **client stub**, that represents the server procedure in the client's address space. Similarly, the server is bound with a procedure called the **server stub**. These procedures hide the fact that the procedure call from the client to the server is not local

Step 1 is the client calling the client stub. This call is a local procedure call, with the parameters pushed onto the stack in the normal way.

Step 2 is the client stub packing the parameters into a message and making a system call to send the message. Packing the parameters is called **marshaling**.

Step 3 is the operating system sending the message from the client machine to the server machine.

Step 4 is the operating system passing the incoming packet to the server stub.

Finally, step 5 is the server stub calling the server procedure with the unmarshaled parameters.

The reply traces the same path in the other direction.

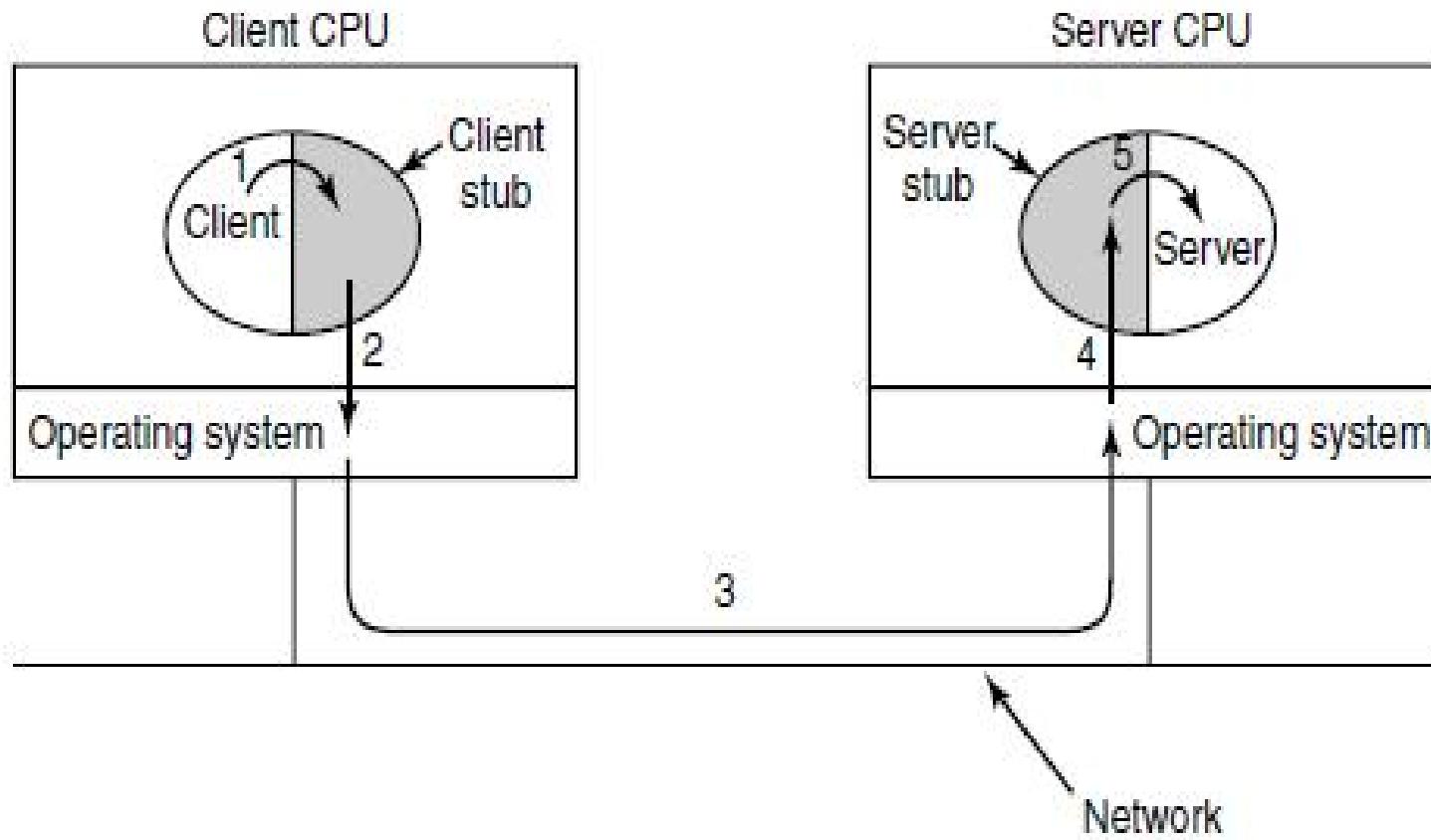


Figure 6-29. Steps in making a remote procedure call. The stubs are shaded.

## Problems with RPC:

- 1 With RPC, passing pointers is impossible because the client and server are in different address spaces.
- 2 It is essentially impossible for the client stub to marshal the parameters: it has no way of determining how large they are.
- 3 A third problem is that it is not always possible to deduce the types of the parameters, not even from a formal specification or the code itself.(exa: printf)
- 4 A fourth problem relates to the use of global variables. Normally, the calling and called procedure can communicate by using global variables, in addition to communicating via parameters. But if the called procedure is moved to a remote machine, the code will fail because the global variables are no longer shared

## Real-Time Transport Protocols

Client-server RPC is one area in which UDP is widely used.

Another one is for real-time multimedia applications.

- Internet radio,
- Internet telephony,
- music-on-demand,
- videoconferencing,
- video-on-demand,

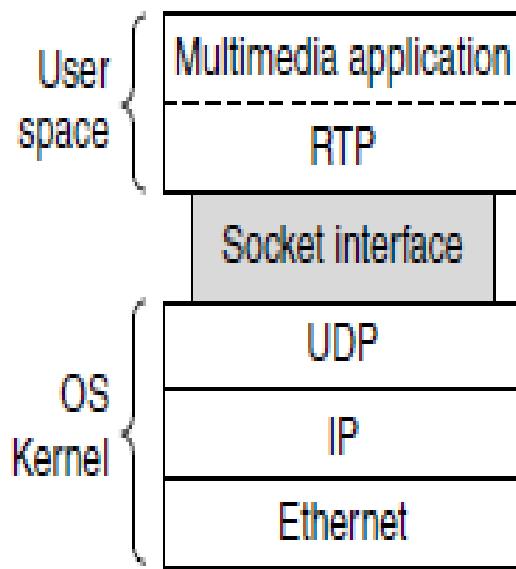
and other multimedia applications became more commonplace, people have discovered that each application was reinventing more or less the same real-time transport protocol.

It gradually became clear that having a generic real-time transport protocol for multiple applications would be a good idea.

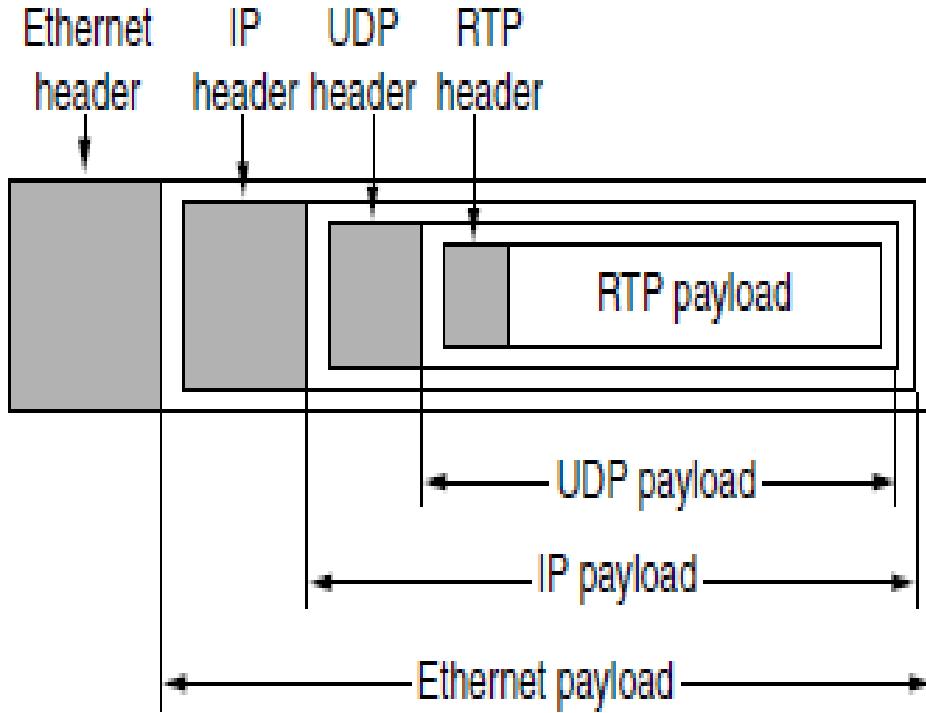
Thus was **RTP (Real-time Transport Protocol)** born. It is described in RFC 3550 and is now in widespread use for multimedia applications. We will describe two aspects of real-time transport.

The first is the RTP protocol for transporting audio and video data in packets.

The second is the processing that takes place, mostly at the receiver, to play out the audio and video at the right time..



(a)



(b)

Figure 6-30. (a) The position of RTP in the protocol stack. (b) Packet nesting.

RTP normally runs in user space over UDP (in the operating system). It operates as follows. The multimedia application consists of multiple audio, video, text, and possibly other streams. These are fed into the RTP library, which is in user space along with the application. This library multiplexes the streams and encodes them in RTP packets, which it stuffs into a socket.

On the operating system side of the socket, UDP packets are generated to wrap the RTP packets and handed to IP for transmission over a link such as Ethernet.

The reverse process happens at the receiver. The multimedia application eventually receives multimedia data from the RTP library. It is responsible for playing out the media. The protocol stack for this situation is shown in Fig. 6-30(a). The packet nesting is shown in Fig. 6-30(b).

## RTP—The Real-time Transport Protocol

The basic function of RTP is to multiplex several real-time data streams onto a single stream of UDP packets. The UDP stream can be sent to a single destination (unicasting) or to multiple destinations (multicasting). Because RTP just uses normal UDP, its packets are not treated specially by the routers unless some normal IP quality-of-service features are enabled. In particular, there are no special guarantees about delivery, and packets may be lost, delayed, corrupted, etc.

The RTP format contains several features.

Each packet sent in an RTP stream is given a number one higher than its predecessor. This numbering allows the destination to determine if any packets are missing.

RTP has no acknowledgements, and no mechanism to request retransmissions.

Each RTP payload may contain multiple samples, and they may be coded any way that the application wants. To allow for interworking, RTP defines several profiles (e.g., a single audio stream), and for each profile, multiple encoding formats may be allowed

Another facility many real-time applications need is time stamping. Not only does time stamping reduce the effects of variation in network delay, but it

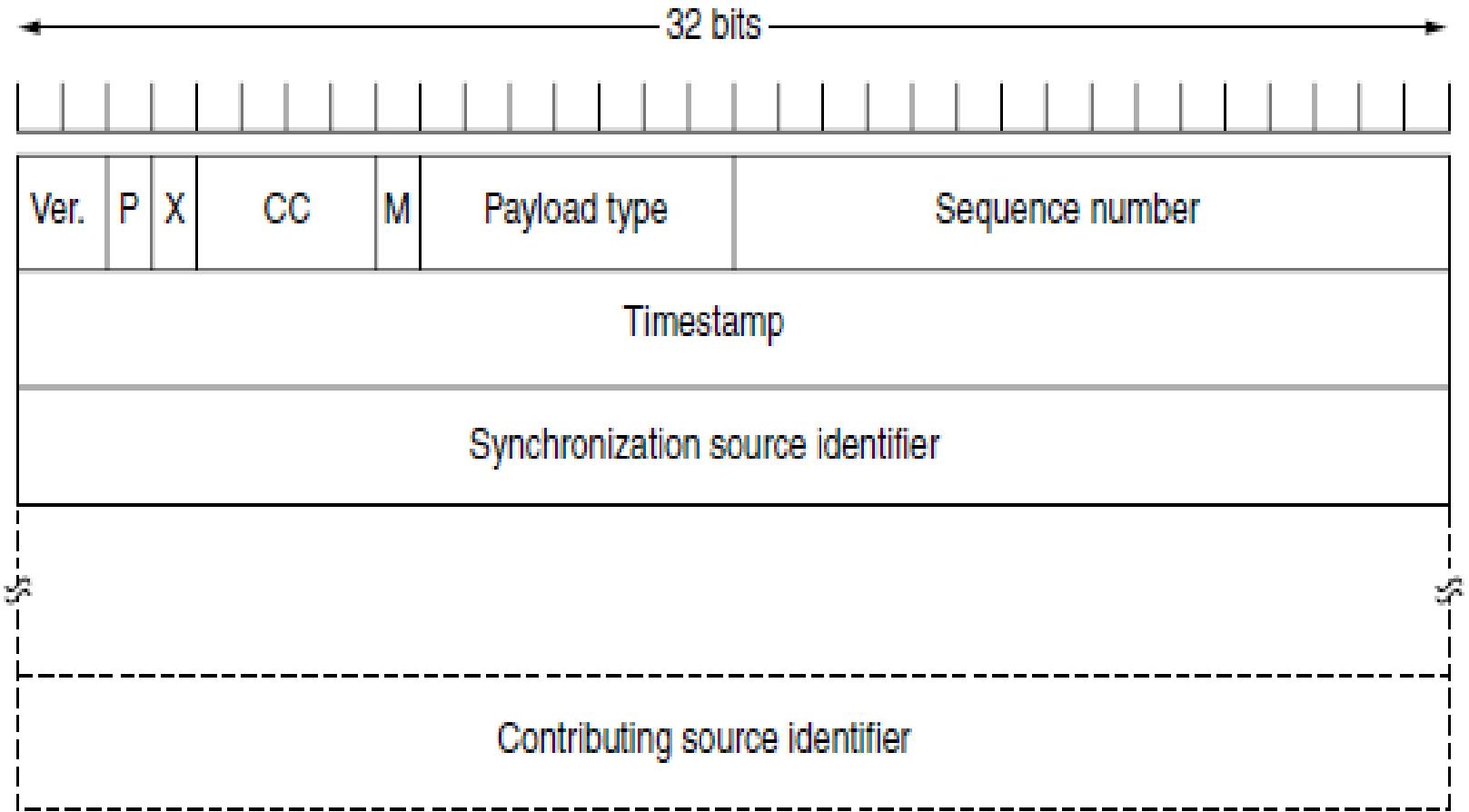


Figure 6-31. The RTP header.

It consists of three 32-bit words and potentially some extensions. The first word contains the *Version field*, which is already at 2. The *P bit* indicates that the packet has been padded to a multiple of 4 bytes. The last padding byte tells how many bytes were added. The *X bit* indicates that an extension header is present. The *CC field* tells how many contributing sources are present, from 0 to 15. The *M bit* is an application-specific marker bit. It can be used to mark the start of a video frame, the start of a word in an audio channel, or something else that the application understands. The *Payload type field* tells which encoding algorithm has been used (e.g., uncompressed 8-bit audio, MP3, etc.). Since every packet carries this field, the encoding can change during transmission. The *Sequence number* is just a counter that is incremented on each RTP packet sent. It is used to detect lost packets. The *Timestamp*, this value can help reduce timing variability called jitter at the receiver by decoupling the playback from the packet arrival time. The *Synchronization source identifier* tells which stream the packet belongs to. It is the method used to multiplex and demultiplex multiple data streams onto a single stream of UDP packets. Finally, the *Contributing source identifiers*, if any, are used when i

## RTCP—The Real-time Transport Control Protocol

RTP has a little sister protocol (little sibling protocol?) called RTCP (Real time Transport Control Protocol). It is defined along with RTP in RFC 3550 and handles feedback, synchronization, and the user interface. It does not transport any media samples.

The first function can be used to provide feedback on delay, variation in delay or jitter, bandwidth, congestion, and other network properties to the sources.

This information can be used by the encoding process to increase the data rate (and give better quality) when the network is functioning well and to cut back the data rate when there is trouble in the network. By providing continuous feedback, It provides the best quality

The *Payload type field* is used to tell the destination what encoding algorithm is used for the current packet, making it possible to vary it on demand.

RTCP also handles inter stream synchronization. The problem is that different streams may use different clocks, with different granularities and different drift rates. RTCP can be used to keep them in sync.

Finally, RTCP provides a way for naming the various sources (e.g., in ASCII text). This information can be displayed on the receiver's screen to indicate who is talking at the moment.

## Playout with Buffering and Jitter Control

Once the media information reaches the receiver, it must be played out at the right time. Even if the packets are injected with exactly the right intervals between them at the sender, they will reach the receiver with different relative times. This variation in delay is called **jitter**. Even a small amount of packet jitter can cause distracting media artifacts, such as jerky video frames and unintelligible audio, if the media is simply played out as it arrives.

The solution to this problem is to **buffer** packets at the receiver before they are played out to reduce the jitter.

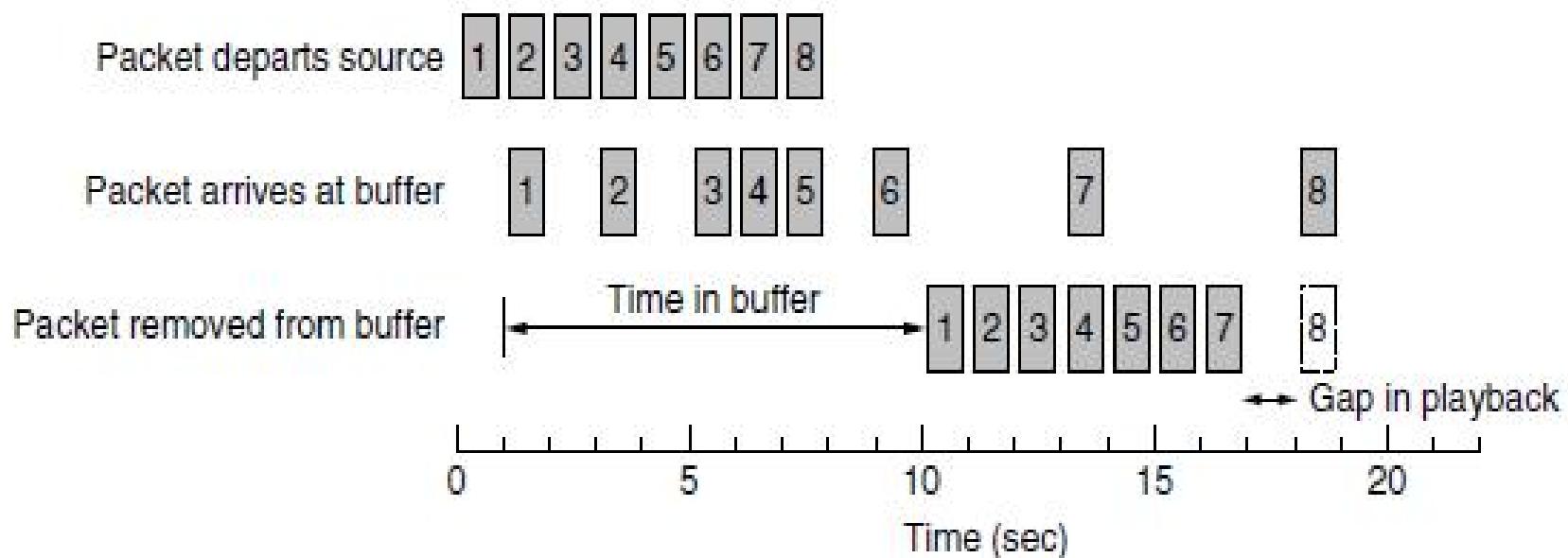


Figure 6-32. Smoothing the output stream by buffering packets.

A key consideration for smooth playout is the **playback point**, or how long to wait at the receiver for media before playing it out. Deciding how long to wait depends on the jitter. The difference between a low-jitter and high-jitter connection is shown in Fig. The average delay may not differ greatly between the two, but if there is high jitter the playback point may need to be much further out to capture 99% of the packets than if there is low jitter.

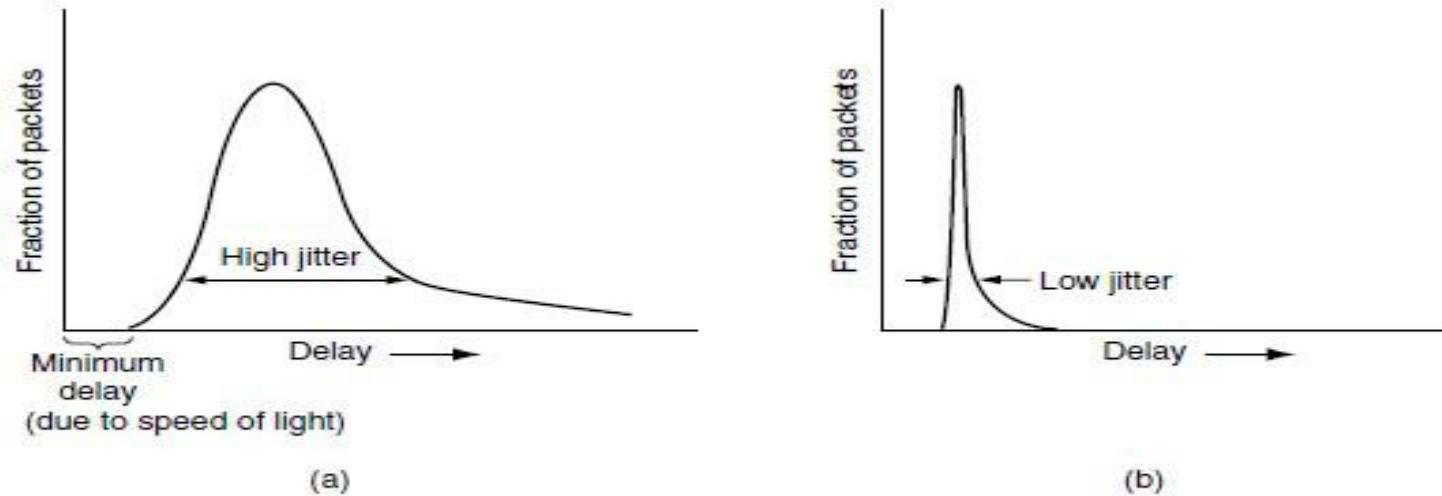


Figure 6-33. (a) High jitter. (b) Low jitter.

One way to avoid this problem for audio is to adapt the playback point between **talk spurts**, in the gaps in a conversation. No one will notice the difference between a short and slightly longer silence

## TELNET

It is client/server application program. TELNET is an abbreviation for *TERminaL NETwork*. TELNET enables the establishment of a connection to a remote system in such a way that the local terminal appears to be a terminal at the remote system.

### *Timesharing Environment*

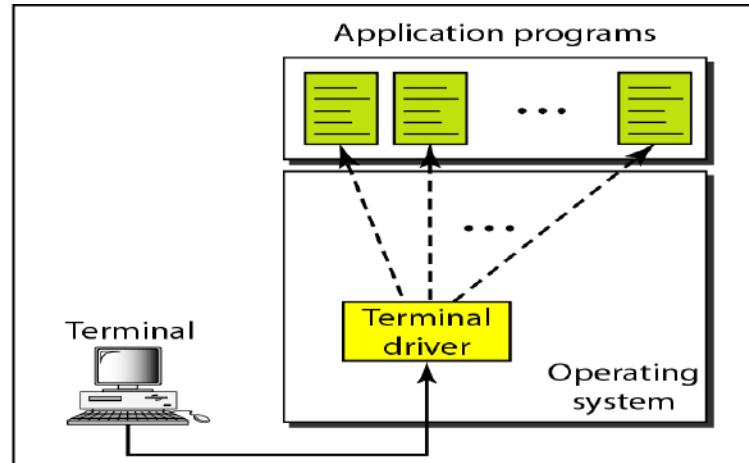
A large computer supports multiple users. The interaction between a user and the computer occurs through a terminal, which is usually a combination of keyboard, monitor, and mouse.

### *Logging*

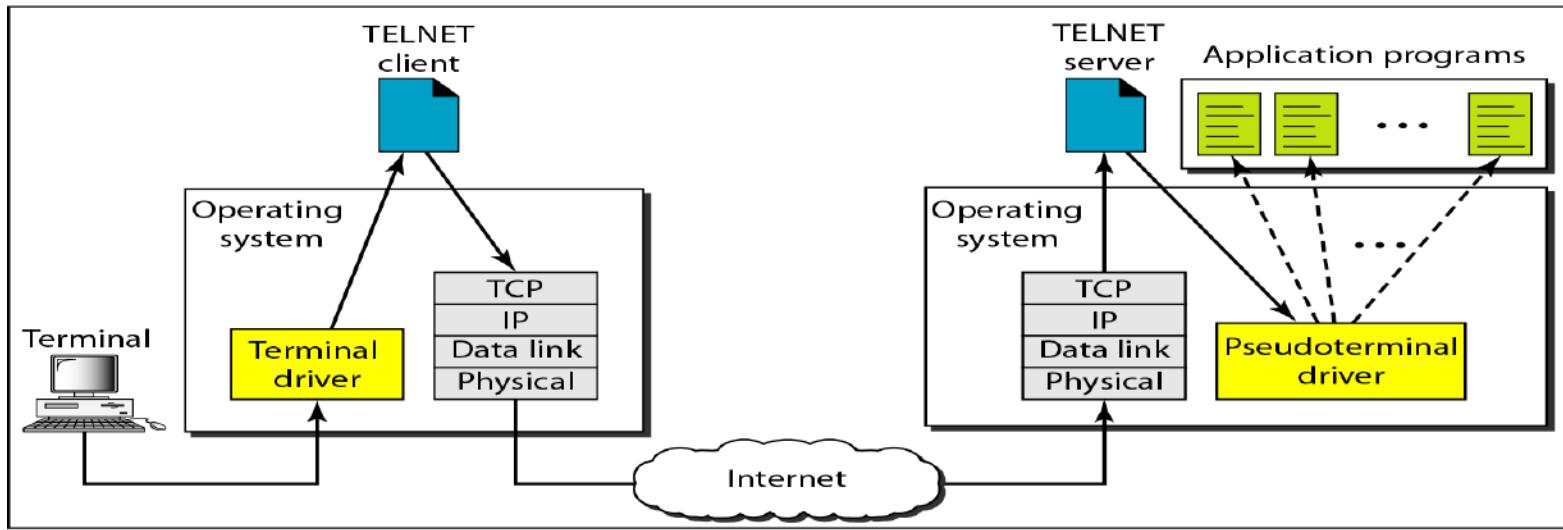
To access the system, the user logs into the system with a user id or log-in name. The system also includes password checking to prevent an unauthorized user from accessing the resources.

Local login

Remote login



a. Local log-in



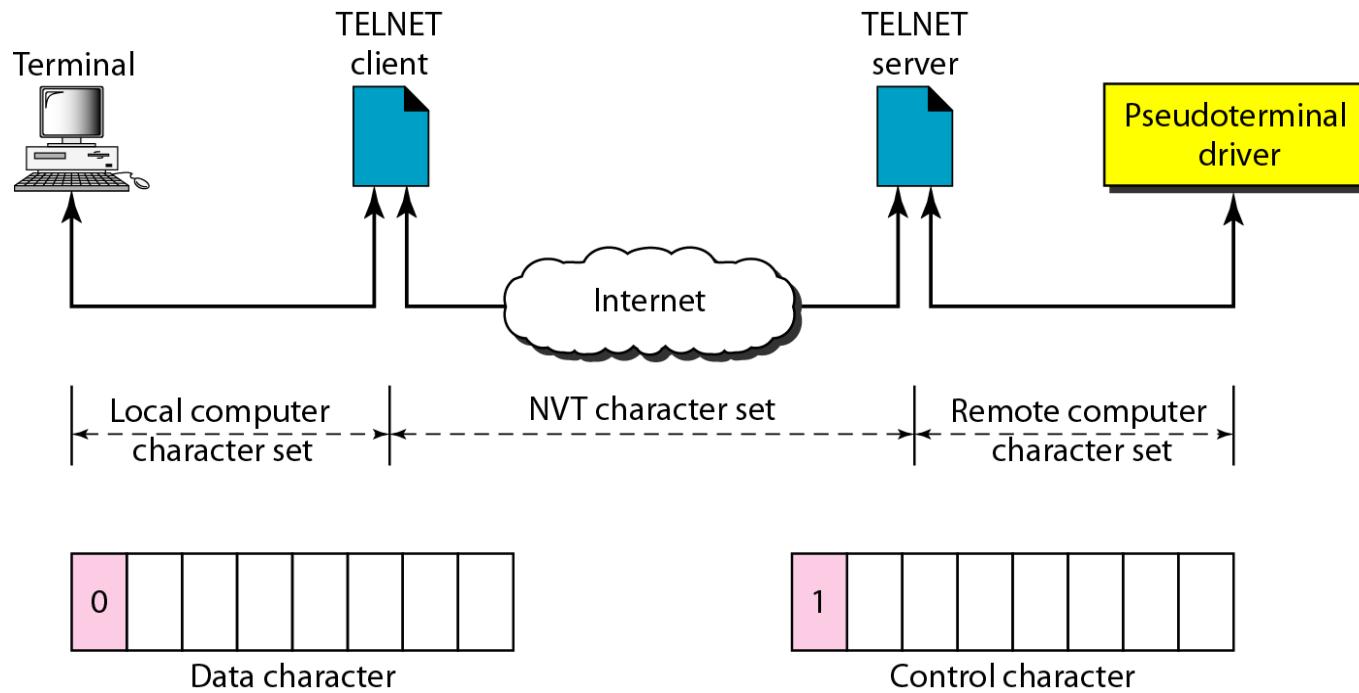
b. Remote log-in

When a user logs into a local timesharing system, it is called **local log-in**. As a user types at a terminal or at a workstation running a terminal emulator, the keystrokes are accepted by the terminal driver. The terminal driver passes the characters to the operating system. The operating system, in turn, interprets the combination of characters and invokes the desired application program or utility.

When a user wants to access an application program or utility located on a remote Machine, it is called **remote log-in**. Here the TELNET client and server programs come into use. The user sends the keystrokes to the terminal driver, where the local operating system accepts the characters but does not interpret them. The characters are sent to the TELNET client, which transforms the characters to a universal character set called network virtual terminal (NVT) characters and delivers them to the local TCP/IP protocol stack.

The commands or text, in NVT form, travel through the Internet and arrive at the TCP/IP stack at the remote machine. Here the characters are delivered to the operating system and passed to the TELNET server, which changes the characters to the corresponding characters understandable by the remote computer. However, the characters cannot be passed directly to the operating system because the remote operating system is not designed to receive characters from a TELNET server: It is designed to receive characters from a terminal driver. The solution is to add a piece of

## *Concept of NVT(network virtual terminal)*





# Data Communications and Networking

Fourth Edition

Forouzan

## WWW and HTTP

*The WWW today is a distributed client/server service, in which a client using a browser can access a service using a server. However, the service provided is distributed over many locations called sites as shown in fig.*

*Topics discussed in this section:*

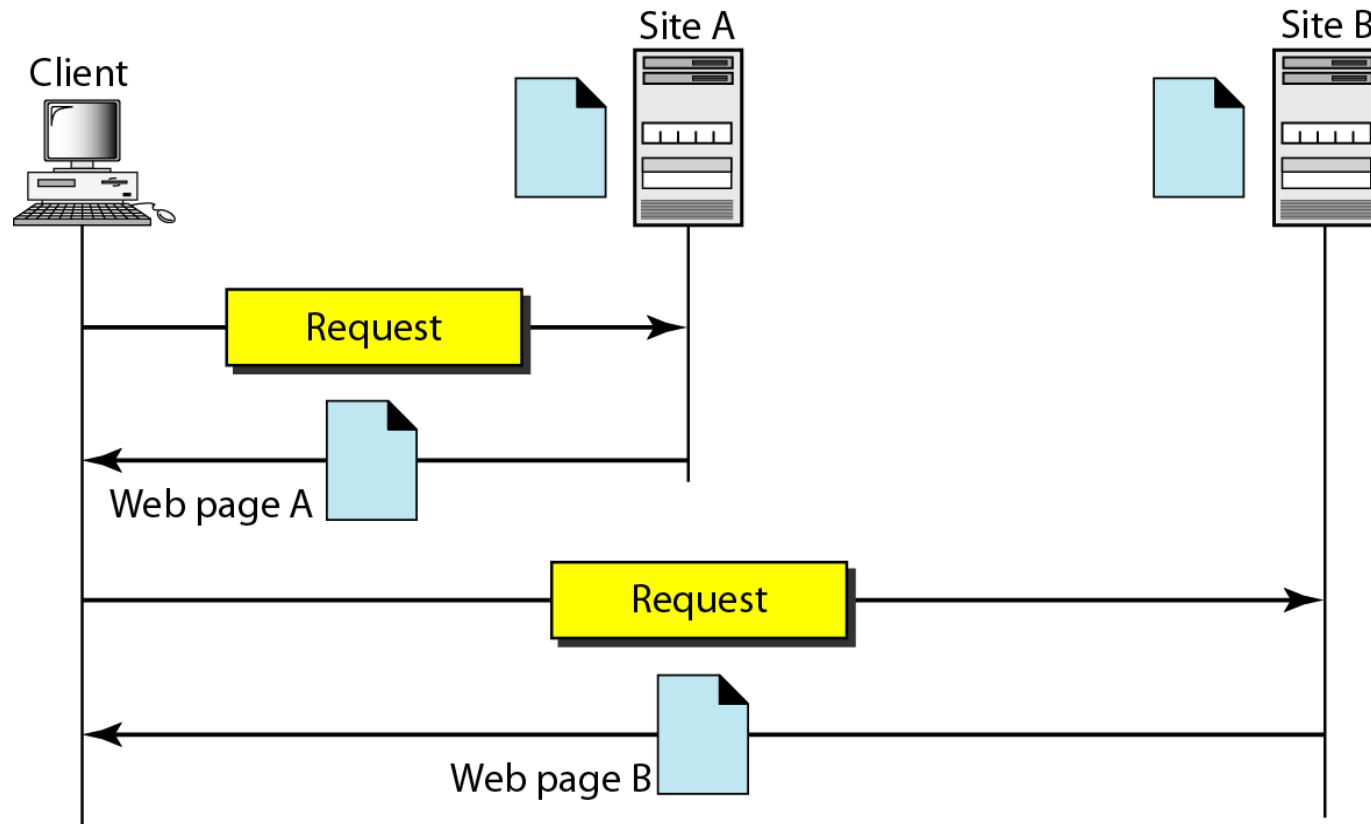
Client (Browser)

Server

Uniform Resource Locator

Cookies

**Figure 27.1** *Architecture of WWW*



## Client (Browser)

A variety of vendors offer commercial browsers that interpret and display a Web document, and all use nearly the same architecture.

Each browser usually consists of three parts: a controller, client protocol, and interpreters.

The controller receives input from the keyboard or the mouse and uses the client programs to access the document.

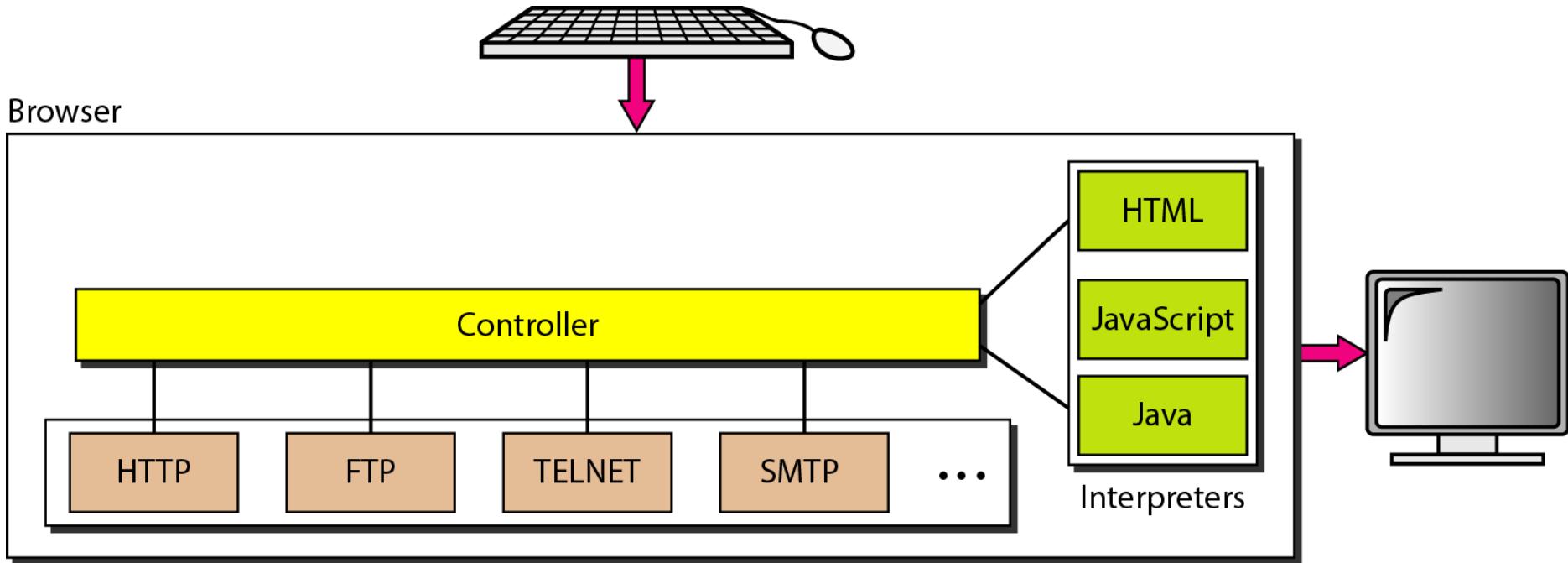
After the document has been accessed, the controller uses one of the interpreters to display the document on the screen. The interpreter can be HTML, Java, or JavaScript, depending on the type of document

The client protocol can be one of the protocols described previously such as FTP or HTTP.

## Server

The Web page is stored at the server. Each time a client request arrives, the corresponding document is sent to the client. To improve efficiency, servers normally store requested files in a cache in memory; memory is faster to access than disk. A server can also become more efficient through multithreading or multiprocessing. In this case, a server can answer more than one request at a time.

Figure 27.2 *Browser*



## Uniform Resource Locator

A client that wants to access a Web page needs the address. To facilitate the access of documents distributed throughout the world, HTTP uses locators. The uniform resource locator (URL) is a standard for specifying any kind of information on the Internet. The URL defines four things: protocol, host computer, port, and path.

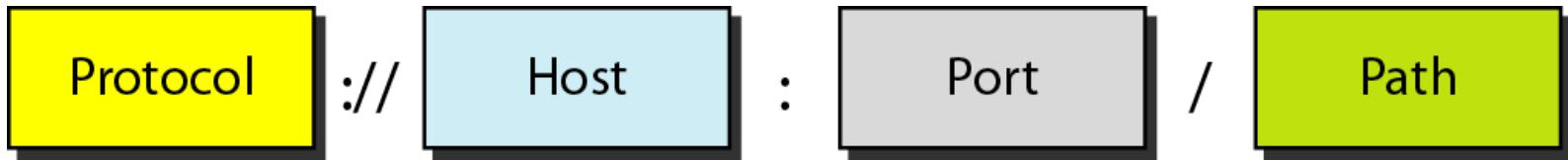
The *protocol* is the client/server program used to retrieve the document. Many different protocols can retrieve a document; among them are FTP or HTTP. The most common today is HTTP.

The host is the computer on which the information is located, although the name of the computer can be an alias. Web pages are usually stored in computers, and computers are given alias names that usually begin with the characters "www".

The URL can optionally contain the port number of the server. If the *port* is included, it is inserted between the host and the path, and it is separated from the host by a colon.

Path is the pathname of the file where the information is located. Note that the path can itself contain slashes that, in the UNIX operating system, separate the directories from the subdirectories and files.

## Figure 27.3 URL



An HTTP cookie (also called web cookie, Internetcookie, browser cookie or simply cookie, the latter which is not to be confused with the literal definition), is a small piece of data sent from a website and stored in a user's web browser while the user is browsing that website

*The documents in the WWW can be grouped into three broad categories: static, dynamic, and active. The category is based on the time at which the contents of the document are determined.*

### Topics discussed in this section:

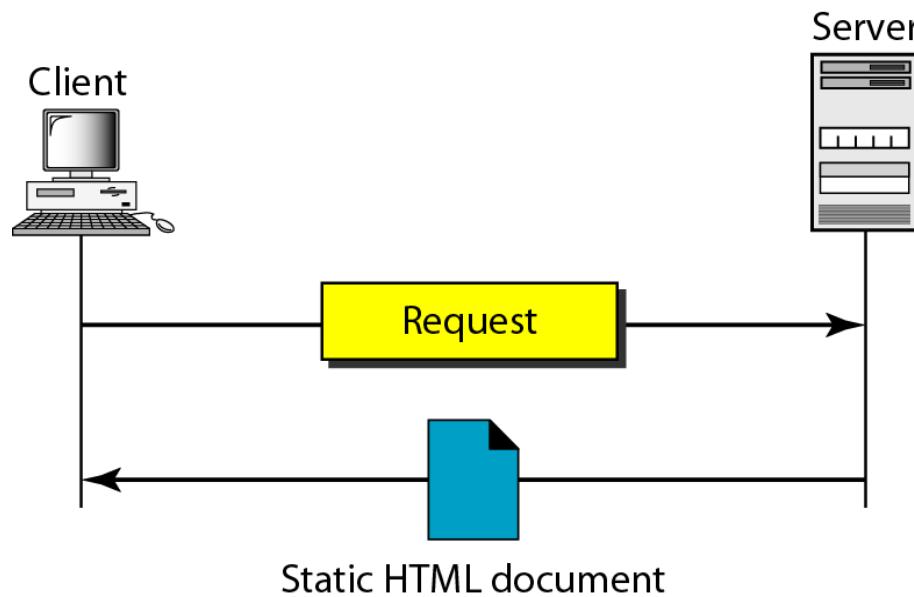
Static Documents

Dynamic Documents

Active Documents

## Static Documents

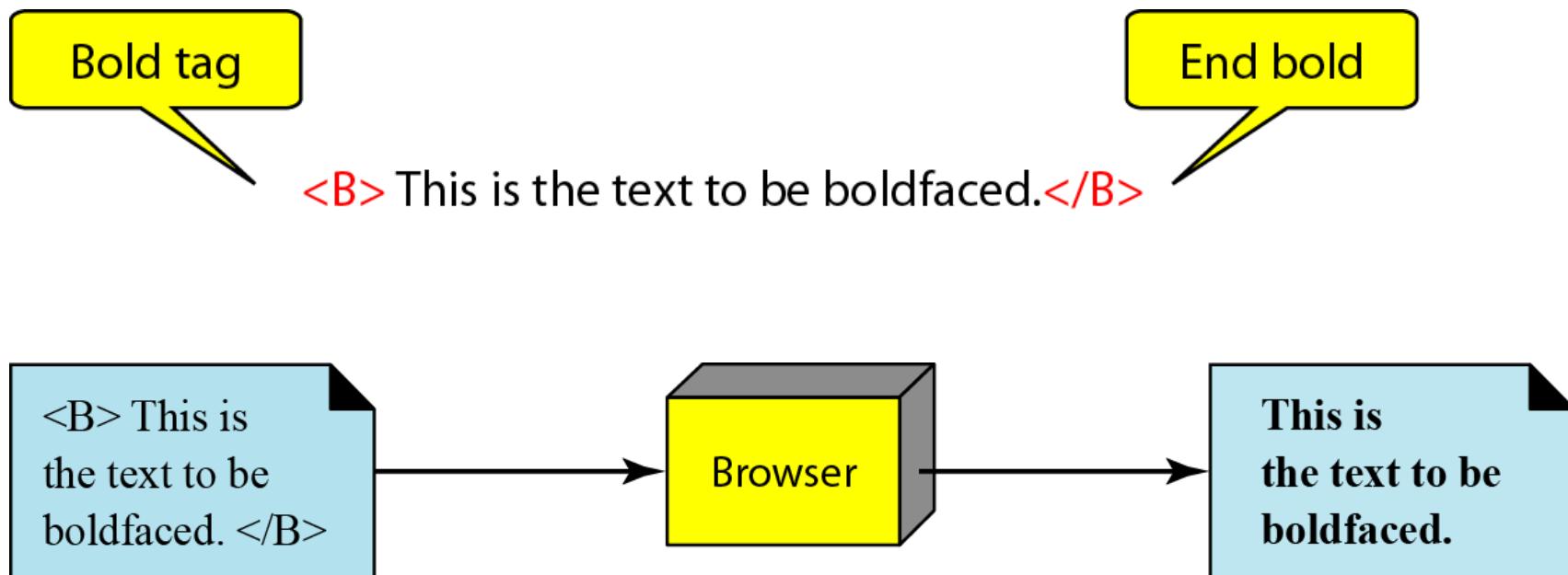
Static documents are fixed-content documents that are created and stored in a server. The client can get only a copy of the document. When a client accesses the document, a copy of the document is sent. The user can then use a browsing program to display the document



**Figure 27.5** *Boldface tags*

## *HTML*

Hypertext Markup Language (HTML) is a language for creating Web pages.



## Figure 27.7 Beginning and ending tags

< TagName Attribute = Value Attribute = Value ... >

a. Beginning tag

< /TagName >

b. Ending tag

## Dynamic Documents

A dynamic document is created by a Web server whenever a browser requests the document. When a request arrives, the Web server runs an application program or a script that creates the dynamic document. The server returns the output of the program or script as a response to the browser that requested the document.

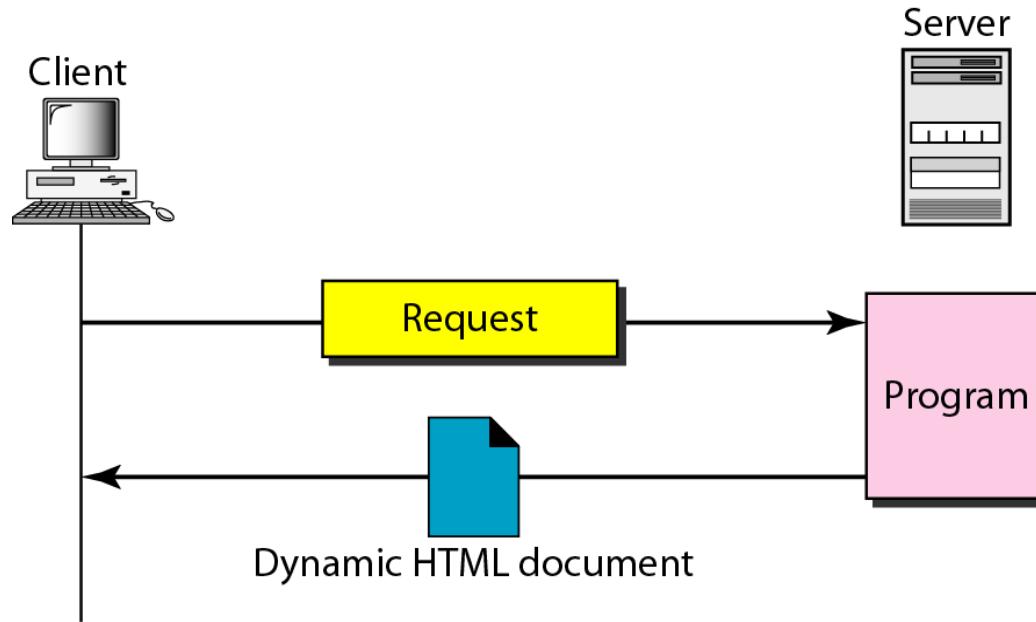
A very simple example of a dynamic document is the retrieval of the time and date from a server. Time and date are kinds of information that are dynamic in that they change from moment to moment. The client can ask the server to run a program such as the *date program in UNIX and send the result of the program to the client*.

### Common Gateway Interface (CGI)

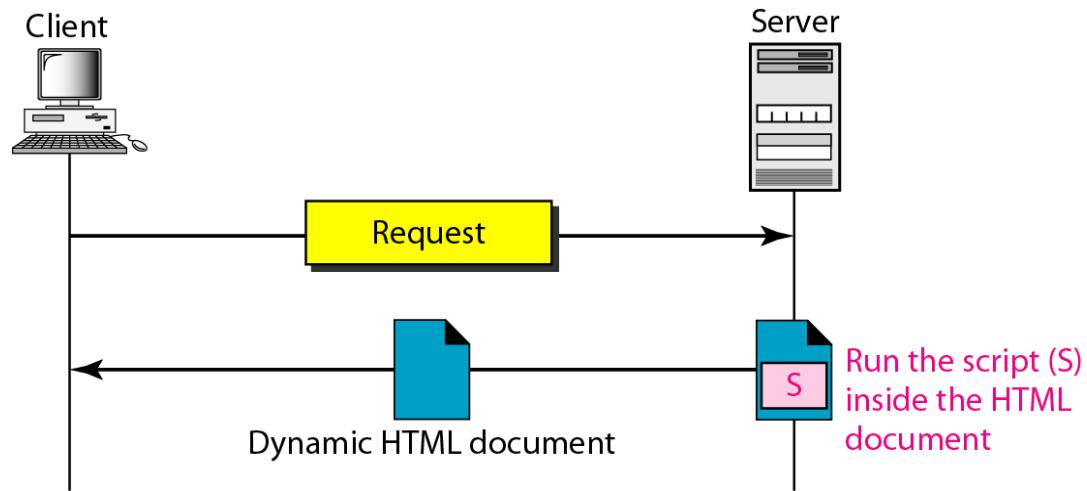
The Common Gateway Interface (CGI) is a technology that creates and handles dynamic documents.

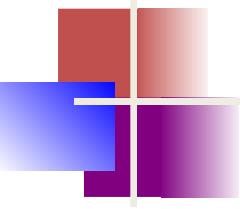
Hypertext Preprocessor (pHP), which uses the Perl language; Java Server Pages (JSP), which uses the Java language for scripting; Active Server Pages (ASP), a Microsoft product which uses Visual Basic language for scripting; and ColdFusion, which embeds SQL database queries in the HTML document

**Figure 27.8** Dynamic document using CGI



**Figure 27.9** Dynamic document using server-site script





## *Note*

---

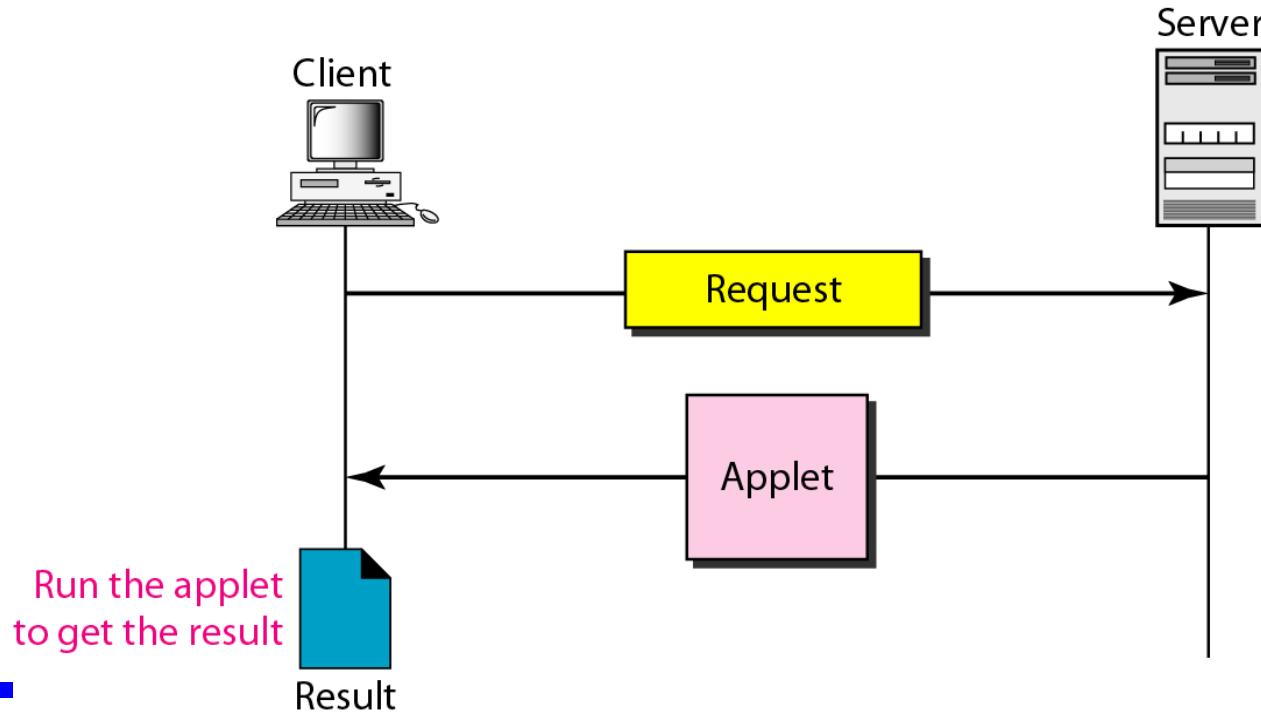
Dynamic documents are sometimes referred to as server-site dynamic documents.

---

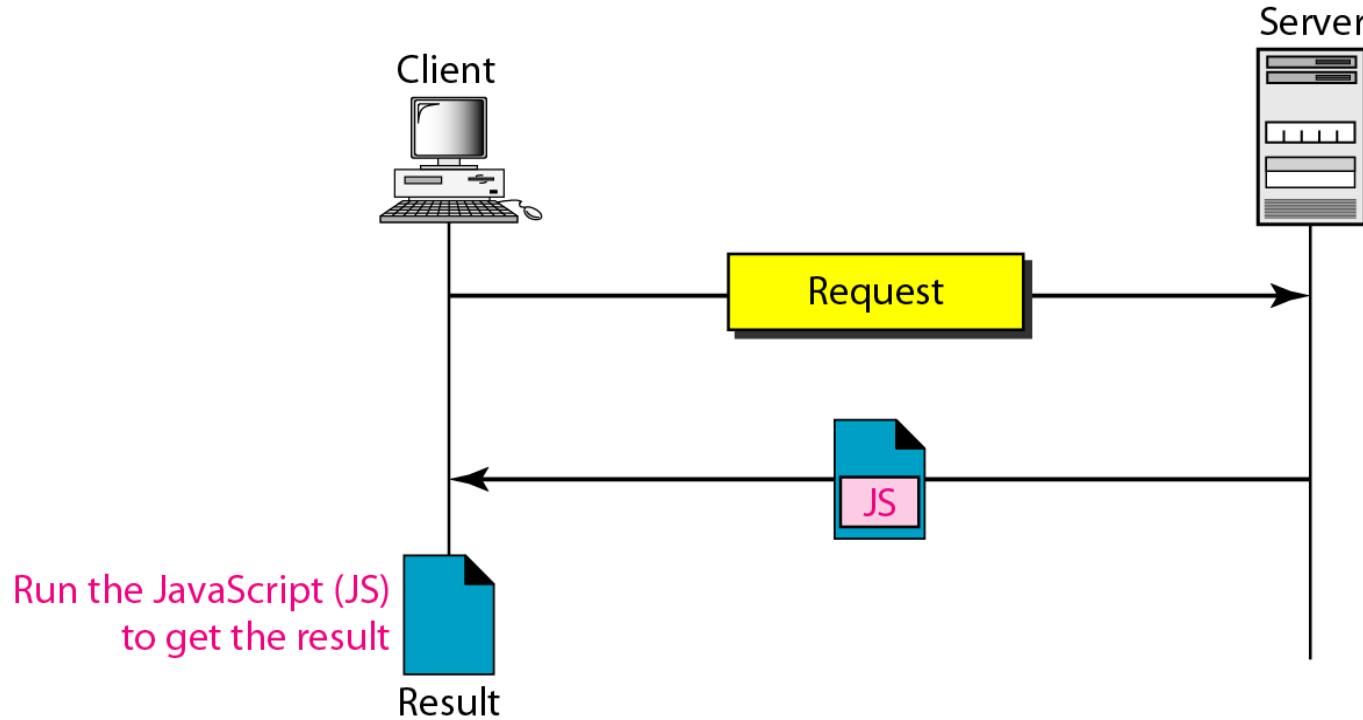
**Figure 27.10** Active document using Java applet

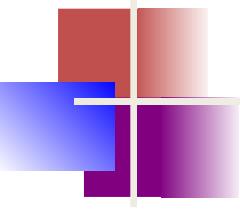
## Active Documents

For many applications, we need a program or a script to be run at the client site. These are called active documents



**Figure 27.11** Active document using client-site script



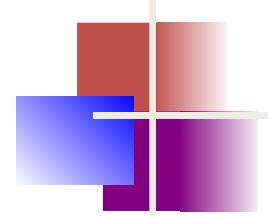


## *Note*

---

Active documents are sometimes referred to as client-site dynamic documents.

---



## *Note*

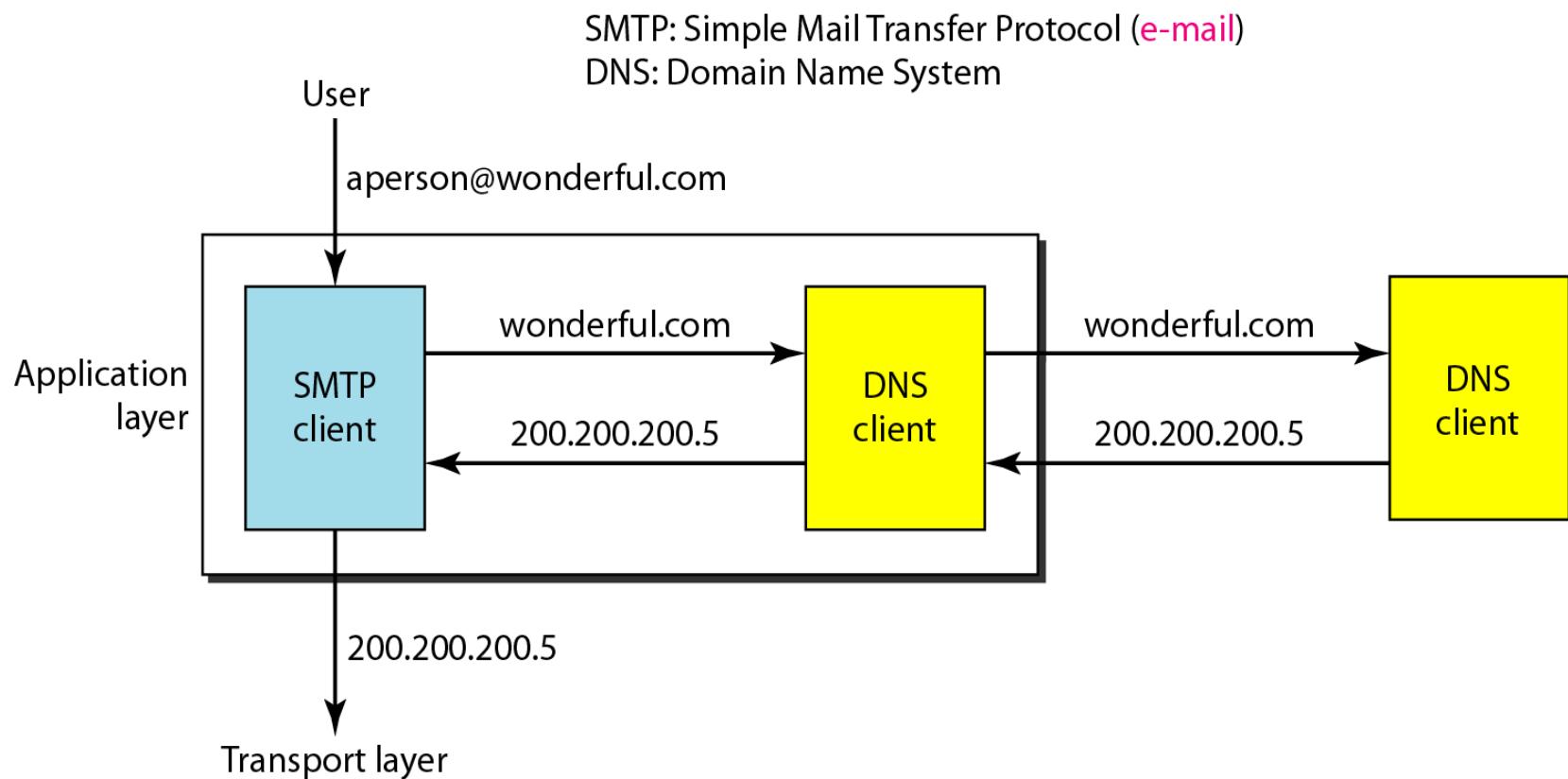
---

HTTP version 1.1 specifies a persistent connection by default.

---

## DNS (Domain Name System)

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.



## NAME SPACE

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.

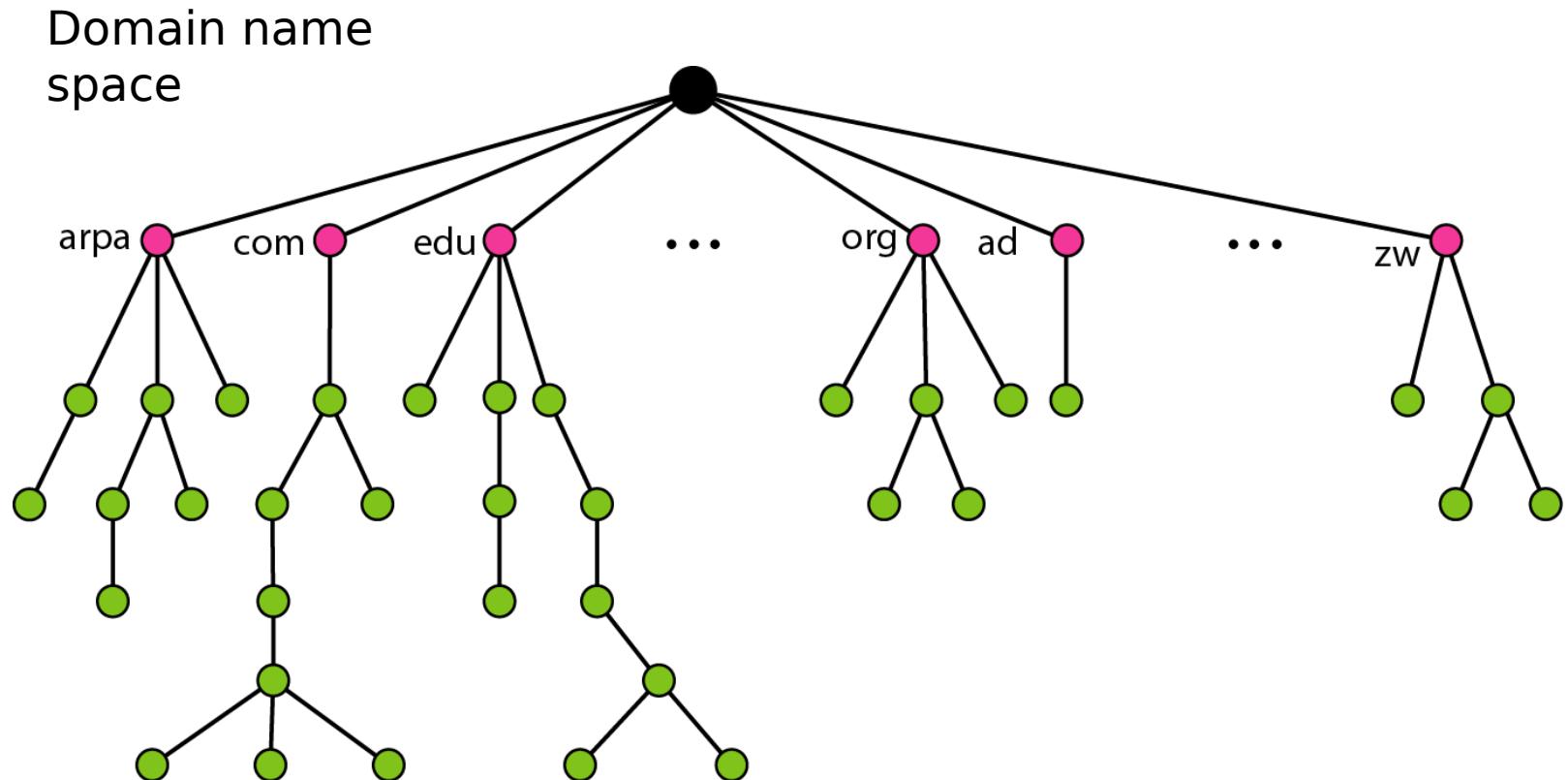
### Hierarchical Name Space

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.

Exa:      *challenger.jhda.edu*,      *challenger.berkeley.edu*,      and  
*challenger.smart.com*

## 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. Below Figure shows some domain names

-

FQDN

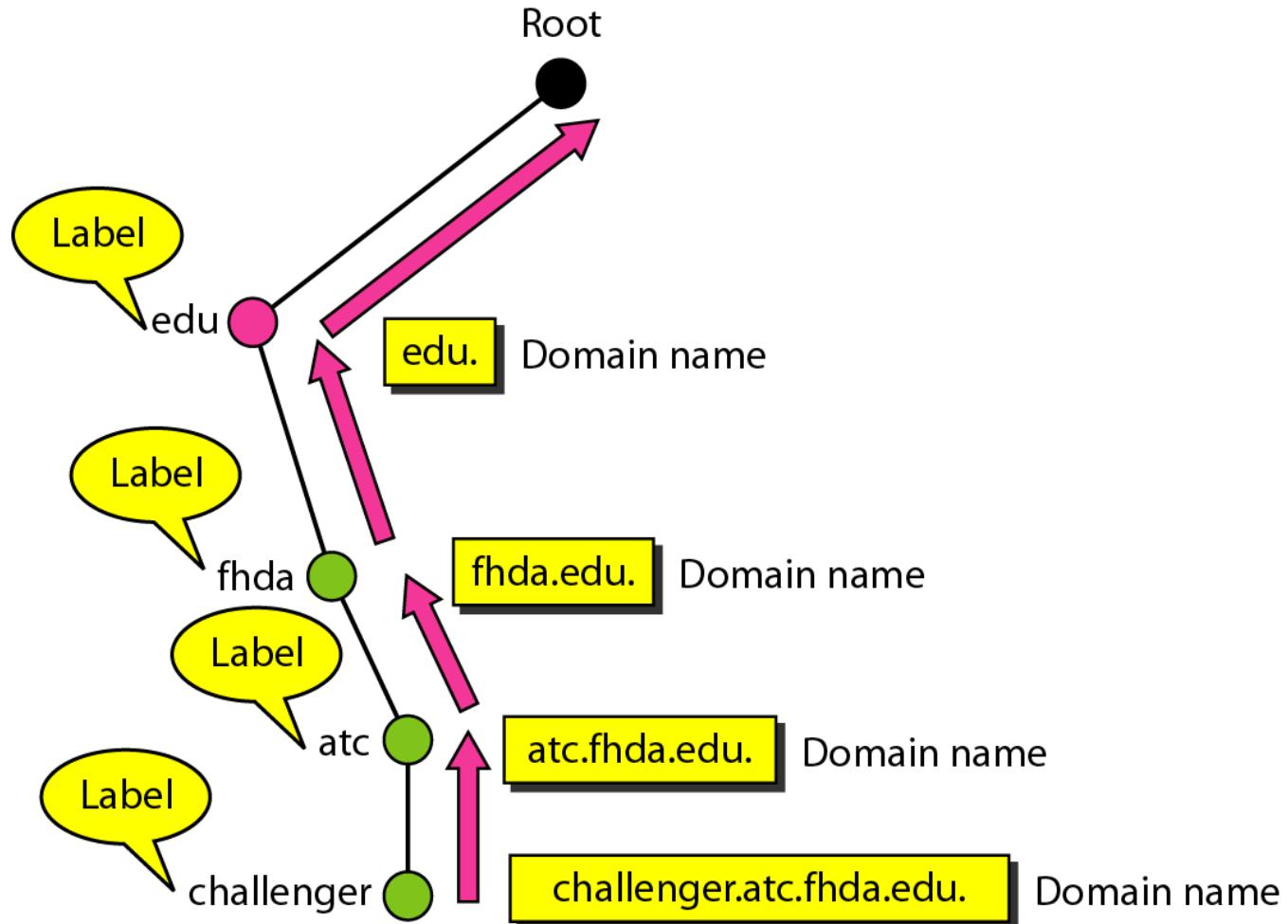
PQDN

challenger.atc.fhda.edu.  
cs.hmme.com.  
www.funny.int.

challenger.atc.fhda.edu  
cs.hmme  
www

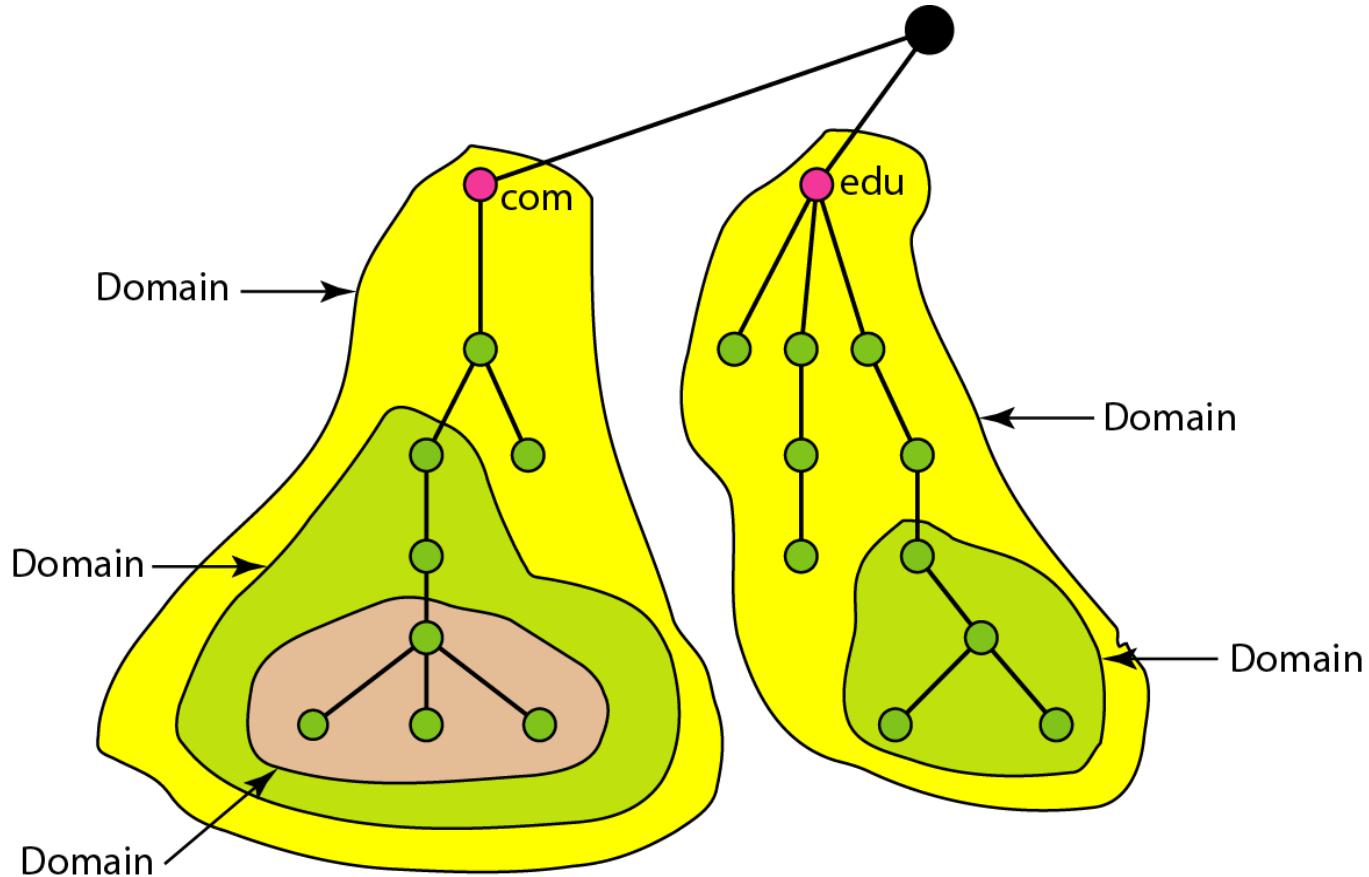
## *Domain names and labels*

---



## Domain

A domain is a subtree of the domain name space. The name of the domain is the domain name of the node at the top of the subtree.

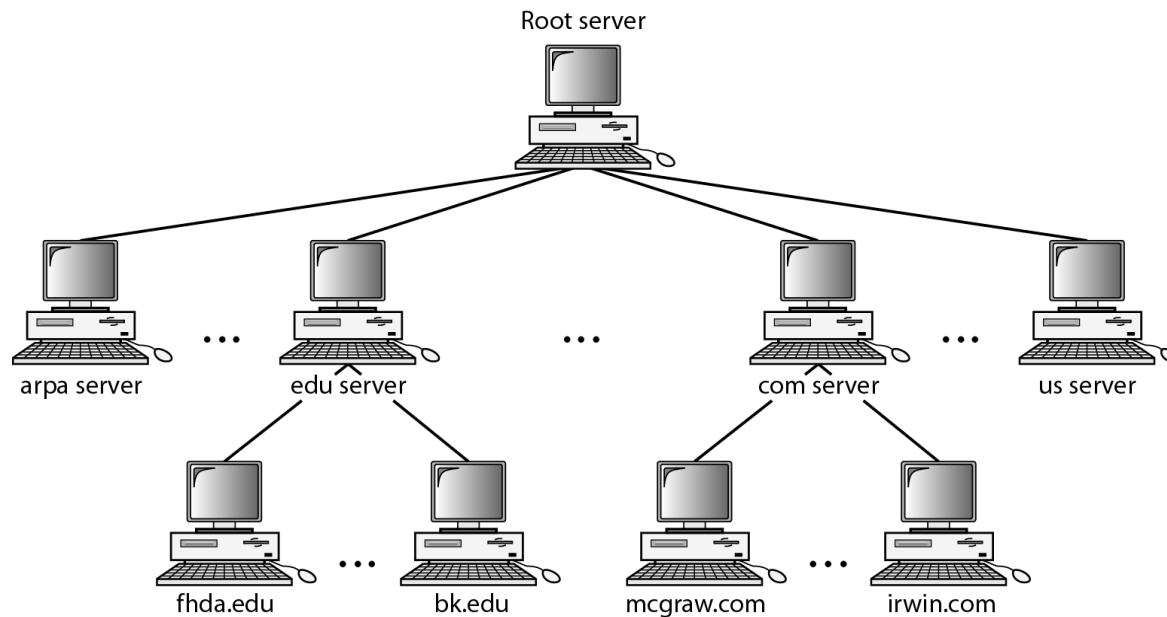


## DISTRIBUTION OF NAME SPACE:

The information contained in the domain name space must be stored. However, it is very inefficient and also unreliable to have just one computer store such a huge amount of information. In this section, we discuss the distribution of the domain name space

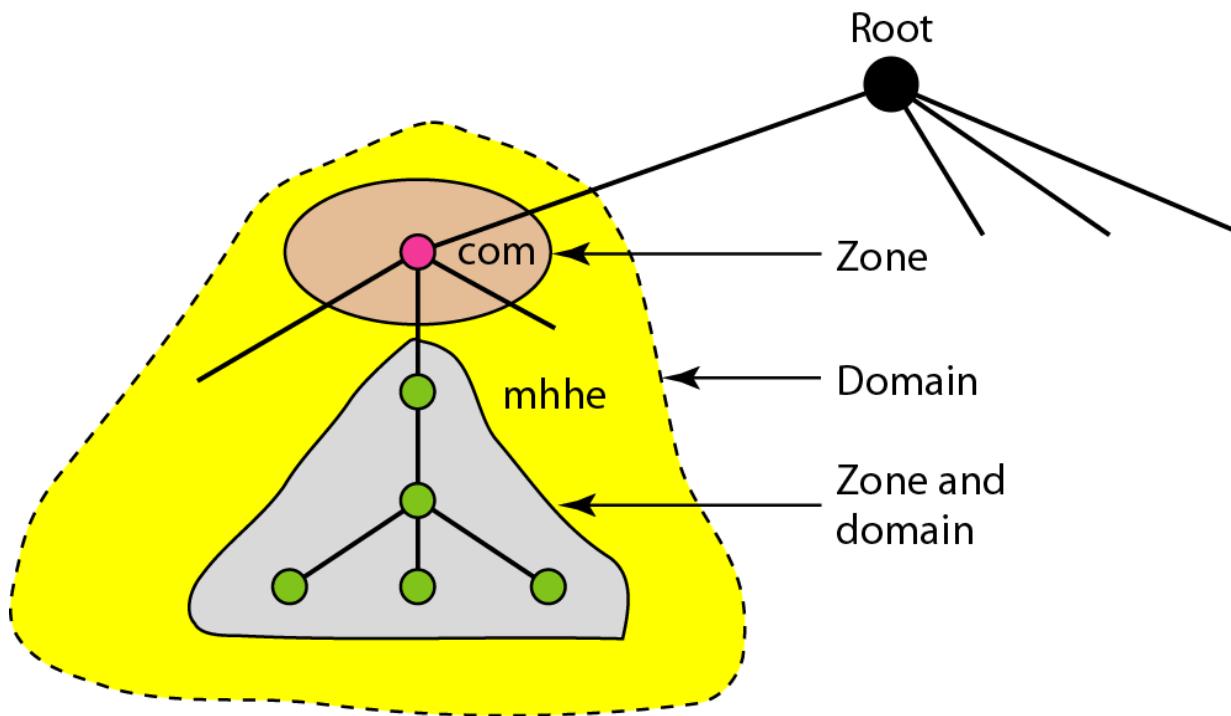
### 1 Hierarchy of Name Servers

distribute the information among many computers called DNS servers. we let the root stand alone and create as many domains (subtrees) as there are first-level nodes



## 2 Zone

Since the complete domain name hierarchy cannot be stored on a single server, it is divided among many servers. What a server is responsible for or has authority over is called a zone. We can define a zone as a contiguous part of the entire tree



### 3 Root Server

A root server is a server whose zone consists of the whole tree. A root server usually does not store any information about domains but delegates its authority to other servers, keeping references to those servers. There are several root servers, each covering the whole domain name space. The servers are distributed all around the world.

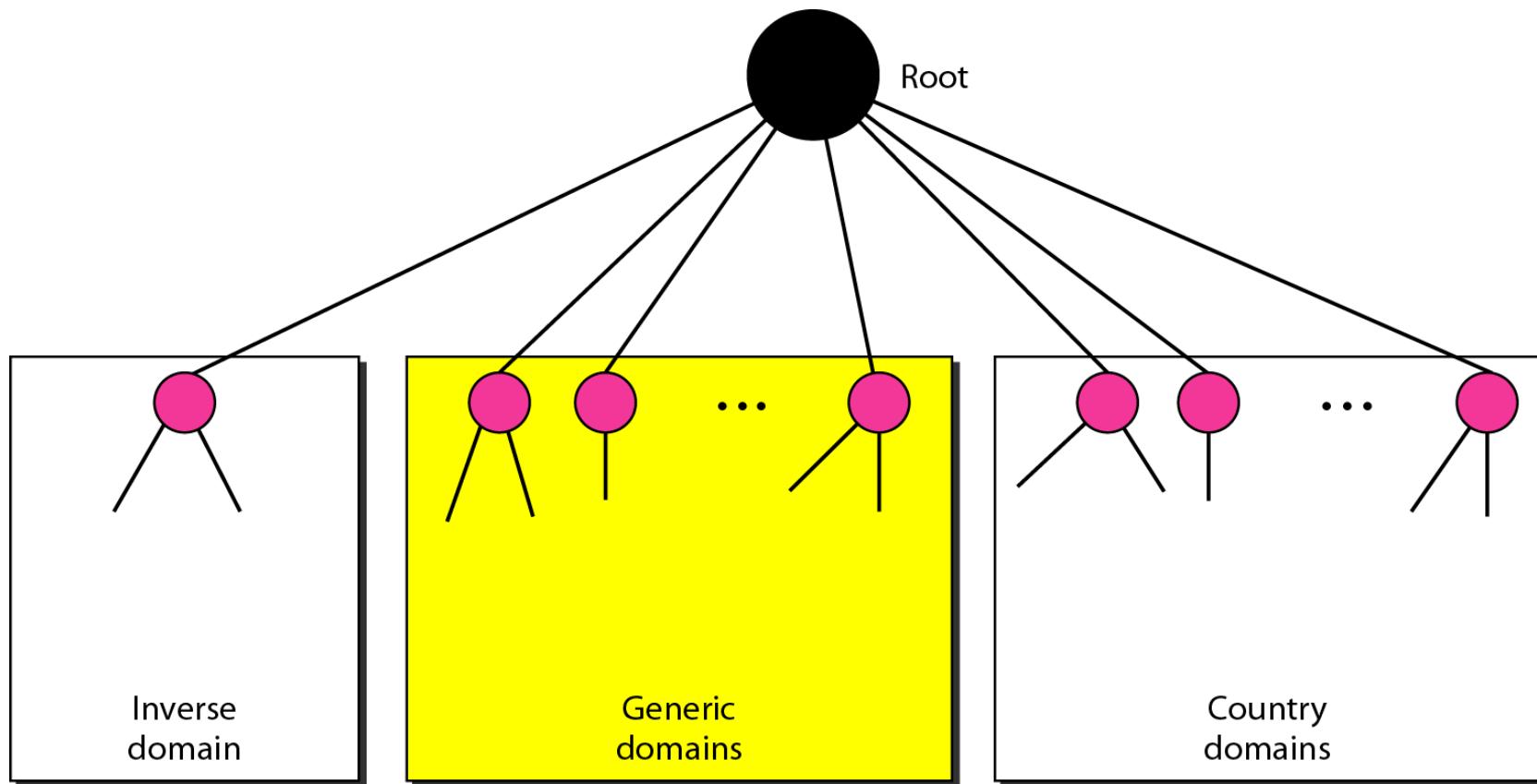
### 4 Primary and Secondary Servers

A primary server is a server that stores a file about the zone for which it is an authority. It is responsible for creating, maintaining, and updating the zone file. It stores the zone file on a local disk

A secondary server is a server that transfers the complete information about a zone from another server (primary or secondary) and stores the file on its local disk. The secondary server neither creates nor updates the zone files

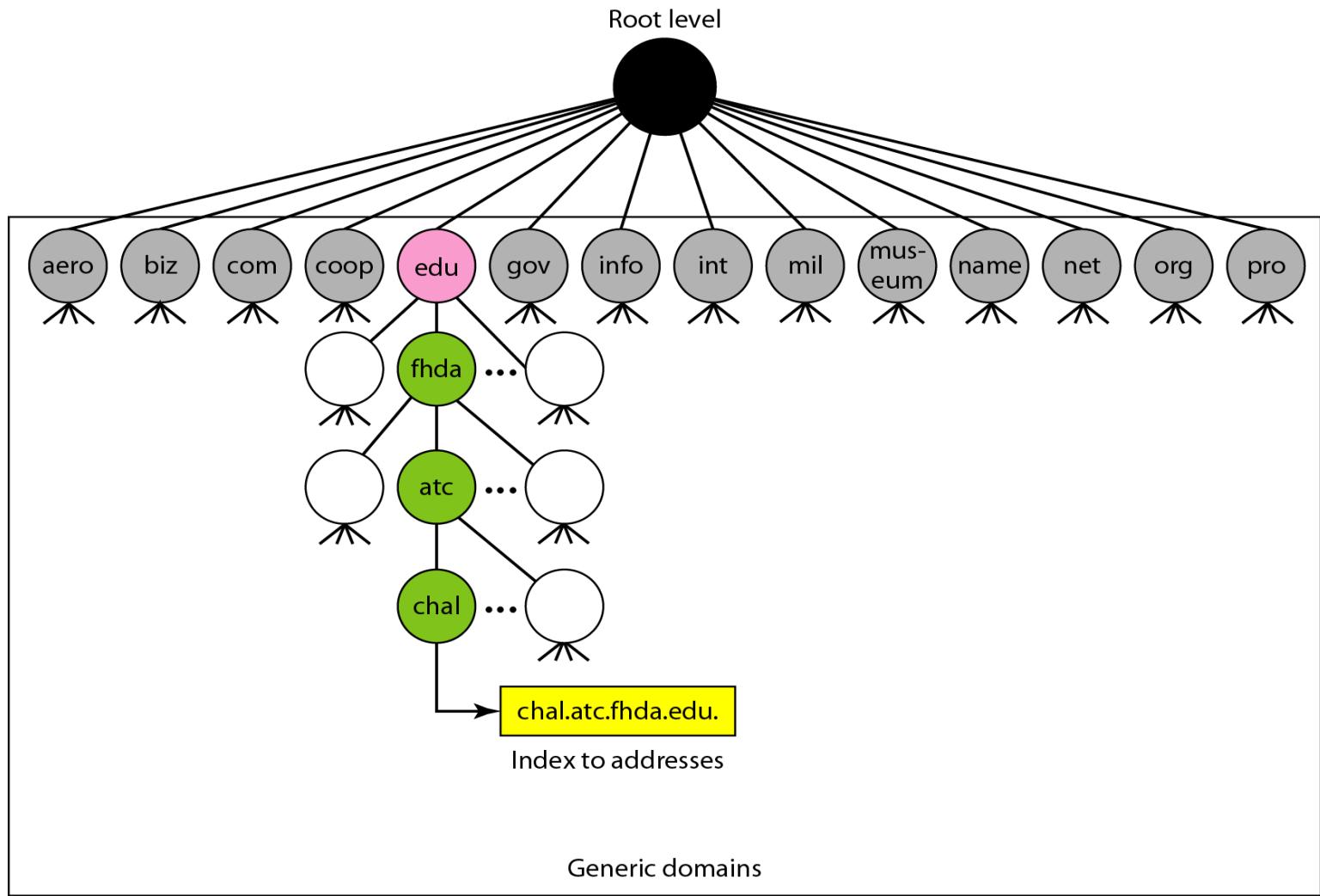
## DNS IN THE INTERNET

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



## 1 Generic Domains

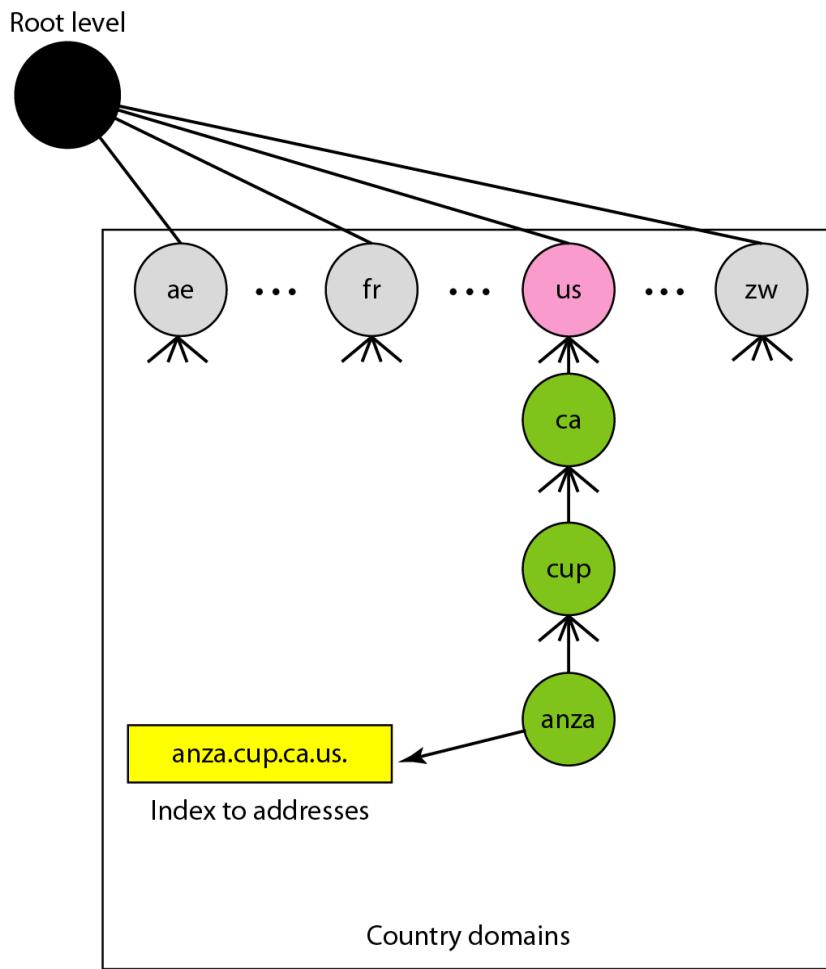
The generic domains define registered hosts according to their generic behavior. Each node in the tree defines a domain, which is an index to the domain name space database



<i>Label</i>	<i>Description</i>
<b>aero</b>	Airlines and aerospace companies
<b>biz</b>	Businesses or firms (similar to “com”)
<b>com</b>	Commercial organizations
<b>coop</b>	Cooperative business organizations
<b>edu</b>	Educational institutions
<b>gov</b>	Government institutions
<b>info</b>	Information service providers
<b>int</b>	International organizations
<b>mil</b>	Military groups
<b>museum</b>	Museums and other nonprofit organizations
<b>name</b>	Personal names (individuals)
<b>net</b>	Network support centers
<b>org</b>	Nonprofit organizations
<b>pro</b>	Professional individual organizations

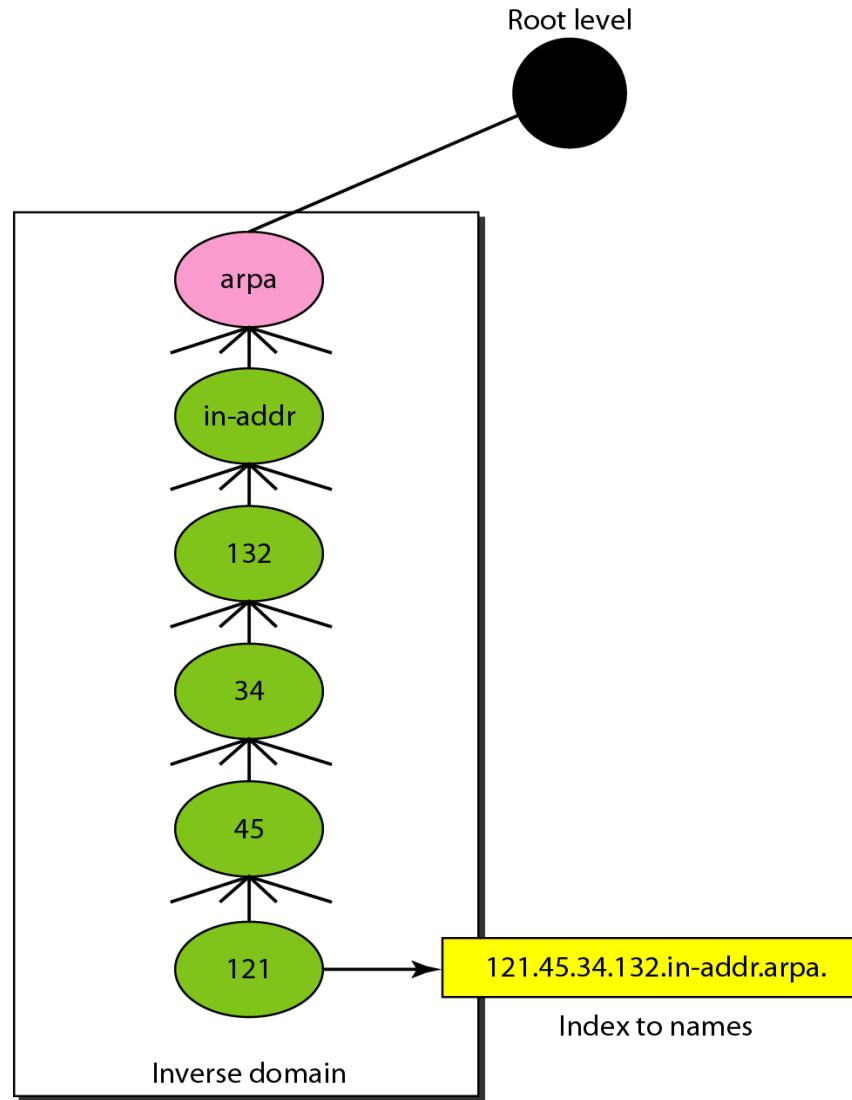
## 2 Country Domains

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.).



### 3 Inverse Domain

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



## RESOLUTION

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

### **1 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.

### **2 Mapping Names to Addresses**

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

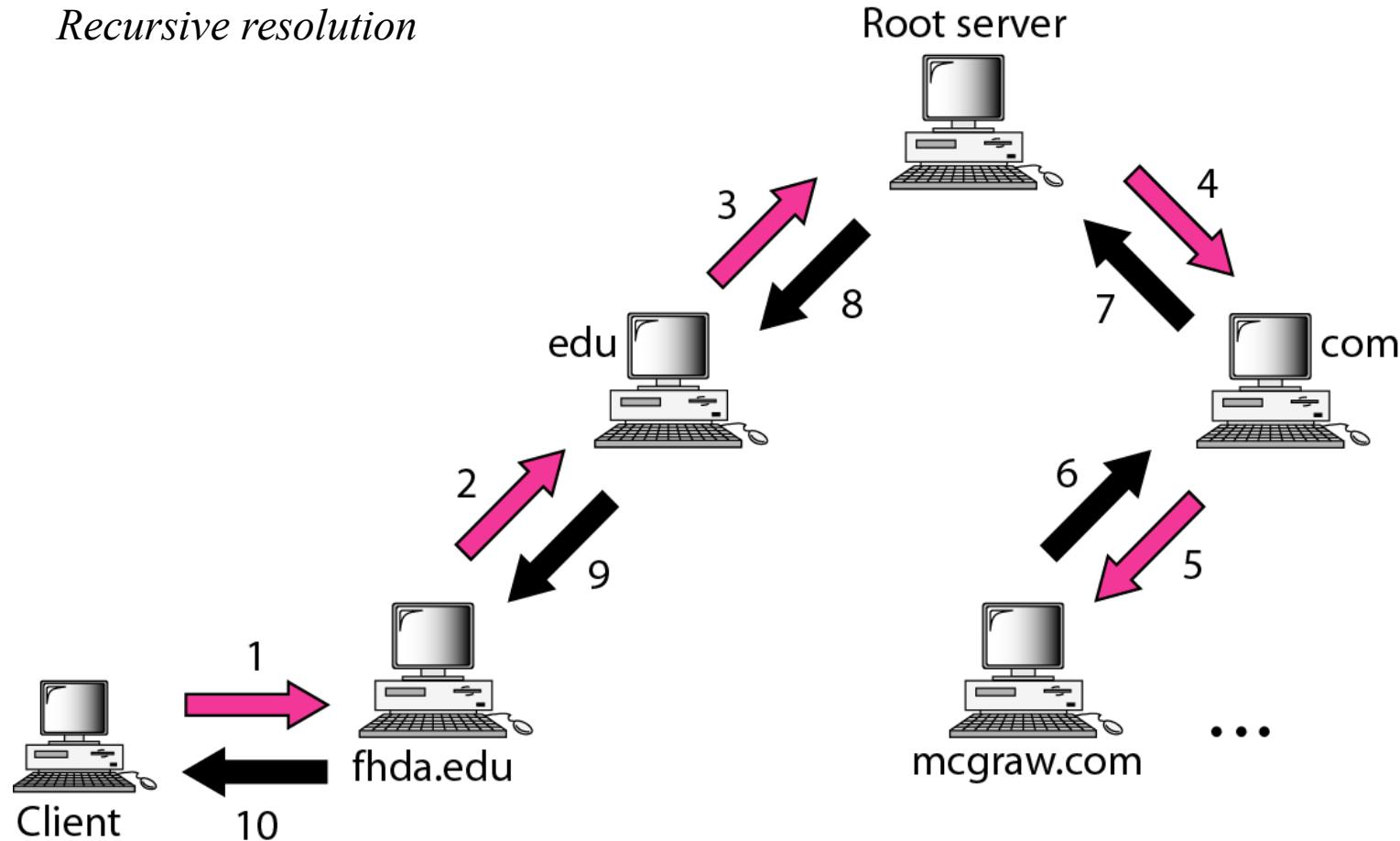
### **3 Mapping Addresses to Names**

To answer queries of this kind, DNS uses the inverse domain

### **4 Recursive Resolution**

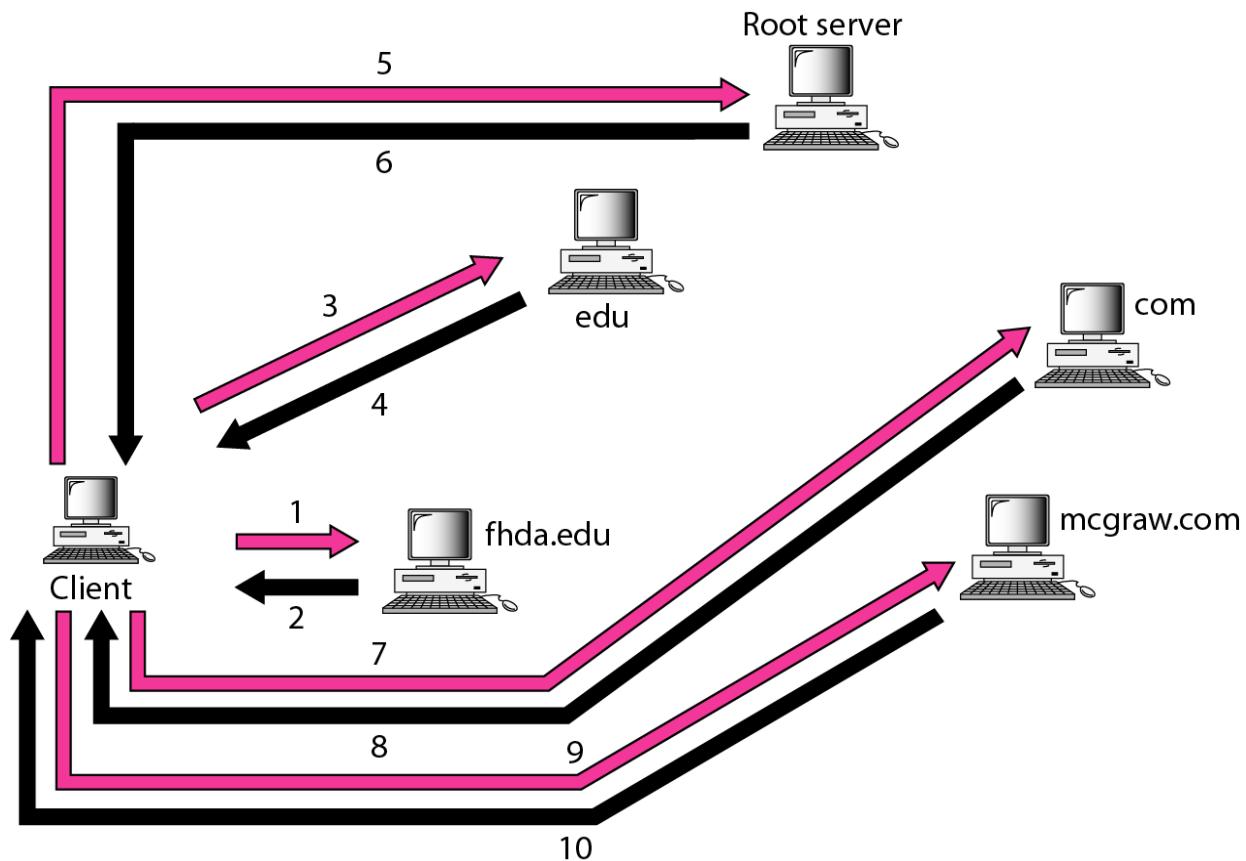
The client (resolver) can ask for a recursive answer from a name server. This means that the resolver expects the server to supply the final answer.. When the query is finally resolved, the response travels back until it finally reaches the requesting client. This is called recursive resolution and is shown in FIG

*Recursive resolution*



## 5 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



## 6 Caching

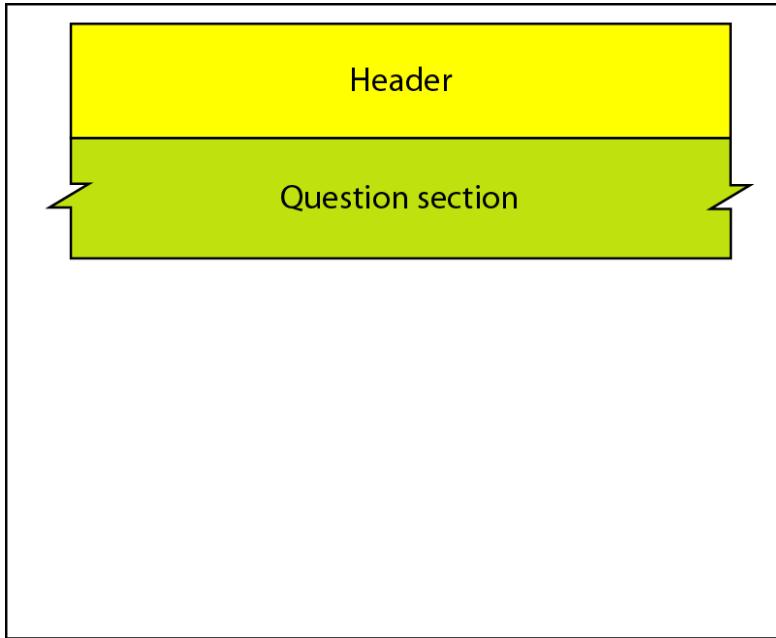
Each time a server receives a query for a name that is not in its domain, it needs to search its database for a server IP address. Reduction of this search time would increase efficiency. DNS handles this with a mechanism called caching

## DNS MESSAGES

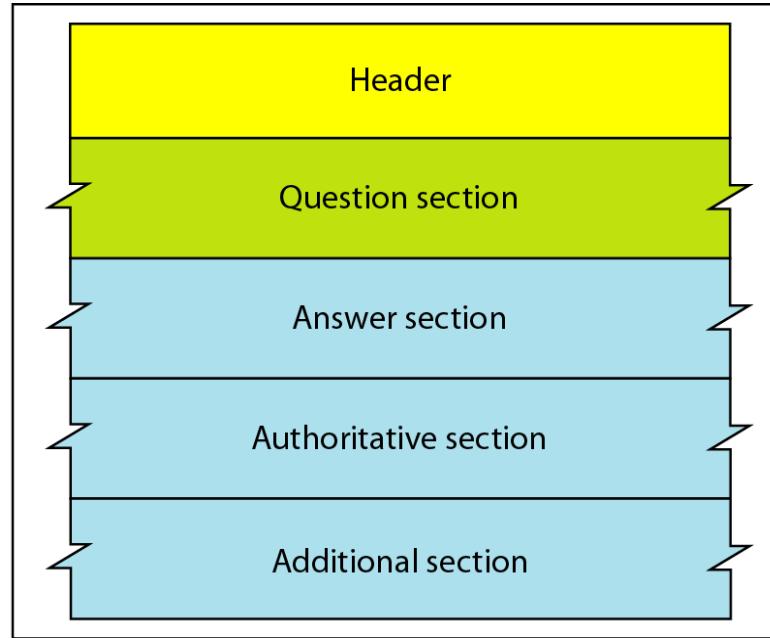
DNS has two types of messages: query and response. Both types have the same format.

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



a. Query



b. Response

## Header

Both query and response messages have the same header format with some fields set to zero for the query messages. The header is 12 bytes,

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)

## **TYPES OF RECORDS**

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.

### **Question Record**

A question record is used by the client to get information from a server..

### **Resource Record**

Each domain name (each node on the tree) is associated with a record called the resource record. The server database consists of resource records. Resource records are also what is returned by the server to the client.

## **REGISTRARS**

How are new domains added to DNS? This is done through a registrar, a commercial entity accredited by ICANN. A registrar first verifies that the requested domain name is unique and then enters it into the DNS database. A fee is charged. Today, there are many registrars; their names and addresses can be found at <http://www.intenic.net>

## **DYNAMIC DOMAIN NAME SYSTEM (DDNS)**

In DNS, when there is a change, such as adding a new host, removing a host, or changing an IP address, the change must be made to the DNS master file. The size of today's Internet does not allow for this kind of manual operation.

The DNS master file must be updated dynamically. The Dynamic Domain Name System (DDNS) therefore was devised to respond to this need.

## **ENCAPSULATION**

DNS can use either UDP or TCP. In both cases the well-known port used by the server is port 53.

# **COMPUTER NETWORKS**

**FIFTH EDITION**

## **PROBLEM SOLUTIONS**

**ANDREW S. TANENBAUM**

*Vrije Universiteit  
Amsterdam, The Netherlands*

and

**DAVID WETHERALL**

*University of Washington  
Seattle, WA*

**PRENTICE HALL**

*Upper Saddle River, NJ*



**SOLUTIONS TO CHAPTER 1 PROBLEMS**

1. The dog can carry 21 gigabytes, or 168 gigabits. A speed of 18 km/hour equals 0.005 km/sec. The time to travel distance  $x$  km is  $x/0.005 = 200x$  sec, yielding a data rate of  $168/200x$  Gbps or  $840/x$  Mbps. For  $x < 5.6$  km, the dog has a higher rate than the communication line.
  - (i) If dog's speed is doubled, maximum value of  $x$  is also doubled.
  - (ii) If tape capacity is doubled, value of  $x$  is also doubled.
  - (iii) If data rate of the transmission line is doubled, value of  $x$  is halved.
2. The LAN model can be grown incrementally. If the LAN is just a long cable, it cannot be brought down by a single failure (if the servers are replicated). It is probably cheaper. It provides more computing power and better interactive interfaces.
3. A transcontinental fiber link might have many gigabits/sec of bandwidth, but the latency will also be high due to the speed of light propagation over thousands of kilometers. In contrast, a 56-kbps modem calling a computer in the same building has low bandwidth and low latency.
4. A uniform delivery time is needed for voice as well as video, so the amount of jitter in the network is important. This could be expressed as the standard deviation of the delivery time. Having short delay but large variability is actually worse than a somewhat longer delay and low variability. For financial transaction traffic, reliability and security are very important.
5. No. The speed of propagation is 200,000 km/sec or 200 meters/ $\mu$ sec. In 10  $\mu$ sec the signal travels 2 km. Thus, each switch adds the equivalent of 2 km of extra cable. If the client and server are separated by 5000 km, traversing even 50 switches adds only 100 km to the total path, which is only 2%. Thus, switching delay is not a major factor under these circumstances.
6. The request has to go up and down, and the response has to go up and down. The total path length traversed is thus 160,000 km. The speed of light in air and vacuum is 300,000 km/sec, so the propagation delay alone is  $160,000/300,000$  sec or about 533 msec.
7. There is obviously no single correct answer here, but the following points seem relevant. The present system has a great deal of inertia (checks and balances) built into it. This inertia may serve to keep the legal, economic, and social systems from being turned upside down every time a different party comes to power. Also, many people hold strong opinions on controversial social issues, without really knowing the facts of the matter. Allowing poorly reasoned opinions be to written into law may be undesirable. The potential

effects of advertising campaigns by special interest groups of one kind or another also have to be considered. Another major issue is security. A lot of people might worry about some 14-year kid hacking the system and falsifying the results.

8. Call the routers  $A$ ,  $B$ ,  $C$ ,  $D$ , and  $E$ . There are ten potential lines:  $AB$ ,  $AC$ ,  $AD$ ,  $AE$ ,  $BC$ ,  $BD$ ,  $BE$ ,  $CD$ ,  $CE$ , and  $DE$ . Each of these has four possibilities (three speeds or no line), so the total number of topologies is  $4^{10} = 1,048,576$ . At 100 ms each, it takes 104,857.6 sec, or slightly more than 29 hours to inspect them all.
9. Distinguish  $n + 2$  events. Events 1 through  $n$  consist of the corresponding host successfully attempting to use the channel, i.e., without a collision. The probability of each of these events is  $p(1 - p)^{n-1}$ . Event  $n + 1$  is an idle channel, with probability  $(1 - p)^n$ . Event  $n + 2$  is a collision. Since these  $n + 2$  events are exhaustive, their probabilities must sum to unity. The probability of a collision, which is equal to the fraction of slots wasted, is then just  $1 - np(1 - p)^{n-1} - (1 - p)^n$ .
10. Among other reasons for using layered protocols, using them leads to breaking up the design problem into smaller, more manageable pieces, and layering means that protocols can be changed without affecting higher or lower ones. One possible disadvantage is the performance of a layered system is likely to be worse than the performance of a monolithic system, although it is extremely difficult to implement and manage a monolithic system.
11. In the ISO protocol model, physical communication takes place only in the lowest layer, not in every layer.
12. Message and byte streams are different. In a message stream, the network keeps track of message boundaries. In a byte stream, it does not. For example, suppose a process writes 1024 bytes to a connection and then a little later writes another 1024 bytes. The receiver then does a read for 2048 bytes. With a message stream, the receiver will get two messages, of 1024 bytes each. With a byte stream, the message boundaries do not count and the receiver will get the full 2048 bytes as a single unit. The fact that there were originally two distinct messages is lost.
13. Negotiation has to do with getting both sides to agree on some parameters or values to be used during the communication. Maximum packet size is one example, but there are many others.
14. The service shown is the service offered by layer  $k$  to layer  $k + 1$ . Another service that must be present is below layer  $k$ , namely, the service offered to layer  $k$  by the underlying layer  $k - 1$ .

- 15.** The probability,  $P_k$ , of a frame requiring exactly  $k$  transmissions is the probability of the first  $k - 1$  attempts failing,  $p^{k-1}$ , times the probability of the  $k$ -th transmission succeeding,  $(1 - p)$ . The mean number of transmission is then just

$$\sum_{k=1}^{\infty} kP_k = \sum_{k=1}^{\infty} k(1-p)p^{k-1} = \frac{1}{1-p}$$

- 16.** With  $n$  layers and  $h$  bytes added per layer, the total number of header bytes per message is  $hn$ , so the space wasted on headers is  $hn$ . The total message size is  $M + nh$ , so the fraction of bandwidth wasted on headers is  $hn/(M + hn)$ .
- 17.** TCP is connection oriented, whereas UDP is a connectionless service.
- 18.** The two nodes in the upper-right corner can be disconnected from the rest by three bombs knocking out the three nodes to which they are connected. The system can withstand the loss of any two nodes.
- 19.** Doubling every 18 months means a factor of four gain in 3 years. In 9 years, the gain is then  $4^3$  or 64, leading to 38.4 billion hosts. That sounds like a lot, but if every television, cellphone, camera, car, and appliance in the world is online, maybe it is plausible. The average person may have dozens of hosts by then.
- 20.** If the network tends to lose packets, it is better to acknowledge each one separately, so the lost packets can be retransmitted. On the other hand, if the network is highly reliable, sending one acknowledgement at the end of the entire transfer saves bandwidth in the normal case (but requires the entire file to be retransmitted if even a single packet is lost).
- 21.** Having mobile phone operators know the location of users lets the operators learn much personal information about users, such as where they sleep, work, travel and shop. This information might be sold to others or stolen; it could let the government monitor citizens. On the other hand, knowing the location of the user lets the operator send help to the right place in an emergency. It might also be used to deter fraud, since a person who claims to be you will usually be near your mobile phone.
- 22.** The speed of light in coax is about 200,000 km/sec, which is 200 meters/ $\mu$ sec. At 10 Mbps, it takes 0.1  $\mu$ sec to transmit a bit. Thus, the bit lasts 0.1  $\mu$ sec in time, during which it propagates 20 meters. Thus, a bit is 20 meters long here.
- 23.** The image is  $1600 \times 1200 \times 3$  bytes or 5,760,000 bytes. This is 46,080,000 bits. At 56,000 bits/sec, it takes about 822.857 sec. At 1,000,000 bits/sec, it takes 46.080 sec. At 10,000,000 bits/sec, it takes 4.608 sec. At 100,000,000

- bits/sec, it takes about 0.461 sec. At 1,000,000,000 bits/sec it takes about 46 msec.
24. Think about the hidden terminal problem. Imagine a wireless network of five stations,  $A$  through  $E$ , such that each one is in range of only its immediate neighbors. Then  $A$  can talk to  $B$  at the same time  $D$  is talking to  $E$ . Wireless networks have potential parallelism, and in this way differ from Ethernet.
  25. One advantage is that if everyone uses the standard, everyone can talk to everyone. Another advantage is that widespread use of any standard will give it economies of scale, as with VLSI chips. A disadvantage is that the political compromises necessary to achieve standardization frequently lead to poor standards. Another disadvantage is that once a standard has been widely adopted, it is difficult to change,, even if new and better techniques or methods are discovered. Also, by the time it has been accepted, it may be obsolete.
  26. There are many examples, of course. Some systems for which there is international standardization include compact disc players and their discs, digital cameras and their storage cards, and automated teller machines and bank cards. Areas where such international standardization is lacking include VCRs and videotapes (NTSC VHS in the U.S., PAL VHS in parts of Europe, SECAM VHS in other countries), portable telephones, lamps and lightbulbs (different voltages in different countries), electrical sockets and appliance plugs (every country does it differently), photocopiers and paper (8.5 x 11 inches in the U.S., A4 everywhere else), nuts and bolts (English versus metric pitch), etc.
  27. This has no impact on the operations at layers k-1 or k+1.
  28. There is no impact at layer k-1, but operations in k+1 have to be reimplemented.
  29. One reason is request or response messages may get corrupted or lost during transmission. Another reason is the processing unit in the satellite may get overloaded processing several requests from different clients.
  30. Small-sized cells result in large header-to-payload overhead. Fixed-size cells result in wastage of unused bytes in the payload.

## SOLUTIONS TO CHAPTER 2 PROBLEMS

1.  $a_n = \frac{-1}{\pi n}$ ,  $b_n = 0$ ,  $c = 1$ .

2. A noiseless channel can carry an arbitrarily large amount of information, no matter how often it is sampled. Just send a lot of data per sample. For the 4-kHz channel, make 8000 samples/sec. If each sample is 16 bits, the channel can send 128 kbps. If each sample is 1024 bits, the channel can send 8.2 Mbps. The key word here is “noiseless.” With a normal 4 kHz channel, the Shannon limit would not allow this. A signal-to-noise ratio of 30 dB means  $S/N = 1000$ . So, the Shannon limit is about 39.86 kbps.
3. Using the Nyquist theorem, we can sample 12 million times/sec. Four-level signals provide 2 bits per sample, for a total data rate of 24 Mbps.
4. A signal-to-noise ratio of 20 dB means  $S/N = 100$ . Since  $\log_2 101$  is about 6.658, the Shannon limit is about 19.975 kbps. The Nyquist limit is 6 kbps. The bottleneck is therefore the Nyquist limit, giving a maximum channel capacity of 6 kbps.
5. To send a T1 signal we need  $H \log_2(1 + S/N) = 1.544 \times 10^6$  with  $H = 50,000$ . This yields  $S/N = 2^{30} - 1$ , which is about 93 dB.
6. Fiber has many advantages over copper. It can handle much higher bandwidth than copper. It is not affected by power surges, electromagnetic interference, power failures, or corrosive chemicals in the air. It does not leak light and is quite difficult to tap. Finally, it is thin and lightweight, resulting in much lower installation costs. There are some downsides of using fiber over copper. First, it can be damaged easily by being bent too much. Second, optical communication is unidirectional, thus requiring either two fibers or two frequency bands on one fiber for two-way communication. Finally, fiber interfaces cost more than electrical interfaces.
7. Use  $\Delta f = c \Delta\lambda / \lambda^2$  with  $\Delta\lambda = 10^{-7}$  meters and  $\lambda = 10^{-6}$  meters. This gives a bandwidth ( $\Delta f$ ) of 30,000 GHz.
8. The data rate is  $2560 \times 1600 \times 24 \times 60$  bps, which is 5898 Mbps. For simplicity, let us assume 1 bps per Hz. From Eq. (2-3) we get  $\Delta\lambda = \lambda^2 \Delta f / c$ . We have  $\Delta f = 5.898 \times 10^9$ , so  $\Delta\lambda = 3.3 \times 10^{-5}$  microns. The range of wavelengths used is very short.
9. The Nyquist theorem is a property of mathematics and has nothing to do with technology. It says that if you have a function whose Fourier spectrum does not contain any sines or cosines above  $f$ , by sampling the function at a frequency of  $2f$  you capture all the information there is. Thus, the Nyquist theorem is true for all media.
10. Start with  $\lambda f = c$ . We know that  $c$  is  $3 \times 10^8$  m/s. For  $\lambda = 1$  cm, we get 30 GHz. For  $\lambda = 5$  m, we get 60 MHz. Thus, the band covered is 60 MHz to 30 GHz.

11. If the beam is off by 1 mm at the end, it misses the detector. This amounts to a triangle with base 100 m and height 0.001 m. The angle is one whose tangent is thus 0.00001. This angle is about 0.00057 degrees.
12. With 66/6 or 11 satellites per necklace, every 90 minutes 11 satellites pass overhead. This means there is a transit every 491 seconds. Thus, there will be a handoff about every 8 minutes and 11 seconds.
13. Transit time =  $2 \times (\text{Altitude}/\text{Speed of light})$ . The speed of light in air or vacuum is 300,000 km/sec. This evaluates to 239 msec for GEO, 120 msec for MEO, and 5 msec for LEO satellites.
14. The call travels from the North Pole to the satellite directly overhead, and then transits through four other satellites to reach the satellite directly above the South Pole. Down it goes down to earth to the South Pole. The total distance traveled is  $2 \times 750 + 0.5 \times \text{circumference}$  at altitude 750 km. Circumference at altitude 750 km is  $2 \times \pi \times (6371 + 750) = 44,720$  km. So, the total distance traveled is 23,860 km. Time to travel this distance =  $23860/300000 = 79.5$  msec. In addition, switching occurs at six satellites. So, the total switching time is 60 usec. So, the total latency is about 79.56 msec.
15. In NRZ, the signal completes a cycle at most every 2 bits (alternating 1s and 0s). So, the minimum bandwidth need to achieve  $B$  bits/sec data rate is  $B/2$  Hz. In MLT-3, the signal completes a cycle at most every 4 bits (a sequence of 1s), thus requiring at least  $B/4$  Hz to achieve  $B$  bits/sec data rate. Finally, in Manchester encoding, the signal completes a cycle in every bit, thus requiring at least  $B$  Hz to achieve  $B$  bits/sec data rate.
16. Since 4B/5B encoding uses NRZI, there is a signal transition every time a 1 is sent. Furthermore, the 4B/5B mapping (see Figure 2-21) ensures that a sequence of consecutive 0s cannot be longer than 3. Thus, in the worst case, the transmitted bits will have a sequence 10001, resulting in a signal transition in 4 bits.
17. The number of area codes was  $8 \times 2 \times 10$ , which is 160. The number of prefixes was  $8 \times 8 \times 10$ , or 640. Thus, the number of end offices was limited to 102,400. This limit is not a problem.
18. Each telephone makes 0.5 calls/hour at 6 minutes each. Thus, a telephone occupies a circuit for 3 minutes/hour. Twenty telephones can share a circuit, although having the load be close to 100% ( $\rho = 1$  in queuing terms) implies very long wait times. Since 10% of the calls are long distance, it takes 200 telephones to occupy a long-distance circuit full time. The interoffice trunk has  $1,000,000/4000 = 250$  circuits multiplexed onto it. With 200 telephones per circuit, an end office can support  $200 \times 250 = 50,000$  telephones. Supporting such a large number of telephones may result in significantly long

wait times. For example, if 5,000 (10% of 50,000) users decide to make a long-distance telephone call at the same time and each call lasts 3 minutes, the worst-case wait time will be 57 minutes. This will clearly result in unhappy customers.

19. The cross-section of each strand of a twisted pair is  $\pi/4$  square mm. A 10-km length of this material, with two strands per pair has a volume of  $2\pi/4 \times 10^{-2} \text{ m}^3$ . This volume is about  $15,708 \text{ cm}^3$ . With a specific gravity of 9.0, each local loop has a mass of 141 kg. The phone company thus owns  $1.4 \times 10^9 \text{ kg}$  of copper. At \$6 each, the copper is worth about 8.4 billion dollars.
20. Like a single railroad track, it is half duplex. Oil can flow in either direction, but not both ways at once. A river is an example of a simplex connection while a walkie-talkie is another example of a half-duplex connection.
21. Traditionally, bits have been sent over the line without any error-correcting scheme in the physical layer. The presence of a CPU in each modem makes it possible to include an error-correcting code in layer 1 to greatly reduce the effective error rate seen by layer 2. The error handling by the modems can be done totally transparently to layer 2. Many modems now have built-in error correction. While this significantly reduces the effective error rate seen at layer 2, errors at layer 2 are still possible. This can happen, for example, because of loss of data as it is transferred from layer 1 to layer 2 due lack of buffer space.
22. There are four legal values per baud, so the bit rate is twice the baud rate. At 1200 baud, the data rate is 2400 bps.
23. Since there are 32 symbols, 5 bits can be encoded. At 1200 baud, this provides  $5 \times 1200 = 6000 \text{ bps}$ .
24. Two, one for upstream and one for downstream. The modulation scheme itself just uses amplitude and phase. The frequency is not modulated.
25. There are 10 4000 Hz signals. We need nine guard bands to avoid any interference. The minimum bandwidth required is  $4000 \times 10 + 400 \times 9 = 43,600 \text{ Hz}$ .
26. A sampling time of 125  $\mu\text{sec}$  corresponds to 8000 samples per second. According to the Nyquist theorem, this is the sampling frequency needed to capture all the information in a 4-kHz channel, such as a telephone channel. (Actually the nominal bandwidth is somewhat less, but the cutoff is not sharp.)
27. The end users get  $7 \times 24 = 168$  of the 193 bits in a frame. The overhead is therefore  $25/193 = 13\%$ . From Figure 2-41, percent overhead in OC-1 is  $(51.84 - 49.536)/51.84 = 3.63\%$ . In OC-768, percent overhead is  $(39813.12 -$

$$38043.648)/39813.12 = 4.44\%.$$

- 28.** In both cases 8000 samples/sec are possible. With dabit encoding, 2 bits are sent per sample. With T1, 7 bits are sent per period. The respective data rates are 16 kbps and 56 kbps.
- 29.** Ten frames. The probability of some random pattern being 0101010101 (on a digital channel) is 1/1024.
- 30.** A coder accepts an arbitrary analog signal and generates a digital signal from it. A demodulator accepts a modulated sine wave only and generates a digital signal.
- 31.** A drift rate of  $10^{-9}$  means 1 second in  $10^9$  seconds or 1 nsec per second. At OC-1 speed, say, 50 Mbps, for simplicity, a bit lasts for 20 nsec. This means it takes only 20 seconds for the clock to drift off by 1 bit. Consequently, the clocks must be continuously synchronized to keep them from getting too far apart. Certainly every 10 sec, preferably much more often.
- 32.** The lowest bandwidth link (1 Mbps) is the bottleneck.  
 One-way latency =  $4 \times (35800/300000) = 480$  msec.  
 Total time =  $1.2 + 233/220 + 0.48 = 8193.68$  sec.
- 33.** Again, the lowest-bandwidth link is the bottleneck.  
 Number of packets =  $230/216 = 214$ .  
 One way latency =  $480 + 3 \times 0.001 = 480.003$  msec.  
 Total bits transmitted =  $233 + 214 * 28 = 233 + 222$ .  
 Total time =  $(233 + 222) / 220 + 0.48 = 8196.48$  sec.
- 34.** Of the 90 columns, 86 are available for user data in OC-1. Thus, the user capacity is  $86 \times 9 = 774$  bytes/frame. With 8 bits/byte, 8000 frames/sec, and 3 OC-1 carriers multiplexed together, the total user capacity is  $3 \times 774 \times 8 \times 8000$ , or 148.608 Mbps. For an OC-3072 line:  
 Gross data rate =  $51.84 \times 3072 = 159252.48$  Mbps.  
 SPE data rate =  $50.112 \times 3072 = 153944.064$  Mbps.  
 User data rate =  $49.536 \times 3072 = 152174.592$  Mbps.
- 35.** VT1.5 can accommodate  $8000 \text{ frames/sec} \times 3 \text{ columns} \times 9 \text{ rows} \times 8 \text{ bits} = 1.728$  Mbps. It can be used to accommodate DS-1. VT2 can accommodate  $8000 \text{ frames/sec} \times 4 \text{ columns} \times 9 \text{ rows} \times 8 \text{ bits} = 2.304$  Mbps. It can be used to accommodate European CEPT-1 service. VT6 can accommodate  $8000 \text{ frames/sec} \times 12 \text{ columns} \times 9 \text{ rows} \times 8 \text{ bits} = 6.912$  Mbps. It can be used to accommodate DS-2 service.
- 36.** The OC-12c frames are  $12 \times 90 = 1080$  columns of 9 rows. Of these,  $12 \times 3 = 36$  columns are taken up by line and section overhead. This leaves an SPE of 1044 columns. One SPE column is taken up by path overhead,

leaving 1043 columns for user data. Since each column holds 9 bytes of 8 bits, an OC-12c frame holds 75,096 user data bits. With 8000 frames/sec, the user data rate is 600.768 Mbps.

37. The three networks have the following properties:

Star: best case = 2, average case = 2, worst case = 2.

Ring: best case = 1, average case =  $n/4$ , worst case =  $n/2$ .

Full interconnect: best case = 1, average case = 1, worst case = 1.

38. With circuit switching, at  $t = s$  the circuit is set up, at  $t = s + x/b$  the last bit is sent, at  $t = s + x/b + kd$  the message arrives. With packet switching, the last bit is sent at  $t = x/b$ . To get to the final destination, the last packet must be retransmitted  $k - 1$  times by intermediate routers, with each retransmission taking  $p/b$  sec, so the total delay is  $x/b + (k - 1)p/b + kd$ . Packet switching is faster if  $s > (k - 1)p/b$ . In addition to the faster transmission under these conditions, packet switching is preferable when fault-tolerant transmission in the presence of switch failures is desired.
39. The total number of packets needed is  $x/p$ , so the total data + header traffic is  $(p + h)x/p$  bits. The source requires  $(p + h)x/pb$  sec to transmit these bits. The retransmissions of the last packet by the intermediate routers take up a total of  $(k - 1)(p + h)/b$  sec. Adding up the time for the source to send all the bits, plus the time for the routers to carry the last packet to the destination, thus clearing the pipeline, we get a total time of  $(p + h)x/pb + (p + h)(k - 1)/b$  sec. Minimizing this quantity with respect to  $p$ , we find  $p = \sqrt{hx/(k - 1)}$ .
40. Each cell has six neighbors. If the central cell uses frequency group A, its six neighbors can use B, C, B, C, B, and C, respectively. In other words, only three unique cells are needed. Consequently, each cell can have 280 frequencies.
41. First, initial deployment simply placed cells in regions where there was a high density of human or vehicle population. Once they were there, the operators often did not want to go to the trouble of moving them. Second, antennas are typically placed on tall buildings or mountains. Depending on the exact location of such a structure, the area covered by a cell may be irregular due to obstacles near the transmitter. Third, some communities or property owners do not allow building a tower at a location where the center of a cell falls. In such cases, directional antennas are placed at a location not at the cell center. In the case of regular shapes, there is typically a buffer two cells wide where a frequency assigned to a cell is not reused in order to provide good separation and low interference. When the shapes of cells are irregular, the number of cells separating two cells that are using the same frequency is variable, depending on the width of the intermediate cells. This makes frequency

assignment much more complicated.

42. If we assume that each microcell is a circle 100 m in diameter, each cell has an area of  $2500\pi$ . If we take the area of San Francisco,  $1.2 \times 10^8 \text{ m}^2$ , and divide it by the area of 1 microcell, we get 15,279 microcells. Of course, it is impossible to tile the plane with circles (and San Francisco is decidedly three-dimensional), but with 20,000 microcells we could probably do the job.
43. Frequencies cannot be reused in adjacent cells, so when a user moves from one cell to another, a new frequency must be allocated for the call. If a user moves into a cell, all of whose frequencies are currently in use, the user's call must be terminated.
44. The result is obtained by negating each of  $A$ ,  $B$ , and  $C$  and then adding the three chip sequences. Alternatively, the three can be added and then negated. The result is  $(+3 +1 +1 -1 -3 -1 -1 +1)$ .
45. When two elements match, their product is  $+1$ . When they do not match, their product is  $-1$ . To make the sum 0, there must be as many matches as mismatches. Thus, two chip sequences are orthogonal if exactly half of the corresponding elements match and exactly half do not match.
46. Just compute the four normalized inner products:

$$\begin{aligned} (-1 +1 -3 +1 -1 -3 +1 +1) \bullet (-1 -1 -1 +1 +1 -1 +1 +1)/8 &= 1 \\ (-1 +1 -3 +1 -1 -3 +1 +1) \bullet (-1 -1 +1 -1 +1 +1 +1 -1)/8 &= -1 \\ (-1 +1 -3 +1 -1 -3 +1 +1) \bullet (-1 +1 -1 +1 +1 +1 -1 -1)/8 &= 0 \\ (-1 +1 -3 +1 -1 -3 +1 +1) \bullet (-1 +1 -1 -1 -1 +1 -1)/8 &= 1 \end{aligned}$$

The result is that  $A$  and  $D$  sent 1 bits,  $B$  sent a 0 bit, and  $C$  was silent.

47. Here are the chip sequences:

$$\begin{aligned} (+1 +1 +1 +1 +1 +1 +1 +1) \\ (+1 -1 +1 -1 +1 -1 +1 -1) \\ (+1 +1 -1 -1 +1, +1 -1 -1) \\ (+1 -1 -1 +1 +1 -1 -1 +1) \end{aligned}$$

48. Ignoring speech compression, a digital PCM telephone needs 64 kbps. If we divide 10 Gbps by 64 kbps we get 156,250 houses per cable. Current systems have hundreds of houses per cable.
49. A 2-Mbps downstream bandwidth guarantee to each house implies at most 50 houses per coaxial cable. Thus, the cable company will need to split up the existing cable into 100 coaxial cables and connect each of them directly to a fiber node.

50. The upstream bandwidth is 37 MHz. Using QPSK with 2 bits/Hz, we get 74 Mbps upstream. Downstream we have 200 MHz. Using QAM-64, this is 1200 Mbps. Using QAM-256, this is 1600 Mbps.
51. The downstream data rate of a cable user is the smaller of the downstream cable bandwidth and the bandwidth of the communication medium between the cable modem and the user's PC. If the downstream cable channel works at 27 Mbps, the downstream data rate of the cable user will be
- 10 Mbps.
  - 27 Mbps.
  - 27 Mbps.

This is assuming that the communication medium between cable modem and the user's PC is not shared with any other user. Usually, cable operators specify 10-Mbps Ethernet because they do not want one user sucking up the entire bandwidth.

### SOLUTIONS TO CHAPTER 3 PROBLEMS

1. Since each frame has a chance of 0.8 of getting through, the chance of the whole message getting through is  $0.8^{10}$ , which is about 0.107. Call this value  $p$ . The expected number of transmissions for an entire message is then

$$E = \sum_{i=1}^{\infty} ip(1-p)^{i-1} = p \sum_{i=1}^{\infty} i(1-p)^{i-1}$$

To reduce this, use the well-known formula for the sum of an infinite geometric series,

$$S = \sum_{i=1}^{\infty} \alpha^i = \frac{1}{1-\alpha}$$

Differentiate both sides with respect to  $\alpha$  to get

$$S' = \sum_{i=1}^{\infty} i\alpha^{i-1} = \frac{1}{(1-\alpha)^2}$$

Now use  $\alpha = 1 - p$  to get  $E = 1/p$ . Thus, it takes an average of 1/0.107, or about 9.3 transmissions.

2. The solution is
- 00000100 01000111 11100011 11100000 01111110
  - 01111110 01000111 11100011 11100000 11100000 01111110

0111110

(c) 01111110 01000111 110100011 111000000 01111010 01111110

3. After stuffing, we get A B ESC ESC C ESC ESC ESC FLAG ESC FLAG D.
4. The maximum overhead occurs when the payload consists of only ESC and FLAG bytes. In this case, there will be 100% overhead.
5. If you could always count on an endless stream of frames, one flag byte might be enough. But what if a frame ends (with a flag byte) and there are no new frames for 15 minutes? How will the receiver know that the next byte is actually the start of a new frame and not just noise on the line? The protocol is much simpler with starting and ending flag bytes.
6. The output is 011110111110011111010.
7. If the propagation delay is very long, as in the case of a space probe on or near Mars or Venus, forward error correction is indicated. It is also appropriate in a military situation in which the receiver does not want to disclose its location by transmitting. If the error rate is low enough that an error-correcting code is good enough, it may also be simpler. Finally, real-time systems cannot tolerate waiting for retransmissions.
8. Making one change to any valid character cannot generate another valid character due to the nature of parity bits. Making two changes to even bits or two changes to odd bits will give another valid character, so the distance is 2.
9. Parity bits are needed at positions 1, 2, 4, 8, and 16, so messages that do not extend beyond bit 31 (including the parity bits) fit. Thus, 5 parity bits are sufficient. The bit pattern transmitted is 011010110011001110101.
10. If we number the bits from left to right starting at bit 1, in this example bit 2 (a parity bit) is incorrect. The 12-bit value transmitted (after Hamming encoding) was 0xA4F. The original 8-bit data value was 0xAF.
11. A single error will cause both the horizontal and vertical parity checks to be wrong. Two errors will also be easily detected. If they are in different rows, the row parity will catch them. If they are in the same row, the column parity will catch them. Three errors will also be detected. If they are in the same row or column, that row's or column's parity will catch them. If two errors are in the same row, the column parity of at least one of them will catch the error. If two errors are in the same column, the row parity of at least one of them will catch the error. A 4-bit error in which the four error bits lie on the four corners of a rectangle cannot be caught.
12. From Eq. (3-1), we know that 10 check bits are needed for each block in case of using Hamming code. Total bits transmitted per block is 1010 bits. In case of error detection mechanism, one parity bit is transmitted per block. Suppose

error rate is  $x$  per bit. However, a block may encounter a bit error  $1000x$  times. Every time an error is encountered, 1001 bits have to be retransmitted. So, total bits transmitted per block is  $1001 + 1000x \times 1001$  bits. For error detection and retransmission to be better,  $1001 + 1000x \times 1001 < 1010$ . So, the error rate must be less than  $9 \times 10^{-6}$ .

- 13.** Describe an error pattern as a matrix of  $n$  rows by  $k$  columns. Each of the correct bits is a 0, and each of the incorrect bits is a 1. With four errors per block, each block will have exactly four 1s. How many such blocks are there? There are  $nk$  ways to choose where to put the first 1 bit,  $nk - 1$  ways to choose the second, and so on, so the number of blocks is  $nk(nk - 1)(nk - 2)(nk - 3)$ . Undetected errors only occur when the four 1 bits are at the vertices of a rectangle. Using Cartesian coordinates, every 1 bit is at a coordinate  $(x, y)$ , where  $0 \leq x < k$  and  $0 \leq y < n$ . Suppose that the bit closest to the origin (the lower-left vertex) is at  $(p, q)$ . The number of legal rectangles is  $(k - p - 1)(n - q - 1)$ . The total number of rectangles can be found by summing this formula for all possible  $p$  and  $q$ . The probability of an undetected error is then the number of such rectangles divided by the number of ways to distribute the 4 bits:

$$\frac{\sum_{p=0}^{k-2} \sum_{q=0}^{n-2} (k-p-1)(n-q-1)}{nk(nk-1)(nk-2)(nk-3)}$$

- 14.** When the first 1 goes in, 11 comes out and  $S_1$  stores the 1. Then 0 goes in and 01 comes out, with  $S_2$  now storing a 1 and  $S_1$  storing the 0. The complete output sequence, including these initial values is 11 01 00 10 10 00 11 00.

- 15.** To obtain the checksum, we need to calculate the ones complement sum of words, which is same as sum modulo  $2^4$  and adding any overflow of high order bits back into low-order bits:

$$\begin{aligned} 0011 + 1010 &= 1101 \\ 1101 + 1100 &= 1001 + 1 = 1010 \\ 1010 + 1001 &= 0011 + 1 = 1100. \end{aligned}$$

So, the Internet checksum is 1100.

- 16.** The remainder is  $x^2 + x + 1$ .

- 17.** The frame is 10011101. The generator is 1001. The message after appending three zeros is 10011101000. The remainder on dividing 10011101000 by 1001 is 100. So, the actual bit string transmitted is 10011101100. The received bit stream with an error in the third bit from the left is 10111101100. Dividing this by 1001 produces a remainder of 100, which is different from 0. Thus, the receiver detects the error and can ask for a retransmission. If the

- transmitted bit stream is converted to any multiple of 1001, the error will not be detected. A trivial example is if all ones in the bit stream are inverted to zeros.
18. The CRC checksum polynomial is or degree 32, so
    - (a) Yes. CRC catches all single-bit errors.
    - (b) Yes. CRC catches all double-bit errors for any reasonably long message.
    - (c) No. CRC may not be able catch all even number of isolated bit errors.
    - (d) Yes. CRC catches all odd number of isolated bit errors.
    - (e) Yes. CRC catches all burst errors with burst lengths less than or equal to 32.
    - (f) No. CRC may not be able to catch a burst error with burst length greater than 32.
  19. Yes, it is possible. The reason is that an acknowledgement frame may arrive correctly, but after the sender's timer has expired. This can happen if the receiver gets delayed in sending the acknowledgement frame, because its CPU is overloaded processing other jobs in the system.
  20. Efficiency will be 50% when the time required to transmit the frame equals the round-trip propagation delay. At a transmission rate of 4 bits/msec, 160 bits takes 40 msec. For frame sizes above 160 bits, stop-and-wait is reasonably efficient.
  21. It can happen. Suppose that the sender transmits a frame and a garbled acknowledgement comes back quickly. The main loop will be executed a second time and a frame will be sent while the timer is still running.
  22. To operate efficiently, the sequence space (actually, the sender's window size) must be large enough to allow the transmitter to keep transmitting until the first acknowledgement has been received. The propagation time is 18 ms. At T1 speed, which is 1.536 Mbps (excluding the 1 header bit), a 64-byte frame takes 0.300 msec. Therefore, the first frame fully arrives 18.3 msec after its transmission was started. The acknowledgement takes another 18 msec to get back, plus a small (negligible) time for the acknowledgement to arrive fully. In all, this time is 36.3 msec, so the transmitter must have enough window space to keep going for 36.3 msec. A frame takes 0.3 ms, so it takes 121 frames to fill the pipe. Seven-bit sequence numbers are needed.
  23. Let the sender's window be  $(S_l, S_u)$  and the receiver's be  $(R_l, R_u)$ . Let the window size be  $W$ . The relations that must hold are:
 
$$0 \leq S_u - S_l + 1 \leq W$$

$$R_u - R_l + 1 = W$$

$$S_l \leq R_l \leq S_u + 1$$

24. The protocol would be incorrect. Suppose that 3-bit sequence numbers are in use. Consider the following scenario:

*A* just sent frame 7.

*B* gets the frame and sends a piggybacked ACK.

*A* gets the ACK and sends frames 0–6, all of which get lost.

*B* times out and retransmits its current frame, with the ACK 7.

Look at the situation at *A* when the frame with  $r.ack = 7$  arrives. The key variables are  $AckExpected = 0$ ,  $r.ack = 7$ , and  $NextFrameToSend = 7$ . The modified *between* would return *true*, causing *A* to think the lost frames were being acknowledged.

25. Yes. It might lead to deadlock. Suppose that a batch of frames arrived correctly and was accepted. The receiver would advance its window. Now suppose that all the acknowledgements were lost. The sender would eventually time out and send the first frame again. The receiver would then send a NAK. If this packet were lost, from that point on, the sender would keep timing out and sending a frame that had already been accepted, but the receiver would just ignore it. Setting the auxiliary timer results in a correct acknowledgement being sent back eventually instead, which resynchronizes.
26. It would lead to deadlock because this is the only place that incoming acknowledgements are processed. Without this code, the sender would keep timing out and never make any progress.
27. Link utilization =  $(1/(1 + 2BD))$   
 $BD$  = bandwidth-delay product / frame size  
 $delay = (9 \times 10^{10})/(3 \times 10^8) = 300$  sec  
bandwidth-delay product =  $64 \times 300 = 19.2$  Gb  
 $BD = 19200000 / 256 = 75000$   
So, link utilization is  $6.67 \times 10^{-4}\%$
28. For a send window size  $w$ , link utilization is  $w/(1 + 2BD)$ . So, for 100% link utilization,  $w = 150001$ .
29. Consider the following scenario. *A* sends 0 to *B*. *B* gets it and sends an ACK, but the ACK gets lost. *A* times out and repeats 0, but now *B* expects 1, so it sends a NAK. If *A* merely resent  $r.ack + 1$ , it would be sending frame 1, which it has not gotten yet.
30. Suppose *A* sent *B* a frame that arrived correctly, but there was no reverse traffic. After a while *A* would time out and retransmit. *B* would notice that the sequence number was incorrect, since it would be below  $FrameExpected$ . Consequently, it would send a NAK, which carries an acknowledgement number. Each frame would be sent exactly two times.

- 31.** No. This implementation fails. With  $\text{MaxSeq} = 4$ , we get  $\text{NrBufs} = 2$ . The even sequence numbers use buffer 0 and the odd ones use buffer 1. This mapping means that frames 4 and 0 both use the same buffer. Suppose that frames 0–3 are received and acknowledged. The receiver's window now contains 4 and 0. If 4 is lost and 0 arrives, it will be put in buffer 0 and  $\text{arrived}[0]$  will be set to *true*. The loop in the code for *FrameArrival* will be executed once, and an out-of-order message will be delivered to the host. This protocol requires  $\text{MaxSeq}$  to be odd to work properly. However, other implementations of sliding window protocols do not all have this property.
- 32.** Let  $t = 0$  denote the start of transmission. At  $t = 1$  msec, the first frame has been fully transmitted. At  $t = 271$  msec, the first frame has fully arrived. At  $t = 272$  msec, the frame acknowledging the first one has been fully sent. At  $t = 542$  msec, the acknowledgement-bearing frame has fully arrived. Thus, the cycle is 542 msec. A total of  $k$  frames are sent in 542 msec, for an efficiency of  $k/542$ . Hence, for
- (a)  $k = 1$ , efficiency =  $1/542 = 0.18\%$ .
  - (b)  $k = 7$ , efficiency =  $7/542 = 1.29\%$ .
  - (c)  $k = 4$ , efficiency =  $4/542 = 0.74\%$ .
- 33.** With a 50-kbps channel and 8-bit sequence numbers, the pipe is always full. The number of retransmissions per frame is about 0.01. Each good frame wastes 40 header bits, plus 1% of 4000 bits (retransmission), plus a 40-bit NAK once every 100 frames. The total overhead is 80.4 bits per 3960 data bits, giving  $80.4/(3960 + 80.4) = 1.99\%$ .
- 34.** The transmission starts at  $t = 0$ . At  $t = 4096/64000 \text{ sec} = 64$  msec, the last bit is sent. At  $t = 334$  msec, the last bit arrives at the satellite and the very short ACK is sent. At  $t = 604$  msec, the ACK arrives at the earth. The data rate here is 4096 bits in 604 msec, or about 6781 bps. With a window size of 7 frames, transmission time is 448 msec for the full window, at which time the sender has to stop. At 604 msec, the first ACK arrives and the cycle can start again. Here we have  $7 \times 4096 = 28,672$  bits in 604 msec. The data rate is 47,470.2 bps. Continuous transmission can only occur if the transmitter is still sending when the first ACK gets back at  $t = 604$  msec. In other words, if the window size is greater than 604 msec worth of transmission, it can run at full speed. For a window size of 10 or greater this condition is met, so for any window size of 10 or greater (e.g., 15 or 127) the data rate is 64 kbps.
- 35.** The propagation speed in the cable is 200,000 km/sec, or 200 km/msec, so a 100-km cable will be filled in 500  $\mu$ sec. Each T1 frame is 193 bits sent in 125  $\mu$ sec. This corresponds to four frames, or 772 bits on the cable.

36. PPP was clearly designed to be implemented in software, not in hardware as bit-stuffing protocols such as HDLC nearly always are. With a software implementation, working entirely with bytes is much simpler than working with individual bits. In addition, PPP was designed to be used with modems, and modems accept and transmit data in units of 1 byte, not 1 bit.
37. At its smallest, each frame has 2 flag bytes, 1 protocol byte, and 2 checksum bytes, for a total of 5 overhead bytes per frame. For maximum overhead, 2 flag bytes, 1 byte each for address and control, 2 bytes for protocol and 4 bytes for checksum. This totals to 10 overhead bytes.
38. The AAL5 frame will consist of 2 PPP protocol bytes, 100 PPP payload bytes, some padding bytes, and 8 trailer bytes. To make this frame size a multiple of 48, the number of padding bytes will be 34. This will result in an AAL5 frame of size 144 bytes. This can fit in three ATM cells. The first ATM cell will contain the 2 PPP protocol bytes and 46 bytes of the IP packet, the second cell will contain the next 48 bytes of the IP packet, and finally, the third ATM cell will contain the last 6 bytes of IP packet, 34 padding bytes, and 8 AAL5 trailer bytes.

### SOLUTIONS TO CHAPTER 4 PROBLEMS

1. The formula is the standard formula for Markov queueing given in Sec. 4.1.1, namely,  $T = 1/(\mu C - \lambda)$ . Here,  $C = 10^8$  and  $\mu = 10^{-4}$ , so  $T = 1/(10000 - \lambda)$  sec. For the three arrival rates, we get (a) 0.1 msec, (b) 0.11 msec, and (c) 1 msec. For case (c) we are operating a queueing system with  $\rho = \lambda/\mu C = 0.9$ , which gives the 10× delay.
2. With pure ALOHA, the usable bandwidth is  $0.184 \times 56$  kbps = 10.3 kbps. Each station requires 10 bps, so  $N = 10300/10 = 1030$  stations.
3. With pure ALOHA, transmission can start instantly. At low load, no collisions are expected so the transmission is likely to be successful. With slotted ALOHA, it has to wait for the next slot. This introduces half a slot time of delay.
4. (a) With  $G = 2$  Poisson's Law gives a probability of  $e^{-2}$ .  
 (b)  $(1 - e^{-G})^k e^{-G} = 0.135 \times 0.865^k$ .  
 (c) The expected number of transmissions is  $e^G = 7.4$ .
5. The number of transmissions is  $E = e^G$ . The  $E$  events are separated by  $E - 1$  intervals of four slots each, so the delay is  $4(e^G - 1)$ . The throughput is given by  $S = Ge^{-G}$ . Thus, we have two parametric equations, one for delay and one for throughput, both in terms of  $G$ . For each  $G$  value, it is possible to find the corresponding delay and throughput, yielding one point on the curve.

6. (a) Signal propagation speed in twin lead is  $2.46 \times 10^8$  m/sec. Signal propagation time for 2 km is  $8.13 \mu\text{sec}$ . So, the length of contention slot is 16.26  $\mu\text{sec}$ . (b) Signal propagation speed in multimode fiber is  $1.95 \times 10^8$  m/s. Signal propagation time for 40 km is  $205.13 \mu\text{sec}$ . So, the length of contention slot is 410.26  $\mu\text{sec}$ .
7. The worst case is where all stations want to send and  $s$  is the lowest-numbered station. Wait time  $N$  bit contention period +  $(N - 1) \times d$  bit for transmission of frames. The total is  $N + (N - 1)d$  bit times.
8. If a higher-numbered station and a lower-numbered station have packets to send at the same time, the higher-numbered station will always win the bid. Thus, a lower-numbered station will be starved from sending its packets if there is a continuous stream of higher-numbered stations ready to send their packets.
9. Stations 2, 3, 5, 7, 11, and 13 want to send. Eleven slots are needed, with the contents of each slot being as follows:

Slot 1: 2, 3, 5, 7, 11, 13  
 Slot 2: 2, 3, 5, 7  
 Slot 3: 2, 3  
 Slot 4: 2  
 Slot 5: 3  
 Slot 6: 5, 7  
 Slot 7: 5  
 Slot 8: 7  
 Slot 9: 11, 13  
 Slot 10: 11  
 Slot 11: 13

10. (a) Since all stations will see  $A$ 's packet, it will interfere with receipt of any other packet by any other station. So, no other communication is possible in this case.  
 (b)  $B$ 's packet will be seen by  $E$ ,  $A$  and  $C$ , by not by  $D$ . Thus,  $E$  can send to  $D$ , or  $A$  can send to  $D$ , or  $C$  can send to  $D$  at the same time.  
 (c) This scenario is same as (b).
11. Yes. Imagine that they are in a straight line and that each station can reach only its nearest neighbors. Then  $A$  can send to  $B$  while  $E$  is sending to  $F$ .
12. (a) Number the floors 1–7. In the star configuration, the router is in the middle of floor 4. Cables are needed to each of the  $7 \times 15 - 1 = 104$  sites. The total length of these cables is

$$4 \sum_{i=1}^{7} \sum_{j=1}^{15} \sqrt{(i-4)^2 + (j-8)^2}$$

or about 1832 meters.

(b) For classic 802.3, 7 horizontal cables 56 m long are needed, plus one vertical cable 24 m long, for a total of 416 m.

13. Classic Ethernet uses Manchester encoding, which means it has two signal periods per bit sent. The data rate is 10 Mbps, so the baud rate is twice that, or 20 megabaud.
14. The signal is a square wave with two values, high (H) and low (L). The pattern is LHLHLHHLHLHLLHHLLHHL.
15. The round-trip propagation time of the cable is 10  $\mu$ sec. A complete transmission has six phases:

1. Transmitter seizes cable (10  $\mu$ sec)
2. Transmit data (25.6  $\mu$ sec)
3. Delay for last bit to get to the end (5.0  $\mu$ sec)
4. Receiver seizes cable (10  $\mu$ sec)
5. Acknowledgement sent (3.2  $\mu$ sec)
6. Delay for last bit to get to the end (5.0  $\mu$ sec)

The sum of these is 58.8  $\mu$ sec. In this period, 224 data bits are sent, for a rate of about 3.8 Mbps.

16. Number the acquisition attempts starting at 1. Attempt  $i$  is distributed among  $2^{i-1}$  slots. Thus, the probability of a collision on attempt  $i$  is  $2^{-(i-1)}$ . The probability that the first  $k-1$  attempts will fail, followed by a success on round  $k$  is

$$P_k = (1 - 2^{-(k-1)}) \prod_{i=1}^{k-1} 2^{-(i-1)}$$

which can be simplified to

$$P_k = (1 - 2^{-(k-1)}) 2^{-(k-1)(k-2)/2}$$

The expected number of rounds is then just  $\sum k P_k$ .

17. The minimum Ethernet frame is 64 bytes, including both addresses in the Ethernet frame header, the type/length field, and the checksum. Since the header fields occupy 18 bytes and the packet is 60 bytes, the total frame size is 78 bytes, which exceeds the 64-byte minimum. Therefore, no padding is used.
18. The maximum wire delay in fast Ethernet is 1/10 as long as in Ethernet.
19. The payload is 1500 bytes, but when the destination address, source address, type/length, and checksum fields are counted, plus the VLAN header, the total is indeed 1522. Prior to VLANs, the total was 1518.

20. The smallest Ethernet frame is 512 bits, so at 1 Gbps we get 1,953,125 or almost 2 million frames/sec. However, this only works when frame bursting is operating. Without frame bursting, short frames are padded to 4096 bits, in which case the maximum number is 244,140. For the largest frame (12,144 bits), there can be as many as 82,345 frames/sec.
21. Gigabit Ethernet has it and so does 802.16. It is useful for bandwidth efficiency (one preamble, etc.) but also when there is a lower limit on frame size.
22. Station *C* is the closest to *A* since it heard the RTS and responded to it by asserting its NAV signal. *D* did not respond, so it must be outside *A*'s radio range.
23. RTS/CTS in 802.11 does not help with the exposed terminals problem. So, given the scenario in Figure 4-11(b), MACA protocol will allow simultaneous communication, *B* to *A* and *C* to *D*, but 802.11 will allow only one of these communications to take place at a time.
24. (a) Each set of ten frames will include one frame from each station. So, all stations will experience a data rate of  $54/50 \text{ Mbps} = 1.08 \text{ Mbps}$ . (b) Each station gets the same amount of time to transmit. So, the 6 Mbps stations will get 0.6 Mbps, 18 Mbps stations will get 1.8 Mbps, and 54 Mbps stations will get 5.4 Mbps.
25. A frame contains 512 bits. The bit error rate is  $p = 10^{-7}$ . The probability of all 512 of them surviving correctly is  $(1 - p)^{512}$ , which is about 0.9999488. The fraction damaged is thus about  $5 \times 10^{-5}$ . The number of frames/sec is  $11 \times 10^6 / 512$  or about 21,484. Multiplying these two numbers together, we get about 1 damaged frame per second.
26. It depends how far away the subscriber is. If the subscriber is close, QAM-64 is used for 120 Mbps. For medium distances, QAM-16 is used for 80 Mbps. For distant stations, QPSK is used for 40 Mbps.
27. One reason is the need for real-time quality of service. If an error is discovered, there is no time for a retransmission. The show must go on. Forward error correction can be used here. Another reason is that on very low-quality lines (e.g., wireless channels), the error rate can be so high that practically all frames would have to be retransmitted, and the retransmissions would probably damage as well. To avoid this, forward error correction is used to increase the fraction of frames that arrive correctly.
28. Like 802.11, WiMAX wirelessly connects devices, including mobile devices to the Internet at Mbps speeds. Also, like 802.11, WiMAX is based on OFDM and MIMO technologies. However, unlike 802.11, WiMAX base stations are much more powerful than 802.11 access points. Also, transmissions in WiMAX are carefully scheduled by the base station for each subscriber

without any possibility of collisions unlike CSMA/CA used in 802.11.

29. It is impossible for a device to be master in two piconets at the same time. Allowing this would create two problems. First, only 3 address bits are available in the header, while as many as seven slaves could be in each piconet. Thus, there would be no way to uniquely address each slave. Second, the access code at the start of the frame is derived from the master's identity. This is how slaves tell which message belongs to which piconet. If two overlapping piconets used the same access code, there would be no way to tell which frame belonged to which piconet. In effect, the two piconets would be merged into one big piconet instead of two separate ones.
30. A Bluetooth frame has an overhead of 126 bits for access code and header, and a settling time of 250 to 260  $\mu$ sec. At the basic data rate, 1 Mbps, a settling time of 250 to 260  $\mu$ sec corresponds to 250 to 260 bits. A slot is 625  $\mu$ sec long, which corresponds to 625 bits at 1 Mbps. So, a maximum of 1875 bits can be transmitted in a 3-slot frame. Out of this, 376 to 386 bits are overhead bits, leaving a maximum of 1499 to 1509 bits for the data field.
31. Bluetooth uses FHSS, just as 802.11 does. The biggest difference is that Bluetooth hops at a rate of 1600 hops/sec, far faster than 802.11.
32. In a 5-slot Bluetooth frame, a maximum of 3125 ( $625 \times 5$ ) bits can be transmitted at basic rate. Out of this, a maximum of 2744 bits are for data. In case of repetition encoding, data is replicated thrice, so the actual data transmitted is about 914 bits. This results in about 29% efficiency.
33. They do not. The dwell time in 802.11 is not standardized, so it has to be announced to new stations that arrive. In Bluetooth, this is always 625  $\mu$ sec. There is no need to announce this. All Bluetooth devices have this hardwired into the chip. Bluetooth was designed to be cheap, and fixing the hop rate and dwell time leads to a simpler chip.
34. We want to maximize the probability that one (and only one) tag responds in a given slot. Consulting Sec. 4.2.4, the best tag probability for 10 tags is 1/10. This occurs when the reader sets Q equal to 10 slots. Consulting Fig. 4-0, the probability that one tag responds is roughly 40%.
35. One key security concern is unauthorized tracking of RFID tags. An adversary with an appropriate RFID reader can track the locations of the items tagged using RFID tags. This becomes quite serious if the item is sensitive in nature, for example, a passport, and the tag can be used to retrieve further information, for example, the nationality and other personal information of the person holding the passport. Another security concern is the ability of a reader to change tag information. This can be used by an adversary to, for example, change the price of a tagged item he plans to buy.

36. The worst case is an endless stream of 64-byte (512-bit) frames. If the back-plane can handle  $10^9$  bps, the number of frames it can handle is  $10^9/512$ . This is 1,953,125 frames/sec.
37. A store-and-forward switch stores each incoming frame in its entirety, then examines it and forwards it. A cut-through switch starts to forward incoming frames before they have arrived completely. As soon as the destination address is in, the forwarding can begin.
38. (a)  $B1$  will forward this packet on ports 2, 3, and 4.  $B2$  will forward it on 1, 2 and 3.  
(b)  $B2$  will forward this packet on ports 1, 3, and 4.  $B1$  will forward it on 1, 2 and 3.  
(c)  $B2$  will not forward this packet on any of its ports, and  $B1$  will not see it.  
(d)  $B2$  will forward this packet on port 2.  $B1$  will not see it.  
(e)  $B2$  will forward this packet on port 4 and  $B1$  will forward it on port 1.  
(f)  $B1$  will forward this packet on ports 1, 3 and 4.  $B2$  will forward it on port 2.
39. Store-and-forward switches store entire frames before forwarding them. After a frame comes in, the checksum can be verified. If the frame is damaged, it is discarded immediately. With cut-through, damaged frames cannot be discarded by the switch because by the time the error is detected, the frame is already gone. Trying to deal with the problem is like locking the barn door after the horse has escaped.
40. A bridge that does not have any station directly connected to any of its ports and is part of a loop is a candidate for not being a part of the spanning tree bridges. This can happen if the shortest paths to the root for all bridges connected to this bridge does not include this bridge.
41. No. Hubs just connect all the incoming lines together electrically. There is nothing to configure. No routing is done in a hub. Every frame coming into the hub goes out on all the other lines.
42. It would work. Frames entering the core domain would all be legacy frames, so it would be up to the first core switch to tag them. It could do this by using MAC addresses or IP addresses. Similarly, on the way out, that switch would have to untag outgoing frames.

## SOLUTIONS TO CHAPTER 5 PROBLEMS

1. File transfer, remote login, and video on demand need connection-oriented service. On the other hand, credit card verification and other point-of-sale terminals, electronic funds transfer, and many forms of remote database ac-

cess are inherently connectionless, with a query going one way and the reply coming back the other way.

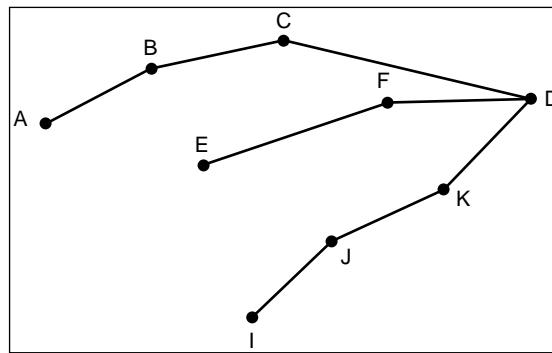
2. Virtual circuit networks most certainly need this capability in order to route connection setup packets from an arbitrary source to an arbitrary destination.
3. The negotiation could set the window size, maximum packet size, data rate, and timer values.
4. Yes. A large noise burst could garble a packet badly. With a  $k$ -bit checksum, there is a probability of  $2^{-k}$  that the error is undetected. If the destination field or, equivalently, virtual-circuit number, is changed, the packet will be delivered to the wrong destination and accepted as genuine. Put in other words, an occasional noise burst could change a perfectly legal packet for one destination into a perfectly legal packet for another destination.
5. Pick a route using the shortest path. Now remove all the arcs used in the path just found, and run the shortest path algorithm again. The second path will be able to survive the failure of any line in the first path, and vice versa. It is conceivable, though, that this heuristic may fail even though two line-disjoint paths exist. To solve it correctly, a max-flow algorithm should be used.
6. Going via  $B$  gives (11, 6, 14, 18, 12, 8).  
 Going via  $D$  gives (19, 15, 9, 3, 9, 10).  
 Going via  $E$  gives (12, 11, 8, 14, 5, 9).

Taking the minimum for each destination except  $C$  gives (11, 6, 0, 3, 5, 8). The outgoing lines are ( $B, B, -, D, E, B$ ).

7. The routing table is 400 bits. Twice a second this table is written onto each line, so 800 bps are needed on each line in each direction.
8. It always holds. If a packet has arrived on a line, it must be acknowledged. If no packet has arrived on a line, it must be sent there. The cases 00 (has not arrived and will not be sent) and 11 (has arrived and will be sent back) are logically incorrect and thus do not exist.
9. The minimum occurs at 15 clusters, each with 16 regions, each region having 20 routers, or one of the equivalent forms, e.g., 20 clusters of 16 regions of 15 routers. In all cases the table size is  $15 + 16 + 20 = 51$ .
10. Conceivably it might go into promiscuous mode, reading all frames dropped onto the LAN, but this is very inefficient. Instead, what is normally done is that the home agent tricks the router into thinking it is the mobile host by responding to ARP requests. When the router gets an IP packet destined for the mobile host, it broadcasts an ARP query asking for the 802.3 MAC-level address of the machine with that IP address. When the mobile host is not around, the home agent responds to the ARP, so the router associates the

mobile user's IP address with the home agent's 802.3 MAC-level address.

11. (a) The reverse path forwarding algorithm takes five rounds to finish. The packet recipients on these rounds are *AC*, *DFIJ*, *DEGHJKN*, *GHKN*, and *LMO*, respectively. A total of 21 packets are generated.  
 (b) The sink tree needs four rounds and 14 packets.
12. Node *F* currently has two descendants, *A* and *D*. It now acquires a third one, *G*, not circled because the packet that follows *IFG* is not on the sink tree. Node *G* acquires a second descendant, in addition to *D*, labeled *F*. This, too, is not circled as it does not come in on the sink tree.
13. Multiple spanning trees are possible. One of them is:



14. Node *H* is three hops from *B*, so it takes three rounds to find the route.
15. The protocol is terrible. Let time be slotted in units of  $T$  sec. In slot 1 the source router sends the first packet. At the start of slot 2, the second router has received the packet but cannot acknowledge it yet. At the start of slot 3, the third router has received the packet, but it cannot acknowledge it either, so all the routers behind it are still hanging. The first acknowledgement can only be sent when the destination host takes the packet from the destination router. Now the acknowledgement begins propagating back. It takes two full transits of the network,  $2(n - 1)T$  sec, before the source router can send the second packet. Thus, the throughput is one packet every  $2(n - 1)T$  sec.
16. Each packet emitted by the source host makes either 1, 2, or 3 hops. The probability that it makes one hop is  $p$ . The probability that it makes two hops is  $p(1 - p)$ . The probability that it makes 3 hops is  $(1 - p)^2$ . The mean path length a packet can expect to travel is then the weighted sum of these three probabilities, or  $p^2 - 3p + 3$ . Notice that for  $p = 0$  the mean is 3 hops and for  $p = 1$  the mean is 1 hop. With  $0 < p < 1$ , multiple transmissions may be needed. The mean number of transmissions can be found by realizing that the probability of a successful transmission all the way is  $(1 - p)^2$ , which we will

call  $\alpha$ . The expected number of transmissions is just

$$\alpha + 2\alpha(1 - \alpha) + 3\alpha(1 - \alpha)^2 + \dots = \frac{1}{\alpha} = \frac{1}{(1 - p)^2}$$

Finally, the total hops used is just  $(p^2 - 3p + 3)/(1 - p)^2$ .

17. First, the ECN method explicitly sends a congestion notification to the source by setting a bit, whereas RED implicitly notifies the source by simply dropping one of its packets. Second, the ECN method drops a packet only when there is no buffer space left, whereas RED drops packets before all the buffer are exhausted.
18. With a token every 5  $\mu$ sec, 200,000 cells/sec can be sent. Each packet holds 48 data bytes or 384 bits. The net data rate is then 76.8 Mbps.
19. The naive answer says that at 6 Mbps it takes  $4/3$  sec to drain an 8 megabit bucket. However, this answer is wrong, because during that interval, more tokens arrive. The correct answer can be obtained by using the formula  $S = C/(M - \rho)$ . Substituting, we get  $S = 8/(6 - 1)$  or 1.6 sec.
20. The bandwidths in MB/sec are as follows: A: 2, B: 0, C: 1, E: 3, H: 3, J: 3, K: 2, and L: 1.
21. Here  $\mu$  is 2 million and  $\lambda$  is 1.5 million, so  $\rho = \lambda/\mu$  is 0.75, and from queueing theory, each packet experiences a delay four times what it would in an idle system. The time in an idle system is 500 nsec, here it is 2  $\mu$ sec. With 10 routers along a path, the queueing plus service time is 20  $\mu$ sec.
22. There is no guarantee. If too many packets are expedited, their channel may have even worse performance than the regular channel.
23. The initial IP datagram will be fragmented into two IP datagrams at I1. No other fragmentation will occur.

Link A-R1:

$Length = 940; ID = x; DF = 0; MF = 0; Offset = 0$

Link R1-R2:

- (1)  $Length = 500; ID = x; DF = 0; MF = 1; Offset = 0$
- (2)  $Length = 460; ID = x; DF = 0; MF = 0; Offset = 60$

Link R2-B:

- (1)  $Length = 500; ID = x; DF = 0; MF = 1; Offset = 0$
- (2)  $Length = 460; ID = x; DF = 0; MF = 0; Offset = 60$

24. If the bit rate of the line is  $b$ , the number of packets/sec that the router can emit is  $b/8192$ , so the number of seconds it takes to emit a packet is  $8192/b$ . To put out 65,536 packets takes  $2^{29}/b$  sec. Equating this to the maximum

- packet lifetime, we get  $2^{29}/b = 10$ . Then,  $b$  is about 53,687,091 bps.
25. Since the information is needed to route every fragment, the option must appear in every fragment.
  26. With a 2-bit prefix, there would have been 18 bits left over to indicate the network. Consequently, the number of networks would have been  $2^{18}$  or 262,144. However, all 0s and all 1s are special, so only 262,142 are available.
  27. The address is 194.47.21.130.
  28. The mask is 20 bits long, so the network part is 20 bits. The remaining 12 bits are for the host, so 4096 host addresses exist.
  29. Each Ethernet adapter sold in stores comes hardwired with an Ethernet (MAC) address in it. When burning the address into the card, the manufacturer has no idea where in the world the card will be used, making the address useless for routing. In contrast, IP addresses are either assigned either statically or dynamically by an ISP or company, which knows exactly how to get to the host getting the IP address.
  30. To start with, all the requests are rounded up to a power of two. The starting address, ending address, and mask are as follows:
    - A: 198.16.0.0 – 198.16.15.255 written as 198.16.0.0/20
    - B: 198.16.16.0 – 198.23.15.255 written as 198.16.16.0/21
    - C: 198.16.32.0 – 198.47.15.255 written as 198.16.32.0/20
    - D: 198.16.64.0 – 198.95.15.255 written as 198.16.64.0/19
  31. They can be aggregated to 57.6.96.0/19.
  32. It is sufficient to add one new table entry: 29.18.0.0/22 for the new block. If an incoming packet matches both 29.18.0.0/17 and 29.18.0.0/22, the longest one wins. This rule makes it possible to assign a large block to one outgoing line but make an exception for one or more small blocks within its range.
  33. The packets are routed as follows:
    - (a) Interface 1
    - (b) Interface 0
    - (c) Router 2
    - (d) Router 1
    - (e) Router 2
  34. After NAT is installed, it is crucial that all the packets pertaining to a single connection pass in and out of the company via the same router, since that is where the mapping is kept. If each router has its own IP address and all traffic belonging to a given connection can be sent to the same router, the map-

- ping can be done correctly and multihoming with NAT can be made to work.
35. You say that ARP does not provide a service to the network layer, it is part of the network layer and helps provide a service to the transport layer. The issue of IP addressing does not occur in the data link layer. Data link layer protocols are like protocols 1 through 6 in Chap. 3, HDLC, PPP, etc. They move bits from one end of a line to the other.
  36. In the general case, the problem is nontrivial. Fragments may arrive out of order and some may be missing. On a retransmission, the datagram may be fragmented in different-sized chunks. Furthermore, the total size is not known until the last fragment arrives. Probably the only way to handle reassembly is to buffer all the pieces until the last fragment arrives and the size is known. Then build a buffer of the right size, and put the fragments into the buffer, maintaining a bit map with 1 bit per 8 bytes to keep track of which bytes are present in the buffer. When all the bits in the bit map are 1, the datagram is complete.
  37. As far as the receiver is concerned, this is a part of new datagram, since no other parts of it are known. It will therefore be queued until the rest show up. If they do not, this one will time out too.
  38. An error in the header is much more serious than an error in the data. A bad address, for example, could result in a packet being delivered to the wrong host. Many hosts do not check to see if a packet delivered to them is in fact really for them. They assume the network will never give them packets intended for another host. Data is sometimes not checksummed because doing so is expensive, and upper layers often do it anyway, making it redundant here.
  39. Yes. The fact that the Minneapolis LAN is wireless does not cause the packets that arrive for her in Boston to suddenly jump to Minneapolis. The home agent in Boston must tunnel them to the foreign agent on the wireless LAN in Minneapolis. The best way to think of this situation is that the user has plugged into the Minneapolis LAN, the same way all the other Minneapolis users have. That the connection uses radio instead of a wire is irrelevant.
  40. With 16 bytes there are  $2^{128}$  or  $3.4 \times 10^{38}$  addresses. If we allocate them at a rate of  $10^{18}$  per second, they will last for  $10^{13}$  years. This number is 1000 times the age of the universe. Of course, the address space is not flat, so they are not allocated linearly, but this calculation shows that even with an allocation scheme that has an efficiency of 1/1000 (0.1 percent), one will never run out.
  41. The *Protocol* field tells the destination host which protocol handler to give the IP packet to. Intermediate routers do not need this information, so it is not needed in the main header. Actually, it is there, but disguised. The *Next*

*header* field of the last (extension) header is used for this purpose.

42. Conceptually, there are no changes. Technically, the IP addresses requested are now bigger, so bigger fields are needed.

## SOLUTIONS TO CHAPTER 6 PROBLEMS

1. The LISTEN call could indicate a willingness to establish new connections but not block. When an attempt to connect was made, the caller could be given a signal. It would then execute, say, OK or REJECT to accept or reject the connection. In our original scheme, this flexibility is lacking.
2. Since the two end points are peers, a separate application-level mechanism is needed that informs the end points at run time about which end will act as server and which end will act as client, as well as their addresses. One way to do this is to have a separate coordinator process that provides this information to the end points before a connection between the end points is established.
3. The dashed line from *PASSIVE ESTABLISHMENT PENDING* to *ESTABLISHED* is no longer contingent on an acknowledgement arriving. The transition can happen immediately. In essence, the *PASSIVE ESTABLISHMENT PENDING* state disappears, since it is never visible at any level.
4. If the client sends a packet to *SERVER\_PORT* and the server is not listening to that port, the packet will not be delivered to the server.
5. The *connect( )* may fail if the server hasn't yet executed its *listen( )* call.
6. One other criteria is how the client is affected by extra delay involved in process server technique. The server for the requested service has to be loaded and probably has to be initialized before the client request can be serviced.
7. (a) The clock takes 32768 ticks, i.e., 3276.8 sec to cycle around. At zero generation rate, the sender would enter the forbidden zone at  $3276.8 - 60 = 3216.8$  sec.  
 (b) At 240 sequence numbers/min, the actual sequence number is  $4t$ , where  $t$  is in sec. The left edge of the forbidden region is  $10(t - 3216.8)$ . Equating these two formulas, we find that they intersect at  $t = 5361.3$  sec.
8. Look at the second duplicate packet in Fig. 6-11(b). When that packet arrives, it would be a disaster if acknowledgements to  $y$  were still floating around.
9. Deadlocks are possible. For example, a packet arrives at  $A$  out of the blue, and  $A$  acknowledges it. The acknowledgement gets lost, but  $A$  is now open while  $B$  knows nothing at all about what has happened. Now the same thing happens to  $B$ , and both are open, but expecting different sequence numbers.

Timeouts have to be introduced to avoid the deadlocks.

10. No. The problem is essentially the same with more than two armies.
11. If the  $AW$  or  $WA$  time is small, the events  $AC(W)$  and  $WC(A)$  are unlikely events. The sender should retransmit in state  $S1$ ; the receiver's order does not matter.
12. Allocation for flow  $A$  will be  $1/2$  on links  $R1R2$  and  $R2R3$ . Allocation for flow  $E$  will  $1/2$  on links  $R1R2$  and  $R2R6$ . All other allocations remain the same.
13. The sliding window is simpler, having only one set of parameters (the window edges) to manage. Furthermore, the problem of a window being increased and then decreased, with the segments arriving in the wrong order, does not occur. However, the credit scheme is more flexible, allowing a dynamic management of the buffering, separate from the acknowledgements.
14. In AIAD and MIMD, the users will oscillate along the efficiency line, but will not converge. MIAD will converge just like AIMD. None of these policies are stable. Decrease policy in AIAD and MIAD is not aggressive, and increase policy in MIAD and MIMD is not gentle.
15. No. IP packets contain IP addresses, which specify a destination machine. Once such a packet arrived, how would the network handler know which process to give it to? UDP packets contain a destination port. This information is essential so they can be delivered to the correct process.
16. It is possible that a client may get the wrong file. Suppose client  $A$  sends a request for file  $f1$  and then crashes. Another client  $B$  then uses the same protocol to request another file  $f2$ . Suppose client  $B$ , running on the same machine as  $A$  (with the same IP address), binds its UDP socket to the same port that  $A$  was using earlier. Furthermore, suppose  $B$ 's request is lost. When the server's reply (to  $A$ 's request) arrives, client  $B$  will receive it and assume that it is a reply its own request.
17. Sending 1000 bits over a 1 Gbps line takes  $1 \mu\text{sec}$ . The speed of light in fiber optics is 200 km/msec, so it takes 0.5 msec for the request to arrive and another 0.5 msec for the reply to get back. In all, 1000 bits have been transmitted in 1 msec. This is equivalent to 1 megabit/sec, or 1/10 of 1% efficiency.
18. At 1 Gbps, the response time is determined by the speed of light. The best that can be achieved is 1 msec. At 1 Mbps, it takes about 1 msec to pump out the 1024 bits, 0.5 msec for the last one to get to the server, and 0.5 msec for the reply to get back in the best case. The best possible RPC time is then 2 msec. The conclusion is that improving the line speed by a factor of 1000 only wins a factor of two in performance. Unless the gigabit line is amazingly cheap, it is probably not worth having for this application.

19. Here are three reasons. First, process IDs are OS-specific. Using process IDs would have made these protocols OS-dependent. Second, a single process may establish multiple channels of communications. A single process ID (per process) as the destination identifier cannot be used to distinguish between these channels. Third, having processes listen on well-known ports is easy, but well-known process IDs are impossible.
20. A client will use RPC over UDP if the operation is idempotent and the length of all parameters or results is small enough to fit in a single UDP packet. On the other hand if the parameters or results are large, or the operation is not idempotent, he will use RPC over TCP.
21. In  $N$ , since the maximum delay is 10 seconds, an appropriate buffer can be chosen to store a little more than 10 seconds of data at destination  $D$ . This will ensure that there will be no jitter experienced. On the other hand, in  $N2$ , a smaller buffer, perhaps 2-3 seconds will be used, but some frames (that experience larger delays) will be dropped.
22. The default segment is 536 bytes. TCP adds 20 bytes and so does IP, making the default 576 bytes in total.
23. Even though each datagram arrives intact, it is possible that datagrams arrive in the wrong order, so TCP has to be prepared to reassemble the parts of a message properly.
24. Each sample occupies 4 bytes. This gives a total of 256 samples per packet. There are 44,100 samples/sec, so with 256 samples/packet, it takes  $44100/256$  or 172 packets to transmit one second's worth of music.
25. Sure. The caller would have to provide all the needed information, but there is no reason RTP could not be in the kernel, just as UDP is.
26. No. A connection is identified only by its sockets. Thus,  $(1, p) - (2, q)$  is the only possible connection between those two ports.
27. The *ACK* bit is used to tell whether the 32-bit field is used. But if it were not there, the 32-bit field would always have to be used, if necessary acknowledging a byte that had already acknowledged. In short, it is not absolutely essential for normal data traffic. However, it plays a crucial role during connection establishment, where it is used in the second and third messages of the three-way handshake.
28. The entire TCP segment must fit in the 65,515-byte payload field of an IP packet. Since the TCP header is a minimum of 20 bytes, only 65,495 bytes are left for TCP data.

29. One way starts out with a LISTEN. If a *SYN* is received, the protocol enters the *SYN RECD* state. The other way starts when a process tries to do an active open and sends a *SYN*. If the other side was opening too, and a *SYN* is received, the *SYN RECD* state is also entered.
30. The first bursts contain 2K, 4K, 8K, and 16K bytes, respectively. The next one is 24 KB and occurs after 40 msec.
31. The next transmission will be 1 maximum segment size. Then 2, 4, and 8. So after four successes, it will be 8 KB.
32. The successive estimates are 29.6, 29.84, 29.256.
33. One window can be sent every 20 msec. This gives 50 windows/sec, for a maximum data rate of about 3.3 million bytes/sec. The line efficiency is then 26.4 Mbps/1000 Mbps or 2.6 percent.
34. The goal is to send  $2^{32}$  bytes in 120 sec or 35,791,394 payload bytes/sec. This is 23,860 1500-byte frames/sec. The TCP overhead is 20 bytes. The IP overhead is 20 bytes. The Ethernet overhead is 26 bytes. This means that for 1500 bytes of payload, 1566 bytes must be sent. If we are to send 23,860 frames of 1566 bytes every second, we need a line of 299 Mbps. With anything faster than this we run the risk of two different TCP segments having the same sequence number at the same time.
35. IP is a network level protocol while TCP is an end-to-end transport level protocol. Any change in the protocol specification of IP must be incorporated on all routers in the Internet. On the other hand, TCP can work fine as long as the two end points are running compatible versions. Thus, it is possible to have many different versions of TCP running at the same time on different hosts, but not this is not the case with IP.
36. A sender may not send more than 255 segments, i.e.,  $255 \times 128 \times 8$  bits, in 30 sec. The data rate is thus no more than 8.704 kbps.
37. Compute the average:  $(270,000 \times 0 + 730,000 \times 1 \text{ msec})/1,000,000$ . It takes 730  $\mu$ sec.
38. It takes  $4 \times 10 = 40$  instructions to copy 8 bytes. Forty instructions takes 40 nsec. Thus, each byte requires 5 nsec of CPU time for copying. The system is thus capable of handling 200 MB/sec or 1600 Mbps. It can handle a 1-Gbps line if no other bottleneck is present. The size of the sequence space is  $2^{64}$  bytes, which is about
39.  $2 \times 10^{19}$  bytes. A 75-Tbps transmitter uses up sequence space at a rate of  $9.375 \times 10^{12}$  sequence numbers per second. It takes 2 million seconds to wrap around. Since there are 86,400 seconds in a day, it will take over 3 weeks to wrap around, even at 75 Tbps. A maximum packet lifetime of less

- than 3 weeks will prevent the problem. In short, going to 64 bits is likely to work for quite a while.
40. With a packet 11.72 times smaller, you get 11.72 times as many per second, so each packet only gets  $6250/11.72$  or 533 instructions.
41. The speed of light in fiber and copper is about 200 km/msec. For a 20-km line, the delay is 100  $\mu$ sec one way and 200  $\mu$ sec round trip. A 1-KB packet has 8192 bits. If the time to send 8192 bits and get the acknowledgement is 200  $\mu$ sec, the transmission and propagation delays are equal. If  $B$  is the bit time, then we have  $8192B = 2 \times 10^{-4}$  sec. The data rate,  $1/B$ , is then about 40 Mbps.
42. The answers are: (1) 18.75 KB, (2) 125 KB, (3) 562.5 KB, (4) 1.937 MB. A 16-bit window size means a sender can send at most 64 KB before having to wait for an acknowledgement. This means that a sender cannot transmit continuously using TCP and keep the pipe full if the network technology used is Ethernet, T3, or STS-3.
43. The round-trip delay is about 540 msec, so with a 50-Mbps channel the bandwidth-product delay is 27 megabits or 3,375,000 bytes. With packets of 1500 bytes, it takes 2250 packets to fill the pipe, so the window should be at least 2250 packets.

## SOLUTIONS TO CHAPTER 7 PROBLEMS

1. They are the DNS name, the IP address, and the Ethernet address.
2. It is not an absolute name, but relative to `.cs.vu.nl`. It is really just a shorthand notation for `laserjet.cs.vu.nl`.
3. The DNS servers provide a mapping between domain names and IP addresses, such that when a request for a Web page is received, the browser can look up in the DNS server the IP address corresponding to the domain name of the requested page, and then download the requested page from that IP address.

If all the DNS servers in the world were to crash at the same time, one would not be able to map between domain names and IP addresses. Therefore, the only way to access Web pages would be by using the IP address of the host server instead of the domain name. Since most of us do not know the IP addresses of the servers we access, this type of situation would make use of the Internet extremely inefficient, if not virtually impossible for most users.
4. DNS is idempotent. Operations can be repeated without harm. When a process makes a DNS request, it starts a timer. If the timer expires, it just makes the request again. No harm is done.

5. The generated name would probably be unique, and should therefore be allowed. However, DNS names *must* be shorter than 256 bytes, as required by the standard. Since together with the *.com* ending the generated name would be longer than 256 characters, it is not permissible.
6. Yes. In fact, in Fig. 7-4 we see an example of a duplicate IP address. Remember that an IP address consists of a network number and a host number. If a machine has two Ethernet cards, it can be on two separate networks, and if so, it needs two IP addresses.
7. There are, obviously, many approaches. One is to turn the top-level server into a server farm. Another is to have 26 separate servers, one for names beginning with *a*, one for *b*, and so on. For some period of time (say, 3 years) after introducing the new servers, the old one could continue to operate to give people a chance to adapt their software.
8. It belongs to the envelope because the delivery system needs to know its value to handle email that cannot be delivered.
9. This is much more complicated than you might think. To start with, about half the world writes the given names first, followed by the family name, and the other half (e.g., China and Japan) do it the other way. A naming system would have to distinguish an arbitrary number of given names, plus a family name, although the latter might have several parts, as in John von Neumann. Then there are people who have a middle initial, but no middle name. Various titles, such as Mr., Miss, Mrs., Ms., Dr., Prof., or Lord, can prefix the name. People come in generations, so Jr., Sr., III, IV, and so on have to be included. Some people use their academic titles in their names, so we need B.A., B.Sc., M.A., M.Sc., Ph.D., and other degrees. Finally, there are people who include certain awards and honors in their names. A Fellow of the Royal Society in England might append FRS, for example. By now we should be able to please even the learned:

Prof. Dr. Abigail Barbara Cynthia Doris E. de Vries III, Ph.D., FRS

10. Naturally, the firm does not want to provide an additional email account for each employee. However, the only thing that needs to be done is to associate the alias *firstname.lastname* with a user's existing email account. This way, when incoming email at the SMTP daemon with a *TO* address of the form *firstname.lastname@lawfirm.com*, all it needs to do is look up what login name this alias corresponds to, and point that email to the mailbox *login@lawfirm.com*.
11. The base64 encoding will break the message into 1520 units of 3 bytes each. Each of these will be encoded as 4 bytes, for a total of 6080 bytes. If these are then broken up into lines of 110 bytes, 56 such lines will be needed, adding 56 CRs and 56 LFs. The total length will then be 6192 bytes.

12. Some examples and possible helpers are application/msexcel (Excel), application/ppt (PowerPoint), audio/midi (MIDI sound), image/tiff (any graphics previewer), and also video/x-dv (QuickTime player).
13. Yes. Use the *message/external-body* subtype and just send the URL of the file instead of the actual file.
14. Each message received in John's work email inbox will be forwarded to his personal inbox, thereby generating an autoreply by the vacation agent, sent to his work inbox. This reply will be seen by the work computer as a new message, and thus be forwarded to the personal mailbox, which in turn, will send another reply to the work inbox. As a result there will be an endless string of messages for each message received in John's work email address (unless the vacation agent is smart enough to reply just once to each sender it sees). However, assuming that the vacation agent logs email addresses to which it has already responded, a single auto-reply will be received by the work email inbox and forwarded back to the personal inbox, and no more canned messages will be generated.
15. The first one is any sequence of one or more spaces and/or tabs. The second, one is any sequence of one or more spaces and/or tabs and/or backspaces, subject to the condition that the net result of applying all the backspaces still leaves at least one space or tab over.
16. The actual replies have to be done by the message transfer agent. When an SMTP connection comes in, the message transfer agent has to check whether a vacation agent is set up to respond to the incoming email, and, if so, send an answer. The user transfer agent cannot do this because it will not even be invoked until the user comes back from vacation.
17. It can do it approximately, but not exactly. Suppose that there are 1024 node identifiers. If node 300 is looking for node 800, it is probably better to go clockwise, but it could happen that there are 20 actual nodes between 300 and 800 going clockwise and only 16 actual nodes between them going counterclockwise. The purpose of the cryptographic hashing function SHA-1 is to produce a very smooth distribution so that the node density is about the same all along the circle. But there will always be statistical fluctuations, so the straightforward choice may be wrong.
18. No. The IMAP program does not actually touch the remote mailbox. It sends commands to the IMAP daemon on the mail server. As long as that daemon understands the mailbox format, it can work. Thus, a mail server could change from one format to another overnight without telling its customers, as long as it simultaneously changes its IMAP daemon so it understands the new format.

19. In the finger table for node 1, the node in entry 4 switches from 20 to 18. In the finger table for node 12, the node in entry 2 switches from 20 to 18. The finger table for node 4 is not affected by the change.
20. It does not use either one, but it is fairly similar in spirit to IMAP because both of them allow a remote client to examine and manage a remote mailbox. In contrast, POP3 just sends the mailbox to the client for processing there.
21. The browser has to be able to know whether the page is text, audio, video, or something else. The MIME headers provide this information.
22. Yes, it is possible. Which helper is started depends on the configuration tables inside the browser, and Firefox and IE may have been configured differently. Furthermore, IE takes the file extension more seriously than the MIME type, and the file extension may indicate a different helper than the MIME type.
23. As mentioned, an IP address is a set of four numbers separated by dots. An example of using an IP address is *http://192.31.231.66/index.html*. The browser uses the fact that a DNS name cannot end with a digit in order to distinguish between a URL using a DNS name and a URL using an IP address, which would always end with a digit.
24. The URL is probably *ftp://www.ma.stanford.edu/ftp/pub/forReview/newProof.pdf*.
25. Do it the way *toms-casino* does: just put a customer ID in the cookie and store the preferences in a database on the server indexed by customer ID. That way, the size of the record is unlimited.
26. Technically, it will work, but it is a terrible idea. All the customer has to do is modify the cookie to get access to someone else's bank account. Having the cookie provide the customer's ID number is safe, but the customer should be required to enter a password to prove his identity.
27. (a) The browser uses the *TITLE* attribute when a user hovers with the mouse over the words "HEADER 1", and displays the value of that attribute as "this is the header".  
(b) The *ALT* attribute is only useful for images, whereas the *TITLE* attribute can be included in any HTML tag. Additionally, the *ALT* attribute is used when the browser cannot find the image which should be displayed, whereas the *TITLE* attribute is used during hover-over. Due to these different uses, an *<img>* tag may include both *ALT* and *TITLE* attributes, though their values would typically be identical.
28. A hyperlink consists of *<a href="...">* and *</a>*. In between them is the clickable text. It is also possible to put an image here. For example:

```
<a href="http://www.abcd.com/foo">  </a>
```

- 29.** Here is one way to do it:

```
<html>
<body>
<a href="mailto:username@DomainName.com"> Click Here to email me </a>
</body>
</html>
```

When a user clicks this link, the user's default email-writing program opens up a "compose message" window including the address "username@DomainName.com" in the *TO* field.

- 30.** One way of writing the XML page is:

```
<?xml version="1.0" ?>
<?xml-stylesheet type="text/xsl" href="student list.xsl"?>
<student list>
  <student>
    <name> Jerry </name>
    <address> 50 Farmington Av </address>
    <sid> 11227766 </sid>
    <gpa> 4.0 </gpa>
  </student>
  <student>
    <name> Elaine </name>
    <address> 5 Gumdrop Lane</address>
    <sid> 37205639 </sid>
    <gpa> 3.0 </gpa>
  </student>
  <student>
    <name> Tessa </name>
    <address> 6 Waterfall St </address>
    <sid> 43720472 </sid>
    <gpa> 3.8 </gpa>
  </student>
</student list>
```

- 31.** (a) There are only 14 annual calendars, depending on the day of the week on which 1 January falls and whether the year is a leap year. Thus, a JavaScript program could easily contain all 14 calendars and a small database of which year gets which calendar. A PHP script could also be used, but it would be slower.

(b) This requires a large database. It must be done on the server by using PHP.

(c) Both work, but JavaScript is faster.

**32.** There are obviously many possible solutions. Here is one:

```
<html>
<head> <title> JavaScript test </title> </head>
<script language="javascript" type="text/javascript">

function response(test_form) {
    var n = 2;
    var has_factors = 0;
    var number = eval(test_form.number.value);
    var limit = Math.sqrt(number);
    while (n++ < limit) if (number % n == 0) has_factors = 1;
    document.open();
    document.writeln("<html> <body>");
    if (has_factors > 0) document.writeln(number, " is not a prime");
    if (has_factors == 0) document.writeln(number, " is a prime");
    document.writeln("</body> </html>");
    document.close();
}
</script>
</head>

<body>
<form name="myform">
Please enter a number: <input type="text" name="number">
<input type="button" value="compute primality"
onclick="response(this.form)">
</form>
</body>
</html>
```

Clearly, this can be improved in various ways, but these require a bit more knowledge of JavaScript.

**33.** The commands sent are as follows:

```
GET /welcome.html HTTP/1.1
Host: www.info-source.com
```

Note the blank line at the end. It is mandatory.

34. Most likely, HTML pages change more often than JPEG files. Lots of sites fiddle with their HTML all the time, but do not change the images much. But the effectiveness relates to not only the hit rate, but also the payoff. There is not much difference between getting a 304 message and getting 500 lines of HTML. The delay is essentially the same in both cases because HTML files are so small. Image files are large, so not having to send one is a big win.
35. No. In the sports case, it is known months in advance that there will be a big crowd at the Web site and replicas can be constructed all over the place. The essence of a flash crowd is that it is unexpected. There was a big crowd at the Florida Web site but not at the Iowa or Minnesota sites. Nobody could have predicted this in advance.
36. Sure. The ISP goes to a number of content providers and gets their permission to replicate their content on the ISP's site. The content provider might even pay for this service. The disadvantage is that it is a lot of work for the ISP to contact many content providers. It is easier to let a CDN do this.
37. Audio needs 1.4 Mbps, which is 175 KB/sec. Two hours are  $2 \times 60 \times 60 = 7,200$  seconds. Therefore, the number of Mbit needed in the CD is 10,080 M-bit, which are 1,260 MB.
38. The true values are  $\sin(2\pi i/32)$  for  $i$  from 1 to 3. Numerically, these sines are 0.195, 0.383, and 0.556. They are represented as 0.250, 0.500, and 0.500, respectively. Thus, the percent errors are 28, 31, and 10 percent, respectively.
39. In theory, it could be used, but Internet telephony is real time. For music, there is no objection to spending 5 minutes to encode a 3-minute song. For real-time speech, that would not work. Psychoacoustic compression could work for telephony, but only if a chip existed that could do the compression on the fly with a delay of around 1 msec.
40. It takes 100 msec to get a pause command to the server, in which time 12,500 bytes will arrive, so the low-water mark should be way above 12,500, probably 50,000 to be safe. Similarly, the high-water mark should be at least 12,500 bytes from the top, but, say, 50,000 would be safer.
41. It depends. If the caller is not behind a firewall and the callee is at a regular telephone, there are no problems at all. If the caller is behind a firewall and the firewall is not picky about what leaves the site, it will also work. If the callee is behind a firewall that will not let UDP packets out, it will not work.
42. The number of bits/sec is just  $1200 \times 800 \times 50 \times 16$  or 768 Mbps.
43. Yes. An error in an I-frame will cause errors in the reconstruction of subsequent P-frames and B-frames. In fact, the error will continue to propagate until the next I-frame.

- 44.** With 50,000 customers each getting two movies per month,, the server outputs 150,000 movies per month or about 5000 per day. If half of these are at 9 P.M., the server must handle about 3330 movies at once. If the server has to transmit 3330 movies at 6 Mbps each, the required bandwidth is 20 Gbps. Using OC-12 connections, with an SPE capacity of 594 Mbps each, at least 34 connections will be needed.
- 45.** The fraction of all references to the first  $r$  movies is given by

$$C/1 + C/2 + C/3 + C/4 + \cdots + C/r$$

Thus, the ratio of the first 1000 to the first 10,000 is

$$\frac{1/1 + 1/2 + 1/3 + 1/4 + \cdots + 1/1000}{1/1 + 1/2 + 1/3 + 1/4 + \cdots + 1/10000}$$

because the  $C$ s cancel out. Evaluating this numerically, we get 7.486/9.788. Thus, about 0.764 of all requests will be for movies in memory. Noteworthy is that Zipf's law implies that a substantial amount of the distribution is in the tail, compared, say, to exponential decay.

### SOLUTIONS TO CHAPTER 8 PROBLEMS

- 1.** will you walk a little faster said a whiting to a snail  
 theres a porpoise close behind us and hes treading on my tail  
 see how eagerly the lobsters and the turtles all advance  
 they are waiting on the shingle will you come and join the dance  
 will you wont you will you wont you will you join the dance  
 will you wont you will you wont you wont you join the dance

From *Alice in Wonderland* (A Whiting and a Snail).

- 2.** Assume that the most frequent plaintext letter is  $e$  and the second most frequent letter is  $t$ . In the ciphertext, the most frequent letter is 'R', and the second most frequent letter is 'K'. Note that the numerical values are  $e = 4$ ;  $K = 10$ ;  $R = 17$ ; and  $t = 19$ . The following equations therefore exist:

$$17 = (4a+b) \text{mod} 26$$

$$10 = (19a+b) \text{mod} 26$$

Thus,  $-7 = 15a \text{ mod } 26$ , which is equivalent to  $19=15a \text{ mod } 26$ . By trial and error, we solve:  $a = 3$ . Then  $17 = (12 + b) \text{ mod } 26$ . By observation,  $b = 5$ .

- 3.** The plaintext is: a digital computer is a machine that can solve problems for people by carrying out instructions given to it.

From *Structured Computer Organization* by A. S. Tanenbaum.

4. By getting hold of the encrypted key, Trudy now knows the length of the key. She can therefore determine how many columns there were in the transposition cipher matrix, and can break the ciphertext into columns. Subsequently, all Trudy has to do in order to decipher the message is try out all the arrangements of the columns until she finds one that makes sense. Assuming that the length of the encrypted key is  $k$  characters, finding the correct arrangement of the columns would require at most  $2^k$  attempts.
5. It is:  
 1010011 0001110 1100010 1010110 1001011 0100110 1111100 0111100 1001010 1111111 1100001
6. You could use ASCII representation of the characters in *Lord of the Rings* to encrypt your messages. This will give you a one-time pad which is as long as the number of bits required to represent all the characters in *Lord of the Rings*. When you are near the end of the book, and your key is almost used up, you use the last portion of the book to send a message announcing the name of the next book you will be using as your one-time pad, and switch to that book for your subsequent messages. By continuing in this routine, because you have an infinite number of books, you also have an infinitely long one-time pad.
7. At 250 Gbps, a bit takes  $4 \times 10^{-12}$  sec to be transmitted. With the speed of light being  $2 \times 10^8$  meters/sec, in 1 bit time, the light pulse achieves a length of 0.8 mm or 800 microns. Since a photon is about 1 micron in length, the pulse is 800 photons long. Thus, we are nowhere near one photon per bit even at 250 Gbps. Only at 200 Tbps do we achieve 1 bit per photon.
8. Half the time Trudy will guess right. All those bits will be regenerated correctly. The other half she will guess wrong and send random bits to Bob. Half of these will be wrong. Thus, 25% of the bits she puts on the fiber will be wrong. Bob's one-time pad will thus be 75% right and 25% wrong.
9. If the intruder had infinite computing power, they would be the same, but since that is not the case, the second one is better. It forces the intruder to do a computation to see if each key tried is correct. If this computation is expensive, it will slow the intruder down.
10. Yes. A contiguous sequence of P-boxes can be replaced by a single P-box. Similarly, for S-boxes.
11. For each possible 56-bit key, decrypt the first ciphertext block. If the resulting plaintext is legal, try the next block, etc. If the plaintext is illegal, try the next key.
12. The equation  $2^n = 10^{16}$  tells us  $n$ , the number of doubling periods needed. Solving, we get  $n = 16 \log_2 10$  or  $n = 53.15$  doubling periods, which is 79.72 years. Just building that machine is quite a way off, and Moore's Law may

not continue to hold for nearly 80 more years.

13. The equation we need to solve is  $2^{256} = 10^n$ . Taking common logarithms, we get  $n = 256 \log 2$ , so  $n = 77$ . The number of keys is thus  $10^{77}$ . The number of stars in our galaxy is about  $10^{12}$  and the number of galaxies is about  $10^8$ , so there are about  $10^{20}$  stars in the universe. The mass of the sun, a typical star, is  $2 \times 10^{33}$  grams. The sun is made mostly of hydrogen and the number of atoms in 1 gram of hydrogen is about  $6 \times 10^{23}$  (Avogadro's number). So the number of atoms in the sun is about  $1.2 \times 10^{57}$ . With  $10^{20}$  stars, the number of atoms in all the stars in the universe is about  $10^{77}$ . Thus, the number of 256-bit AES keys is equal to the number of atoms in the whole universe (ignoring the dark matter). Conclusion: breaking AES-256 by brute force is not likely to happen any time soon.
14. DES mixes the bits pretty thoroughly, so a single bit error in block  $C_i$  will completely garble block  $P_i$ . However, a one bit error in block  $C_i$  will not affect any other blocks, and therefore a single bit error only affects one plaintext block.
15. Unfortunately, every plaintext block starting at  $P_{i+1}$  will be wrong now, since all the inputs to the XOR boxes will be wrong. A framing error is thus much more serious than an inverted bit.
16. Cipher block chaining produces 8 bytes of output per encryption. Cipher feedback mode produces 1 byte of output per encryption. Thus, cipher block chaining is eight times more efficient (i.e., with the same number of cycles you can encrypt eight times as much plaintext).
17. (a) For these parameters,  $z = 48$ , so we must choose  $d$  to be relatively prime to 48. Possible values are: 5, 7, 11, 13, and 17.  
 (b) If  $e$  satisfies the equation  $37e = 1 \pmod{120}$ , then  $37e$  must be 121, 241, 361, 481 etc. Dividing each of these in turn by 37 to see which is divisible by 37, we find that  $481/37 = 13$ , hence  $e = 13$ .  
 (c) With these parameters,  $e = 9$ . To encrypt  $P$  we use the function  $C = P^9 \pmod{55}$ . For  $P = 8, 5, 12, 12$ , and  $15$ ,  $C = 18, 20, 12, 12$ , and  $25$ , respectively.
18. Trudy can look up Alice's and Bob's public key pairs, and retrieve  $n_a$  and  $n_b$ . Because of the properties of the RSA algorithm, Trudy knows that each of these numbers is a multiplication of two primes, and therefore has only two prime factors. As stated in the question, Trudy also knows that one of the prime factors is common to  $n_a$  and  $n_b$ . Thus, Trudy concludes that the Greatest Common Divisor (GCD) of  $n_a$  and  $n_b$  is the common prime factor,  $q$ . All Trudy needs to do in order to break Alice's code is to use the Euclidean algorithm to find the GCD of  $n_a$  and  $n_b$  to obtain  $q$ , and then divide  $n_a$  by the result,  $q$ , to obtain  $p_a$ . Trudy can look up  $e_a$  in Alice's public key pair, and

- can then find a solution to the equation  $d_a \times e_a = 1 \text{ mod } (p-1)(q-1)$ , thereby determining Alice's private key.
19. No. The security is based on having a strong crypto algorithm and a long key. The *IV* is not really essential. The key is what matters.
  20. If Trudy replaces both parts, when Bob applies Alice's public key to the signature, he will get something that is not the message digest of the plaintext. Trudy can put in a false message and she can hash it, but she cannot sign it with Alice's private key.
  21. When a customer, say, Sam, indicates that he wants to buy some pornography, gamble, or whatever, the Mafia order a diamond on Sam's credit card from a jeweler. When the jeweler sends a contract to be signed (presumably including the credit card number and a Mafia post office box as address), the Mafia forward the hash of the jeweler's message to Sam, along with a contract signing up Sam as a pornography or gambling customer. If Sam just signs blindly without noticing that the contract and signature do not match, the Mafia forward the signature to the jeweler, who then ships them the diamond. If Sam later claims he did not order a diamond, the jeweler will be able to produce a signed contract showing that he did.
  22. With 20 students, there are  $(25 \times 24)/2 = 300$  pairs of students. The probability that the students in any pair have the same birthday is  $1/181$ , and the probability that they have different birthdays is  $180/181$ . The probability that all 300 pairs have different birthdays is thus  $(180/181)^{300}$ . This number is about 0.190. If the probability that all pairs are mismatches is 0.190, then the probability that one or more pairs have the same birthday is about 0.810.
  23. The secretary can pick some number (e.g., 32) spaces in the letter, and potentially replace each one by space, backspace, space. When viewed on the terminal, all variants will look alike, but all will have different message digests, so the birthday attack still works. Alternatively, adding spaces at the end of lines, and interchanging spaces and tabs can also be used.
  24. It is doable. Alice encrypts a nonce with the shared key and sends it to Bob. Bob sends back a message encrypted with the shared key containing the nonce, his own nonce, and the public key. Trudy cannot forge this message, and if she sends random junk, when decrypted it will not contain Alice's nonce. To complete the protocol, Alice sends back Bob's nonce encrypted with Bob's public key.
  25. Step 1 is to verify the X.509 certificate using the root CA's public key. If it is genuine, she now has Bob's public key, although she should check the CRL if there is one. But to see if it is Bob on the other end of the connection, she needs to know if Bob has the corresponding private key. She picks a nonce and sends it to him with his public key. If Bob can send it back in plaintext,

she is convinced that it is Bob.

26. First Alice establishes a communication channel with  $X$  and asks  $X$  for a certificate to verify his public key. Suppose  $X$  provides a certificate signed by another CA  $Y$ . If Alice does not know  $Y$ , she repeats the above step with  $Y$ . Alice continues to do this, until she receives a certificate verifying the public key of a CA  $Z$  signed by  $A$  and Alice knows  $A$ 's public key. Note that this may continue until a root is reached, that is,  $A$  is the root. After this Alice verifies the public keys in reverse order starting from the certificate that  $Z$  provided. In each step during verification, she also checks the CRL to make sure that the certificate provided have not been revoked. Finally, after verifying Bob's public key, Alice ensures that she is indeed talking to Bob using the same method as in the previous problem.
27. No AH in transport mode includes the IP header in the checksum. The NAT box changes the source address, ruining the checksum. All packets will be perceived as having errors.
28. The recommended method would be by using HMACs, since they are computationally faster than using RSA. However, this requires establishing a shared key with Bob prior to the transmission of the message.
29. Incoming traffic might be inspected for the presence of viruses. Outgoing traffic might be inspected to see if company confidential information is leaking out. Checking for viruses might work if a good antivirus program is used. Checking outgoing traffic, which might be encrypted, is nearly hopeless against a serious attempt to leak information.
30. The VPN provides security for communication over the Internet, but not within the organization. Therefore, when communicating with Mary regarding R&D purchases, or any other communication which need only be secure from people outside the organization, Jim does not need to use additional encryption or security measures. However, if Jim wants his communication with Mary to be secure also with respect to people inside the organization, such as when communicating with Mary about his salary and the raise he had been promised, additional security measures should be used.
31. In message 2, put  $R_B$  inside the encrypted message instead of outside it. In this way, Trudy will not be able to discover  $R_B$  and the reflection attack will not work.
32. Bob knows that  $g^x \bmod n = 82$ . He computes  $82^3 \bmod 227 = 155$ . Alice knows that  $g^y \bmod n = 125$ . She computes  $125^{12} \bmod 227 = 155$ . The key is 155. The simplest way to do the above calculations is to use the UNIX *bc* program.

33. (a) The information transferred from Alice to Bob is not encrypted, and therefore, there is nothing Bob knows that Trudy does not know. Any response Bob can give, Trudy can also give. Under these circumstances, it is impossible for Alice to tell if she is talking to Bob or to Trudy.  
(b) If  $n$  or  $g$  are secret, and are not known to Trudy, she cannot pretend to be Bob using a man-in-the-middle attack, since she would not be able to perform the correct calculations in order to send a return message to Alice and/or to obtain the correct key.
34. The KDC needs some way of telling who sent the message, hence which decryption key to apply to it.
35. The two random numbers are used for different purposes.  $R_A$  is used to convince Alice she is talking to the KDC.  $R_{A2}$  is used to convince Alice she is talking to Bob later. Both are needed.
36. If AS goes down, new legitimate users will not be able to authenticate themselves, that is, get a TGS ticket. So, they will not be able to access any servers in the organization. Users that already have a TGS ticket (obtained from AS before it went down) can continue to access the servers until their TGS ticket lifetime expires. If TGS goes down, only those users that already have a server ticket (obtained from TGS before it went down) for a server S will be able to access S until their server ticket lifetime expires. In both cases, no security violation will occur.
37. Even if Trudy intercepted the message including  $R_B$  she has no way of using it, since this value will not be used again in the communication between Alice and Bob. Thus, there is no need for Alice and Bob to repeat the protocol with different values in order to ensure the security of their communication. However, Trudy can use the information she gleaned from the intercepted message (and multiple other such messages) to try and figure out how Bob is generating his random numbers. Therefore, next time Alice should remember to encrypt the last message of the protocol.
38. It is not essential to send  $R_B$  encrypted. Trudy has no way of knowing it, and it will not be used again, so it is not really secret. On the other hand, doing it this way allows a tryout of  $K_S$  to make doubly sure that it is all right before sending data. Also, why give Trudy free information about Bob's random number generator? In general, the less sent in plaintext, the better, and since the cost is so low here, Alice might as well encrypt  $R_B$ .
39. The bank sends a challenge (a long random number) to the merchant's computer, which then gives it to the card. The CPU on the card then transforms it in a complex way that depends on the PIN code typed directly into the card. The result of this transformation is given to the merchant's computer for transmission to the bank. If the merchant calls up the bank again to run an-

other transaction, the bank will send a new challenge, so full knowledge of the old one is worthless. Even if the merchant knows the algorithm used by the smart cards, he does not know the customer's PIN code, since it is typed directly into the card. The on-card display is needed to prevent the merchant from displaying: "Purchase price is 49.95" but telling the bank it is 499.95.

40. In order to multicast a PGP message, one would have to encrypt the IDEA key with the public key for each of the users accessing the Internet address. However, if all the users to whom the message is multicast have the same public key, the message can be multicast effectively.
41. No. Suppose the address was a mailing list. Each person would have his or her own public key. Encrypting the IDEA key with just one public key would not work. It would have to be encrypted with multiple public keys.
42. In step 3, the ISP asks for [www.trudy-the-intruder.com](http://www.trudy-the-intruder.com) and it is never supplied. It would be better to supply the IP address to be less conspicuous. The result should be marked as uncacheable so the trick can be used later if necessary.
43. The nonces guard against replay attacks. Since each party contributes to the key, if an intruder tries to replay old messages, the new key generated will not match the old one.
44. The image contains  $2048 \times 512$  pixels. Since each pixel contains 3 low-order bits, the number of bits which can be used for steganographic purposes is  $2048 \times 512 \times 3$ , which equals 3,145,728 bits or 393,216 bytes. The fraction of the file which could be encrypted in the image is approximately 0.16. If the file were compressed to a quarter of its original size, the compressed version would be of size 0.625 Mbyte. Therefore the fraction of the file which could be hidden in the image would be approximately 0.63.
45. Easy. Music is just a file. It does not matter what is in the file. There is room for 294,912 bytes in the low-order bits. MP3s require roughly 1 MB per minute, so about 18 sec of music could fit.
46. The number of bits to be encrypted is  $60 \times 10^6 \times 8 = 480 \times 10^6$  bits. Each pixel of the image can hide 3 bits in it. Therefore, the number of pixels required in order to encrypt the entire file is  $480 \times 10^6 / 3 = 160 \times 10^6 = 160,000,000$  pixels. We want the image to be 3:2 so let the width be  $3x$  and the height be  $2x$ . The number of pixels is then  $6x^2$  which must be 160,000,000. Solving, we get  $x = 5164$  and an image of  $15492 \times 10328$ . If the file were compressed to a third of its original size, the number of bits to be encrypted would be  $160 \times 10^6 / 3 = 53,333,333$ , and the number of pixels needed would be a third of the uncompressed file or  $53,333,333 / 6 = 8946 \times 5962$ . The image would then be  $8946 \times 5962$ .

47. Alice could hash each message and sign it with her private key. Then she could append the signed hash and her public key to the message. People could compare the signature and compare the public key to the one Alice used last time. If Trudy tried to impersonate Alice and appended Alice's public key, she would not be able to get the hash right. If she used her own public key, people would see it was not the same as last time.

# **SOLUTIONS MANUAL**

**DATA AND COMPUTER  
COMMUNICATIONS  
Seventh Edition**

**WILLIAM STALLINGS**

Copyright 2003: William Stallings

© 2003 by William Stallings

All rights reserved. No part of this document may be reproduced, in any form or by any means, or posted on the Internet, without permission in writing from the author

## NOTICE

This manual contains solutions to all of the review questions and homework problems in *Data and Computer Communications, Seventh Edition*. If you spot an error in a solution or in the wording of a problem, I would greatly appreciate it if you would forward the information via email to me at ws@shore.net. An errata sheet for this manual, if needed, is available at <ftp://shell.shore.net/members/w/s/ws/S/>

W.S.

## TABLE OF CONTENTS

Chapter 2:	Protocol Architecture .....	5
Chapter 3:	Data Transmission .....	8
Chapter 4:	Guided and Wireless Transmission .....	12
Chapter 5:	Signal Encoding Techniques .....	16
Chapter 6:	Digital Data Communication Techniques.....	24
Chapter 7:	Data Link Control .....	31
Chapter 8:	Multiplexing .....	38
Chapter 9:	Spread Spectrum .....	43
Chapter 10:	Circuit Switching and Packet Switching.....	46
Chapter 11:	Asynchronous Transfer Mode .....	50
Chapter 12:	Routing in Switched Networks.....	56
Chapter 13:	Congestion Control in Switched Data Networks .....	62
Chapter 14:	Cellular Wireless Networks.....	65
Chapter 15:	Local Area Network Overview.....	69
Chapter 16:	High-Speed LANs.....	77
Chapter 17:	Wireless LANs.....	82
Chapter 18:	Internetwork Protocols .....	84
Chapter 19:	Internetwork Operation.....	91
Chapter 20:	Transport Protocols.....	96
Chapter 21:	Network Security .....	102
Chapter 22:	Distributed Applications.....	106

## CHAPTER 2

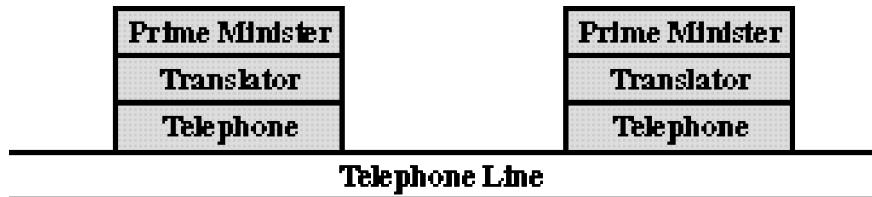
# PROTOCOL ARCHITECTURE

### ANSWERS TO QUESTIONS

- 2.1 The network access layer is concerned with the exchange of data between a computer and the network to which it is attached.
- 2.2 The transport layer is concerned with data reliability and correct sequencing.
- 2.3 A protocol is the set of rules or conventions governing the way in which two entities cooperate to exchange data.
- 2.4 A PDU is the combination of data from the next higher communications layer and control information.
- 2.5 The software structure that implements the communications function. Typically, the protocol architecture consists of a layered set of protocols, with one or more protocols at each layer.
- 2.6 Transmission Control Protocol/Internet Protocol (TCP/IP) are two protocols originally designed to provide low level support for internetworking. The term is also used generically to refer to a more comprehensive collection of protocols developed by the U.S. Department of Defense and the Internet community.
- 2.7 Layering decomposes the overall communications problem into a number of more manageable subproblems.
- 2.8 A router is a device that operates at the Network layer of the OSI model to connect dissimilar networks.

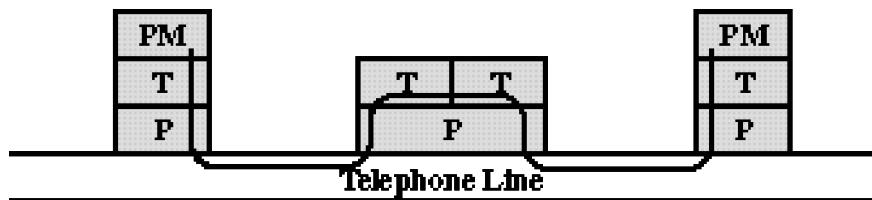
### ANSWERS TO PROBLEMS

- 2.1 The guest effectively places the order with the cook. The host communicates this order to the clerk, who places the order with the cook. The phone system provides the physical means for the order to be transported from host to clerk. The cook gives the pizza to the clerk with the order form (acting as a "header" to the pizza). The clerk boxes the pizza with the delivery address, and the delivery van encloses all of the orders to be delivered. The road provides the physical path for delivery.
- 2.2 a.



The PMs speak as if they are speaking directly to each other. For example, when the French PM speaks, he addresses his remarks directly to the Chinese PM. However, the message is actually passed through two translators via the phone system. The French PM's translator translates his remarks into English and telephones these to the Chinese PM's translator, who translates these remarks into Chinese.

b.



An intermediate node serves to translate the message before passing it on.

- 2.3 Perhaps the major disadvantage is the processing and data overhead. There is processing overhead because as many as seven modules (OSI model) are invoked to move data from the application through the communications software. There is data overhead because of the appending of multiple headers to the data. Another possible disadvantage is that there must be at least one protocol standard per layer. With so many layers, it takes a long time to develop and promulgate the standards.
- 2.4 No. There is no way to be assured that the last message gets through, except by acknowledging it. Thus, either the acknowledgment process continues forever, or one army has to send the last message and then act with uncertainty.
- 2.5 A case could be made either way. First, look at the functions performed at the network layer to deal with the communications network (hiding the details from the upper layers). The network layer is responsible for routing data through the network, but with a broadcast network, routing is not needed. Other functions, such as sequencing, flow control, error control between end systems, can be accomplished at layer 2, because the link layer will be a protocol directly between the two end systems, with no intervening switches. So it would seem that a network layer is not needed. Second, consider the network layer from the point of view of the upper layer using it. The upper layer sees itself attached to an access point into a network supporting communication with multiple devices. The layer for assuring that data sent across a network is delivered to one of a number of other end systems is the network layer. This argues for inclusion of a network layer.

In fact, the OSI layer 2 is split into two sublayers. The lower sublayer is concerned with medium access control (MAC), assuring that only one end system at a time transmits; the MAC sublayer is also responsible for addressing other end

systems across the LAN. The upper sublayer is called Logical Link Control (LLC). LLC performs traditional link control functions. With the MAC/LLC combination, no network layer is needed (but an internet layer may be needed).

- 2.6
  - a. The internet protocol can be defined as a separate layer. The functions performed by IP are clearly distinct from those performed at a network layer and those performed at a transport layer, so this would make good sense.
  - b. The session and transport layer both are involved in providing an end-to-end service to the OSI user, and could easily be combined. This has been done in TCP/IP, which provides a direct application interface to TCP.
- 2.7
  - a. No. This would violate the principle of separation of layers. To layer  $(N - 1)$ , the  $N$ -level PDU is simply data. The  $(N - 1)$  entity does not know about the internal format of the  $N$ -level PDU. It breaks that PDU into fragments and reassembles them in the proper order.
  - b. Each  $N$ -level PDU must retain its own header, for the same reason given in (a).
- 2.8 Suppose that A sends a data packet  $k$  to B and the ACK from B is delayed but not lost. A resends packet  $k$ , which B acknowledges. Eventually A receives 2 ACKs to packet  $k$ , each of which triggers transmission of packet  $(k + 1)$ . B will ACK both copies of packet  $(k + 1)$ , causing A to send two copies of packet  $(k + 2)$ . From now on, 2 copies of every data packet and ACK will be sent.
- 2.9 TFTP can transfer a maximum of 512 bytes per round trip (data sent, ACK received). The maximum throughput is therefore 512 bytes divided by the round-trip time. Source: [STEV94].

## CHAPTER 3

# DATA TRANSMISSION

### ANSWERS TO QUESTIONS

- 3.1 With guided media, the electromagnetic waves are guided along an enclosed physical path whereas unguided media provide a means for transmitting electromagnetic waves but do not guide them.
- 3.2 A continuous or analog signal is one in which the signal intensity varies in a smooth fashion over time while a discrete or digital signal is one in which the signal intensity maintains one of a finite number of constant levels for some period of time and then changes to another constant level.
- 3.3 Amplitude, frequency, and phase are three important characteristics of a periodic signal.
- 3.4  $2\pi$  radians.
- 3.5 The relationship is  $\lambda f = v$ , where  $\lambda$  is the wavelength,  $f$  is the frequency, and  $v$  is the speed at which the signal is traveling.
- 3.6 The spectrum of a signal is the frequencies it contains while the bandwidth of a signal is the width of the spectrum.
- 3.7 Attenuation is the gradual weakening of a signal over distance.
- 3.8 The rate at which data can be transmitted over a given communication path, or channel, under given conditions, is referred to as the channel capacity.
- 3.9 Bandwidth, noise, and error rate.

### ANSWERS TO PROBLEMS

- 3.1 a. If two devices transmit at the same time, their signals will be on the medium at the same time, interfering with each other; i.e., there signals will overlap and become garbled.  
b. See discussion in Section 15.3. on medium access control.
- 3.2 Period =  $1/1000 = 0.001$  s = 1 ms.
- 3.3 a.  $\sin(2\pi ft - \pi) + \sin(2\pi ft + \pi) = 2 \sin(2\pi ft + \pi)$  or  $2 \sin(2\pi ft - \pi)$  or  $-2 \sin(2\pi ft)$

b.  $\sin(2\pi ft) + \sin(2\pi ft - \pi) = 0.$

## 3.4

N	C		D		E		F		G		A		B		C
F	264		297		330		352		396		440		495		528
D		33		33		22		44		44		55		33	
W	1.25		1.11		1		0.93		0.83		0.75		0.67		0.63

N = note; F = frequency (Hz); D = frequency difference; W = wavelength (m)

3.5  $2 \sin(4\pi t + \pi)$ ; A = 2, f = 2,  $\phi = \pi$

- 3.6  $(1 + 0.1 \cos 5t) \cos 100t = \cos 100t + 0.1 \cos 5t \cos 100t$ . From the trigonometric identity  $\cos a \cos b = (1/2)(\cos(a + b) + \cos(a - b))$ , this equation can be rewritten as the linear combination of three sinusoids:  
 $\cos 100t + 0.05 \cos 105t + 0.05 \cos 95t$ . Source: [MOSH89]

- 3.7 We have  $\cos^2 x = \cos x \cos x = (1/2)(\cos(2x) + \cos(0)) = (1/2)(\cos(2x) + 1)$ . Then:  
 $f(t) = (10 \cos t)^2 = 100 \cos^2 t = 50 + 50 \cos(2t)$ . The period of  $\cos(2t)$  is  $\pi$  and therefore the period of  $f(t)$  is  $\pi$ .

- 3.8 If  $f_1(t)$  is periodic with period X, then  $f_1(t) = f_1(t + X) = f_1(t + nX)$  where n is an integer and X is the smallest value such that  $f_1(t) = f_1(t + X)$ . Similarly,  $f_2(t) = f_2(t + Y) = f_2(t + mY)$ . We have  $f(t) = f_1(t) + f_2(t)$ . If  $f(t)$  is periodic with period Z, then  $f(t) = f(t + Z)$ . Therefore  $f_1(t) + f_2(t) = f_1(t + Z) + f_2(t + Z)$ . This last equation is satisfied if  $f_1(t) = f_1(t + Z)$  and  $f_2(t) = f_2(t + Z)$ . This leads to the condition  $Z = nX = mY$  for some integers n and m. We can rewrite this last as  $(n/m) = (Y/X)$ . We can therefore conclude that if the ratio  $(Y/X)$  is a rational number, then  $f(t)$  is periodic.

- 3.9 The signal would be a low-amplitude, rapidly changing waveform.

- 3.10 No transmission medium is capable of transmitting the entire spectrum of frequencies. A real signal therefore is bandlimited, with frequencies above a certain point absent. However, most of the information is in the lower frequencies. This is not a problem if it is remembered that the object of the transmission is to send signals that represent binary 1s and 0s. Even though there will be some distortion because of the loss of higher frequencies, the shape of the original pulse is known (by the specifications for the transmission system). Thus, the receiver will usually be able to distinguish a binary 0 from a binary 1.

- 3.11 A 6-bit code allows only 64 unique characters to be defined. Several *shift lock codes* were defined in various versions of TTS (shift, supershift, unshift). These codes change the meaning of all codes that follow until a new shift lock code appears. Thus, with two shift locks,  $3 \times (64 - 3) = 183$  different codes can be defined. The actual number is less, since some codes, such as space, are "don't-cares" with respect to shift locks.

- 3.12 Refer to the reasoning of Section 3.2. Retaining the vertical resolution of 483 lines, each horizontal line occupies 52.5  $\mu$ sec. A horizontal resolution of H lines results in a maximum of H/2 cycles per line, thus the bandwidth of 5 MHz allows:

$$\begin{aligned} 5 \text{ MHz} &= (H/2) / 52.5 \mu\text{sec} \\ H &= 525 \text{ lines} \end{aligned}$$

Now, if we assume the same horizontal resolution of H = 450, then for a bandwidth of 5 MHz, the duration of one line is:

$$\begin{aligned} 5 \text{ MHz} &= (450/2) / T \\ T &= 45 \mu\text{sec} \end{aligned}$$

allowing 11  $\mu$ sec for horizontal retrace, each line occupies 56.2  $\mu$ sec. The scanning frequency is:

$$\begin{aligned} (1/30 \text{ s/scan}) / V \text{ lines} &= 56.2 \mu\text{sec/line} \\ V &= 593 \text{ lines} \end{aligned}$$

- 3.13 a.  $(30 \text{ pictures/s}) (480 \times 500 \text{ pixels/picture}) = 7.2 \times 10^6 \text{ pixels/s}$

Each pixel can take on one of 32 values and can therefore be represented by 5 bits:

$$R = 7.2 \times 10^6 \text{ pixels/s} \times 5 \text{ bits/pixel} = 36 \text{ Mbps}$$

b. We use the formula:  $C = B \log_2 (1 + SNR)$

$$\begin{aligned} B &= 4.5 \times 10^6 \text{ MHz} = \text{bandwidth, and} \\ SNR_{dB} &= 35 = 10 \log_{10} (SNR), \text{ hence} \\ SNR &= 10^{35/10} = 10^{3.5}, \text{ and therefore} \\ C &= 4.5 \times 10^6 \log_2 (1 + 10^{3.5}) = 4.5 \times 10^6 \times \log_2 (3163) \\ C &= (4.5 \times 10^6 \times 11.63) = 52.335 \times 10^6 \text{ bps} \end{aligned}$$

c. Allow each pixel to have one of ten intensity levels and let each pixel be one of three colors (red, blue, green) for a total of  $10 \times 3 = 30$  levels for each pixel element.

- 3.14  $N = 10 \log k + 10 \log T + 10 \log B$   
 $= -228.6 \text{ dBW} + 10 \log 10^4 + 10 \log 10^7$   
 $= -228.6 + 40 + 70 = -118.6 \text{ dBW}$

Source: [FREE98]

- 3.15 Using Shannon's equation:  $C = B \log_2 (1 + SNR)$   
We have  $W = 300 \text{ Hz}$   $(SNR)_{dB} = 3$   
Therefore,  $SNR = 10^{0.3}$   
 $C = 300 \log_2 (1 + 10^{0.3}) = 300 \log_2 (2.995) = 474 \text{ bps}$

3.16 Using Nyquist's equation:  $C = 2B \log_2 M$

We have  $C = 9600$  bps

a.  $\log_2 M = 4$ , because a signal element encodes a 4-bit word

Therefore,  $C = 9600 = 2B \times 4$ , and  $B = 1200$  Hz

b.  $9600 = 2B \times 8$ , and  $B = 600$  Hz

$$3.17 N = 1.38 \times 10^{-23} \times (50 + 273) \times 10,000 = 4.5 \times 10^{-17} \text{ watts}$$

3.18 Nyquist analyzed the theoretical capacity of a noiseless channel; therefore, in that case, the signaling rate is limited solely by channel bandwidth. Shannon addressed the question of what signaling rate can be achieved over a channel with a given bandwidth, a given signal power, and in the presence of noise.

$$3.19 C = B \log_2 (1 + SNR)$$

$$20 \times 10^6 = 3 \times 10^6 \times \log_2(1 + SNR)$$

$$\log_2(1 + SNR) = 6.67$$

$$1 + SNR = 102$$

$$SNR = 101$$

3.20 a. Output waveform:

$$\sin(2\pi f_1 t) + 1/3 \sin(2\pi(3f_1)t) + 1/5 \sin(2\pi(5f_1)t) + 1/7 \sin(2\pi(7f_1)t)$$

$$\text{where } f_1 = 1/T = 1 \text{ kHz}$$

$$\text{Output power} = 1/2 (1 + 1/9 + 1/25 + 1/49) = 0.586 \text{ watt}$$

b. Output noise power =  $8 \text{ kHz} \times 0.1 \mu\text{Watt}/\text{Hz} = 0.8 \text{ mWatt}$

$$SNR = 0.586/0.0008 = 732.5 \quad (\text{SNR})_{\text{db}} = 28.65$$

$$3.21 (E_b/N_0) = -151 \text{ dBW} - 10 \log 2400 - 10 \log 1500 + 228.6 \text{ dBW} = 12 \text{ dBW}$$

Source: [FREE98]

3.22

Decibels	1	2	3	4	5	6	7	8	9	10
Losses	0.8	0.63	0.5	0.4	0.32	0.25	0.2	0.16	0.125	0.1
Gains	1.25	1.6	2	2.5	3.2	4.0	5.0	6.3	8.0	10

3.23 For a voltage ratio, we have

$$N_{\text{dB}} = 30 = 20 \log(V_2/V_1)$$

$$V_2/V_1 = 10^{30/20} = 10^{1.5} = 31.6$$

$$3.24 \text{ Power (dBW)} = 10 \log(\text{Power}/1\text{W}) = 10 \log 20 = 13 \text{ dBW}$$

## CHAPTER 4

# TRANSMISSION MEDIA

### ANSWERS TO QUESTIONS

- 4.1 The twisting of the individual pairs reduces electromagnetic interference. For example, it reduces crosstalk between wire pairs bundled into a cable.
- 4.2 Twisted pair wire is subject to interference, limited in distance, band width, and data rate.
- 4.3 Unshielded twisted pair (UTP) is ordinary telephone wire, with no form of electromagnetic shielding around the wire. Shielded twisted pair (STP) surrounds the wire with a metallic braid or sheathing that reduces interference.
- 4.4 Optical fiber consists of a column of glass or plastic surrounded by an opaque outer jacket. The glass or plastic itself consists of two concentric columns. The inner column called the core has a higher index of refraction than the outer column called the cladding.
- 4.5 Point-to-point microwave transmission has a high data rate and less attenuation than twisted pair or coaxial cable. It is affected by rainfall, however, especially above 10 GHz. It is also requires line of sight and is subject to interference from other microwave transmission, which can be intense in some places.
- 4.6 Direct broadcast transmission is a technique in which satellite video signals are transmitted directly to the home for continuous operation.
- 4.7 A satellite must use different uplink and downlink frequencies for continuous operation in order to avoid interference.
- 4.8 Broadcast is omnidirectional, does not require dish shaped antennas, and the antennas do not have to be rigidly mounted in precise alignment.
- 4.9 The two functions of an antenna are: (1) For transmission of a signal, radio-frequency electrical energy from the transmitter is converted into electromagnetic energy by the antenna and radiated into the surrounding environment (atmosphere, space, water); (2) for reception of a signal, electromagnetic energy impinging on the antenna is converted into radio-frequency electrical energy and fed into the receiver.
- 4.10 An isotropic antenna is a point in space that radiates power in all directions equally.

- 4.11 A parabolic antenna creates, in theory, a parallel beam without dispersion. In practice, there will be some beam spread. Nevertheless, it produces a highly focused, directional beam.
- 4.12 Effective area and wavelength.
- 4.13 Free space loss.
- 4.14 Refraction is the bending of a radio beam caused by changes in the speed of propagation at a point of change in the medium.
- 4.15 Diffraction occurs at the edge of an impenetrable body that is large compared to the wavelength of the radio wave. The edge in effect becomes a source and waves radiate in different directions from the edge, allowing a beam to bend around an obstacle. If the size of an obstacle is on the order of the wavelength of the signal or less, scattering occurs. An incoming signal is scattered into several weaker outgoing signals in unpredictable directions.

## ANSWERS TO PROBLEMS

4.1 Elapsed time =  $(5000 \text{ km}) / (1000 \text{ km/hr}) = 5 \text{ hours} = 18,000 \text{ seconds}$

Amount of data per diskette =  $1.4 \times 1024^2 \times 8 = 11.74 \times 10^6 \text{ bits/diskette}$

Number of diskettes =  $(10^7 \text{ g}) / (30 \text{ g/diskette}) = 333333 \text{ diskettes}$

$$\text{Data transfer rate} = \frac{(11.74 \times 10^6 \text{ bits/diskette}) \times (333333 \text{ diskettes})}{18,000 \text{ seconds}} = 217 \text{ Mbps}$$

4.2  $10 \log (P_o/P_i) = -20 \text{ dB}$ ; Therefore,  $P_o/P_i = 0.01$

For  $P_i = 0.5 \text{ Watt}$ ,  $P_o = 0.005 \text{ Watt}$

$$\text{SNR} = 0.005 / (4.5 \times 10^{-6}) = 1.11 \times 10^3$$

$$\text{SNR}_{\text{dB}} = 10 \log (1.11 \times 10^3) = 30 \text{ dB}$$

4.3 The allowable power loss is  $10 \times \log 100 = 20 \text{ dB}$

a. From Figure 4.3, the attenuation is about 13 dB per km.

$$\text{Length} = (20 \text{ dB}) / (13 \text{ dB per km}) = 1.5 \text{ km}$$

b. Length =  $(20 \text{ dB}) / (20 \text{ dB per km}) = 1 \text{ km}$

c. Length =  $(20 \text{ dB}) / (2.5 \text{ dB per km}) = 8 \text{ km}$

d. Length =  $(20 \text{ dB}) / (10 \text{ dB per km}) = 2 \text{ km}$

e. Length =  $(20 \text{ dB}) / (0.2 \text{ dB per km}) = 100 \text{ km}$

4.4 An electromagnetic wave cannot penetrate an enclosing conductor. If the outer conductor of a coaxial cable is everywhere held at ground potential, no external disturbance can reach the inner, signal-carrying, conductor.

4.5 From Equation 4.2, the ratio of transmitted power to received power is  
 $P_t/P_r = (4\pi d/\lambda)^2$

If we double the frequency, we halve  $\lambda$ , or if we double the distance, we double  $d$ , so the new ratio for either of these events is:

$$P_t/P_{r2} = (8\pi d/\lambda)^2$$

Therefore:

$$10 \log (P_r/P_{r2}) = 10 \log (2^2) = 6 \text{ dB}$$

Source: [MOSH89]

4.6 We have  $\lambda f = c$ ; in this case  $\lambda \times 30 = 3 \times 10^8 \text{ m/sec}$ , which yields a wavelength of 10,000 km. Half of that is 5,000 km which is comparable to the east-to-west dimension of the continental U.S. While an antenna this size is impractical, the U.S. Defense Department has considered using large parts of Wisconsin and Michigan to make an antenna many kilometers in diameter.

4.7 a. Using  $\lambda f = c$ , we have  $\lambda = (3 \times 10^8 \text{ m/sec})/(300 \text{ Hz}) = 1,000 \text{ km}$ , so that  $\lambda/2 = 500 \text{ km}$ .

b. The carrier frequency corresponding to  $\lambda/2 = 1 \text{ m}$  is given by:  
 $f = c/\lambda = (3 \times 10^8 \text{ m/sec})/(2 \text{ m}) = 150 \text{ MHz}$ .

4.9  $\lambda = 2 \times 2.5 \times 10^{-3} \text{ m} = 5 \times 10^{-3} \text{ m}$

$$f = c/\lambda = (3 \times 10^8 \text{ m/sec})/(5 \times 10^{-3} \text{ m}) = 6 \times 10^{10} \text{ Hz} = 60 \text{ GHz}$$

4.10

Distance (km)	Radio (dB)	Wire (dB)
1	-6	-3
2	-12	-6
4	-18	-12
8	-24	-24
16	-30	-28

4.11 a. First, take the derivative of both sides of the equation  $y^2 = 2px$ :

$$\frac{dy}{dx} y^2 = \frac{dy}{dx} (2px); 2y \frac{dy}{dx} = 2p; \frac{dy}{dx} = \frac{p}{y}$$

Therefore  $\tan \beta = (p/y_1)$ .

b. The slope of PF is  $(y_1 - 0)/(x_1 - (p/2))$ . Therefore:

$$\tan \alpha = \frac{\frac{y_1}{x_1} - \frac{p}{2}}{1 + \frac{\frac{y_1}{x_1} - \frac{p}{2}}{\frac{x_1 - \frac{p}{2}}{2}}} = \frac{y_1^2 - px_1 + \frac{1}{2}p^2}{x_1y_1 - \frac{1}{2}py_1 + py_1}$$

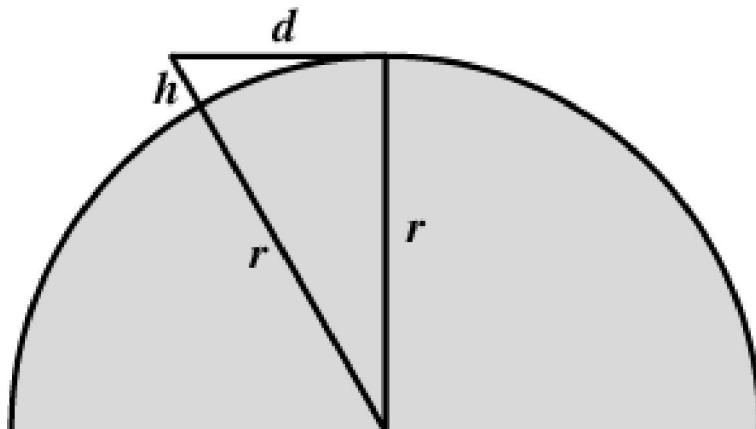
Because  $y_1^2 = 2px_1$ , this simplifies to  $\tan \alpha = (p/y_1)$ .

$$\begin{aligned} 4.12 \quad L_{dB} &= 20 \log(f_{MHz}) + 120 + 20 \log(d_{km}) + 60 - 147.56 \\ &= 20 \log(f_{MHz}) + 20 \log(d_{km}) + 32.44 \end{aligned}$$

- 4.13 a. From Appendix 3A, Power<sub>dBW</sub> = 10 log (Power<sub>W</sub>) = 10 log (50) = 17 dBW  
 Power<sub>dBm</sub> = 10 log (Power<sub>mW</sub>) = 10 log (50,000) = 47 dBm  
 b. Using Equation (4.3),  
 $L_{dB} = 20 \log(900 \times 10^6) + 20 \log(100) - 147.56 = 120 + 59.08 + 40 - 147.56 = 71.52$   
 Therefore, received power in dBm = 47 - 71.52 = -24.52 dBm  
 c.  $L_{dB} = 120 + 59.08 + 80 - 147.56 = 111.52$ ; P<sub>r,dBm</sub> = 47 - 111.52 = -64.52 dBm  
 d. The antenna gain results in an increase of 3 dB, so that P<sub>r,dBm</sub> = -61.52 dBm  
 Source: [RAPP96]

- 4.14 a.  $G = 7A/\lambda^2 = 7Af^2/c^2 = (7 \times \pi \times (0.6)^2 \times (2 \times 10^9)^2)/(3 \times 10^8)^2 = 351.85$   
 $G_{dB} = 25.46 \text{ dB}$   
 b.  $0.1 \text{ W} \times 351.85 = 35.185 \text{ W}$   
 c. Use  $L_{dB} = 20 \log(4\pi) + 20 \log(d) + 20 \log(f) - 20 \log(c) - 10 \log(G_r) - 10 \log(G_t)$   
 $L_{dB} = 21.98 + 87.6 + 186.02 - 169.54 - 25.46 - 25.46 = 75.14 \text{ dB}$   
 The transmitter power, in dBm is  $10 \log(100) = 20$ .  
 The available received signal power is  $20 - 75.14 = -55.14 \text{ dBm}$

4.15



By the Pythagorean theorem:  $d^2 + r^2 = (r + h)^2$   
 Or,  $d^2 = 2rh + h^2$ . The  $h^2$  term is negligible with respect to  $2rh$ , so we use  $d^2 = 2rh$ .

$$\text{Then, } d_{km} = \sqrt{2r_{km}h_{km}} = \sqrt{2r_{km}h_m / 1000} = \sqrt{2 \times 6.37 \times h_m} = 3.57\sqrt{h_m}$$

4.16 For radio line of sight, we use  $d = 3.57\sqrt{Kh}$ , with  $K = 4/3$ , we have  $80^2 = (3.57)^2 \times 1.33 \times h$ . Solving for  $h$ , we get  $h = 378$  m.

4.17 Let  $RI$  = refractive index,  $\alpha$  = angle of incidence,  $\beta$  = angle of refraction

$$(\sin \alpha) / (\sin \beta) = RI_{\text{air}} / RI_{\text{water}} = 1.0003 / (4/3) = 0.75$$

$$\sin \beta = 0.5 / 0.75 = 0.66; \quad \beta = 41.8^\circ$$

## CHAPTER 5

# SIGNAL ENCODING TECHNIQUES

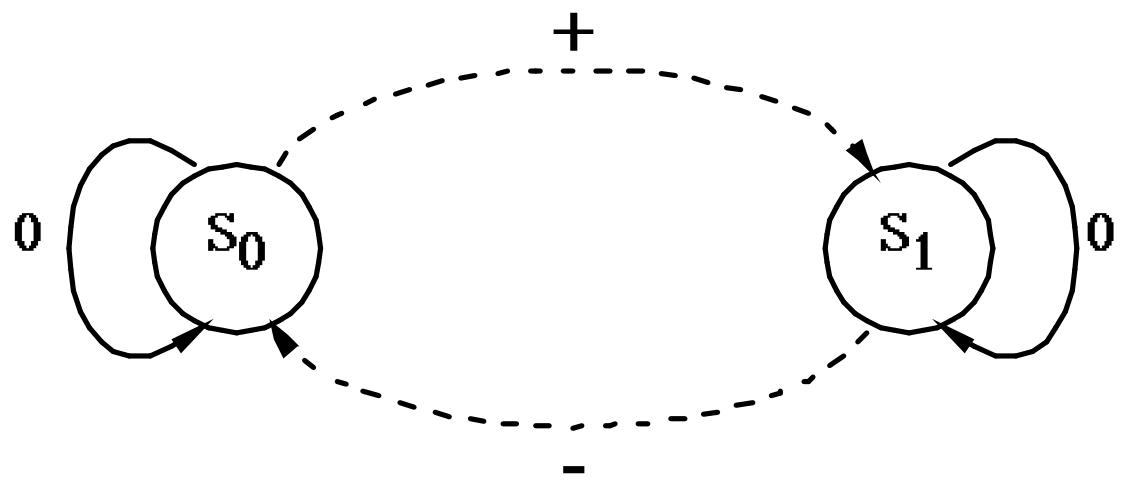
### ANSWERS TO QUESTIONS

- 5.1 Signal spectrum: A lack of high-frequency components means that less bandwidth is required for transmission. In addition, lack of a direct-current (dc) component means that ac coupling via transformer is possible. The magnitude of the effects of signal distortion and interference depend on the spectral properties of the transmitted signal. Clocking: Encoding can be used to synchronize the transmitter and receiver. Error detection: It is useful to have some error detection capability built into the physical signaling encoding scheme. Signal interference and noise immunity: Certain codes exhibit superior performance in the presence of noise. Cost and complexity: The higher the signaling rate to achieve a given data rate, the greater the cost. Some codes require a signaling rate that is in fact greater than the actual data rate.
- 5.2 In differential encoding, the signal is decoded by comparing the polarity of adjacent signal elements rather than determining the absolute value of a signal element.
- 5.3 Non return-to-zero-level (NRZ-L) is a data encoding scheme in which a negative voltage is used to represent binary one and a positive voltage is used to represent binary zero. As with NRZ-L, NRZI maintains a constant voltage pulse for the duration of a bit time. The data themselves are encoded as the presence or absence of a signal transition at the beginning of the bit time. A transition (low to high or high to low) at the beginning of a bit time denotes a binary 1 for that bit time; no transition indicates a binary 0.
- 5.4 For bipolar-AMI scheme, a binary 0 is represented by no line signal, and a binary 1 is represented by a positive or negative pulse. The binary 1 pulses must alternate in polarity. For pseudoternary, a binary 1 is represented by the absence of a line signal, and a binary 0 by alternating positive and negative pulses.
- 5.5 A biphase scheme requires at least one transition per bit time and may have as many as two transitions. In the Manchester code, there is a transition at the middle of each bit period; a low-to-high transition represents a 1, and a high-to-low transition represents a 0. In differential Manchester, the midbit transition is used only to provide clocking. The encoding of a 0 is represented by the presence of a transition at the beginning of a bit period, and a 1 is represented by the absence of a transition at the beginning of a bit period.

- 5.6 In a scrambling technique, sequences that would result in a constant voltage level on the line are replaced by filling sequences that will provide sufficient transitions for the receiver's clock to maintain synchronization. The filling sequence must be recognized by the receiver and replaced with the original data sequence. The filling sequence is the same length as the original sequence, so there is no data rate penalty.
- 5.7 A modem converts digital information into an analog signal, and conversely.
- 5.8 With amplitude-shift keying, binary values are represented by two different amplitudes of carrier frequencies. This approach is susceptible to sudden gain changes and is rather inefficient.
- 5.9 The difference is that offset QPSK introduces a delay of one bit time in the Q stream
- 5.10 QAM takes advantage of the fact that it is possible to send two different signals simultaneously on the same carrier frequency, by using two copies of the carrier frequency, one shifted by  $90^\circ$  with respect to the other. For QAM, each carrier is ASK modulated.
- 5.11 The sampling rate must be higher than twice the highest signal frequency.
- 5.12 Frequency modulation (FM) and phase modulation (PM) are special cases of angle modulation. For PM, the phase is proportional to the modulating signal. For FM, the derivative of the phase is proportional to the modulating signal.

## ANSWERS TO PROBLEMS

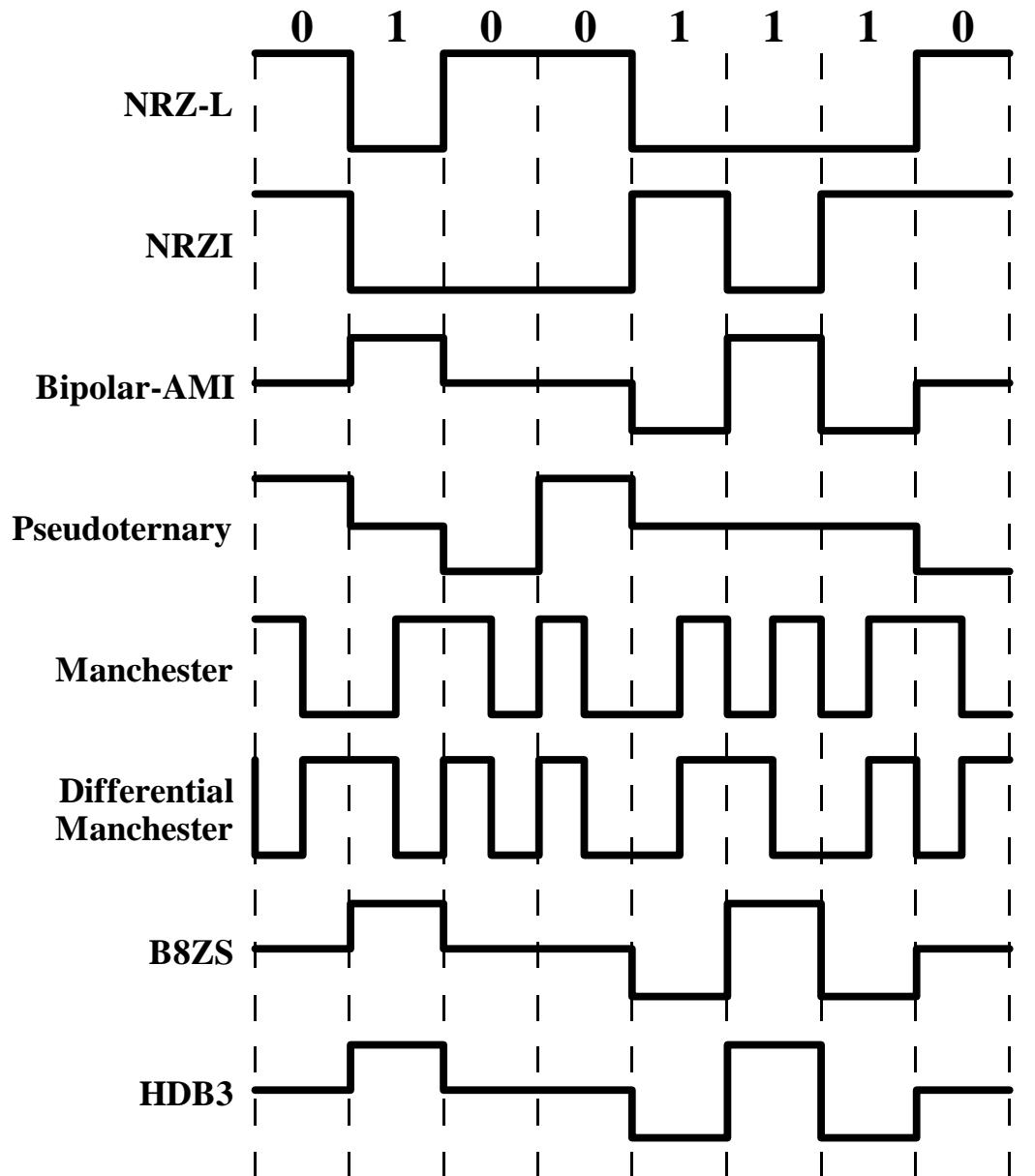
- 5.1 NRZI, Differential Manchester
- 5.2 NRZ-L produces a high level for binary 0 and a low level for binary 1. Map this level as indicated by the definition for 1 and 0 for each of the other codes.
- 5.3 First, E-NRZ provides a minimum transition rate that reduces the dc component. Second, under worst case, E-NRZ provides a minimum of one transition for every 14 bits, reducing the synchronization problem. Third, the parity bit provides an error check. The disadvantages of E-NRZ are added complexity and the overhead of the extra parity bit.
- 5.4



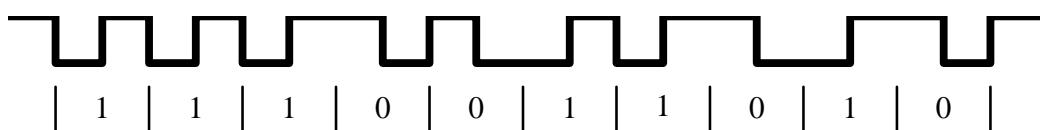
In this diagram, a dashed arrow represents a binary 0 and a solid arrow represents a binary 1. The labels -, 0, and + are used to indicate the line voltages: negative, zero, and positive, respectively.

- 5.5 a.  $c_m = b_m - b_{m-1} = (a_m + b_{m-1}) - b_{m-1} = a_m$   
 b. Bipolar-AMI

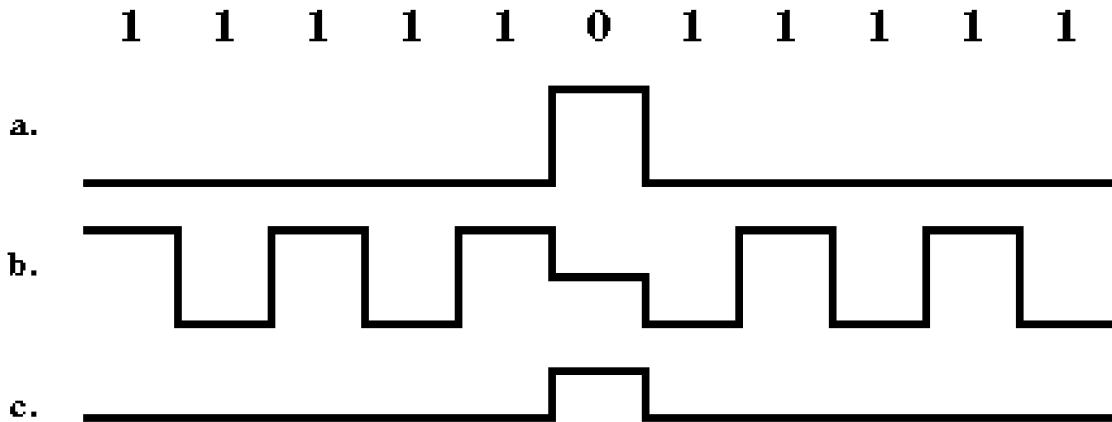
5.6



5.7 With Manchester, there is always a transition in the middle of a bit period.



5.8



- 5.9 The error is at bit position 7, where there is a negative pulse. For AMI, positive and negative pulses are used alternately for binary 1. The pulse in position 1 represents the third binary 1 in the data stream and should have a positive value.

5.10

+ - 0 + - 0 - +  
↑ ↑  
BPV

could have been produced by either

+ - 0 + - + - +  
↓ error: converted to 0  
+ - 0 + - 0 - +

or

+ - 0 + 0 0 - +  
↓ error: converted to -  
+ - 0 + - 0 - +

5.11  $s(t) = d_1(t)\cos w_c t + d_2(t)\sin w_c t$

Use the following identities:  $\cos 2\alpha = 2\cos^2 \alpha - 1$ ;  $\sin 2\alpha = 2\sin \alpha \cos \alpha$

$$\begin{aligned} s(t) \cos w_c t &= d_1(t)\cos^2 w_c t + d_2(t)\sin w_c t \cos w_c t \\ &= (1/2)d_1(t) + (1/2)d_1(t) \cos 2w_c t + (1/2)d_2(t) \sin 2w_c t \end{aligned}$$

Use the following identities:  $\cos 2\alpha = 1 - 2\sin^2 \alpha$ ;  $\sin 2\alpha = 2\sin \alpha \cos \alpha$

$$\begin{aligned}s(t) \sin w_c t &= d_1(t) \cos w_c t \sin w_c t + d_2(t) \sin^2 w_c t \\ &= (1/2)d_1(t) \sin 2w_c t + (1/2)d_2(t) - (1/2)d_2(t) \cos 2w_c t\end{aligned}$$

All terms at  $2w_c$  are filtered out by the low-pass filter, yielding:

$$y_1(t) = (1/2)d_1(t); \quad y_2(t) = (1/2)d_2(t)$$

5.12  $T_s$  = signal element period;  $T_b$  = bit period;  $A$  = amplitude = 0.005

a.  $T_s = T_b = 10^{-5}$  sec

$$P = \frac{1}{T_s} \int_0^{T_s} s^2(t) dt = \frac{A^2}{2}$$

$$E_b = P \times T_b = P \times T_s = \frac{A^2}{2} \times T_s; \quad N_0 = 2.5 \times 10^{-8} \times T_s$$

$$\frac{E_b}{N_0} = \frac{(A^2/2) \times T_s}{2.5 \times 10^{-8} \times T_s} = 500; \quad (E_b/N_0)_{dB} = 10 \log 500 = 27 \text{ dB}$$

b.

$$T_b = \frac{T_s}{2}; \quad E_b = P \times \frac{T_s}{2}; \quad N_0 = 2.5 \times 10^{-8} \times T_s$$

$$(E_b/N_0) = 250; \quad (E_b/N_0)_{dB} = 10 \log 250 = 24 \text{ dB}$$

5.13 Each signal element conveys two bits. First consider NRZ-L. It should be clear that in this case,  $D = R/2$ . For the remaining codes, one must first determine the average number of pulses per bit. For example, for Biphase-M, there is an average of 1.5 pulses per bit. We have pulse rate of  $P$  which yields a data rate of

$$R = P/1.5$$

$$D = P/2 = (1.5 \times R)/2 = 0.75 \times R$$

5.14  $E_b/N_0 = (S/N)(B/R)$

$$S/N = (R/B)(E_b/N_0) = 1 \times (E_b/N_0)$$

$$(S/N)_{dB} = (E_b/N_0)_{dB}$$

For FSK and ASK, from Figure 5.4,  $(E_b/N_0)_{dB} = 13.5 \text{ dB}$

$$(S/N)_{dB} = 13.5 \text{ dB}$$

For PSK, from Figure 5.4,  $(E_b/N_0)_{dB} = 10.5$

$$(S/N)_{dB} = 10.5 \text{ dB}$$

For QPSK, the effective bandwidth is halved, so that

$$(R/B) = 2$$

$$(R/B)_{dB} = 3$$

$$(S/N)_{dB} = 3 + 10.5 = 13.5 \text{ dB}$$

- 5.15 For ASK,  $B_T = (1 + r)R = (1.5)2400 = 3600 \text{ Hz}$   
 For FSK,  $B_T = 2 \Delta f + (1 + r)R = 2(2.5 \times 10^3) + (1.5)2400 = 8600 \text{ Hz}$

- 5.16 For multilevel signaling  $B_T = [(1 + r)/\log_2 L]R$   
 For 2400 bps QPSK,  $\log_2 L = \log_2 4 = 2$

$$B_T = (2/2)2400 = 2400 \text{ Hz}, \text{ which just fits the available bandwidth}$$

$$\text{For 8-level 4800 bps signaling, } \log_2 L = \log_2 8 = 3$$

$$B_T = (2/3)(4800) = 3200 \text{ Hz, which exceeds the available bandwidth}$$

- 5.17 As was mentioned in the text, analog signals in the voice band that represent digital data have more high frequency components than analog voice signals. These higher components cause the signal to change more rapidly over time. Hence, DM will suffer from a high level of slope overload noise. PCM, on the other hand, does not estimate changes in signals, but rather the absolute value of the signal, and is less affected than DM.
- 5.18 No. The demodulator portion of a modem expects to receive a very specific type of waveform (e.g., ASK) and would not produce meaningful output with voice input. Thus, it would not function as the coder portion of a codec. The case against using a codec in place of a modem is less easily explained, but the following intuitive argument is offered. If the decoder portion of a codec is used in place of the modulator portion of a modem, it must accept an arbitrary bit pattern, interpret groups of bits as a sample, and produce an analog output. Some very wide value swings are to be expected, resulting in a strange-looking waveform. Given the effects of noise and attenuation, the digital output produced at the receiving end by the coder portion of the codec will probably contain many errors.

- 5.19 From the text,  $(SNR)_{dB} = 6.02 n + 1.76$ , where  $n$  is the number of bits used for quantization. In this case,  $(SNR)_{dB} = 60.2 + 1.76 = 61.96 \text{ dB}$ .

- 5.20 a.  $(SNR)_{dB} = 6.02 n + 1.76 = 30 \text{ dB}$   
 $n = (30 - 1.76)/6.02 = 4.69$   
 Rounded off,  $n = 5 \text{ bits}$   
 This yields  $2^5 = 32 \text{ quantization levels}$
- b.  $R = 7000 \text{ samples/s} \times 5 \text{ bits/sample} = 35 \text{ Kbps}$

- 5.21 The maximum slope that can be generated by a DM system is  $\delta/T_s = \delta f_s$   
 where  $T_s$  = period of sampling;  $f_s$  = frequency of sampling  
 Consider that the maximum frequency component of the signal is

$$w(t) = A \sin 2\pi f_a t$$

The slope of this component is  $dw(t)/dt = A 2 \pi f_a \cos 2 \pi f_a t$

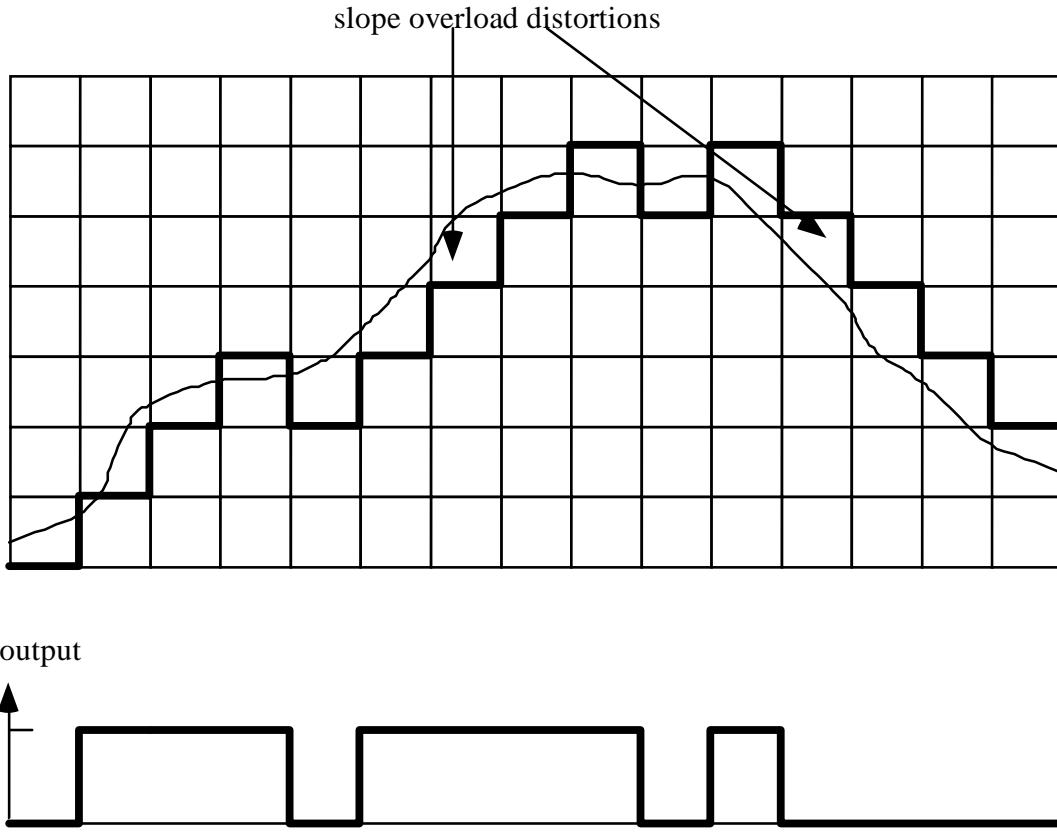
and the maximum slope is  $A 2 \pi f_a$ . To avoid slope overload, we require that

$$\delta f_s > A 2 \pi f_a \quad \text{or} \quad \delta > \frac{2\pi f_a A}{f_s}$$

Source: [COUC01]

- 5.22 a. A total of  $2^8$  quantization levels are possible, so the normalized step size is  $2^{-8} = 0.003906$ .
- b. The actual step size, in volts, is:  $0.003906 \times 10V = 0.03906V$
- c. The maximum normalized quantized voltage is  $1 - 2^{-8} = 0.9961$ . Thus the actual maximum quantized voltage is:  
 $0.9961 \times 10V = 9.961V$
- d. The normalized step size is  $2^{-8}$ . The maximum error that can occur is one-half the step size. Therefore, the normalized resolution is:  
 $\pm 1/2 \times 2^{-8} = \pm 0.001953$
- e. The actual resolution is  $\pm 0.001953 \times 10V = \pm 0.01953V$
- f. The percentage resolution is  $\pm 0.001953 \times 100\% = \pm 0.1953\%$

5.23



$$5.24 \quad s(t) = A_c \cos[2\pi f_c t + \phi(t)] = 10 \cos [(10^8)\pi t + 5 \sin 2\pi(10^3)t]$$

Therefore,  $\phi(t) = 5 \sin 2\pi(10^3)t$ , and the maximum phase deviation is 5 radians. For frequency deviation, recognize that the change in frequency is determined by the derivative of the phase:

$$\phi'(t) = 5 (2\pi) (10^3) \cos 2\pi(10^3)t$$

which yields a frequency deviation of  $\Delta f = (1/2\pi)[ 5 (2\pi) (10^3)] = 5 \text{ kHz}$

$$5.25 \quad \text{a. } s(t) = A_c \cos[2\pi f_c t + n_p m(t)] = 10 \cos [2\pi(10^6)t + 0.1 \sin (10^3)\pi t]$$

$$A_c = 10; f_c = 10^6$$

$$10 m(t) = 0.1 \sin (10^3)\pi t, \text{ so } m(t) = 0.01 \sin (10^3)\pi t$$

$$\text{b. } s(t) = A_c \cos[2\pi f_c t + \phi(t)] = 10 \cos [2\pi(10^6)t + 0.1 \sin (10^3)\pi t]$$

$$A_c = 10; f_c = 10^6$$

$$\phi(t) = 0.1 \sin (10^3)\pi t, \text{ so } \phi'(t) = 100\pi \cos (10^3)\pi t = n_f m(t) = 10 m(t)$$

$$\text{Therefore } m(t) = 10\pi \cos (10^3)\pi t$$

$$5.26 \quad \text{a. For AM, } s(t) = [1 + m(t)] \cos(2\pi f_c t)$$

$$s_1(t) = [1 + m_1(t)] \cos(2\pi f_c t); \quad s_2(t) = [1 + m_2(t)] \cos(2\pi f_c t)$$

$$\text{For the combined signal } m_c(t) = m_1(t) + m_2(t),$$

$$s_c(t) = [1 + m_1(t) + m_2(t)] \cos(2\pi f_c t) = s_1(t) + s_2(t) - 1, \text{ which is a linear combination of } s_1(t) \text{ and } s_2(t).$$

b. For PM,  $s(t) = A \cos(2\pi f_c t + n_p m(t))$

$$s_1(t) = A \cos(2\pi f_c t + n_p m_1(t)); \quad s_2(t) = A \cos(2\pi f_c t + n_p m_2(t))$$

For the combined signal  $m_c(t) = m_1(t) + m_2(t)$ ,

$s_c(t) = A \cos(2\pi f_c t + n_p [m_1(t) + m_2(t)])$ , which is not a linear combination of  $s_1(t)$  and  $s_2(t)$ .

# CHAPTER 6

## DIGITAL DATA COMMUNICATION TECHNIQUES

### ANSWERS TO QUESTIONS

- 6.1 The beginning of a character is signaled by a start bit but with a value of binary zero. A stop (binary one) follows the character.
- 6.2 Asynchronous transmission requires an overhead of two or three bits per character, and is, therefore, significantly less efficient than synchronous transmission.
- 6.3 One possibility is to provide a separate clock line between transmitter and receiver. One side (transmitter or receiver) pulses the line regularly with one short pulse per bit time. The other side uses these regular pulses as a clock. An other alternative is to embed the clocking information in the data signal. For digital signals, this can be accomplished with Manchester or differential Manchester encoding. For analog signals, a number of techniques can be used; for example, the carrier frequency itself can be used to synchronize the receiver based on the phase of the carrier.
- 6.4 A check bit appended to an array of binary digits to make the sum of all the binary digits, including the check bit, always odd (odd parity) or always even (even parity).
- 6.5 An error detecting code in which the code is the remainder resulting from dividing the bits to be checked by a predetermined binary number.
- 6.6 The CRC has more bits and therefore provides more redundancy. That is, it provides more information that can be used to detect errors.
- 6.7 Modulo 2 arithmetic, polynomials, and digital logic.
- 6.8 It is possible. You could design a code in which all codewords are at least a distance of 3 from all other codewords, allowing all single-bit errors to be corrected. Suppose that some but not all codewords in this code are at least a distance of 5 from all other codewords. Then for those particular codewords, but not the others, a double-bit error could be corrected.
- 6.9 An  $(n, k)$  block code encodes  $k$  data bits into  $n$ -bit codewords.
- 6.10 Data circuit-terminating equipment (DCE) mediates between a user's data terminal equipment (DTE) and a network or transmission facility.

## ANSWERS TO PROBLEMS

- 6.1 a. Each character has 25% overhead. For 10,000 characters, there are 20,000 extra bits. This would take an extra  $20,000/2400 = 8.33$  seconds.  
 b. The file takes 10 frames or 480 additional bits. The transmission time for the additional bits is  $480/2400 = 0.2$  seconds.  
 c. Ten times as many extra bits and ten times as long for both.  
 d. The number of overhead bits would be the same, and the time would be decreased by a factor of 4 =  $9600/2400$ .
- 6.2 For each case, compute the fraction g of transmitted bits that are data bits. Then the maximum effective data rate R is:  $R = gx$ , where x is the data rate on the line.
- There are 7 data bits, 1 start bit, 1.5 stop bits, and 1 parity bit  

$$g = 7/(1 + 7 + 1 + 1.5) = 7/10.5 = 0.67$$
  

$$R = 0.67x$$
  - Each frame contains 48 control bits + 128 information bits = 176 bits. The number of characters is  $128/8 = 16$ , and the number of data bits is  $16 \times 7 = 112$ .  

$$R = (112/176)B = 0.64x$$
  - Each frame contains  $48 + 1024 = 1072$  bits. The number of characters is  $1024/8 = 128$ , and the number of data bits is  $128 \times 7 = 896$ .  

$$R = (896/1072)B = 0.84x$$
- 6.3 Use 1-bit START and STOP bits. Write down a few dozen characters. Choose a zero in the first characters as a START bit, count out eight bits, call the next bit STOP (even if it is a zero), look for the next zero, and call that the START bit of the next character. Since some 1's will intervene before you find that zero, you will have moved the starting point of the framing process. Eventually, you will achieve proper framing.
- 6.4 Not for asynchronous transmission. The stop bit is needed so that the start bit can be recognized as such. The start bit is the synchronization event, but it must be recognizable. The start bit is always a 0, and the stop bit is always a 1, which is also the idle state of the line. When a start bit occurs, it is guaranteed to be different from the current state of the line.
- 6.5 Let the bit duration be T. Then a frame is  $12T$  long. Let a clock period be  $T'$ . The last bit (bit 12) is sampled at  $11.5T'$ . For a fast running clock, the condition to satisfy is

$$11.5T' > 11T \Rightarrow \frac{T}{T'} < \frac{11.5}{11} = 1.045 \Rightarrow f_{clock} < 1.045f_{bit}$$

For a slow running clock, the condition to satisfy is

$$11.5T' < 12T \Rightarrow \frac{T}{T'} > \frac{11.5}{12} = 0.958 \Rightarrow f_{clock} > 0.958f_{bit}$$

Therefore, the overall condition:  $0.958 f_{bit} < f_{clock} < 1.045 f_{bit}$

- 6.6 In worst case conditions, the two clocks will drift in opposite directions. The resultant accuracy is 2 minutes in 1 year or:  
$$2/(60 \times 24 \times 365) = 0.0000038$$
The allowable error is 0.4  
Therefore, number of bits is  $0.4/0.0000038 = 105,000$  bits

6.7 The inclusion of a parity bit extends the message length. There are more bits that can be in error since the parity bit is now included. The parity bit may be in error when there are no errors in the corresponding data bits. Therefore, the inclusion of a parity bit with each character would change the probability of receiving a correct message.

6.8 Any arithmetic scheme will work if applied in exactly the same way to the forward and reverse process. The modulo 2 scheme is easy to implement in circuitry. It also yields a remainder one bit smaller than binary arithmetic.

6.9 a. We have:

$\Pr [\text{single bit in error}] = 10^{-3}$

$\Pr [\text{single bit not in error}] = 1 - 10^{-3} = 0.999$

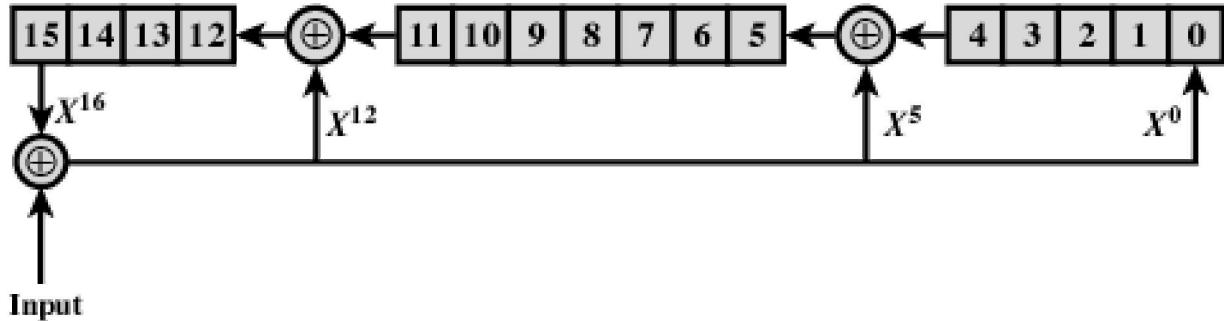
$\Pr [8 \text{ bits not in error}] = (1 - 10^{-3})^8 = (0.999)^8 = 0.992$

$\Pr [\text{at least one error in frame}] = 1 - (1 - 10^{-3})^8 = 0.008$

b.  $\Pr [\text{at least one error in frame}] = 1 - (1 - 10^{-3})^{10} = 1 - (0.999)^{10} = 0.01$

6.10a.

b.



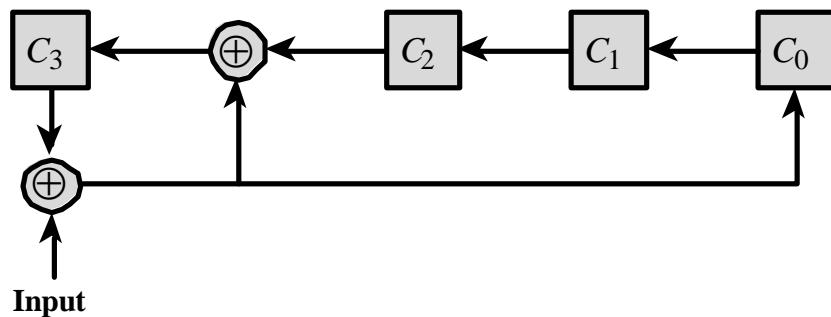
Shift	Shift Register																Input
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1	1
2	0	0	1	0	0	0	0	0	0	1	0	0	0	1	0	0	0
3	0	1	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0
4	1	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0
5	0	0	0	1	0	0	1	0	0	0	1	1	0	0	0	1	0
6	0	0	1	0	0	1	0	0	0	1	1	0	0	0	1	0	0
7	0	1	0	0	1	0	0	0	1	1	0	0	0	1	0	0	0
8	1	0	0	1	0	0	0	1	1	0	0	0	1	0	0	0	0
9	0	0	1	1	0	0	1	1	0	0	1	1	0	0	0	1	0
10	0	1	1	0	0	1	1	0	0	1	1	0	0	0	1	0	0
11	1	1	0	0	1	1	0	0	1	1	0	0	0	1	0	0	0
12	1	0	0	0	1	0	0	1	1	0	1	0	1	0	0	1	0
13	0	0	0	0	0	0	1	1	0	1	1	1	0	0	1	1	0
14	0	0	0	0	0	1	1	0	1	1	1	0	0	1	1	0	0
15	0	0	0	0	1	1	0	1	1	1	0	0	1	1	0	0	0
16	0	0	0	1	1	0	1	1	1	0	0	1	1	0	0	0	0
	CRC																

- 6.11 At the conclusion of the data transfer, just before the CRC pattern arrives, the shift register should contain the identical CRC result. Now, the bits of the incoming CRC are applied at point C<sub>4</sub> (Figure 6.5). Each 1 bit will merge with a 1 bit (exclusive-or) to produce a 0; each 0 bit will merge with a 0 bit to produce a zero.

6.12

$$\begin{array}{r}
 \begin{array}{c} 10110110 \\ \hline 110011 \end{array} \overline{\quad} \\
 \begin{array}{c} 1110001100000 \\ -110011 \\ \hline 101111 \\ -110011 \\ \hline 111000 \\ -110011 \\ \hline 101100 \\ -110011 \\ \hline 111110 \\ -110011 \\ \hline \text{CRC} = 11010 \end{array}
 \end{array}$$

6.13 a.



b. Data = 1 0 0 1 1 0 1 1 1 0 0

$M(X) = 1 + X^3 + X^4 + X^6 + X^7 + X^8$

$X^4M(X) = X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^4$

$$\frac{X^4M(X)}{P(X)} = X^{12} + X^{11} + X^{10} + X^8 + X^7 + \frac{X^2}{P(X)}$$

$R(X) = X^2$

$T(X) = X^4M(X) + R(X) = X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^4 + X^2$

Code = 0 0 1 0 1 0 0 1 1 0 1 1 0 0

c. Code = 0 0 1 0 1 0 0 0 1 0 1 1 1 0 0

$\frac{T(X)}{P(X)}$  yields a nonzero remainder

- 6.14 a. Divide  $X^{10} + X^7 + X^4 + X^3 + X + 1$  by  $X^4 + X + 1$ . The remainder is  $X^3 + X^2$ . The CRC bits are 1100. The string 100100110111100 is sent.  
 b. The string 000110110111100 is received, corresponding to  $X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^3 + X^2$ . The remainder after division by  $X^4 + X + 1$  is  $X^3 + X^2 + X$ ,

- which is nonzero. The errors are detected.
- c. The string 000010110111100 is received, corresponding to  $X^{10} + X^8 + X^7 + X^5 + X^4 + X^3 + X^2$ . The remainder after division by  $X^4 + X + 1$  is zero. The errors are not detected.
- 6.15 a. The multiplication of  $M(X)$  by  $X^{16}$  corresponds to shifting  $M(X)$  16 places and thus providing the space for a 16-bit FCS. The addition of  $X^k L(X)$  to  $X^{16} M(X)$  inverts the first 16 bits of  $G(X)$  (one's complements). The addition of  $L(X)$  to  $R(X)$  inverts all of the bits of  $R(X)$ .
- b. The HDLC standard provides the following explanation. The addition of  $X^k L(X)$  corresponds to a value of all ones. This addition protects against the obliteration of leading flags, which may be non-detectable if the initial remainder is zero. The addition of  $L(X)$  to  $R(X)$  ensures that the received, error-free message will result in a unique, non-zero remainder at the receiver. The non-zero remainder protects against the potential non-detectability of the obliteration of trailing flags.
- c. The implementation is the same as that shown in Solution 6.10b, with the following strategy. At both transmitter and receiver, the initial content of the register is preset to all ones. The final remainder, if there are no errors, will be 0001 1101 0000 1111.

6.16

a.

	00000	10101	01010
00000	0	2	2
10101	3	0	5
01010	2	5	0

b.

	000000	010101	101010	110110
000000	0	3	3	4
010101	3	0	6	6
101010	3	6	0	3
110110	4	6	3	0

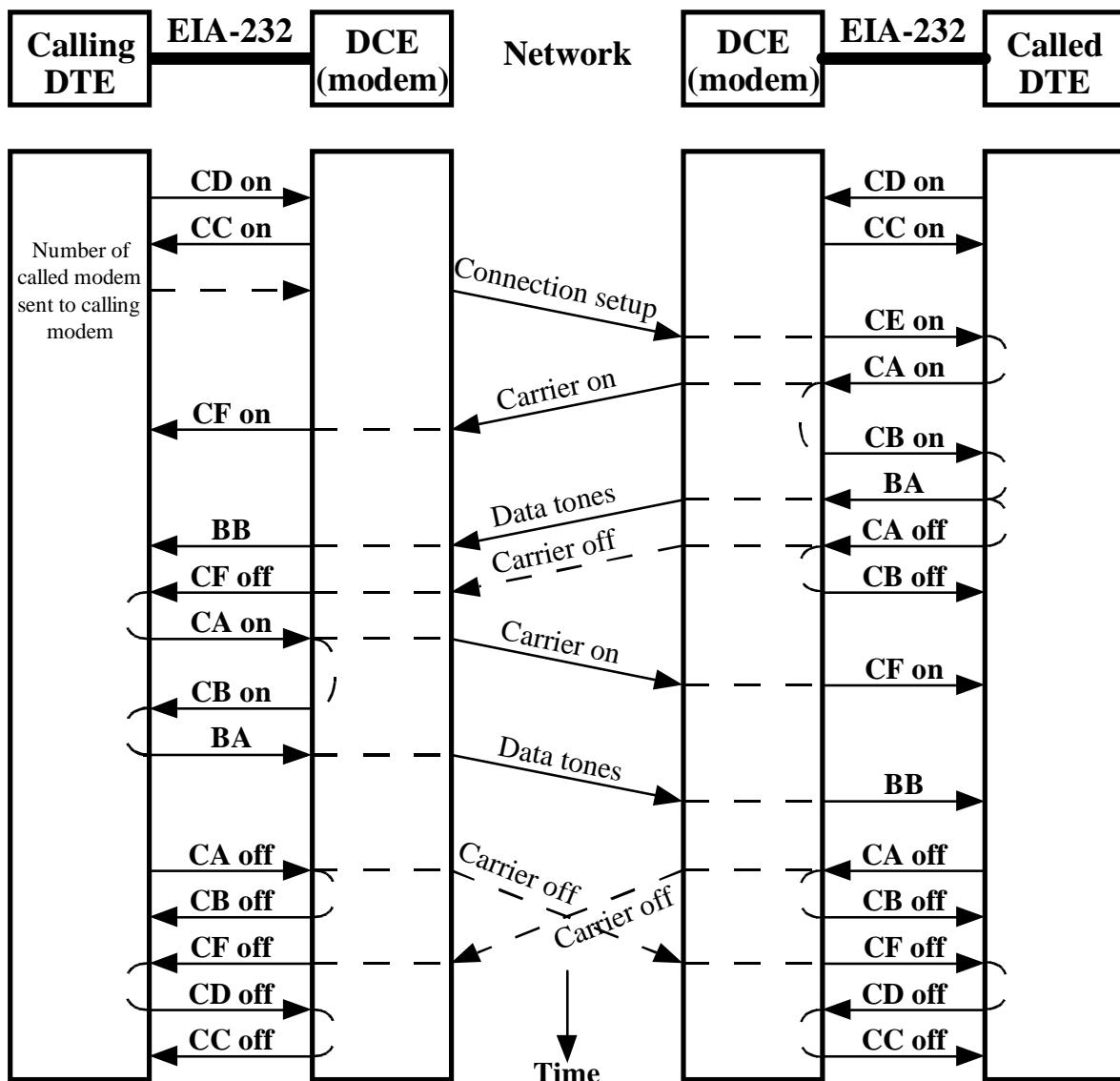
6.17 a.  $p(v|w) = \beta^{d(w,v)}(1 - \beta)^{(n - d(w,v))}$

b. If we write  $d_i = d(w_i, v)$ , then  $\frac{p(v|w_1)}{p(v|w_2)} = \frac{\beta^{d_1} (1 - \beta)^{n-d_1}}{\beta^{d_2} (1 - \beta)^{n-d_2}} = \left(\frac{1-\beta}{\beta}\right)^{d_2-d_1}$

c. If  $0 < \beta < 0.5$ , then  $(1 - \beta)/\beta > 1$ . Therefore, by the equation of part b,  $p(v|w_1)/p(v|w_2) > 1$  if and only if  $d_1 < d_2$ .

- 6.18 Suppose that the minimum distance between codewords is at least  $2t + 1$ . For a codeword  $w$  to be decoded as another codeword  $w'$ , the received sequence must be at least as close to  $w'$  as to  $w$ . For this to happen, at least  $t + 1$  bits of  $w$  must be in error. Therefore all errors involving  $t$  or fewer digits are correctable.

6.19



- 6.20 If a device asserts Request to Send, it will get back a Clear to Send and the other device will get a Carrier Detect. If a device asserts Data Terminal Ready, the other device is alerted with a Data Set Ready and a Ring Indicator. Data transmitted by one side is received by the other. In order to operate a synchronous data link without a modem, clock signals need to be supplied. The transmitter and Receive Timing leads are cross-connected for this purpose.
- 6.21 Circuit SD (Send Data) and Circuit RD (Receive Data) are disconnected or isolated from the remote DTE at the interface and connected to each other in the remote DCE.

## CHAPTER 7

# DATA LINK CONTROL PROTOCOLS

### ANSWERS TO QUESTIONS

- 7.1 Frame synchronization: The beginning and end of each frame must be recognizable. Flow control: The sending station must not send frames at a rate faster than the receiving station can absorb them. Error control: Bit errors introduced by the transmission system should be corrected. Addressing: On a multipoint line, such as a local area network (LAN), the identity of the two stations involved in a transmission must be specified. Control and data on same link: The receiver must be able to distinguish control information from the data being transmitted. Link management: The initiation, maintenance, and termination of a sustained data exchange require a fair amount of coordination and cooperation among stations. Procedures for the management of this exchange are required.
- 7.2 The function performed by a receiving entity to limit the amount or rate of data that is sent by a transmitting entity.
- 7.3 A flow control protocol in which the sender transmits a block of data and then awaits an acknowledgment before transmitting the next block.
- 7.4 (1) The buffer size of the receiver may be limited. (2) The longer the transmission, the more likely that there will be an error, necessitating retransmission of the entire frame. With smaller frames, errors are detected sooner, and a smaller amount of data needs to be retransmitted. (3) On a shared medium, such as a LAN, it is usually desirable not to permit one station to occupy the medium for an extended period, thus causing long delays at the other sending stations.
- 7.5 A method of flow control in which a transmitting station may send numbered packets within a window of numbers. The window changes dynamically to allow additional packets to be sent.
- 7.6 The stop & wait approach requires acknowledgments after each frame. The sliding window flow control technique can send multiple frames before waiting for an acknowledgment. Efficiency can be greatly improved by allowing multiple frames to be in transit at the same time.
- 7.7 The inclusion of an acknowledgment to a previously received packet in an outgoing data packet.
- 7.8 Error control refers to mechanisms to detect and correct errors that occur in the transmission of frames.

- 7.9 Error detection; positive acknowledgment; retransmission after timeout; negative acknowledgment.
- 7.10 A feature that automatically initiates a request for retransmission when an error in transmission is detected.
- 7.11 Stop-and-wait ARQ: Based on stop-and-wait flow control. A station retransmits on receipt of a duplicate acknowledgment or as a result of a timeout. Go-back-N ARQ: Based on sliding-window flow control. When an error is detected, the frame in question is retransmitted, as well as all subsequent frames that have been previously transmitted. Selective-reject ARQ. Based on sliding-window flow control. When an error is detected, only the frame in question is retransmitted.
- 7.12 Primary station: Responsible for controlling the operation of the link. Frames issued by the primary are called commands. Secondary station: Operates under the control of the primary station. Frames issued by a secondary are called responses. The primary maintains a separate logical link with each secondary station on the line. Combined station: Combines the features of primary and secondary. A combined station may issue both commands and responses.
- 7.13 Normal response mode (NRM): Used with an unbalanced configuration. The primary may initiate data transfer to a secondary, but a secondary may only transmit data in response to a command from the primary. Asynchronous balanced mode (ABM): Used with a balanced configuration. Either combined station may initiate transmission without receiving permission from the other combined station. Asynchronous response mode (ARM): Used with an unbalanced configuration. The secondary may initiate transmission without explicit permission of the primary. The primary still retains responsibility for the line, including initialization, error recovery, and logical disconnection.
- 7.14 The flag field delimits the beginning and end of a frame.
- 7.15 Data transparency refers to the ability to include arbitrary bit patterns in the data field of a frame without any pattern being confused with part of the control information in the frame. This is achieved by bit stuffing.
- 7.16 Information frames (I-frames) carry the data to be transmitted for the user (the logic above HDLC that is using HDLC). Additionally, flow and error control data, using the ARQ mechanism, are piggybacked on an information frame. Supervisory frames (S-frames) provide the ARQ mechanism when piggybacking is not used. Unnumbered frames (U-frames) provide supplemental link control functions.

## ANSWERS TO PROBLEMS

- 7.1 a. Because only one frame can be sent at a time, and transmission must stop until an acknowledgment is received, there is little effect in increasing the size of the message if the frame size remains the same. All that this would affect is connect and disconnect time.
- b. Increasing the number of frames would decrease frame size (number of bits/frame). This would lower line efficiency, because the propagation time is unchanged but more acknowledgments would be needed.
- c. For a given message size, increasing the frame size decreases the number of frames. This is the reverse of (b).

7.2 Let L be the number of bits in a frame. Then, using Equation 7.5 of Appendix 7A:

$$a = \frac{\text{Propagation Delay}}{\text{Transmission Time}} = \frac{20 \times 10^{-3}}{L/(4 \times 10^3)} = \frac{80}{L}$$

Using Equation 7.4 of Appendix 7A:

$$U = \frac{1}{1+2a} = \frac{1}{1+(160/L)} \geq 0.5$$

$$L \geq 160$$

Therefore, an efficiency of at least 50% requires a frame size of at least 160 bits.

7.3  $a = \frac{\text{Propagation Delay}}{L/R} = \frac{270 \times 10^{-3}}{10^3/10^6} = 270$

- a.  $U = 1/(1+2a) = 1/541 = 0.002$
- b. Using Equation 7.6:  $U = W/(1+2a) = 7/541 = 0.013$
- c.  $U = 127/541 = 0.23$
- d.  $U = 255/541 = 0.47$

7.4 A → B: Propagation time =  $4000 \times 5 \mu\text{sec} = 20 \text{ msec}$

$$\text{Transmission time per frame} = \frac{1000}{100 \times 10^3} = 10 \text{ msec}$$

B → C: Propagation time =  $1000 \times 5 \mu\text{sec} = 5 \text{ msec}$

$$\text{Transmission time per frame} = x = 1000/R$$

R = data rate between B and C (unknown)

A can transmit three frames to B and then must wait for the acknowledgment of the first frame before transmitting additional frames. The first frame takes 10 msec to transmit; the last bit of the first frame arrives at B 20 msec after it was transmitted, and therefore 30 msec after the frame transmission began. It will take an additional 20 msec for B's acknowledgment to return to A. Thus, A can transmit 3 frames in 50 msec.

B can transmit one frame to C at a time. It takes  $5 + x$  msec for the frame to be received at C and an additional 5 msec for C's acknowledgment to return to A.

Thus, B can transmit one frame every  $10 + x$  msec, or 3 frames every  $30 + 3x$  msec.  
Thus:

$$30 + 3x = 50$$

$$x = 6.66 \text{ msec}$$

$$R = 1000/x = 150 \text{ kbps}$$

### 7.5 Round trip propagation delay of the link = $2 \times L \times t$

Time to transmit a frame =  $B/R$

To reach 100% utilization, the transmitter should be able to transmit frames continuously during a round trip propagation time. Thus, the total number of frames transmitted without an ACK is:

$$N = \left\lceil \frac{2 \times L \times t}{B/R} + 1 \right\rceil, \quad \text{where } \lceil X \rceil \text{ is the smallest integer greater than or equal to } X$$

This number can be accommodated by an M-bit sequence number with:

$$M = \lceil \log_2(N) \rceil$$

### 7.6 In fact, NAK is not needed at all, since the sender will time out if it fails to receive an ACK. The NAK improves efficiency by informing the sender of a bad frame as early as possible.

### 7.7 Assume a 2-bit sequence number:

1. Station A sends frames 0, 1, 2 to station B.
2. Station B receives all three frames and cumulatively acknowledges with RR 3.
3. Because of a noise burst, the RR 3 is lost.
4. A times out and retransmits frame 0.
5. B has already advanced its receive window to accept frames 3, 0, 1, 2. Thus it assumes that frame 3 has been lost and that this is a new frame 0, which it accepts.

### 7.8 Use the following formulas:

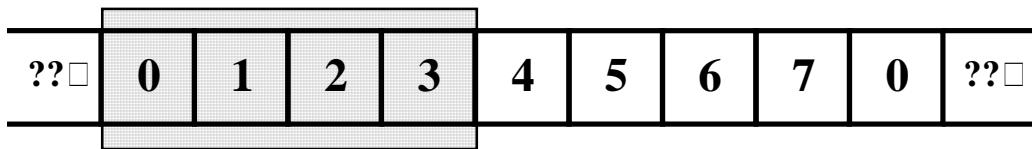
a	0.1	1.	10	100
S&W	$(1 - P)/1.2$	$(1 - P)/3$	$(1 - P)/21$	$(1 - P)/201$
GBN (7)	$(1-P)/(1+0.2P)$	$(1-P)/(1+2P)$	$7(1-P)/21(1+6P)$	$7(1 - P)/201(1+6P)$
GBN (127)	$(1-P)/(1+0.2P)$	$(1-P)/(1+2P)$	$(1 - P)/(1+20P)$	$127(1-P)/201(1+126P)$
SREJ (7)	$1 - P$	$1 - P$	$7(1 - P)/21$	$7(1 - P)/201$

SREJ (127)	1 - P	1 - P	1 - P	127(1 - P)/201
------------	-------	-------	-------	----------------

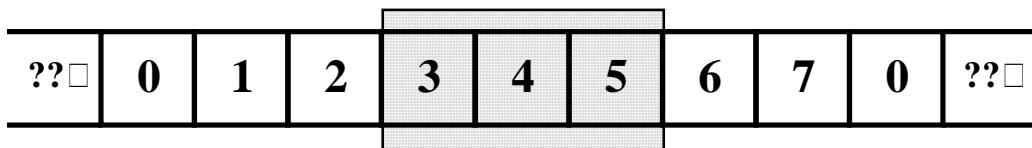
For a given value of  $a$ , the utilization values change very little as a function of  $P$  over a reasonable range (say  $10^{-3}$  to  $10^{-12}$ ). We have the following approximate values for  $P = 10^{-6}$ :

a	0.1	1.0	10	100
Stop-and-wait	0.83	0.33	0.05	0.005
GBN (7)	1.0	1.0	0.33	0.035
GBN (127)	1.0	1.0	1.0	0.63
SREJ (7)	1.0	1.0	0.33	0.035
SREJ (127)	1.0	1.0	1.0	0.63

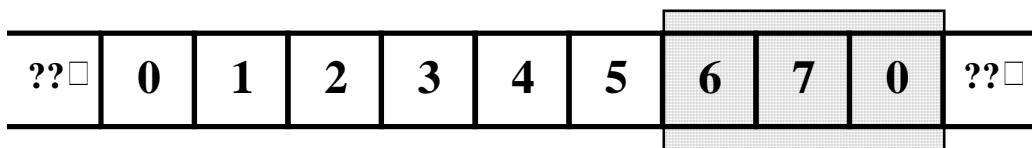
7.9 a.



b.



c.



7.10 A lost SREJ frame can cause problems. The sender never knows that the frame was not received, unless the receiver times out and retransmits the SREJ.

7.11 From the standard: "A SREJ frame shall not be transmitted if an earlier REJ exception condition has not been cleared (To do so would request retransmission of a data frame that would be retransmitted by the REJ operation)." In other words, since the REJ requires the station receiving the REJ to retransmit the rejected frame and all subsequent frames, it is redundant to perform a SREJ on a frame that is already scheduled for retransmission.

Also from the standard: "Likewise, a REJ frame shall not be transmitted if one or more earlier SREJ exception conditions have not been cleared." The REJ frame indicates the acceptance of all frames prior to the frame rejected by the REJ frame. This would contradict the intent of the SREJ frame or frames.

7.12 Let  $t_1$  = time to transmit a single frame

$$t_1 = \frac{1024 \text{ bits}}{10^6 \text{ bps}} = 1.024 \text{ m sec}$$

The transmitting station can send 7 frames without an acknowledgment. From the beginning of the transmission of the first frame, the time to receive the acknowledgment of that frame is:

$$t_2 = 270 + t_1 + 270 = 541.024 \text{ msec}$$

During the time  $t_2$ , 7 frames are sent.

$$\text{Data per frame} = 1024 - 48 = 976$$

$$\text{Throughput} = \frac{7 \times 976 \text{ bits}}{541.024 \times 10^{-3} \text{ sec}} = 12.6 \text{ kbps}$$

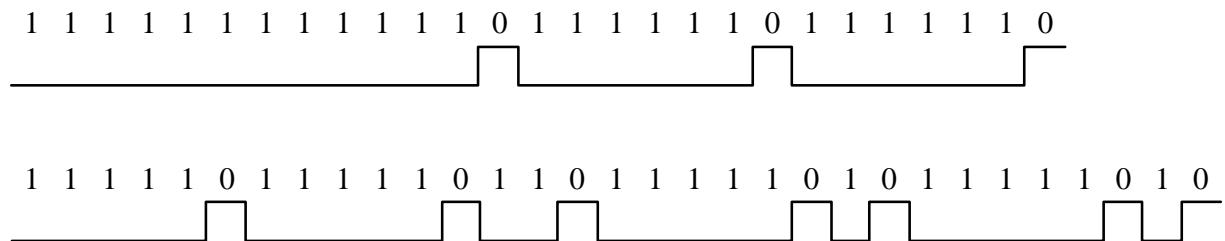
7.13 No, because the field is of known fixed length.

7.14 The following enhancements are possibilities:

- Always transmit an integral number of octets
- Include a length field
- Do not use the same flag to close one frame and open another
- Ignore any frame containing fewer than 32 bits
- Ignore any frame ending in seven or more ones

The length field is used to compare the number of octets received with the number transmitted. Any discrepancies result in discarding the frame. The last three enhancements allow the rejection of frames when the closing flag has been destroyed.

7.15



A problem with NRZ-L is its lack of synchronization capability: a long sequence of 1's or 0's yields a constant output voltage with no transitions. Bit-stuffing at least eliminates the possibility of a long string of 1's.

- 7.16  $N(R) = 2$ . This is the number of the next frame that the secondary station expects to receive.
- 7.17 One example of such a scheme is the multilink procedure (MLP) defined as part of layer 2 of X.25. The same frame format as for LAPB is used, with one additional field:

Flag	Address	Control	MLC	Packet	FCS	Flag
------	---------	---------	-----	--------	-----	------

The multilink control (MLC) field is a 16-bit field that contains a 12-bit sequence number that is unique across all links. The MLC and packet fields form a multilink protocol (MLP) frame. Once an MLP frame is constructed, it is assigned to a particular link and further encapsulated in a LAPB frame, as shown above. The LAPB control field includes, as usual, a sequence number unique to that link.

The MLC field performs two functions. First, LAPB frames sent out over different links may arrive in a different order from that in which they were first constructed by the sending MLP. The destination MLP will buffer incoming frames and reorder them according to MLP sequence number. Second, if repeated attempts to transmit a frame over one link fails, the DTE or DCE will send the frame over one or more other links. The MLP sequence number is needed for duplicate detection in this case.

- 7.18 The selective-reject approach would burden the server with the task of managing and maintaining large amounts of information about what has and has not been successfully transmitted to the clients; the go-back-N approach would be less of a burden on the server.

## CHAPTER 8

# MULTIPLEXING

### ANSWERS TO QUESTIONS

- 8.1 Multiplexing is cost-effective because the higher the data rate, the more cost-effective the transmission facility.
- 8.2 Interference is avoided under frequency division multiplexing by the use of guard bands, which are unused portions of the frequency spectrum between subchannels.
- 8.3 Echo cancellation is a signal processing technique that allows transmission of digital signals in both directions on a single transmission line simultaneously. In essence, a transmitter must subtract the echo of its own transmission from the incoming signal to recover the signal sent by the other side.
- 8.4 Downstream: from the carrier's central office to the customer's site; upstream: from customer to carrier.
- 8.5 A synchronous time division multiplexer interleaves bits from each signal and takes turns transmitting bits from each of the signals in a round-robin fashion.
- 8.6 A statistical time division multiplexer is more efficient than a synchronous time division multiplexer because it allocates time slots dynamically on demand and does not dedicate channel capacity to inactive low speed lines.
- 8.7 The basic difference between North American and international TDM carrier standards is that the North American DS-1 carrier has 24 channels while the international standard is 30 channels. This explains the basic difference between the 1.544 Mbps North American standard and the 2.048 Mbps international standard.
- 8.8 As load increases, the buffer size and delay increase until the load approximates the capacity of the shared channel when both become infinite.

### ANSWERS TO PROBLEMS

- 8.1
  - a. The available bandwidth is  $3100 - 400 = 2700$  Hz. A scheme such as depicted in Figure 8.4 can be used, with each of the four signals modulated onto a different 500-Hz portion of the available bandwidth.
  - b. Each 500-Hz signal can be sampled at a rate of 1 kHz. If 4-bit samples are used, then each signal requires 4 kbps, for a total data rate of 16 kbps. This scheme

will work only if the line can support a data rate of 16 kbps in a bandwidth of 2700 Hz.

- 8.2 In FDM, part of the channel is assigned to a source all of the time. In time-division multiplexing, all of the channel is assigned to the source for a fraction of the time.
- 8.3 In many cases, the cost of the transmission medium is large compared to the cost of a single transmitter/receiver pair or a modulator/demodulator pair. If there is spare bandwidth, then the incremental cost of the transmission can be negligible. The new station pair is simply added to an unused subchannel. If there is no unused subchannel it may be possible to redivide the existing subchannels creating more subchannels with less bandwidth. If, on the other hand, a new pair causes a complete new line to be added, then the incremental cost is large indeed.
- 8.4 Although it seems logical to think of bits being separated as they come in and then switched unchanged onto the transmission channel, this is not necessarily the way it happens. What the multiplexer receives from attached stations are several bit streams from different sources. What the multiplexer sends over the multiplexed transmission line is a bit stream from the multiplexer. As long as the multiplexer sends what can be interpreted as a bit stream to the demultiplexer at the other end, the system will work. The multiplexer, for example, may use a self-clocking signal. The incoming stream may be, on the other hand, encoded in some other format. The multiplexer receives and understands the incoming bits and sends out its equivalent set of multiplexed bits.
- 8.5 The purpose of the start and stop bits is to delimit the data bits of a character in asynchronous transmission. In synchronous TDM, using character interleaving, the character is placed in a time slot that is one character wide. The character is delimited by the bounds of the time slot, which are defined by the synchronous transmission scheme. Thus, no further delimiters are needed. When the character arrives at its destination, the start and stop bits can be added back if the receiver requires these.
- 8.6 Synchronous TDM is a technique to divide the medium to which it is applied into time slots which are used by multiple inputs. TDM's focus is on the medium rather than the information which travels on the medium. Its services should be transparent to the user. It offers no flow or error control. These must be provided on an individual-channel basis by a link control protocol.
- 8.7 This bit carries must carry a repetitive bit pattern that enables the receiver to determine whether or not it has lost synchronization. The actual bit pattern is 01010101... If a receiver gets out of synchronization it can scan for this pattern and resynchronize. This pattern would be unlikely to occur in digital data. Analog sources cannot generate this pattern. It corresponds to a sine wave at 4,000 Hz and would be filtered out from a voice channel that is band limited.

- 8.8 There is one control bit per channel per six frames. Each frame lasts 125  $\mu$ sec. Thus:

$$\text{Data Rate} = 1/(6 \times 125 \times 10^{-6}) = 1.33 \text{ kbps}$$

- 8.9 Assuming 4 kHz per voice signal, the required bandwidth for FDM is  $24 \times 4 = 96$  kHz. With PCM, each voice signal requires a data rate of 64 kbps, for a total data rate of  $24 \times 64 = 1.536$  Mbps. At 1 bps/Hz, this requires a bandwidth of 1.536 MHz.
- 8.10 The structure is that of Figure 8.8, with one analog signal and four digital signals. The 500-Hz analog signal is converted into a PAM signal at 1 kHz; with 4-bit encoding, this becomes a 4-kbps PCM digital bit stream. A simple multiplexing technique is to use a 260-bit frame, with 200 bits for the analog signal and 15 bits for each digital signal, transmitted at a rate of 5.2 kbps or 20 frames per second. Thus the PCM source transmits at  $(20 \text{ frames/sec}) \times (200 \text{ bits/frame}) = 4000 \text{ bps}$ . Each digital source transmits at  $(20 \text{ frames/sec}) \times (15 \text{ bits/frame}) = 300 \text{ bps}$ .

- 8.11 a.  $n = 7 + 1 + 1 + 2 = 11 \text{ bits/character}$   
 b. Available capacity =  $2400 \times 0.97 = 2328 \text{ bps}$

If we use 20 terminals sending one character at a time in TDM plus a synchronization character, the total capacity used is:

$$21 \times 110 \text{ bps} = 2310 \text{ bps available capacity}$$

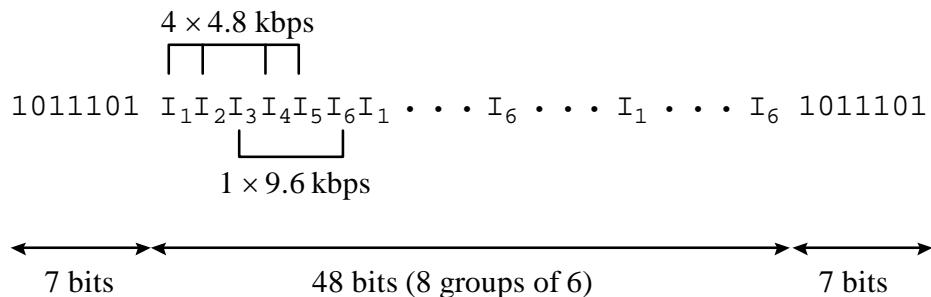
- c. One SYN character, followed by 20 11-bit terminal characters, followed by stuff bits.

- 8.12 The capacity of the T1 line is 1.544 Mbps. The available capacity is  $1.544 \times 0.99 = 1.52856 \text{ Mbps} = \text{AC}$ .
- a.  $\text{AC}/110 = 13,896$
  - b.  $\text{AC}/300 = 5,095$
  - c.  $\text{AC}/1200 = 1273$
  - d.  $\text{AC}/9600 = 159$
  - e.  $\text{AC}/64000 = 23$

If the sources were active only 10% of the time, a statistical multiplexer could be used to boost the number of devices by a factor of about seven or eight in each case. This is a practical limit based on the performance characteristics of a statistical multiplexer.

- 8.13 Synchronous TDM:  $9600 \text{ bps} \times 10 = 96 \text{ kbps}$   
 Statistical TDM:  $9600 \text{ bps} \times 10 \times 0.5/0.8 = 60 \text{ kbps}$

- 8.14 a.



- b.  $7/(48 + 7) \times 100 = 12.7\%$
- c.  $(6 \times 4.8 \text{ kbps}) \times ((48 + 7)/48) = 33 \text{ kbps} = R_0$
- d. If the receiver is on the framing pattern (no searching), the minimum reframe time is 12 frame times (the algorithm takes 12 frames to decide it is "in frame").

$$\text{Frame time} = T_f = (55 \text{ bits/frame}) / (R_0 \text{ seconds/bit}) = 1.67 \text{ ms}$$

$$\text{Minimum reframe time} = 12T_f = 20 \text{ ms}$$

For maximum reframe time, the system is at the worst possible position, having just missed the framing pattern. Hence it must search the maximum number of bits (55) to find it. Each search takes  $12T_f$ . Therefore,

$$\text{Maximum reframe time} = 55(12T_f) = 1.1 \text{ s.}$$

Assuming the system is random, the reframing is equally likely to start on any bit position. Hence on the average it starts in the middle or halfway between the best and worst cases.

$$\text{Average reframe time} = (1.1 + 0.02)/2 = 0.56 \text{ s}$$

- 8.15 The four terminals could easily be multiplexed onto one voice grade line. Therefore, the channel cost will be only one-fourth, since one channel rather than four is now needed. The same reasoning applies to termination charges.  
The present solution requires eight low speed modems (four pairs of modems). The new solution requires two higher-speed modems and two multiplexers.  
The reliability of the multiplexed solution may be somewhat less. The new system does not have the redundancy of the old system. A failure anywhere except at the terminals will cause a complete loss of the system.

- 8.16 No. Each multiplexer also acts as a buffer. It can accept bits in asynchronous form, buffer them and transmit them in synchronous form, and vice versa.

- 8.17 Voice sampling rate =  $2 \times 4 \text{ kHz} = 8 \text{ kHz}$ ; 6 bits/sample

Thus:	30 voices channels:	$30 \times 8 \times 6 =$	1440 kbps
	1 synchronous bit/channel:	$30 \times 8 =$	240 kbps
	1 synchronous bit/frame:	$1 \times 8 =$	8 kbps
TOTAL			1688 kbps

8.18 a. Assume a continuous stream of STDM frames. Then:

Bit rate for data portion of frame =  $L$  bits/second

Frame rate in frames per second =  $(C \text{ bits/second})/(F \text{ bits/frame})$

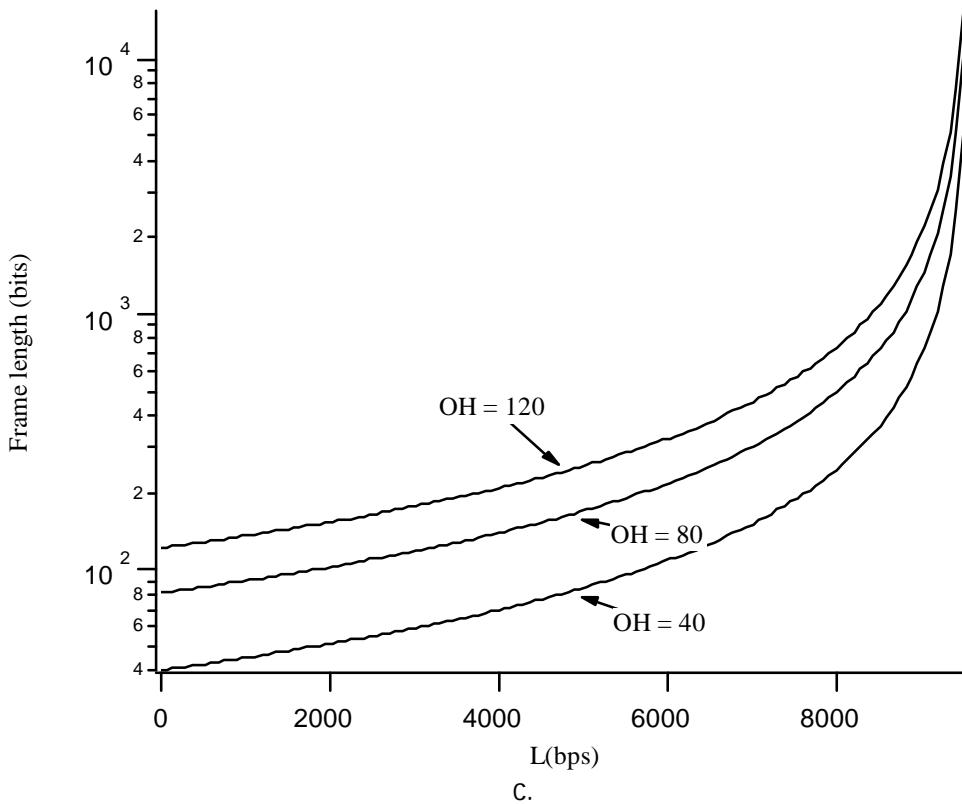
Bit rate for overhead =  $(OH \text{ bits/frame}) \times (C/F \text{ frames/second})$

Total data rate =  $C = L + ((C \times OH)/F)$  bits/second

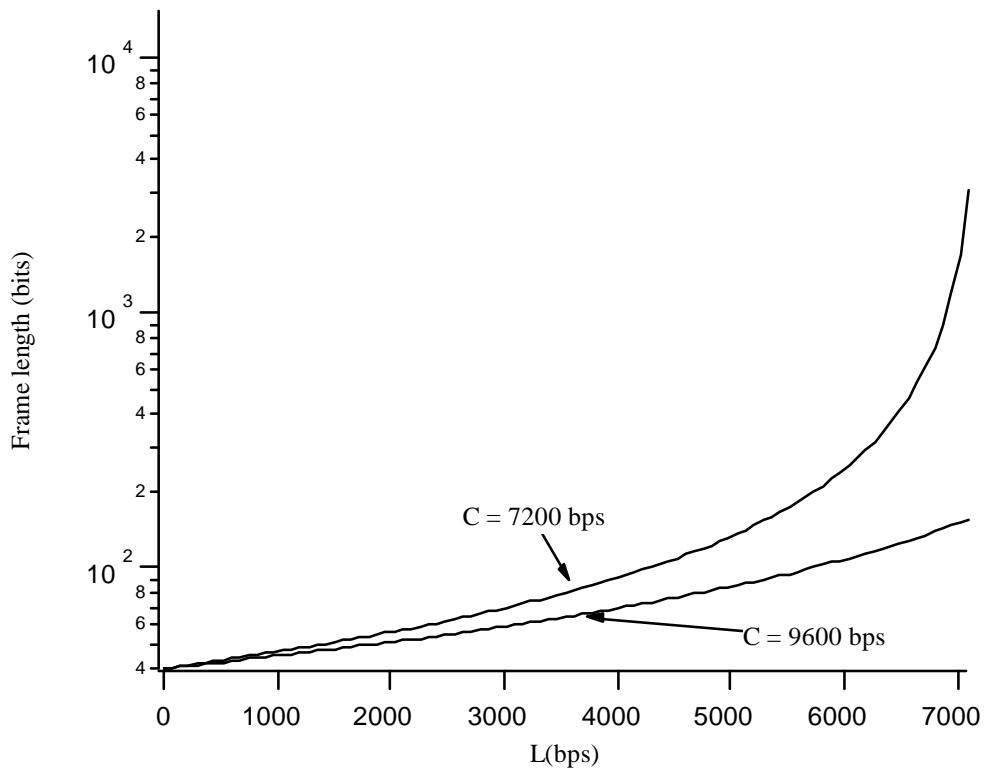
$F = (C \times OH)/(C - L)$  bits

If we fix the number of overhead bits ( $OH$ ), we can vary the percent of overhead by varying  $F$ .

b.



C.



- 8.19 A field can be delimited by a count or by a delimiter that does not occur in the data. If a delimiter is used, bit or character-stuffing may be needed.

## CHAPTER 9

### SPREAD SPECTRUM

### ANSWERS TO QUESTIONS

- 9.1 The bandwidth is wider after the signal has been encoded using spread spectrum.
- 9.2 (1) We can gain immunity from various kinds of noise and multipath distortion. (2) It can also be used for hiding and encrypting signals. Only a recipient who knows the spreading code can recover the encoded information. (3) Several users can independently use the same higher bandwidth with very little interference, using code division multiple access (CDMA).
- 9.3 With frequency hopping spread spectrum (FHSS), the signal is broadcast over a seemingly random series of radio frequencies, hopping from frequency to frequency at fixed intervals. A receiver, hopping between frequencies in synchronization with the transmitter, picks up the message.
- 9.4 Slow FHSS = multiple signal elements per hop; fast FHSS = multiple hops per signal element.
- 9.5 With direct sequence spread spectrum (DSSS), each bit in the original signal is represented by multiple bits in the transmitted signal, using a spreading code.
- 9.6 For an  $N$ -bit spreading code, the bit rate after spreading (usually called the chip rate) is  $N$  times the original bit rate.
- 9.7 CDMA allows multiple users to transmit over the same wireless channel using spread spectrum. Each user uses a different spreading code. The receiver picks out one signal by matching the spreading code.

### ANSWERS TO PROBLEMS

- 9.1 a. We have  $C = B \log_2 (1 + \text{SNR})$ . For  $\text{SNR} = 0.1$ ,  $C = 0.41 \text{ MHz}$ ; For  $\text{SNR} = 0.01$ ,  $C = 3.9 \text{ MHz}$ ; for  $\text{SNR} = 0.001$ ,  $C = 38.84 \text{ MHz}$ . Thus, to achieve the desired SNR, the signal must be spread so that 56 KHz is carried in very large bandwidths.
- b. For 1 bps/Hz, the equation  $C = B \log_2 (1 + \text{SNR})$  becomes  $\log_2 (1 + \text{SNR}) = 1$ . Solving for SNR, we have  $\text{SNR} = 1$ . Thus a far higher SNR is required without spread spectrum.
- 9.2 The total number of tones, or individual channels is:  

$$W_s/f_d = (400 \text{ MHz})/(100 \text{ Hz}) = 4 \times 10^6$$

The minimum number of PN bits =  $\lceil \log_2 (4 \times 10^6) \rceil = 22$

where  $\lceil x \rceil$  indicates the smallest integer value not less than x. Source: [SKLA01]

9.3  $W_s = 1000 f_d$ ;  $W_d = 4 f_d$ ; Using Equation 7.3,  $G_p = W_s/W_d = 250 = 24 \text{ dB}$

- 9.4 a. Period of the PN sequence is  $2^4 - 1 = 15$   
 b. MFSK  
 c.  $L = 2$   
 d.  $M = 2^L = 4$   
 e.  $k = 3$   
 f. slow FHSS  
 g.  $2^k = 8$   
 h.

Time	0	1	2	3	4	5	6	7	8	9	10	11
Input data	0	1	1	1	1	1	0	0	0	1	1	0
Frequency	$f_1$		$f_3$		$f_3$		$f_2$		$f_0$		$f_2$	

Time	12	13	14	15	16	17	18	19
Input data	0	1	1	1	1	0	1	0
Frequency	$f_1$		$f_3$		$f_2$		$f_2$	

Source: [HAYK01]

- 9.5 a. Period of the PN sequence is  $2^4 - 1 = 15$   
 b. MFSK  
 c.  $L = 2$   
 d.  $M = 2^L = 4$   
 e.  $k = 3$   
 f. fast FHSS  
 g.  $2^k = 8$   
 h. Same as for Problem 9.4

- 9.6 a. This is from the example in Section 6.2.

$$\begin{array}{llll} f_1 = 75 \text{ kHz } 000 & f_2 = 125 \text{ kHz } 001 & f_3 = 175 \text{ kHz } 010 & f_4 = 225 \text{ kHz } 011 \\ f_5 = 275 \text{ kHz } 100 & f_6 = 325 \text{ kHz } 101 & f_7 = 375 \text{ kHz } 110 & f_8 = 425 \text{ kHz } 111 \end{array}$$

- b. We need three more sets of 8 frequencies. The second set can start at 475 kHz, with 8 frequencies separated by 50 kHz each. The third set can start at 875 kHz, and the fourth set at 1275 kHz.

- 9.7. a.  $C_0 = 1110010$ ;  $C_1 = 0111001$ ;  $C_2 = 1011100$ ;  $C_3 = 0101110$ ;  $C_4 = 0010111$ ;  
 $C_5 = 1001011$ ;  $C_6 = 1100101$   
 b. C1 output = -7; bit value = 0  
 c. C2 output = +9; bit value = 1

- 9.8 Let us start with an initial seed of 1. The first generator yields the sequence:

1, 6, 10, 8, 9, 2, 12, 7, 3, 5, 4, 11, 1, ...

The second generator yields the sequence:

1, 7, 10, 5, 9, 11, 12, 6, 3, 8, 4, 2, 1, ...

Because of the patterns evident in the second half of the latter sequence, most people would consider it to be less random than the first sequence.

- 9.9 When  $m = 2^k$ , the right-hand digits of  $X_n$  are much less random than the left-hand digits. See [KNUT98], page 13 for a discussion.
- 9.10 Many packages make use of a linear congruential generator with  $m = 2^k$ . As discussed in the answer to Problem 9.9, this leads to results in which the right-hand digits are much less random than the left-hand digits. Now, if we use a linear congruential generator of the following form:

$$X_{n+1} = (aX_n + c) \bmod m$$

then it is easy to see that the scheme will generate all even integers, all odd integers, or will alternate between even and odd integers, depending on the choice for  $a$  and  $c$ . Often,  $a$  and  $c$  are chosen to create a sequence of alternating even and odd integers. This has a tremendous impact on the simulation used for calculating  $\pi$ . The simulation depends on counting the number of pairs of integers whose greatest common divisor is 1. With truly random integers, one-fourth of the pairs should consist of two even integers, which of course have a gcd greater than 1. This never occurs with sequences that alternate between even and odd integers. To get the correct value of  $\pi$  using Cesaro's method, the number of pairs with a gcd of 1 should be approximately 60.8%. When pairs are used where one number is odd and the other even, this percentage comes out too high, around 80%, thus leading to the too small value of  $\pi$ . For a further discussion, see Danilowicz, R. "Demonstrating the Dangers of Pseudo-Random Numbers," *SIGCSE Bulletin*, June 1989.

# CHAPTER 10

## CIRCUIT SWITCHING AND PACKET SWITCHING

### ANSWERS TO QUESTIONS

- 10.1 It is advantageous to have more than one possible path through a network for each pair of stations to enhance reliability in case a particular path fails.
- 10.2 Subscribers: the devices that attach to the network, such as telephones and modems. Subscriber line: the link between the subscriber and the network. Exchanges: the switching centers in the network. Trunks: the branches between exchanges. Trunks carry multiple voice-frequency circuits using either FDM or synchronous TDM.
- 10.3 Telephone communications.
- 10.4 With inchannel signaling, the same channel is used to carry control signals as is used to carry the call to which the control signals relate. With common channel signaling, control signals are carried over paths completely independent of the voice channels.
- 10.5 (1) Line efficiency is greater, because a single node-to-node link can be dynamically shared by many packets over time. (2) A packet-switching network can perform data-rate conversion. Two stations of different data rates can exchange packets because each connects to its node at its proper data rate. (3) When traffic becomes heavy on a circuit-switching network, some calls are blocked; that is, the network refuses to accept additional connection requests until the load on the network decreases. On a packet-switching network, packets are still accepted, but delivery delay increases. (4) Priorities can be used. Thus, if a node has a number of packets queued for transmission, it can transmit the higher-priority packets first. These packets will therefore experience less delay than lower-priority packets.
- 10.6 In the datagram approach, each packet is treated independently, with no reference to packets that have gone before. In the virtual circuit approach, a preplanned route is established before any packets are sent. Once the route is established, all the packets between a pair of communicating parties follow this same route through the network.
- 10.7 There is a significant relationship between packet size and transmission time. As a smaller packet size is used, there is a more efficient "pipelining" effect, as shown in Figure 10.14. However, if the packet size becomes too small, then the transmission is less efficient, as shown in Figure 10.14d.

10.8 Transmission, processing, and queuing delays.

10.9 Frame relay is much simplified, compared to X.25. The major differences are that frame relay uses out-of-channel signaling while X.25 uses all in-channel control. In frame relay there is no "hop-by-hop" flow control or error control as there is in X.25. If a frame error is detected it is just dropped rather than being retransmitted. Similarly, on an end-to-end basis, there is no error control or flow control except what is provided by higher level protocols outside of frame relay. Finally, frame relay is a two level (physical and link) layer protocol and the multiplexing of logical channels takes place at Level 2 rather than in the Level 3 Packet Layer as in X.25.

10.10 Because frame relay is so much simpler than X.25, the processing to support switching can be reduced and higher data rates than for X.25, up to several megabits per second, can be supported. On the other hand, because of the lack of hop-by-hop flow control, the user of frame relay has fewer tools to manage network congestion. The effective use of frame relay also depends on the channels being relatively error free. For example, this is true for fiber optics, but probably not for most forms of broadcast, wireless transmission.

## ANSWERS TO PROBLEMS

10.1 Each telephone makes 0.5 calls/hour at 6 minutes each. Thus a telephone occupies a circuit for 3 minutes per hour. Twenty telephones can share a circuit (although this 100% utilization implies long queuing delays). Since 10% of the calls are long distance, it takes 200 telephones to occupy a long distance (4 kHz) channel full time. The interoffice trunk has  $10^6/(4 \times 10^3) = 250$  channels. With 200 telephones per channel, an end office can support  $200 \times 250 = 50,000$  telephones. Source: [TANE03]

10.2 SS7 could be implemented using circuit switching. This would have the merit of providing minimum delay between control points for the exchange of control information. However, the circuits would have to be set up, which takes time. Alternatively, the circuits could be permanently maintained, which consumes resources.

10.3 The argument ignores the overhead of the initial circuit setup and the circuit teardown.

10.4 a. Circuit Switching

$$T = C_1 + C_2 \text{ where}$$

$C_1$  = Call Setup Time

$C_2$  = Message Delivery Time

$C_1$  = S = 0.2

$C_2$  = Propagation Delay + Transmission Time

$$\begin{aligned}
 &= N \times D + L/B \\
 &= 4 \times 0.001 + 3200/9600 = 0.337 \\
 T &= 0.2 + 0.337 = 0.537 \text{ sec}
 \end{aligned}$$

### Datagram Packet Switching

$$\begin{aligned}
 T &= D_1 + D_2 + D_3 + D_4 \quad \text{where} \\
 D_1 &= \text{Time to Transmit and Deliver all packets through first hop} \\
 D_2 &= \text{Time to Deliver last packet across second hop} \\
 D_3 &= \text{Time to Deliver last packet across third hop} \\
 D_4 &= \text{Time to Deliver last packet across forth hop}
 \end{aligned}$$

There are  $P - H = 1024 - 16 = 1008$  data bits per packet. A message of 3200 bits require four packets ( $3200 \text{ bits}/1008 \text{ bits/packet} = 3.17$  packets which we round up to 4 packets).

$$\begin{aligned}
 D_1 &= 4 \times t + p \text{ where} \\
 t &= \text{transmission time for one packet} \\
 p &= \text{propagation delay for one hop} \\
 D_1 &= 4 \times (P/B) + D \\
 &= 4 \times (1024/9600) + 0.001 \\
 &= 0.428 \\
 D_2 &= D_3 = D_4 = t + p \\
 &= (P/B) + D \\
 &= (1024/9600) + 0.001 = 0.108 \\
 T &= 0.428 + 0.108 + 0.108 + 0.108 \\
 &= 0.752 \text{ sec}
 \end{aligned}$$

### Virtual Circuit Packet Switching

$$\begin{aligned}
 T &= V_1 + V_2 \text{ where} \\
 V_1 &= \text{Call Setup Time} \\
 V_2 &= \text{Datagram Packet Switching Time} \\
 T &= S + 0.752 = 0.2 + 0.752 = 0.952 \text{ sec}
 \end{aligned}$$

#### b. Circuit Switching vs. Datagram Packet Switching

$$\begin{aligned}
 T_c &= \text{End-to-End Delay, Circuit Switching} \\
 T_c &= S + N \times D + L/B \\
 T_d &= \text{End-to-End Delay, Datagram Packet Switching} \\
 N_p &= \text{Number of packets} = \left\lceil \frac{L}{P-H} \right\rceil \\
 T_d &= D_1 + (N-1)D_2 \\
 D_1 &= \text{Time to Transmit and Deliver all packets through first hop} \\
 D_2 &= \text{Time to Deliver last packet through a hop} \\
 D_1 &= N_p(P/B) + D \\
 D_2 &= P/B + D
 \end{aligned}$$

$$T = (N_p + N - 1)(P/B) + N \times D$$

$$T = T_d$$

$$S + L/B = (N_p + N - 1)(P/B)$$

Circuit Switching vs. Virtual Circuit Packet Switching

$$T_V = \text{End-to-End Delay, Virtual Circuit Packet Switching}$$

$$T_V = S + T_d$$

$$T_C = T_V$$

$$L/B = (N_p + N - 1)(P/B)$$

Datagram vs. Virtual Circuit Packet Switching

$$T_d = T_V - S$$

10.5 From Problem 10.4, we have

$$T_d = (N_p + N - 1)(P/B) + N \times D$$

For maximum efficiency, we assume that  $N_p = L/(P - H)$  is an integer. Also, it is assumed that  $D = 0$ . Thus

$$T_d = (L/(P - H) + N - 1)(P/B)$$

To minimize as a function of  $P$ , take the derivative:

$$0 = dT_d / (dP)$$

$$0 = (1/B)(L/(P - H) + N - 1) - (P/B)L/(P - H)^2$$

$$0 = L(P - H) + (N - 1)(P - H)^2 - LP$$

$$0 = -LH + (N - 1)(P - H)^2$$

$$(P - H)^2 = LH/(N - 1)$$

$$P = H + \sqrt{\frac{LH}{N - 1}}$$

10.6 Yes. A large noise burst could create an undetected error in the packet. With an  $N$ -bit CRC, the probability of an undetected error is on the order of  $2^{-N}$ . If such an error occurs and alters a destination address field or virtual circuit identifier field, the packet would be misdelivered.

10.7 The layer 2 flow control mechanism regulates the total flow of data between DTE and DCE. Either can prevent the other from overwhelming it. The layer 3 flow control mechanism regulates the flow over a single virtual circuit. Thus, resources in either the DTE or DCE that are dedicated to a particular virtual circuit can be protected from overflow.

10.8 Yes. Errors are caught at the link level, but this only catches transmission errors. If a packet-switching node fails or corrupts a packet, the packet will not be delivered correctly. A higher-layer end-to-end protocol, such as TCP, must provide end-to-end reliability, if desired.

10.9 On each end, a virtual circuit number is chosen from the pool of locally available numbers and has only local significance. Otherwise, there would have to be global management of numbers.

$$10.10 \quad k = 2 + 2 \times \frac{T_{td} + R_u}{8 \times L_d} = 2 + 2a$$

where the variable  $a$  is the same one defined in Chapter 7. In essence, the upper part of the fraction is the length of the link in bits, and the lower part of the fraction is the length of a frame in bits. So the fraction tells you how many frames can be laid out on the link at one time. Multiplying by 2 gives you the round-trip length of the link. You want your sliding window to accommodate that number of frames so that you can continue to send frames until an acknowledgment is received. Adding 1 to that total takes care of rounding up to the next whole number of frames. Adding 2 instead of 1 is just an additional margin of safety. See Figure 7.11.

# CHAPTER 11

## ASYNCHRONOUS TRANSFER MODE

### ANSWERS TO QUESTIONS

- 11.1 The most obvious feature of ATM compared to frame relay is that ATM makes use of a 53 byte fixed length cell while the frame in frame relay is much longer, and may vary in length, both in its header and its data fields. Additionally, error checking is only done on the header in ATM rather than on the whole cell or frame. Virtual channels of ATM that follow the same route through the network are bundled into paths. A similar mechanism is not used in frame relay.
- 11.2 ATM is even more streamlined than frame relay in its functionality, and can support data rates several orders of magnitude greater than frame relay.
- 11.3 A virtual channel is a logical connection similar to virtual circuit in X.25 or a logical channel in frame relay. In ATM, virtual channels which have the same endpoints can be grouped into virtual paths. All the circuits in virtual paths are switched together; this offers increased efficiency, architectural simplicity, and the ability to offer enhanced network services.
- 11.4 Simplified network architecture: Network transport functions can be separated into those related to an individual logical connection (virtual channel) and those related to a group of logical connections (virtual path). Increased network performance and reliability: The network deals with fewer, aggregated entities. Reduced processing and short connection setup time: Much of the work is done when the virtual path is set up. By reserving capacity on a virtual path connection in anticipation of later call arrivals, new virtual channel connections can be established by executing simple control functions at the endpoints of the virtual path connection; no call processing is required at transit nodes. Thus, the addition of new virtual channels to an existing virtual path involves minimal processing. Enhanced network services: The virtual path is used internal to the network but is also visible to the end user. Thus, the user may define closed user groups or closed networks of virtual channel bundles.
- 11.5 Quality of service: A user of a VCC is provided with a Quality of Service specified by parameters such as cell loss ratio (ratio of cells lost to cells transmitted) and cell delay variation. Switched and semipermanent virtual channel connections: A switched VCC is an on-demand connection, which requires a call control signaling for setup and tearing down. A semipermanent VCC is one that is of long duration and is set up by configuration or network management action. Cell sequence integrity: The sequence of transmitted cells within a VCC is preserved. Traffic

parameter negotiation and usage monitoring: Traffic parameters can be negotiated between a user and the network for each VCC. The input of cells to the VCC is monitored by the network to ensure that the negotiated parameters are not violated.

11.6 Same as for a VCC, plus: Virtual channel identifier restriction within a VPC: One or more virtual channel identifiers, or numbers, may not be available to the user of the VPC but may be reserved for network use. Examples include VCCs used for network management.

11.7 Generic flow control: used to assist the customer in controlling the flow of traffic for different qualities of service; virtual path identifier: constitutes a routing field for the network; virtual channel identifier: used for routing to and from the end user; payload type: indicates the type of information in the information field; cell loss priority bit: is used to provide guidance to the network in the event of congestion; header error control: used for both error control and synchronization.

11.8 Cell-based: No framing is imposed. The interface structure consists of a continuous stream of 53-octet cells. SDH-based: imposes a synchronous TDM structure on the ATM cell stream.

11.9 Constant bit rate: provides a fixed data rate that is continuously available during the connection lifetime, with a relatively tight upper bound on transfer delay. Real-time variable bit rate: intended for time-sensitive applications; that is, those requiring tightly constrained delay and delay variation. The principal difference between applications appropriate for rt-VBR and those appropriate for CBR is that rt-VBR applications transmit at a rate that varies with time.. Non-real-time variable bit rate: with this service, the end system specifies a peak cell rate, a sustainable or average cell rate, and a measure of how bursty or clumped the cells may be. With this information, the network can allocate resources to provide relatively low delay and minimal cell loss.. Available bit rate: designed to improve the service provided to bursty sources that would otherwise use UBR. An application using ABR specifies a peak cell rate (PCR) that it will use and a minimum cell rate (MCR) that it requires.. Unspecified bit rate: a best-effort service. Guaranteed frame rate: designed to optimize the handling of frame-based traffic.

11.10 Handling of transmission errors; segmentation and reassembly, to enable larger blocks of data to be carried in the information field of ATM cells; handling of lost and misinserted cell conditions; flow control and timing control.

## ANSWERS TO PROBLEMS

## 11.1

<b>Controlling → controlled</b>		<b>Controlled → controlling</b>	
0 0 0 0	NO_HALT, NULL	0 0 0 0	Terminal is uncontrolled. Cell is assigned or on an uncontrolled ATM connection.
1 0 0 0	HALT, NULL_A, NULL_B	0 0 0 1	Terminal is controlled. Cell is unassigned or on an uncontrolled ATM connection.
0 1 0 0	NO_HALT, SET_A, NULL_B	0 1 0 1	Terminal is controlled. Cell on a controlled ATM connection Group A.
1 1 0 0	HALT, SET_A, NULL_B	0 0 1 1	Terminal is controlled. Cell on a controlled ATM connection Group B.
0 0 1 0	NO_HALT, NULL_A, SET_B		
1 0 1 0	HALT, NULL_A, SET_B		
0 1 1 0	NO_HALT, SET_A, SET_B		
1 1 1 0	HALT, SET_A, SET_B		

All other values are ignored.

- 11.2 a. We reason as follows. A total of  $X$  octets are to be transmitted. This will require a total of  $\left\lceil \frac{X}{L} \right\rceil$  cells. Each cell consists of  $(L + H)$  octets, where  $L$  is the number of data field octets and  $H$  is the number of header octets. Thus

$$N = \frac{X}{\left\lceil \frac{X}{L} \right\rceil (L + H)}$$

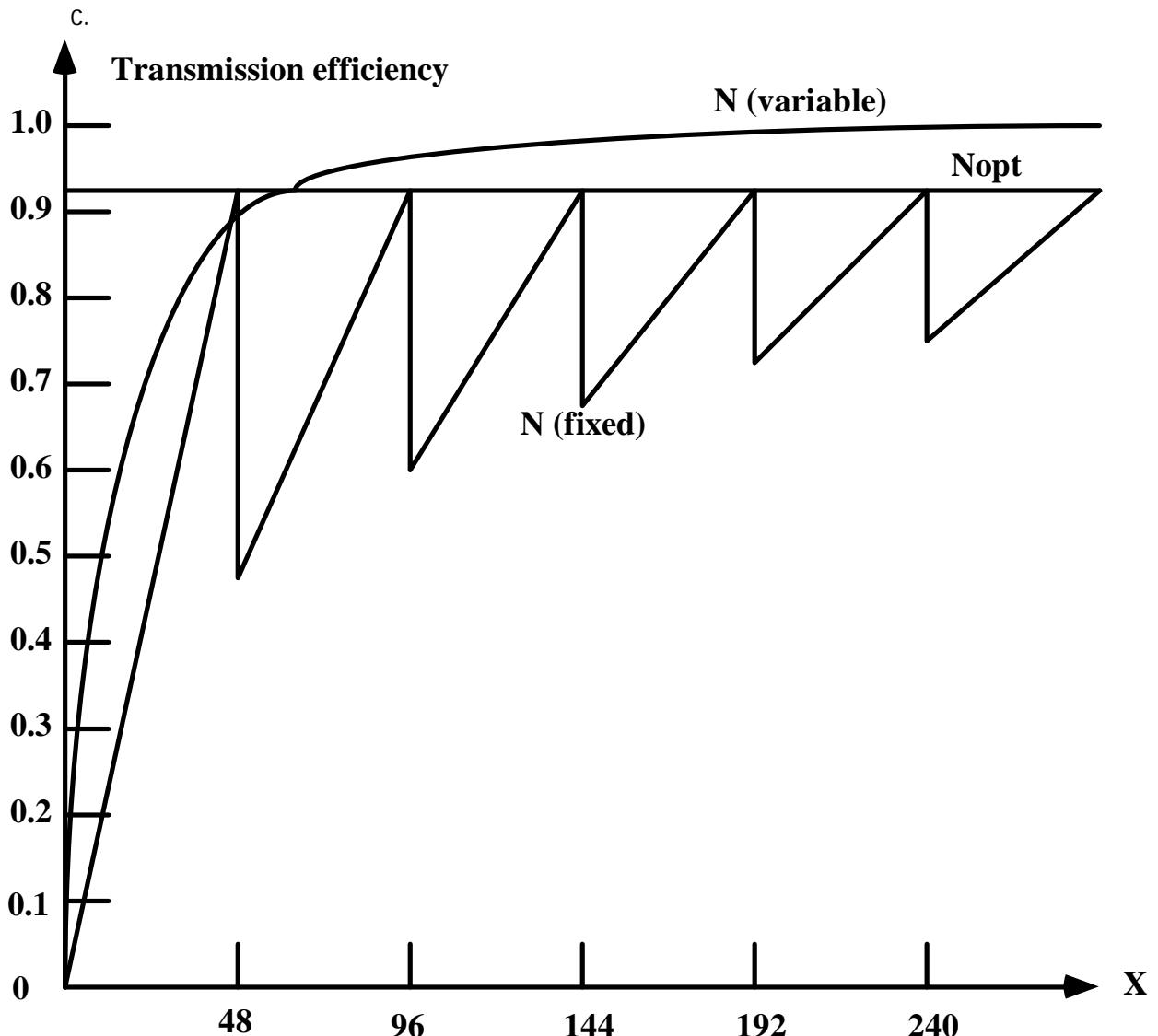
The efficiency is optimal for all values of  $X$  which are integer multiples of the cell information size. In the optimal case, the efficiency becomes

$$N_{\text{opt}} = \frac{X}{\left( \frac{X}{L} \right) (L + H)} = \frac{L}{L + H}$$

For the case of ATM, with  $L = 48$  and  $H = 5$ , we have  $N_{\text{opt}} = 0.91$

- b. Assume that the entire  $X$  octets to be transmitted can fit into a single variable-length cell. Then

$$N = \frac{X}{X + H + H_v}$$



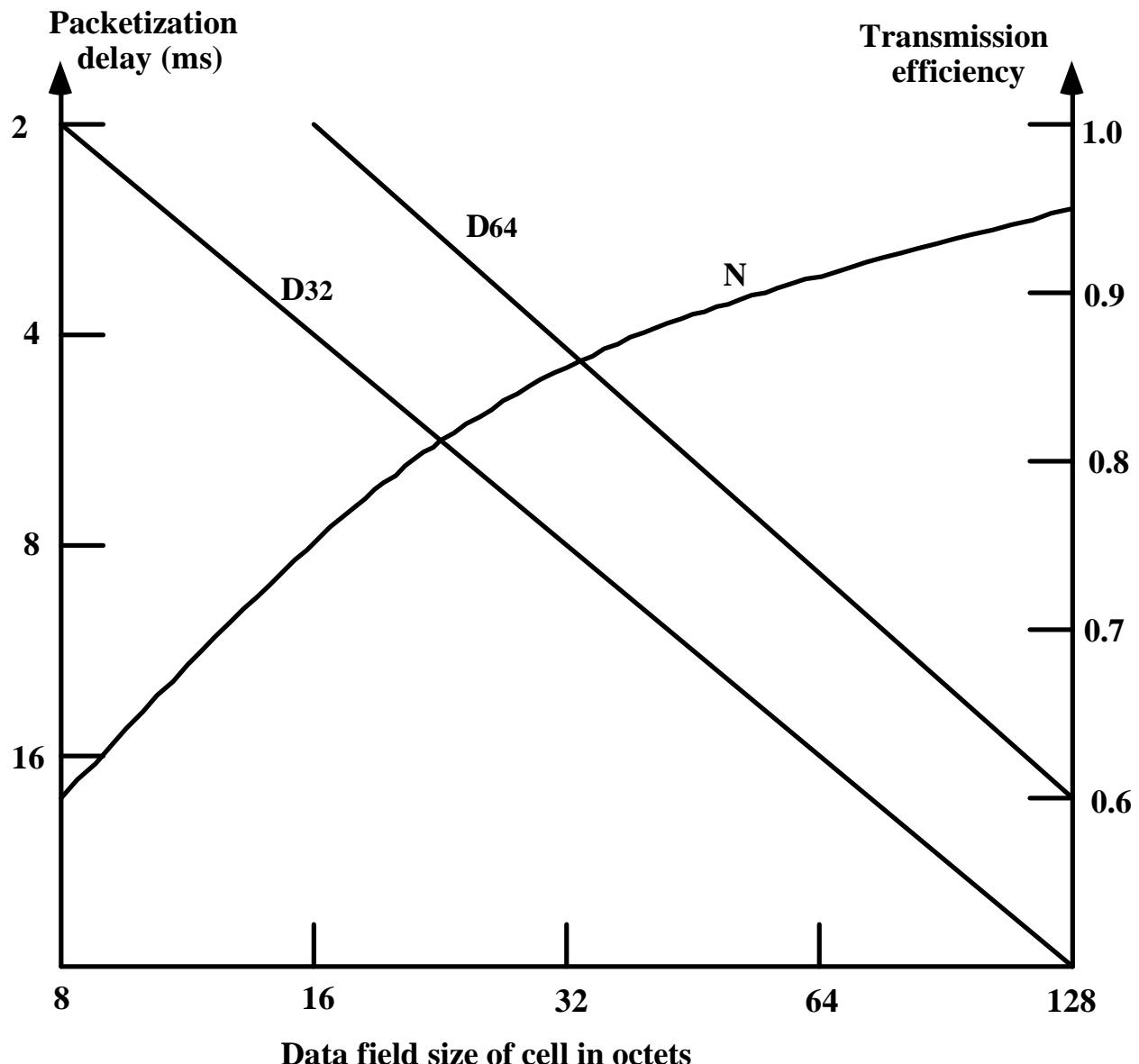
$N$  for fixed-sized cells has a sawtooth shape. For long messages, the optimal achievable efficiency is approached. It is only for very short cells that efficiency is rather low. For variable-length cells, efficiency can be quite high, approaching 100% for large  $X$ . However, it does not provide significant gains over fixed-length cells for most values of  $X$ .

11.3 a. As we have already seen in Problem 11.2:

$$N = \frac{L}{L + H}$$

$$\text{b. } D = \frac{8 \times L}{R}$$

C.



A data field of 48 octets, which is what is used in ATM, seems to provide a reasonably good tradeoff between the requirements of low delay and high efficiency. Source: [PRYC96]

- 11.4 a. The transmission time for one cell through one switch is  $t = (53 \times 8)/(43 \times 10^6) = 9.86\mu s$ .
- b. The maximum time from when a typical video cell arrives at the first switch (and possibly waits) until it is finished being transmitted by the 5th and last one is  $2 \times 5 \times 9.86\mu s = 98.6\mu s$ .
- c. The average time from the input of the first switch to clearing the fifth is  $(5 + 0.6 \times 5 \times 0.5) \times 9.86\mu s = 64.09\mu s$ .

- d. The transmission time is always incurred so the jitter is due only to the waiting for switches to clear. In the first case the maximum jitter is  $49.3\mu s$ . In the second case the average jitter is  $64.09 - 49.3 = 14.79\mu s$ .
- 11.5 a. The reception of a valid BOM is required to enter reassembly mode, so any subsequent COM and EOM cells are rejected. Thus, the loss of a BOM results in the complete loss of the PDU.
- b. Incorrect SN progression between SAR-PDUs reveals the loss of a COM, except for the cases covered by (c) and (d) below. The result is that at least the first 40 octets of the AAL-SDU is correctly received, namely the BOM that originally began the reassembly, and any subsequent COMs up to the point where cell loss occurred. Data from the BOM through the last SAR-PDU received with a correct SN can be passed up to the AAL user as a partial CPCS-PDU.
  - c. The SN wraps around so that there is no incorrect SN progression, but the loss of data is detected by the CPCS-PDU being undersized. In this case, only the BOM may be legitimately retrieved.
  - d. The same answer as for (c).
- 11.6 a. If the BOM of the second block arrives before the EOM of the first block, then the partially reassembled first block must be released. The entire partially reassembled CPCS-PDU received to that point is considered valid and passed to the AAL user along with an error indication. This mechanism works when just the EOM is lost or when a cell burst knocks out some COMs followed by the EOM.
- b. This might be detected by the SN mechanism. If not, the loss will be detected when the Btag and Etag fields fail to match. The Length indication may fail to pick up this error if the cell burst loses as many cells as are added by concatenation the two CPCS-PDU fragments. In this case only the first BOM may be legitimately retrieved.
- 11.7 a. Single bit errors are not picked up until the CPCS-PDU CRC is calculated, and they result in the discarding of the entire CPCS-PDU.
- b. Loss of a cell with SDU=0 is detected by an incorrect CRC. If the CRC fails to catch the error, the Length field mismatch ensures that the CPCS-PDU is discarded.
  - c. Loss of a cell with SDU=1 is detectable in three ways. First, the SAR-PDUs of the following CPCS-PDU may be appended to the first, resulting in a CRC error or Length mismatch being flagged when the second CPCS-PDU trailer arrives. Second, the AAL may enforce a length limit which, if exceeded while appending the second CPCS-PDU, can flag an error and cause the assembled data to be discarded. Third, a timer may be attached to the CPCS-PDU reassembly; if it expires before the CPCS-PDU is completely received, the assembled data is discarded.

See [ARMI93] for a further discussion of the issues raised in problems 11.5 through 11.7.

## COMPUTER NETWORKS

1. The transport layer protocols used for real time multimedia, file transfer, DNS and email, respectively are **GATE 2013**

- (A) TCP, UDP, UDP and TCP
- (B) UDP, TCP, TCP and UDP
- (C) UDP, TCP, UDP and TCP
- (D) TCP, UDP, TCP and UDP

**ANS: C**

**SOL:** **Multimedia** can be unreliable but has to be fast so UDP, **File transfer** has to be secure & reliable so uses TCP, **DNS** can be both TCP and UDP, **E mail** uses TCP for reliability.

2. Using public key cryptography, X adds a digital signature  $\sigma$  to message M, encrypts  $\langle M, \sigma \rangle$ , and sends it to Y, where it is decrypted. Which one of the following sequences of keys is used for the operations? **GATE 2013**

- (A) Encryption: X's private key followed by Y's private key; Decryption: X's public key followed by Y's public key
- (B) Encryption: X's private key followed by Y's public key; Decryption: X's public key followed by Y's private key
- (C) Encryption: X's public key followed by Y's private key; Decryption: Y's public key followed by X's private key
- (D) Encryption: X's private key followed by Y's public key; Decryption: Y's private key followed by X's public key

**ANS: D**

3. Assume that source S and destination D are connected through two intermediate routers labeled R. Determine how many times each packet has to visit the network layer and the data link layer during a transmission from S to D. **GATE 2013**

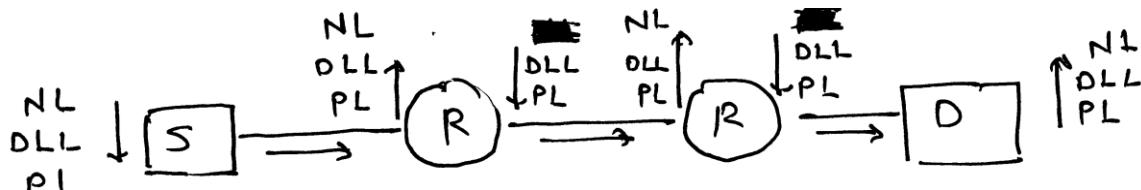


- (A) Network layer – 4 times and Data link layer – 4 times
- (B) Network layer – 4 times and Data link layer – 3 times
- (C) Network layer – 4 times and Data link layer – 6 times
- (D) Network layer – 2 times and Data link layer – 6 times

**ANS: C**

**SOL:** Therefore, Network layer – 4 times

Data link layer – 6 times



4. Determine the maximum length of the cable (in km) for transmitting data at a rate of 500 Mbps in an Ethernet LAN with frames of size 10,000 bits. Assume the signal speed in the cable to be 2,00,000 km/s. **GATE 2013**

(A) 1      (B) 2    (C) 2.5      (D) 5

**ANS: B**

**SOL:** Propagation time = Transmission time + Collision signal time

$$\frac{\text{Frame size}}{\text{Propagation time}} = \frac{\text{Length}}{\text{Signal speed}} + \frac{\text{Length}}{\text{Signal speed}}$$

$$\frac{10000 \text{ bit}}{500 * 10^6 \text{ bits/sec}} = \frac{2 * \text{length}}{2 * 10^5 \text{ km/sec}}$$

$$\text{Length} = 2 \text{ km}$$

5. In an IPv4 datagram, the M bit is 0, the value of HLEN is 10, the value of total length is 400 and the fragment offset value is 300. The position of the datagram, the sequence numbers of the first and the last bytes of the payload, respectively are **GATE 2013**

(A) Last fragment, 2400 and 2789  
 (B) First fragment, 2400 and 2759  
 (C) Last fragment, 2400 and 2759  
 (D) Middle fragment, 300 and 689

**ANS: C**

**SOL:** Since M bit is 0, so there is no fragment after this fragment. Hence this fragment is the last fragment.

Now, HLEN defines the length of header in datagram. Since Hlen is 10 so, size of header is  $10 * 4 = 40 \text{ B}$

$$\begin{aligned}\text{Length of data} &= \text{Total length} - \text{Header length} \\ &= 400 - 40 \\ &= 360 \text{ B}\end{aligned}$$

Now, fragment offset of data in original datagram is measured in units of 8 B. so to find first Byte of this fragment, First byte/8 = fragment offset

First byte =  $300 * 8 = 2400 \text{ B}$  and since length of data is 360 B.

So, last byte on this datagram will be 2759

6. In the IPv4 addressing format, the number of networks allowed under Class C addresses is **GATE 2012**

- (A)  $2^{14}$       (B)  $2^7$       (C)  $2^{21}$       (D)  $2^{24}$

**ANS: C**

**SOL:** For class C address, size of network field is 24 bits. But first 3 bits are fixed as 110; hence total number of networks possible is  $2^{21}$ .

7. Which of the following transport layer protocols is used to support electronic mail?

**GATE 2012**

- (A) SMTP      (B) IP      (C) TCP      (D) UDP

**ANS:C**

**SOL:** E-mail uses SMTP in application layer to transfer mail. And SMTP uses TCP to transfer data in transport layer.

8. The protocol data unit (PDU) for the application layer in the Internet stack is

**GATE 2012**

- (A) Segment      (B) Datagram      (C) Message      (D) Frame

**ANS: C**

**SOL:** Protocol Data Unit (PDU)  
Application layer – Message  
Transport layer – Segment  
Network layer – Datagram  
Data Link layer – Frame  
Ans = Message

9. Consider an instance of TCP's Additive Increase Multiplicative Decrease (AIMD) algorithm where the window size at the start of the slow start phase is 2 MSS and the threshold at the start of the first transmission is 8 MSS. Assume that a timeout occurs during the fifth transmission. Find the congestion window size at the end of the tenth transmission.

**GATE 2012**

- (A) 8 MSS      (B) 14 MSS      (C) 7 MSS      (D) 12 MSS

**ANS: C**

**SOL:** Given threshold = 8

Time = 1, during first transmission, window size = 2 (slow start phase)

Time = 2, congestion window size = 4 (double the no. of acknowledgments)

Time = 3, congestion window size is = 8

Time = 4, congestion window size = 9, after threshold (increase by one additive increase)

Time = 5, transmits 10 MSS, but time out occurs congestion windw size = 10

Hence threshold = (congestion window size)/2=10/2 = 5

Time = 6, transmits 2

Time = 7, transmits 4

Time = 8, transmits 5(threshold is 5)

Time = 9, transmits 6

Time = 10, transmits 7

During 10<sup>th</sup> transmission, it transmits 7 segments hence at the end of the 10<sup>th</sup> transmission the size of congestion window is 7 MSS.

10. Consider a source computer (S) transmitting a file of size  $10^6$  bits to a destination computer (D) over a network of two routers (R1 and R2) and three links (L1, L2, and L3). L1 connects S to R1; L2 connects R1 to R2; and L3 connects R2 to D. Let each link be of length 100km. Assume signals travel over each line at a speed of  $10^8$  meters per second. Assume that the link bandwidth on each link is 1Mbps. Let the file be broken down into 1000 packets each of size 1000 bits. Find the total sum of transmission and propagation delays in transmitting the file from S to D?

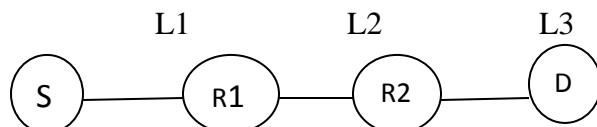
**GATE**

**2012**

- (A) 1005ms    (B) 1010ms    (C) 3000ms    (D) 3003ms

**ANS: A**

**SOL:**



Transmission delay for 1<sup>st</sup> packet from each of S, R<sub>1</sub> and R<sub>2</sub> will take 1 ms

Propagation delay on each link l1,l2 and l3 for one packet is 1ms

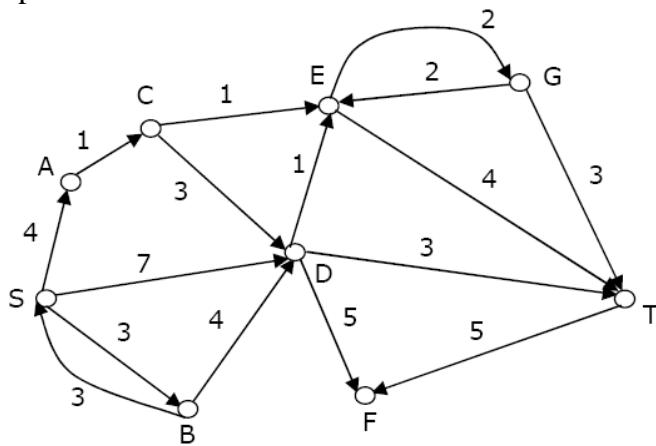
Therefore the sum of transmission delay and propagation delay on each link for one packet is 2ms.

The first packet reaches the destination at 6<sup>th</sup> ms

The second packet reaches the destination at 7<sup>th</sup> ms

So, inductively we can say that 1000<sup>th</sup> packet reaches the destination at 1005<sup>th</sup> ms.

11. Consider the directed graph shown in the figure below. There are multiple shortest paths between vertices S and T. Which one will be reported by Dijkstra's shortest path algorithm? Assume that, in any iteration, the shortest path to a vertex v is updated only when a strictly shortest path to v is discovered.



**GATE**

**2012**

- (A) SDT    (B) SBDT    (C) SACDT    (D) SACET

**ANS: D**

**SOL:** Let  $d[v]$  represent the shortest path distance computed from 'S' Initially  $d[S] = 0$ ,  $d[A] = \infty$ ,  $d[B] = \infty$ ,  $\dots$ ,  $d[T] = \infty$

and let  $P[v]$  represent the predecessor of  $v$  in the shortest path from 'S' to ' $v$ ' and let  $P[v] = 1$  denote that currently predecessor of ' $v$ ' has not been computed

→ Let  $Q$  be the set of vertices for which shortest path distance has not been computed

→ Let  $W$  be the set of vertices for which shortest path distance has not been computed

→ So initially,  $Q = \{S, A, B, C, D, E, F, G, T\}$ ,  $W = \emptyset$

We will use the following procedure

Repeat until  $Q$  is empty{

1.  $u =$  choose a vertex from  $Q$  with minimum  $d[u]$  value

2.  $Q = Q - u$

3. update all the adjacent vertices of  $u$

4.  $W = W \cup \{u\}$

}  $d[S] = 0$ ,  $d[A] = \infty$ ,  $d[B] = \infty$ ,  $\dots$ ,  $d[T] = \infty$

**Iteration 1:** Step 1:  $u = S$

Step 2:  $Q = \{A, B, C, D, E, F, G, T\}$

Step 3: final values after adjustment

$d[S] = 0$ ,  $d[A] = 4$ ,  $d[B] = 3$ ,  $d[C] = \infty$ ,  $d[D] = 7$ ,  $d[E] = \infty \dots$ ,  $d[T] = \infty$

$P[A] = S$ ,  $P[B] = S$ ,  $P[C] = 1$ ,  $P[D] = S$ ,  $P[E] = 1 \dots$ ,  $P[T] = 1$

Step 4:  $W = \{S\}$

**Iteration 2:** Step 1:  $u = S$

Step 2:  $Q = \{A, C, D, E, F, G, T\}$

Step 3: final values after adjustment

$d[S] = 0$ ,  $d[A] = 4$ ,  $d[B] = 3$ ,  $d[C] = \infty$ ,  $d[D] = 7$ ,  $d[E] = \infty \dots$ ,  $d[T] = \infty$

$P[A] = S$ ,  $P[B] = S$ ,  $P[C] = 1$ ,  $P[D] = S$ ,  $P[E] = 1 \dots$ ,  $P[T] = 1$

Step 4:  $W = \{S, B\}$

**Iteration 3:** Step 1:  $u = A$

Step 2:  $Q = \{C, D, E, F, G, T\}$

Step 3: final values after adjustment

$d[S] = 0$ ,  $d[A] = 4$ ,  $d[B] = 3$ ,  $d[C] = 5$ ,  $d[D] = 7$ ,  $d[E] = \infty \dots$ ,  $d[T] = \infty$

$P[A] = S$ ,  $P[B] = S$ ,  $P[C] = A$ ,  $P[D] = S$ ,  $P[E] = 1 \dots$ ,  $P[T] = 1$

Step 4:  $W = \{S, B, A\}$

**Iteration 4:** Step 1:  $u = C$

Step 2:  $Q = \{D, E, F, G, T\}$

Step 3: final values after adjustment

$d[S] = 0$ ,  $d[A] = 4$ ,  $d[B] = 3$ ,  $d[C] = 5$ ,  $d[D] = 7$ ,  $d[E] = 6$ ,  $\dots$ ,  $d[T] = \infty$

$P[A] = S$ ,  $P[B] = S$ ,  $P[C] = A$ ,  $P[D] = S$ ,  $P[E] = C$ ,  $\dots$ ,  $P[T] = 1$

Step 4:  $W = \{S, B, A, C\}$

**Iteration 5:** Step 1:  $u = E$

Step 2:  $Q = \{D, F, G, T\}$

Step 3: final values after adjustment

$d[S] = 0$ ,  $d[A] = 4$ ,  $d[B] = 3$ ,  $d[C] = 5$ ,  $d[D] = 7$ ,  $d[E] = 6$ ,  $d[F] = \infty$ ,  $d[G]$

$= 8$ ,  $d[T] = 10$   $P[A] = S$ ,  $P[B] = S$ ,  $P[C] = A$ ,  $P[D] = S$ ,  $P[E] = C$ ,  $P[F] = 1$ ,

$$P[G] = E, P[T] = E$$

Step 4:  $W = \{S, B, A, C, E\}$

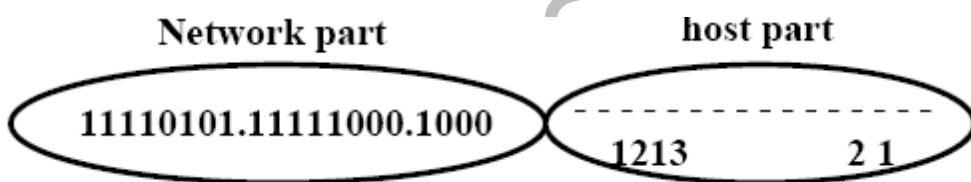
After iteration 5, we can observe that  $P[T] = E, P[E] = C, P[C] = A, P[A] = S$ , So the shortest path from S to T is SACET

12. An Internet Service Provider (ISP) has the following chunk of CIDR-based IP addresses available with it: 245.248.128.0/20. The ISP wants to give half of this chunk of addresses to Organization A, and a quarter to Organization B, while retaining the remaining with itself. Which of the following is a valid allocation of addresses to A and B? **GATE 2012**

- (A) 245.248.136.0/21 and 245.248.128.0/22
- (B) 245.248.128.0/21 and 245.248.128.0/22
- (C) 245.248.132.0/22 and 245.248.132.0/21
- (D) 245.248.136.0/24 and 245.248.132.0/21

**ANS: A**

**SOL:**



Since half of 4096 host addresses must be given to organization A, we can set 12<sup>th</sup> bit to 1 and include that bit into network part of organization A, so the valid allocation of addresses to A is 245.248.136.0/21

Now for organization B, 12<sup>th</sup> bit is set to '0' but since we need only half of 2048 addresses, 13<sup>th</sup> bit can be set to '0' and include that bit into network part of organization B so the valid allocation of addresses to B is 245.248.128.0/22

13. Consider different activities related to email.

- m1: Send an email from a mail client to a mail server
- m2: Download an email from mailbox server to a mail client
- m3: Checking email in a web browser

Which is the application level protocol used in each activity?

**GATE 2011**

- (A) m1: HTTP m2: SMTP m3: POP
- (B) m1: SMTP m2: FTP m3: HTTP
- (C) m1: SMTP m2: POP m3: HTTP
- (D) m1: POP m2: SMTP m3: IMAP

**ANS: C**

**SOL:** Mail client uses SMTP (Simple Mail Transfer Protocol) to send mail. (The client need not be web based. So, HTTP may not be involved here). POP (Post Office Protocol) is used to retrieve mail from mail server. HTTP (Hypertext transfer protocol) is used to transfer a HTML page containing the mail message that can be viewed on a web browser.

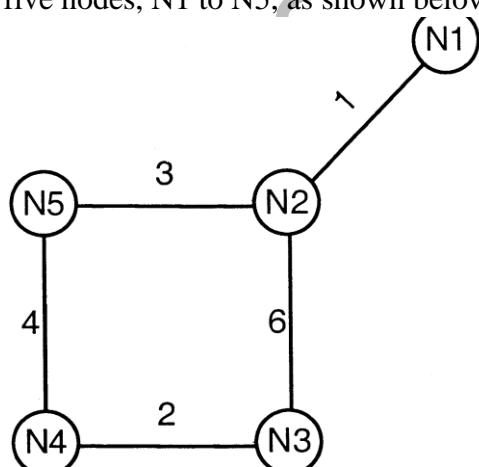
14. A layer-4 firewall (a device that can look at all protocol headers up to the transport layer)  
**CANNOT**  
**GATE 2011**

- (A) Block entire HTTP traffic during 9:00PM and 5:00AM
- (B) Block all ICMP traffic
- (C) Stop incoming traffic from a specific IP address but allow outgoing traffic to the same IP address
- (D) Block TCP traffic from a specific user on a multi-user system during 9:00PM and 5:00AM

**ANS: A**

**Statement for Linked Questions 15 and 16**

Consider a network with five nodes, N1 to N5, as shown below.



The network uses a Distance Vector Routing protocol. Once the routes have stabilized, the distance vectors at different nodes are as following.

- N1: (0, 1, 7, 8, 4)
- N2: (1, 0, 6, 7, 3)
- N3: (7, 6, 0, 2, 6)
- N4: (8, 7, 2, 0, 4)
- N5: (4, 3, 6, 4, 0)

Each distance vector is the distance of the best known path at the instance to nodes, N1 to N5, where the distance to itself is 0. Also, all links are symmetric and the cost is identical in both directions. In each round, all nodes exchange their distance vectors with their respective neighbors. Then all nodes update their distance vectors. In between two rounds, any change in cost of a link will cause the two incident nodes to change only that entry in their distance vectors.

15. The cost of link N2-N3 reduces to 2(in both directions). After the next round of updates, what will be the new distance vector at node, N3?

**GATE 2011**

- (A) (3, 2, 0, 2, 5)      (B) (3, 2, 0, 2, 6)      (C) (7, 2, 0, 2, 5)      (D) (7, 2, 0, 2, 6)

**ANS: A**

16. After the update in the previous question, the link N1-N2 goes down. N2 will reflect this change immediately in its distance vector as cost,  $\infty$ . After the **NEXT ROUND** of update, what will be the cost to N1 in the distance vector of N3?

**GATE 2011**

- (A) 3      (B) 9      (C) 10      (D)  $\infty$

**ANS: C**

17. One of the header fields in an IP datagram is the Time to Live (TTL) field. Which of the following statements best explains the need for this field?

**GATE 2010**

- (A) It can be used to prioritize packets
- (B) It can be used to reduce delays
- (C) It can be used to optimize throughput
- (D) It can be used to prevent packet looping

**ANS: D**

**SOL:** Whenever Time to live field reaches ‘0’ we discard the packet, so that we can prevent it from looping.

18. Which one of the following is not a client server application?

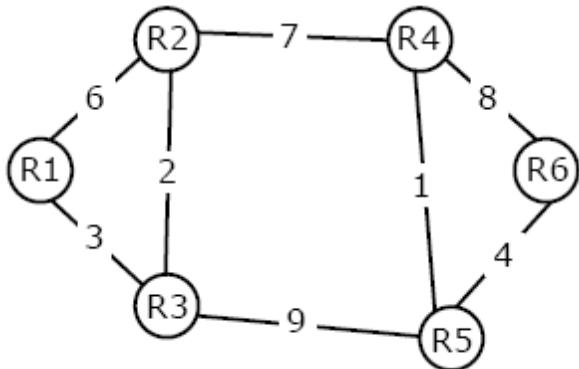
**GATE 2010**

- (A) Internet chat      (B) Web browsing      (C) E-mail      (D) Ping

**ANS: D**

#### **Statement for Linked Answer Questions: 19 & 20**

Consider a network with 6 routers R1 to R6 connected with links having weights as shown in the following diagram



19. All the routers use the distance vector based routing algorithm to update their routing tables. Each router starts with its routing table initialized to contain an entry for each neighbor with the weight of the respective connecting link. After all the routing tables stabilize, how many links in the network will never be used for carrying any data? **GATE 2010**

(A) 4      (B) 3      (C) 2      (D) 1

**ANS: D**

**SOL:** In Distance vector, the Router will update its routing tables by exchanging the information from all its neighbors. After all the routing tables stabilize the routing Table for 'R1' will not have any entry to Router R6., so that link will not be used. So one link.

20. Suppose the weights of all unused links in the previous question are changed to 2 and the distance vector algorithm is used again until all routing tables stabilize. How many links will now remain unused? **GATE 2010**

(A) 0      (B) 1      (C) 2      (D) 3

**ANS: A**

21. Which of the following statement(s) is / are correct regarding Bellman-Ford shortest path algorithm?

P. Always finds a negative weighted cycle, if one exists.  
Q. Finds whether any negative weighted cycle is reachable from the source.

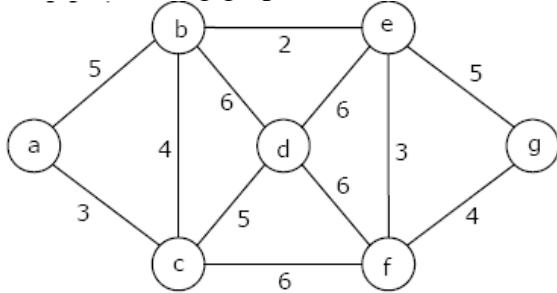
**GATE 2009**

(A) P only      (B) Q only      (C) both P and Q      (D) Neither P nor Q

**ANS: B**

**SOL:** The algorithm identifies a negative weight cycle iff it is reachable from Source.

22. Consider the following graph:



Which one of the following is NOT the sequence of edges added to the minimum spanning tree using Kruskal's algorithm?

**GATE 2009**

- (A) (b,e) (e,f) (a,c) (b,c) (f,g) (c,d)      (B) (b,e) (e,f) (a,c) (f,g) (b,c) (c,d)  
(C) (b,e) (a,c) (e,f) (b,c) (f,g) (c,d)      (D) (b,e) (e,f) (b,c) (a,c) (f,g) (c,d)

**ANS: D**

**SOL:** Weight of edge (a,c) is less than (b,c) . So it cannot come after (b,c)

23. In the RSA public key cryptosystem, the private and public keys are (e, n) and (d, n) respectively, where  $n=p \cdot q$  and p and q are large primes. Besides, n is public and p and q are private. Let M be an integer such that  $0 < M < n$  and  $\Phi(n) = (p-1)(q-1)$ . Now consider the following equations.

I.  $M' = M^e \bmod n$   
 $M = (M')^d \bmod n$

II.  $ed \equiv 1 \bmod n$

III.  $ed \equiv 1 \bmod \Phi(n)$

IV.  $M' = M^e \bmod \Phi(n)$   
 $M = (M')^d \bmod \Phi(n)$

Which of the above equations correctly represent RSA cryptosystem?

**GATE 2009**

- (A) I and II      (B) I and III      (C) II and IV      (D) III and IV

**ANS: B**

#### Statement for Linked Answer Questions: 24 & 25

Frames of 1000 bits are sent over a 106 bps duplex link between two hosts. The propagation time is 25ms. Frames are to be transmitted into this link to maximally pack them in transit (within the link).

24. What is the minimum number of bits (i) that will be required to represent the sequence numbers distinctly? Assume that no time gap needs to be given between transmission of two frames.

**GATE 2009**

- (A) i=2                    (B) i=3                    (C) i=4                    (D) i=5  
**ANS: D**

**SOL:** The transmission time for a frame is  $1000/1\text{Mbps} = 1 \text{ ms}$ . As the propagation time is 25 ms, the sender can transmit 25 packets before the first packet reaches the destination. Therefore the number of bits required to represent 25 packets is 5.

25. Suppose that the sliding window protocol is used with the sender window size of  $2^i$ , where i is the number of bits identified in the earlier part and acknowledgements are always piggy backed. After sending  $2^i$  frames, what is the minimum time the sender will have to wait before starting transmission of the next frame? (Identify the closest choice ignoring the frame processing time.)

**GATE 2009**

- (A) 16ms                    (B) 18ms                    (C) 20ms                    (D) 22ms

**ANS: B**

**SOL:** Sliding window size is 32 as i=5. The sender can expect an Ack after one RTT. Here Round trip time is 50ms. Therefore the sender has to wait at least  $50-32= 18\text{ms}$  before transmission of the next frame.

26. What is the maximum size of data that the application layer can pass on to the TCP layer below?

**GATE 2008**

- (A) Any size                    (B)  $2^{16}$  bytes-size of TCP header  
(C)  $2^{16}$  bytes                    (D) 1500 bytes

**ANS: A**

**SOL:** Application layer can pass any length data. TCP layer will divide that data into frames.

27. Which of the following system calls results in the sending of SYN packets?

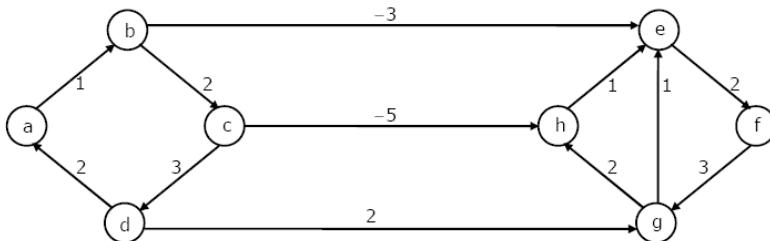
**GATE 2008**

- (A) socket                    (B) bind                    (C) listen                    (D) connect

**ANS: D**

**SOL:** In the process of establishing a connection between two endpoints, the user process on active end point invokes the connect() system call. The active end point then sends a SYN packet. The passive end point invokes an accept() system call and sends ACK to the other system then the connection is established.

28. Dijkstra's single source shortest path algorithm when run from vertex 'a' in the following graph, computes the correct shortest path distance to



- (A) only vertex a  
 (C) only vertices a, b, c, d

- (B) only vertices a, e, f, g, h  
 (D) all the vertices

**ANS: D**

**SOL:** Even though the graph has negative weights, it correctly computes the shortest path to all the vertices. There will not be any problem with the Dijkstra's algorithm operating on negative edge weights as long as the shortest path distance computed for the currently removed vertex is the actual shortest path distance.

29. In the slow start phase of the TCP congestion control algorithm, the size of the congestion window **GATE 2008**

- (A) Does not increase  
 (C) Increases quadratically

- (B) increases linearly  
 (D) increases exponentially

**ANS: B**

30. If a class B network on the Internet has a subnet mask of 255.255.248.0, what is the maximum number of hosts per subnet? **GATE 2008**

- (A) 1022                    (B) 1023                    (C) 2046                    (D) 2047

**ANS: C**

**SOL:** Number of bits for subnet mask = 21

Number of bits for host = 11

$$\text{Number of hosts} = 2^{11} - 2 = 2046$$

31. A computer on a 10Mbps network is regulated by a token bucket. The token bucket is filled at a rate of 2Mbps. It is initially filled to capacity with 16Megabits. What is the maximum duration for which the computer can transmit at the full 10Mbps? **GATE 2008**

- (A) 1.6 seconds                    (B) 2 seconds                    (C) 5 seconds                    (D) 8 seconds

**ANS: B**

**SOL:** If the capacity of the token bucket is C bytes, Token arrival rate is R bytes/sec, and the Maximum possible transmission rate is M bytes/sec then the time(S) in seconds it is possible to transmit is  $S = C/(M-R)$  seconds , so  $16/(10-2) = 2$  seconds.

32. In Ethernet when Manchester encoding is used, the bit rate is:

  - (A) Half the baud rate.
  - (B) Twice the baud rate.
  - (C) Same as the baud rate.
  - (D) None of the above

GATE 2007

**ANS:** A

**SOL:** In Ethernet when Manchester encoding is used, the bit rate is half of the baud rate.

33. Which one of the following uses UDP as the transport protocol?

GATE 2007

- (A) HTTP      (B) Telnet      (C) DNS      (D) SMTP

**ANS: C**

**SOL:** DNS queries are normally short and they need fast responses. Therefore UDP is a better option as a transport protocol for DNS.

34. There are  $n$  stations in a slotted LAN. Each station attempts to transmit with a probability  $p$  in each time slot. What is the probability that ONLY one station transmits in a given time slot? **GATE 2007**

- (A)  $np(1-p)^{n-1}$       (B)  $(1-p)^{n-1}$       (C)  $p(1-p)^{n-1}$       (D)  $1-(1-p)^{n-1}$

ANS: A

**SOL:** The probability that only one station transmits in a given slot is, probability that station 1 transmits, stations 2 to n are not transmitting + station 2 is transmitting and station 1, stations 3 to n are not transmitting + .....

$$\begin{aligned} \text{which is, } & p(1-p)(1-P) \dots + (1-p)p(1-p)(1-p) \dots + (1-p)(1-p)p(1-p) \dots + \dots \text{ ntimes} \\ = & p(1-p)^{(n-1)} + p(1-p)^{(n-1)} + p(1-p)^{(n-1)} + p(1-p)^{(n-1)} + p(1-p)^{(n-1)} + \dots \text{ n times} \\ = & np(1-p)^{(n-1)} \end{aligned}$$

35. In a token ring network the transmission speed is 7.10 bps and the propagation speed is 200 metres/Rs. The 1-bit delay in this network is equivalent to: **GATE 2007**



**ANS: C**

**SOL:** The time taken to transmit 1 bit is 0.1micro second. Propagation speed is 200meters/microsecond.Therefore 1 bit delay implies 20 metres.

36. The address of a class B host is to be split into subnets with a 6-bit subnet number. What is the maximum number of subnets and the maximum number of hosts in each subnet?

- (A) 62 subnets and 262142 hosts.  
(B) 64 subnets and 262142 hosts.  
(C) 62 subnets and 1022 hosts.  
(D) 64 subnets and 1024 hosts.

**ANS:** C

**SOL:** Maximum number of subnets is  $2^6 - 2 = 62$ . Maximum number of hosts is  $2^{10} - 2 = 1022$ . Actually at present, subnets with addresses all 0's and all 1's can also be used.

37. The message 11001001 is to be transmitted using the CRC polynomial  $3x + 1$  to protect it from errors. The message that should be transmitted is: **GATE 2007**



**ANS: B**

**SOL:** The divisor is 1001. After dividing the given data 11001001 by 1001, the remainder is 011 which is the CRC. Therefore the transmitted data is, data+CRC which is 11001001011.

38. The distance between two stations M and N is L kilometers. All frames are K bits long. The propagation delay per kilometer is t seconds. Let R bits/second be the channel capacity. Assuming that processing delay is negligible, the minimum number of bits for the sequence number field in a frame for maximum utilization, when the sliding window protocol is used, is:

- (A)  $\left\lceil \log_2 \frac{2LtR+2K}{K} \right\rceil$

(B)  $\left\lceil \log_2 \frac{2LtR}{K} \right\rceil$

(C)  $\left\lceil \log_2 \frac{2LtR+K}{K} \right\rceil$

(D)  $\left\lceil \log_2 \frac{2LtR+K}{2K} \right\rceil$

**ANS:** A

**SOL:** The distance between the stations is L kilometers.

Propagation delay per kilometer is  $t$  seconds. Therefore the total propagation delay is  $Lt$  seconds.

The round trip time is  $2 \times \text{propagation delay} = 2Lt$  seconds.

Maximum utilization can be achieved by transmitting data for the whole round trip time. The size of data that can be transmitted for Round trip time is Time \* bandwidth =  $2LtR$ .

other than round trip time a packet is transmitted, and an ack. is also transmitted when the packet is received. The size in bits is 2k.

Therefore total size in bits is  $2LtR + 2k$ . Number of packets is  $(2LtR+2k)/k$ . Number of bits required to represent these packets is  $\log(\text{Number of packets})$ .

39. Match the following:

- |          |                       |
|----------|-----------------------|
| (P) SMTP | (1) Application layer |
| (Q) BGP  | (2) Transport layer   |
| (R) TCP  | (3) Data link layer   |
| (S) PPP  | (4) Network layer     |
|          | (5) Physical layer    |

GATE 2007

(C) P - 1 Q - 4 R - 2 S - 5

(D) P - 2 Q - 4 R - 1 S - 3

**ANS: B**

**SOL:** SMTP is an application layer protocol. TCP is the transport layer protocol. BGP is network layer protocol and PPP is the data link layer protocol.

40. For which one of the following reasons does Internet Protocol (IP) use the time-to-live (TTL) field in the IP datagram header?

**GATE 2006**

- (A) Ensure packets reach destination within that time
- (B) Discard packets that reach later than that time
- (C) Prevent packets from looping indefinitely
- (D) Limit the time for which a packet gets queued in intermediate routers.

**ANS: C**

**SOL:** Time to live indicates the time or maximum number of hops the packet is allowed to make before it is discarded. Normally it is set to twice the maximum length path from source to destination.

41. Station A uses 32 byte packets to transmit messages to Station B using a sliding window protocol. The round trip delay between A and B is 80 milliseconds and the bottleneck bandwidth on the path between A and B is 128 kbps. What is the optimal window size that A should use?

**GATE 2006**

- (A) 20
- (B) 40
- (C) 160
- (D) 320

**ANS: B**

**SOL:** The packet size is 32 bytes =  $32 \times 8$  bits.

Round trip time is 80ms. Bandwidth is 128kbps. Therefore in 1 RTT, the source can transmit  $128\text{ kbps} \times 80\text{ ms}$  bits of data.

The data possible to be transmitted divided by the packet size gives the window size, which is  $(128\text{k} \times 80\text{ ms})/(32 \times 8)$  = 40.

42. Two computers C1 and C2 are configured as follows. C1 has IP address 203.197.2.53 and net mask 255.255.128.0. C2 has IP address 203.197.75.201 and net mask 255.255.192.0. Which one of the following statements is true?

**GATE 2006**

- (A) C1 and C2 both assume they are on the same network
- (B) C2 assumes C1 is on same network, but C1 assumes C2 is on a different network
- (C) C1 assumes C2 is on same network, but C2 assumes C1 is on a different network
- (D) C1 and C2 both assume they are on different networks.

**ANS: C**

**SOL:** IP address of C1 is 203.197.2.53 and Subnet mask is 255.255.128.0 ending gives the network id which is 203.197.0.0.

When C1 sees the ipaddress 203.197.75.201, to find the network id it will and with its subnet mask, which gives 203.197.0.0.

So C1 assumes that C2 is on the same network with C1.

Similarly, IP address of C2 is 203.197.75.201, subnet mask is 255.255.192.0 ending gives the network id which is 203.197.64.0.

When this computer looks at IP address of C1, to find the network id, it will and with its network mask giving 203.197.0.0.

Therefore C1 assumes that C2 is on the same network with C2, but C2 assumes C1 is on a different network.

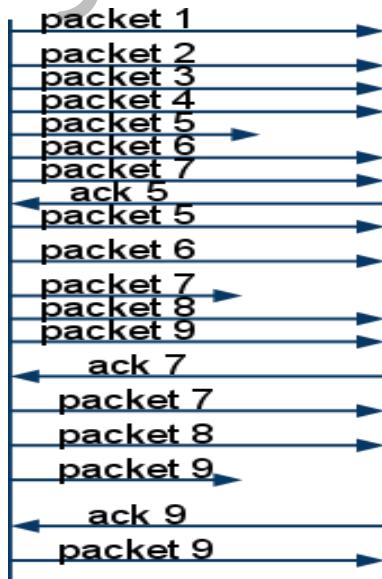
43. Station A needs to send a message consisting of 9 packets to Station B using a sliding window (window size 3) and go-back-n error control strategy. All packets are ready and immediately available for transmission. If every 5th packet that A transmits gets lost (but no acks from B ever get lost), then what is the number of packets that A will transmit for sending the message to B?

**GATE 2006**

- (A) 12                    (B) 14                    (C) 16                    (D) 18

**ANS: C**

**SOL:**



Assume that correctly transmitted packet is acknowledged at the same time. When a packet is lost, the receiver waits for certain duration but the sender can send up to its window size. The total number of packets sent is 16.

44. Packets of the same session may be routed through different paths in:

- (a) TCP, but not UDP
- (b) TCP and UDP
- (c) UDP, but not TCP
- (d) Neither TCP nor UDP

**ANS: B**

**SOL:** Packet is the Network layer Protocol Data Unit (PDU). TCP and UDP are Transport layer protocols. Packets of same session may be routed through different routes. Most networks don't use static routing, but use some form of adaptive routing where the paths used to route two packets for same session may be different due to congestion on some link, or some other reason.

45. The address resolution protocol (ARP) is used for:

**GATE 2005**

- (a) Finding the IP address from the DNS
- (b) Finding the IP address of the default gateway
- (c) Finding the IP address that corresponds to a MAC address
- (d) Finding the MAC address that corresponds to an IP address

**ANS: D**

46. The maximum window size for data transmission using the selective reject protocol with n-bit frame sequence numbers is:

**GATE 2005**

- (a)  $2^n$
- (b)  $2^{n-1}$
- (c)  $2^n - 1$
- (d)  $2^{n-2}$

**ANS: B**

47. In a network of LANs connected by bridges, packets are sent from one LAN to another through intermediate bridges. Since more than one path may exist between two LANs, packets may have to be routed through multiple bridges. Why is the spanning tree algorithm used for bridge-routing?

**GATE 2005**

- (a) For shortest path routing between LANs
- (b) For avoiding loops in the routing paths
- (c) For fault tolerance
- (d) For minimizing collisions

**ANS: B**

48. An organization has a class B network and wishes to form subnets for 64 departments. The subnet mask would be:

**GATE 2005**

- (a) 255.255.0.0
- (b) 255.255.64.0
- (c) 255.255.128.0
- (d) 255.255.252.0

**ANS: D**

**SOL:** The size of network ID is 16 bit in class B networks. So bits after 16th bit must be used to create 64 departments. Total 6 bits are needed to identify 64 different departments. Therefore, subnet mask will be 255.255.252.0.

49. Suppose the round trip propagation delay for a 10 Mbps Ethernet having 48-bit jamming signal is 46.4 ms. The minimum frame size is:

GATE 2005

- (a) 94      (b) 416      (c) 464      (d) 512

**ANS: C**

**SOL:** Transmission Speed = 10Mbps.

Round trip propagation delay = 46.4 ms

$$\text{The minimum frame size} = (\text{Round Trip Propagation Delay}) * (\text{Transmission Speed}) \\ \equiv 10 * (10^6) * 46.4 * (10^{-3}) = 464 * 10^3 = 464 \text{ Kbit.}$$

The concept behind the above formula is collision detection. Consider a situation where a node A wants to send a frame to another node B. When Node A begins transmitting, the signal must propagate the network length. In the worst-case collision scenario, Node B begins to transmit just before the signal for Node A's frame reaches it. The collision signal of Node A and Node B's frame must travel back to Node A for Node A to detect that a collision has occurred.

The time it takes for a signal to propagate from one end of the network to the other is known as the propagation delay. In this worst-case collision scenario, the time that it takes for Node A to detect that its frame has been collided with is twice the propagation delay. Node A's frame must travel all the way to Node B, and then the collision signal must travel all the way from Node B back to Node A. This time is known as the slot time. An Ethernet node must be transmitting a frame for the slot time for a collision with that frame to be detected. This is the reason for the minimum Ethernet frame size.

50. Choose the best matching between Group 1 and Group 2

## **Group -1**

## **Group – 2**

- |                    |  |
|--------------------|--|
| P. Data link layer | 1. Ensures reliable transport of data over a physical Point-to-point link                                    |
| Q. Network layer   | 2. Encodes/decodes data for physical transmission  |
| R. Transport layer | 3. Allows end-to-end communication between two processes<br>4. Routes data from one network node to the next |

**ANS:** A

**SOL:** Transport layer is responsible for end to end communication, creation of sockets. Network layer routes the data from one node to other, till it reaches the destination. Datalink layer ensures reliable data transfer by error correction, duplication check, ordered delivery etc.

51. Which of the following is NOT true with respect to a transparent bridge and a router?

- (a) Both bridge and router selectively forward data packets
  - (b) A bridge uses IP addresses while a router uses MAC addresses
  - (c) A bridge builds up its routing table by inspecting incoming packets
  - (d) A router can connect between a LAN and a WAN

**ANS: B**

**SOL:** Bridge is the device which work at data link layer whereas router works at network layer. Both selectively forward packets, build routing table & connect between LAN & WAN but since bridge works at data link it uses MAC addresses to route whereas router uses IP addresses.

52. The routing table of a router is shown below:

Destination	Subnet Mask	Interface
128.75.43.0	255.255.255.0	Eth0
128.75.43.0	255.255.255.128	Eth1
192.12.17.5	255.255.255.255	Eth3
Default		Eth2

On which interface will the router forward packets addressed to destinations 128.75.43.16 and 192.12.17.10 respectively?

- (a) Eth1 and Eth2
  - (b) Eth0 and Eth2
  - (c) Eth0 and Eth3
  - (d) Eth1 and Eth3

**ANS: C**

**SOL:** Given IP Address

128.75.43.16. (1)  
Eth 0 128.75.43.0. (2)

Mask 255.255.255.0.  
Equation (1) & (2) both are of same network.  
192.12.17.10.

- (1)  
Eth3      192.12.17.5  
(2)  
Mask

255.255.255.255

Equation (1) & (2) both are of same network.

53. Which of the following assertions is FALSE about the Internet Protocol (IP)? **GATE 2003**

- (A) It is possible for a computer to have multiple IP addresses
- (B) IP packets from the same source to the same destination can take different routes in the network
- (C) IP ensures that a packet is discarded if it is unable to reach its destination within a given number of hops
- (D) The packet source cannot set the route of an outgoing packets; the route is determined only by the routing tables in the routers on the way

**ANS: A**

**SOL:** Internet protocol ensures that a packet is forwarded if it is able to reach its destination within a given no. of hops. One computer can have multiple IP addresses also packets having same source & destination can take different routes.

Source doesn't decide where to route the packet, but it is decided by the routing tables at intermediate routers.

54. Which of the following functionalities must be implemented by a transport protocol over and above the network protocol? **GATE 2003**

- (A) Recovery from packet losses
- (B) Detection of duplicate packets
- (C) Packet delivery in the correct order
- (D) End to end connectivity

**ANS: D**

**SOL:** Transport protocols are mainly for providing end to end connections by making sockets. Recovery from packet loss & delivery in correct order, duplication is checked by Data link layer.

### EXERCISE QUESTIONS

1. Frames of 1000 bits are sent over a  $10^6$  bps duplex link between two hosts. The propagation time is 25ms. Frames are to be transmitted into this link to maximally pack them in transit (within the link). **GATE 2009**

What is the minimum number of bits ( $l$ ) that will be required to represent the sequence numbers distinctly? Assume that no time gap needs to be given between transmission of two frames.

- 1)  $l = 2$

2)  $l = 3$

$$3) l = 4$$

$$4) l = 5$$

ANSWER: 4

2. In a packet switching network, packets are routed from source to destination along a single path having two intermediate nodes. If the message size is 24 bytes and each packet contains a header of 3 bytes, then the optimum packet size is:

(a) 4

- ANSWER:**

3. Suppose the round trip propagation delay for a 10 Mbps Ethernet having 48-bit jamming signal, is 16.4  $\mu$ s. The minimum frame size is:

(a) 94                    (b) 116                    (c) 161                    (d) 512

(a) 94

- ANSWER:

4. Assuming that the packets are correctly delivered, how many bytes, including headers, are delivered to the IP layer at the destination for one application message in the best case? Consider only data packets.

5. What is the rate at which application data is transferred to host HC?  
Ignore errors, acknowledgements, and other overheads.  
(a) 325.5 Kbps      (b) 354.5 Kbps      (c) 499.6 Kbps      (d) 512.0 Kbps

6. Which of the following assertions is FALSE about the Internet Protocol (IP)? 3 GATE 2003

- 1) It is possible for a computer to have multiple IP addresses
  - 2) IP packets from the same source to the same destination can take different routes in the network
  - 3) IP ensures that a packet is discarded if it is unable to reach its destination within a given number of hops
  - 4) The packet source cannot set the route of an outgoing packets; the route is determined only by the routing tables in the routers on the way

ANSWER: 1

7. A 2 km long broadcast LAN has  $10^7$  bps bandwidth and uses CSMA/CD. The signal travels along the wire at  $2 \times 10^8$  m/s. What is the minimum packet size that can be used on this network ?

- 1) 50 bytes
  - 2) 100 bytes

- 3) 200 bytes
- 4) None of these

ANSWER: 4

8. Host A is sending data to host B over a full duplex link. A and B are using the sliding window protocol for flow control. The send and receive window sizes are 5 packets each. Data packets (sent only from A to B) are all 1000 bytes long and the transmission time for such a packet is 50  $\mu$ s. Acknowledgement packets (sent only from B to A) are very small and require negligible transmission time. The propagation delay over the link is 200  $\mu$ s. What is the maximum achievable throughput in this communication ?
- 1)  $7.69 \times 10^6$  bps
  - 2)  $11.11 \times 10^6$  bps
  - 3)  $12.33 \times 10^6$  bps
  - 4)  $15.00 \times 10^6$  bps

ANSWER: 2

9. Purpose of a start bit in RS 232 serial communication protocol is GATE 1997
- (a) to synchronize receiver for receiving every byte
  - (b) to synchronize receiver for receiving a sequence of bytes
  - (c) a parity bit
  - (d) to synchronize receiver for receiving the last byte

ANSWER:

10. Which of the following services use TCP?

- 1. DHCP
- 2. SMTP
- 3. HTTP
- 4. TFTP
- 5. FTP

A.1 and 2

B.2, 3 and 5

C.1, 2 and 4

D.1, 3 and 4

49 .What layer in the TCP/IP stack is equivalent to the Transport layer of the OSI model?

- A.Application
- B.Host-to-Host
- C.Internet
- D.Network Access

11. Which of the following describe the DHCP Discover message?

- 1. It uses FF:FF:FF:FF:FF as a layer 2 broadcast.
- 2. It uses UDP as the Transport layer protocol.
- 3. It uses TCP as the Transport layer protocol.

4. It does not use a layer 2 destination address.

- |            |            |
|------------|------------|
| A. 1 only  | B. 1 and 2 |
| C. 3 and 4 | D. 4 only  |

12. You want to implement a mechanism that automates the IP configuration, including IP address, subnet mask, default gateway, and DNS information. Which protocol will you use to accomplish this?

- A.SMTP      B.SNMP      C.DHCP      D.ARP

13. Which of the following allows a router to respond to an ARP request that is intended for a remote host?

- A. Gateway DP  
B. Reverse ARP (RARP)  
C. Proxy ARP  
D. Inverse ARP (IARP)

14. The DoD model (also called the TCP/IP stack) has four layers. Which layer of the DoD model is equivalent to the Network layer of the OSI model?

- A. Application  
B. Host-to-Host  
C. Internet  
D. Network Access

15. Which of the following services use UDP?

1. DHCP  
2. SMTP  
3. SNMP  
4. FTP  
5. HTTP  
6. TFTP  
  
A. 1, 3 and 6  
B. 2 and 4  
C. 1, 2 and 4  
D. All of the above

16. Which of the following statements are true regarding the command ip route 172.16.4.0 255.255.255.0 192.168.4.2?

1. The command is used to establish a static route.  
2. The default administrative distance is used.  
3. The command is used to configure the default route.  
4. The subnet mask for the source address is 255.255.255.0.  
  
A. 1 and 2  
B. 2 and 4

- C. 3 and 4
- D. All of the above

17. Which statement is true regarding classless routing protocols?

- 1. The use of discontiguous networks is not allowed.
  - 2. The use of variable length subnet masks is permitted.
  - 3. RIPv1 is a classless routing protocol.
  - 4. IGRP supports classless routing within the same autonomous system.
  - 5. RIPv2 supports classless routing.
- A. 1, 3 and 5
  - B. 3 and 4
  - C. 2 and 5
  - D. None of the above

18. Which two of the following are true regarding the distance-vector and link-state routing protocols?

- 1. Link state sends its complete routing table out all active interfaces on periodic time intervals.
  - 2. Distance vector sends its complete routing table out all active interfaces on periodic time intervals.
  - 3. Link state sends updates containing the state of its own links to all routers in the internetwork.
  - 4. Distance vector sends updates containing the state of its own links to all routers in the internetwork.
- A. 1 only
  - B. 3 only
  - C. 2 and 3 only
  - D. None of the above

19. Which of the following is an example of a standard IP access list?

- A. access-list 110 permit host 1.1.1.1
- B. access-list 1 deny 172.16.10.1 0.0.0.0
- C. access-list 1 permit 172.16.10.1 255.255.0.0
- D. access-list standard 1.1.1.1

59. What flavor of Network Address Translation can be used to have one IP address allow many users to connect to the global Internet?

- A. NAT   B. Static   C. Dynamic   D. PAT

20. You have 10 users plugged into a hub running 10Mbps half-duplex. There is a server connected to the switch running 10Mbps half-duplex as well. How much bandwidth does each host have to the server?

- A. 100 kbps
- B. 1 Mbps
- C. 2 Mbps

- D. 10 Mbps
21. What protocol does PPP use to identify the Network layer protocol?  
A. NCP B. ISDN  
C. HDLC D. LCP
22. The Acme Corporation is implementing dial-up services to enable remote-office employees to connect to the local network. The company uses multiple routed protocols, needs authentication of users connecting to the network, and since some calls will be long distance, needs callback support. Which of the following protocols is the best choice for these remote services?  
A. 802.1  
B. Frame Relay  
C. HDLC  
D. PPP  
E. PAP6.
23. Which encapsulations can be configured on a serial interface?  
1. Ethernet  
2. Token Ring  
3. HDLC  
4. Frame Relay  
5. PPP  
  
A. 1 and 4  
B. 2 only  
C. 3, 4 and 5  
D. All of the above
24. Which of the following describes the creation of private networks across the Internet, enabling privacy and tunneling of non-TCP/IP protocols?  
A. HDLC B. Cable  
C. VPN D. IPSec  
E. xDSL
25. Which of the following describes an industry-wide standard suite of protocols and algorithms that allows for secure data transmission over an IP-based network that functions at the layer 3 Network layer of the OSI model?  
A. HDLC B. Cable  
C. VPN D. IPSec  
E. xDSL17.
26. Which of the following encapsulates PPP frames in Ethernet frames and uses common PPP features like authentication, encryption, and compression?  
A. PPP

- B. PPPoA
- C. PPPoE
- D. Token Ring

27. Routers operate at layer \_\_\_\_\_. LAN switches operate at layer \_\_\_\_\_. Ethernet hubs operate at layer \_\_\_\_\_. Word processing operates at layer \_\_\_\_\_.  
  - A. 3, 3, 1, 7
  - B. 3, 2, 1, none
  - C. 3, 2, 1, 7
  - D. 3, 3, 2, none
28. Which of the following describe router functions?  
  - A. Packet switching
  - B. Packet filtering
  - C. Internetwork communication
  - D. Path selection
  - E. All of the above
29. A receiving host has failed to receive all of the segments that it should acknowledge. What can the host do to improve the reliability of this communication session?  
  - A. Send a different source port number.
  - B. Restart the virtual circuit.
  - C. Decrease the sequence number.
  - D. Decrease the window size.
30. Why does the data communication industry use the layered OSI reference model?  
  - 1. It divides the network communication process into smaller and simpler components, thus aiding component development, design, and troubleshooting.
  - 2. It enables equipment from different vendors to use the same electronic components, thus saving research and development funds.
  - 3. It supports the evolution of multiple competing standards and thus provides business opportunities for equipment manufacturers.
  - 4. It encourages industry standardization by defining what functions occur at each layer of the model.  
  - A. 1 only
  - B. 1 and 4
  - C. 2 and 3
  - D. 3 only