18CSC302J- Computer Networks

Unit I

Syllabus - Unit I

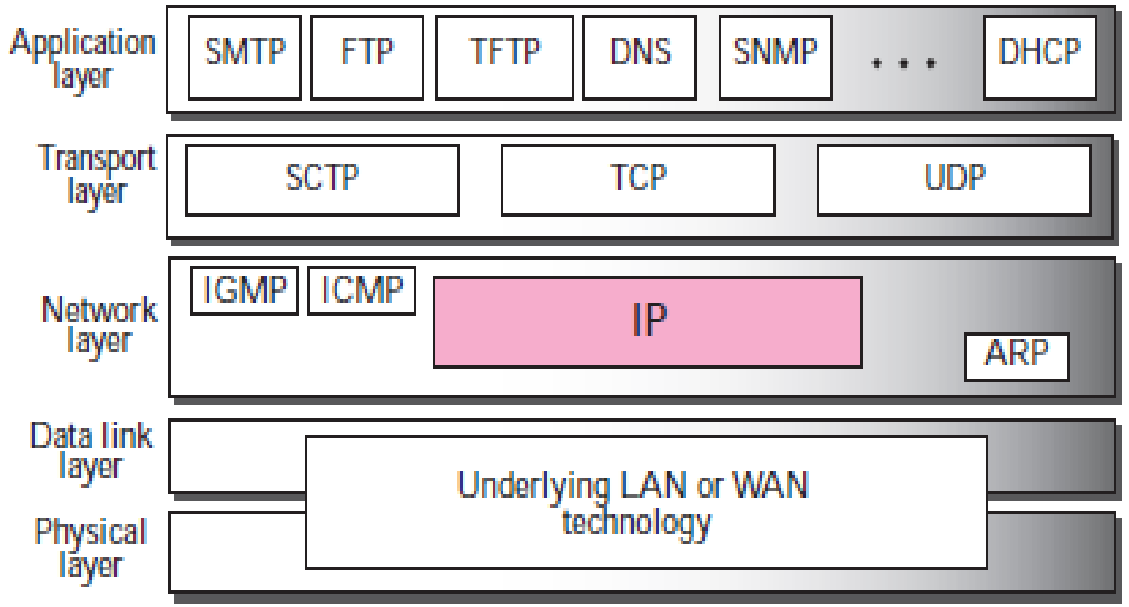
- IP Header
- IP Fragmentation
- ARP
- RARP
- ICMP
  - Introduction
  - Messages
  - Debugging Tools
  - ICMP Package
- UDP Datagram
  - Characteristics
- TCP Header
  - TCP Connection Establishment Process
  - Error Control
  - Congestion Control
  - Flow Control
- Multicasting & Multicast Routing Protocols
- Stream Control Transmission Protocol

IP Header & IP Fragmentation

IP Header & IP Fragmentation

IP– An Introduction

- The **Internet Protocol (IP)** is the transmission mechanism used by the TCP/IP protocols at the network layer Operates at higher level



Position of IP in TCP/IP protocol suite

IP Header & IP Fragmentation

IP– An Introduction

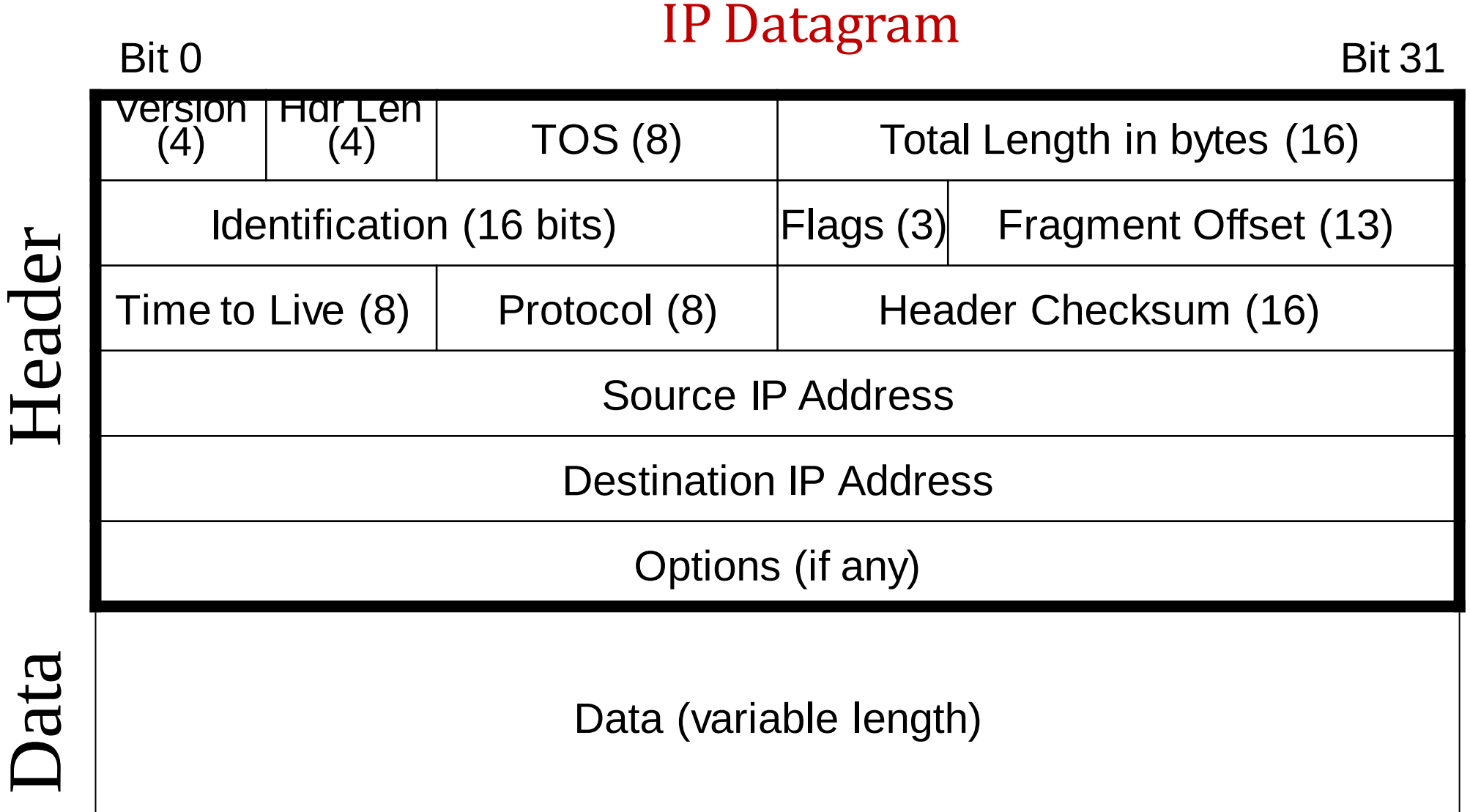
- IP is an unreliable and connectionless datagram protocol—a **best-effort delivery** service.
- The term *best-effort* means that IP packets can be corrupted, lost, arrive out of order, or delayed and may create congestion for the network.
- If reliability is important, IP must be paired with a reliable protocol such as TCP.

IP Header & IP Fragmentation

IP– An Introduction

- example
  - ✓ The post office does its best to deliver the mail but does not always succeed. If an unregistered letter is lost, it is up to the sender or would-be recipient to discover the loss and rectify the problem.
  - ✓ The post office itself does not keep track of every letter and cannot notify a sender of loss or damage.
- IP is also a connectionless protocol for a packet switching network that uses the datagram approach
- This means that each datagram is handled independently, and each datagram can follow a different route to the destination.

IP Header & IP Fragmentation



IP Header & IP Fragmentation

- Version
  - Version number of IP protocol
  - Current version is Version 4
  - Version 6 has different header format

IP Packet Header

Bit 0 Bit 31

Version (4)	Hdr Len (4)	TOS (8)	Total Length in bytes (16)
Identification (16 bits)		Flags (3)	Fragment Offset (13)
Time to Live (8)	Protocol (8)	Header Checksum (16)	
Source IP Address			
Destination IP Address			
Options (if any)			

IP Header & IP Fragmentation

- Header Length (in 32 bit words)
  - Indicates end of header and beginning of payload
  - If no options, Header length = 5

- Bit 0 Bit 31

Version (4)	Hdr Len (4)	TOS (8)	Total Length in bytes (16)
Identification (16 bits)		Flags (3)	Fragment Offset (13)
Time to Live (8)	Protocol (8)	Header Checksum (16)	
Source IP Address			
Destination IP Address			
Options (if any)			

IP Header & IP Fragmentation

- Type of Service (TOS)
  - Allows different types of service to be requested
  - Initially, meaning was not well defined
  - Currently being defined (diffserv)

Bit 0 Bit 31

Version (4)	Hdr Len (4)	TOS (8)	Total Length in bytes (16)	
Identification (16 bits)			Flags (3)	Fragment Offset (13)
Time to Live (8)	Protocol (8)		Header Checksum (16)	
Source IP Address				
Destination IP Address				
Options (if any)				

IP Header & IP Fragmentation

- Packet Length (in Bytes)
  - Unambiguously specify end of packet
  - Max packet size = 2^{16} = 65,535 Bytes

Bit 0 Bit 31

Version (4)	Hdr Len (4)	TOS (8)	Total Length in bytes (16)	
Identification (16 bits)			Flags (3)	Fragment Offset (13)
Time to Live (8)		Protocol (8)	Header Checksum (16)	
Source IP Address				
Destination IP Address				
Options (if any)				

IP Header & IP Fragmentation

IP Packet Header

- These three fields for Fragmentation Control

Bit 0 Bit 31

Version (4)	Hdr Len (4)	TOS (8)	Total Length in bytes (16)
Identification (16 bits)		Flags (3)	Fragment Offset (13)
Time to Live (8)	Protocol (8)	Header Checksum (16)	
Source IP Address			
Destination IP Address			
Options (if any)			

IP Header & IP Fragmentation

IP Packet Header

- Time to Live
  - Initially set by sender (up to 255)
  - Decremented by each router
  - Discard when TTL = 0 to avoid infinite routing loops

Version (4)	Hdr Len (4)	TOS (8)	Total Length in bytes (16)	
Identification (16 bits)			Flags (3)	Fragment Offset (13)
Time to Live (8)		Protocol (8)	Header Checksum (16)	
Source IP Address				
Destination IP Address				
Options (if any)				

IP Header & IP Fragmentation

IP Packet Header

- Protocol
  - Value indicates what is in the data field
  - Example: TCP or UDP

Bit 0 Bit 31

Version (4)	Hdr Len (4)	TOS (8)	Total Length in bytes (16)	
Identification (16 bits)			Flags (3)	Fragment Offset (13)
Time to Live (8)		Protocol (8)	Header Checksum (16)	
Source IP Address				
Destination IP Address				
Options (if any)				

IP Header & IP Fragmentation

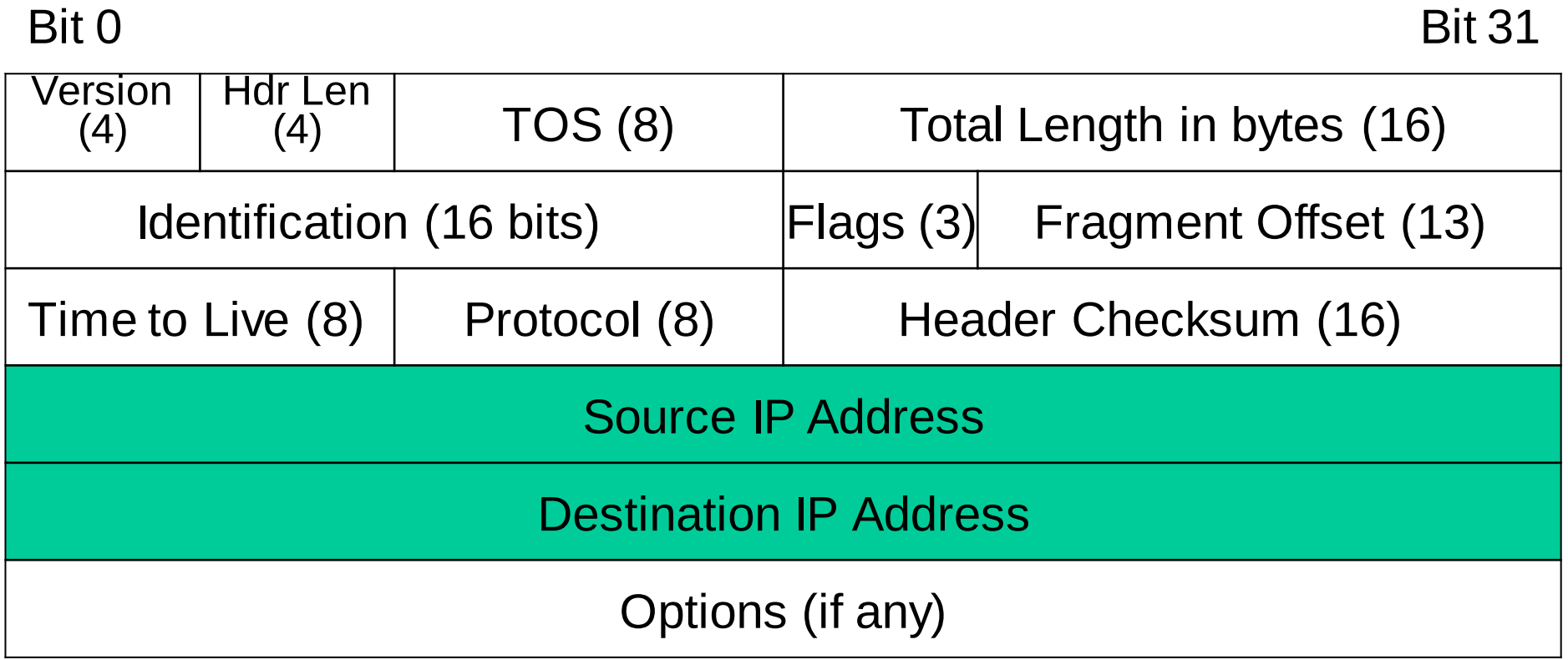
- Header Checksum
  - Checks for error in the header only
  - Bad headers can harm the network
  - If error found, packet is simply discarded

- Bit 31

Version (4)	Hdr Len (4)	TOS (8)	Total Length in bytes (16)	
Identification (16 bits)			Flags (3)	Fragment Offset (13)
Time to Live (8)		Protocol (8)	Header Checksum (16)	
Source IP Address				
Destination IP Address				
Options (if any)				

IP Header & IP Fragmentation

- Source and Destination IP Addresses
  - Strings of 32 ones and zeros



IP Header & IP Fragmentation

IP Packet Header

- Options
  - Example: timestamp, record route, source route

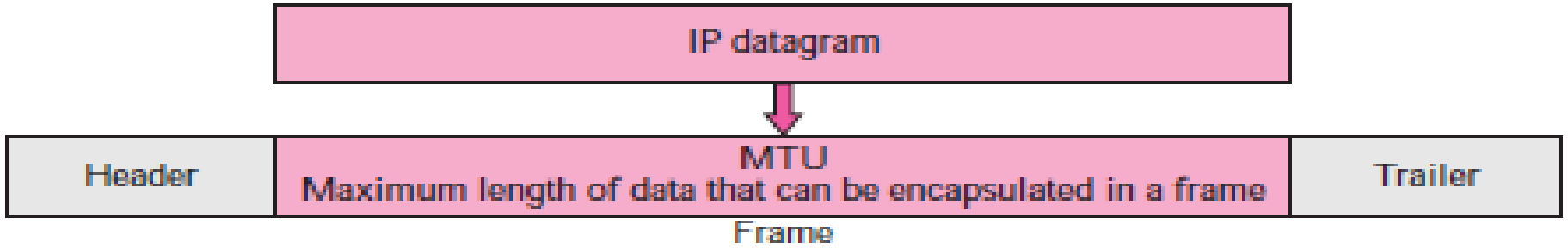
Bit 0 Bit 31

Version (4)	Hdr Len (4)	TOS (8)	Total Length in bytes (16)
Identification (16 bits)		Flags (3)	Fragment Offset (13)
Time to Live (8)	Protocol (8)	Header Checksum (16)	
Source IP Address			
Destination IP Address			
Options (if any)			

IP Header & IP Fragmentation

IP Fragmentation & Reassembly

- Maximum Transmission Unit (MTU)
  - Largest IP packet a network will accept
  - Arriving IP packet may be larger (max IP packet size = 65,535 bytes)
- Sender or router will split the packet into multiple fragments
- Destination will reassemble the packet
- IP header fields used to identify and order related fragments



IP Header & IP Fragmentation

IP Fragmentation & Reassembly

- Divide the datagram to make it possible to pass through these networks called **fragmentation.**
- A fragmented datagram may itself be fragmented if it encounters a network with an even smaller MTU.
- A datagram can be fragmented by the source host or any router in the path
- the reassembly of the datagram, however, is done only by the destination host

IP Header & IP Fragmentation

- Identification
  - – All fragments of a single datagram have the same identification number

- Bit 0 Bit 31

Version (4)	Hdr Len (4)	TOS (8)	Total Length in bytes (16)	
Identification (16 bits)			Flags (3)	Fragment Offset (13)
Time to Live (8)		Protocol (8)	Header Checksum (16)	
Source IP Address				
Destination IP Address				
Options (if any)				

IP Header & IP Fragmentation

- Flags:
  - 1st bit: reserved, must be zero
  - 2nd bit: DF -- Do Not Fragment
  - 3rd bit: MF -- More Fragments

Bit 0 Bit 31

Version (4)	Hdr Len (4)	TOS (8)	Total Length in bytes (16)
Identification (16 bits)		Flags (3)	Fragment Offset (13)
Time to Live (8)	Protocol (8)	Header Checksum (16)	
Source IP Address			
Destination IP Address			
Options (if any)			

IP Header & IP Fragmentation

- Fragment Offset (in units of 8 bytes)
  - Used for reassembly of packet
  - 1st fragment has offset = 0

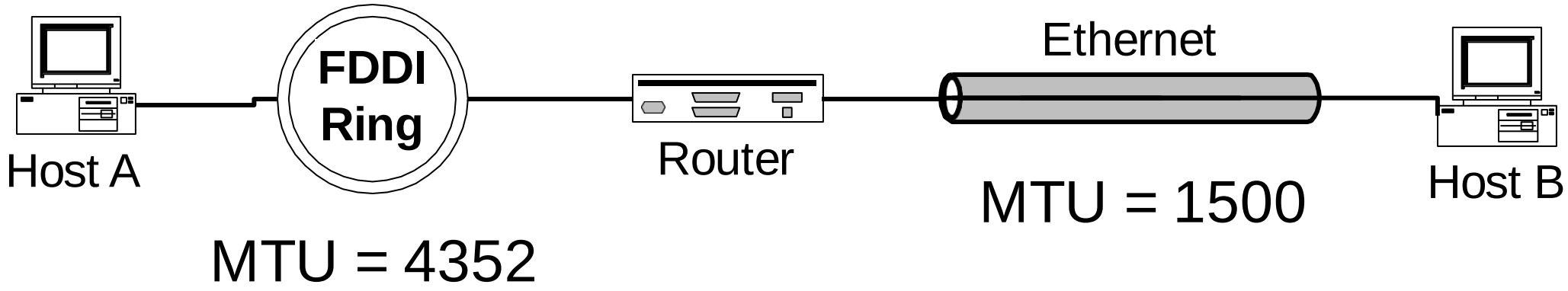
Bit 0 Bit 31

Version (4)	Hdr Len (4)	TOS (8)	Total Length in bytes (16)
Identification (16 bits)		Flags (3)	Fragment Offset (13)
Time to Live (8)	Protocol (8)	Header Checksum (16)	
Source IP Address			
Destination IP Address			
Options (if any)			

IP Header & IP Fragmentation

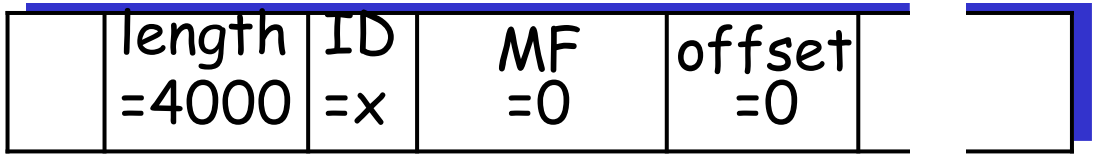
IP Fragmentation Example

- Host A wants to send to Host B an IP datagram of size = 4000 Bytes

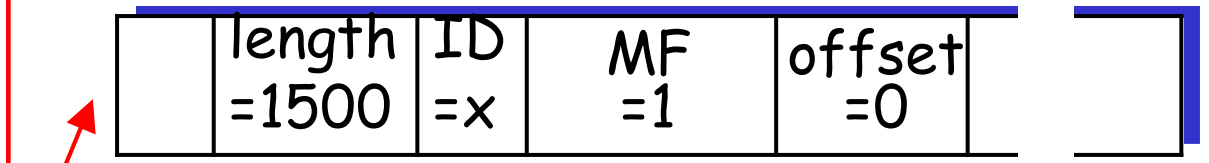


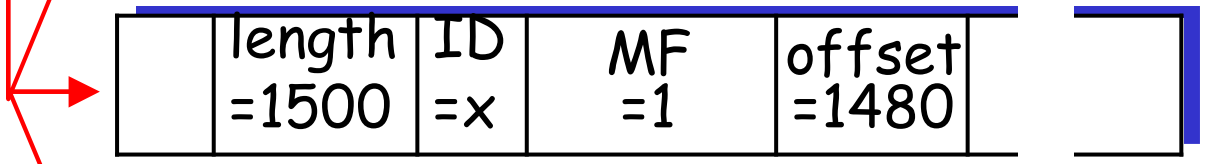
IP Header & IP Fragmentation

IP Fragmentation Example



One large datagram becomes several smaller datagrams





IP Header & IP Fragmentation

Multiple Fragmenting Points

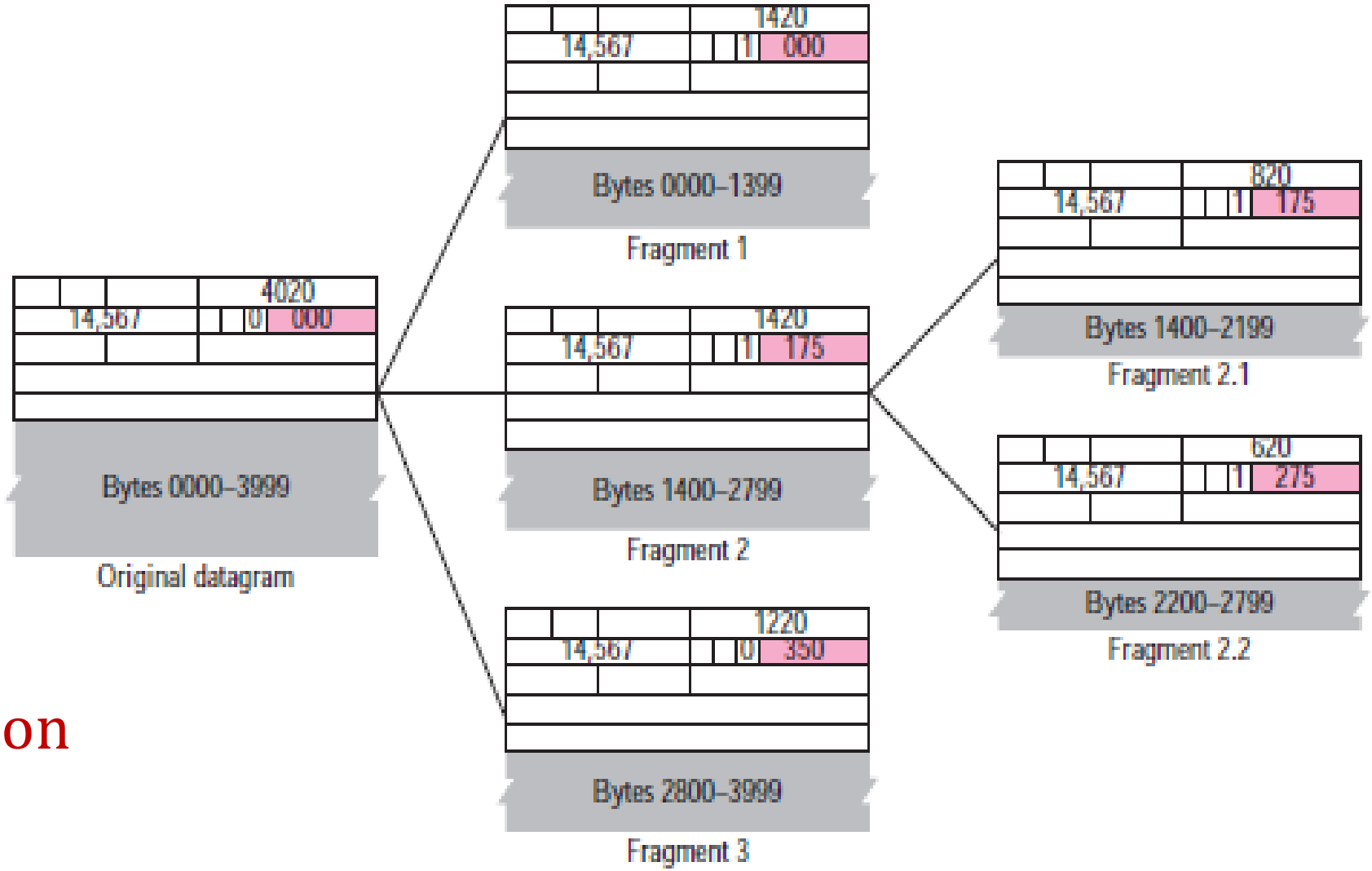
Let MTUs along internet path be

- 1500
- 1500
- 1000
- 1500
- 576
- 1500

Result: fragmentation can occur twice

IP Header & IP Fragmentation

Multiple Fragmenting Points



Detailed fragmentation example

IP Header & IP Fragmentation

- The figure shows what happens if a fragment itself is fragmented.

  In this case the value of the offset field is always relative to the original datagram.

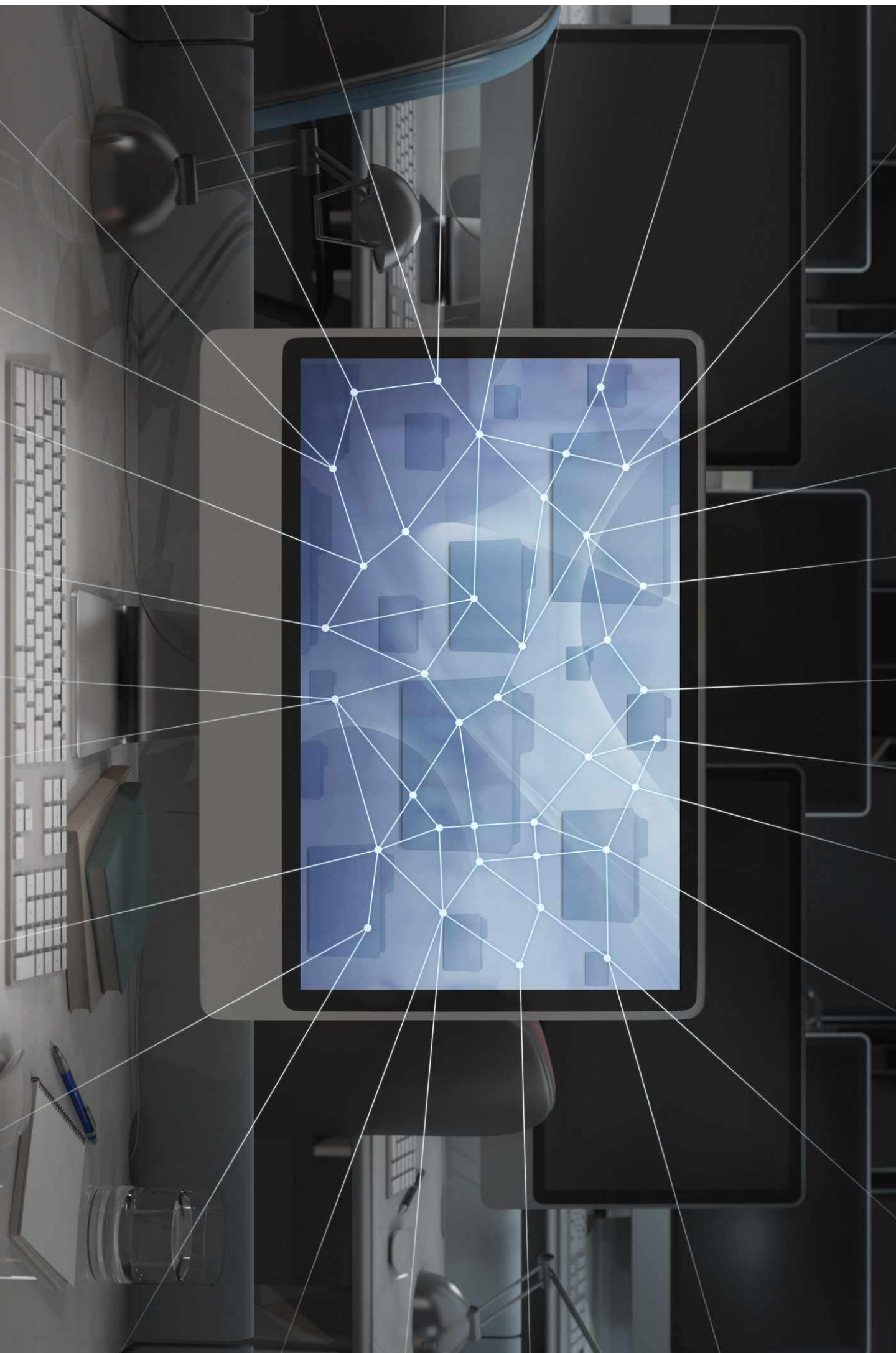
  a. The first fragment has an offset field value of zero.

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

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

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

ARP & RARP



ARP & RARP

ARP– An Introduction

- Logical Addresses
  - ✓ The hosts and routers are recognized at the network level by their *logical addresses*
    - A **logical address** is an internet address
    - Called a *logical* address because it is usually implemented in software
    - The logical addresses in the TCP/IP are called **IP address** and are 32 bits long

ARP & RARP

ARP– An Introduction

- Physical Address
  - ✓ However, hosts/routers are recognized at the physical layer by their physical address
    - A **physical address** is an local address
    - Called a physical address because it is usually implemented in hardware
    - Examples
      - 48-bit MAC addresses in Ethernet

ARP & RARP

ARP– An Introduction

- Translation

- We need both the physical address and the logical address for packet delivery.
- Thus, we need to be able to map a logical address to its corresponding physical address and vice versa
- Solutions
  - ***Static mapping***
  - ***Dynamic mapping***

ARP & RARP

ARP– An Introduction

- Static Mapping
  - Create a table that associates a logical address with a physical address and store in each machine
  - However, physical addresses may change A machine could change its NIC resulting in a new physical address
  - In some LANs, such as Local Talk, the physical address changes every time the computer is turned on.
  - A mobile station can move from one physical network to another, resulting in a change in its physical address

ARP & RARP

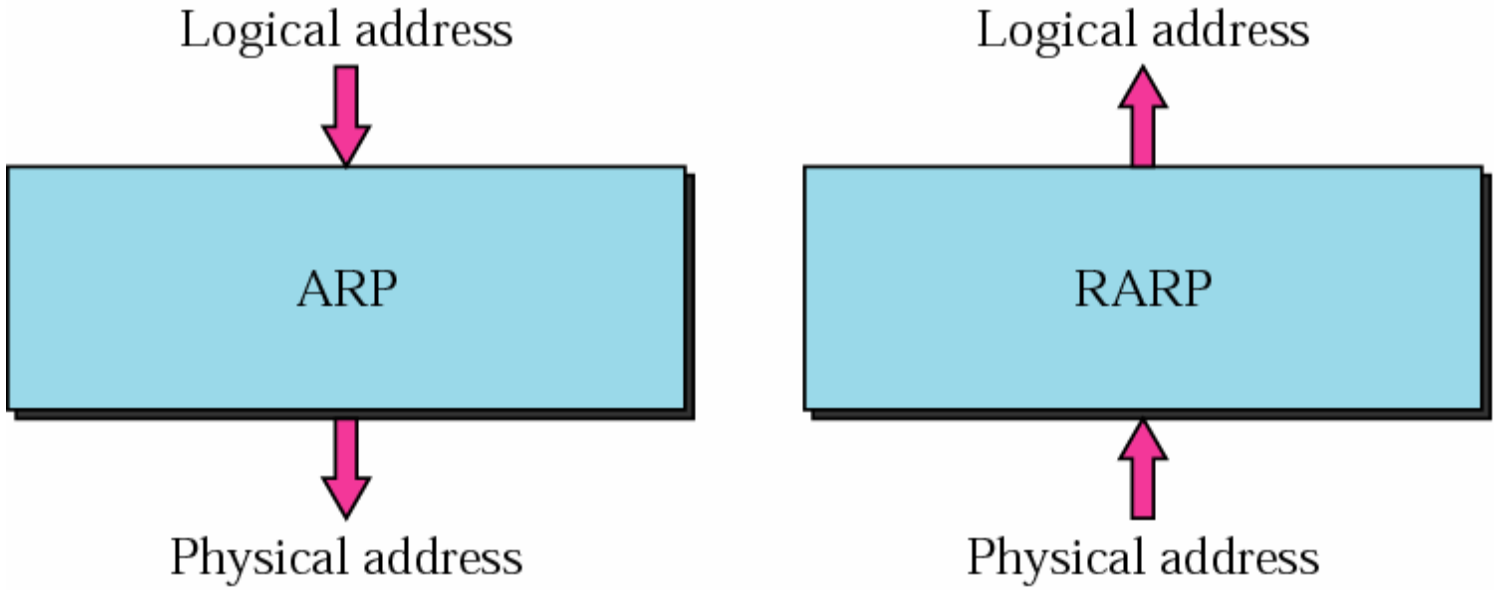
ARP– An Introduction

- Dynamic Mapping
- Use a protocol to find another address
- ARP: Address Resolution Protocol
  - Map a logical address to a physical address
- RARP: Reverse Address Resolution Protocol
  - Map a physical address to a logical address

ARP & RARP

ARP– An Introduction

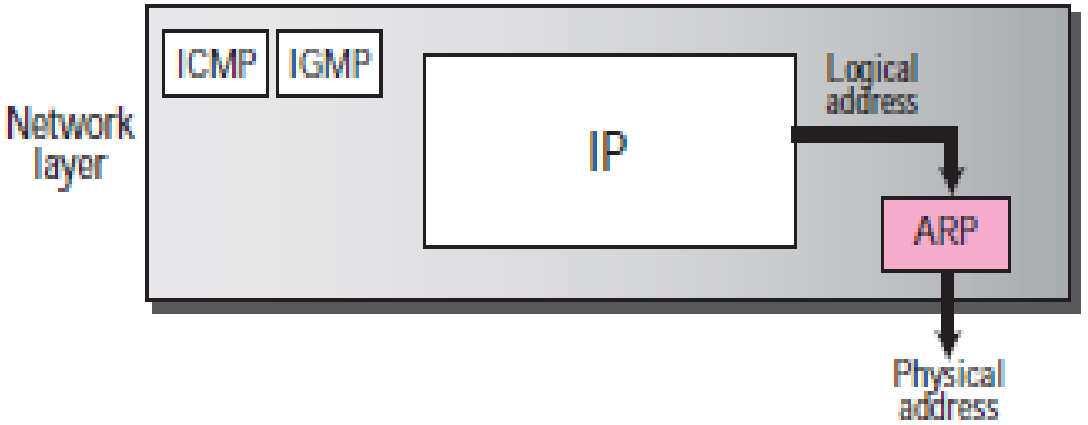
ARP and RARP



ARP & RARP

ARP– An Introduction

Position of ARP and RARP in TCP/IP Protocol Suite



ARP & RARP

ARP– An Introduction

ARP Operation

- To find the physical address of another host or router on its network
  - ✓ Send an ARP request message
- ARP request message
  - ✓ The physical address of the sender
  - ✓ The IP address of the sender
  - ✓ The physical address of the receiver is ***0s***
  - ✓ The IP address of the receiver

ARP & RARP

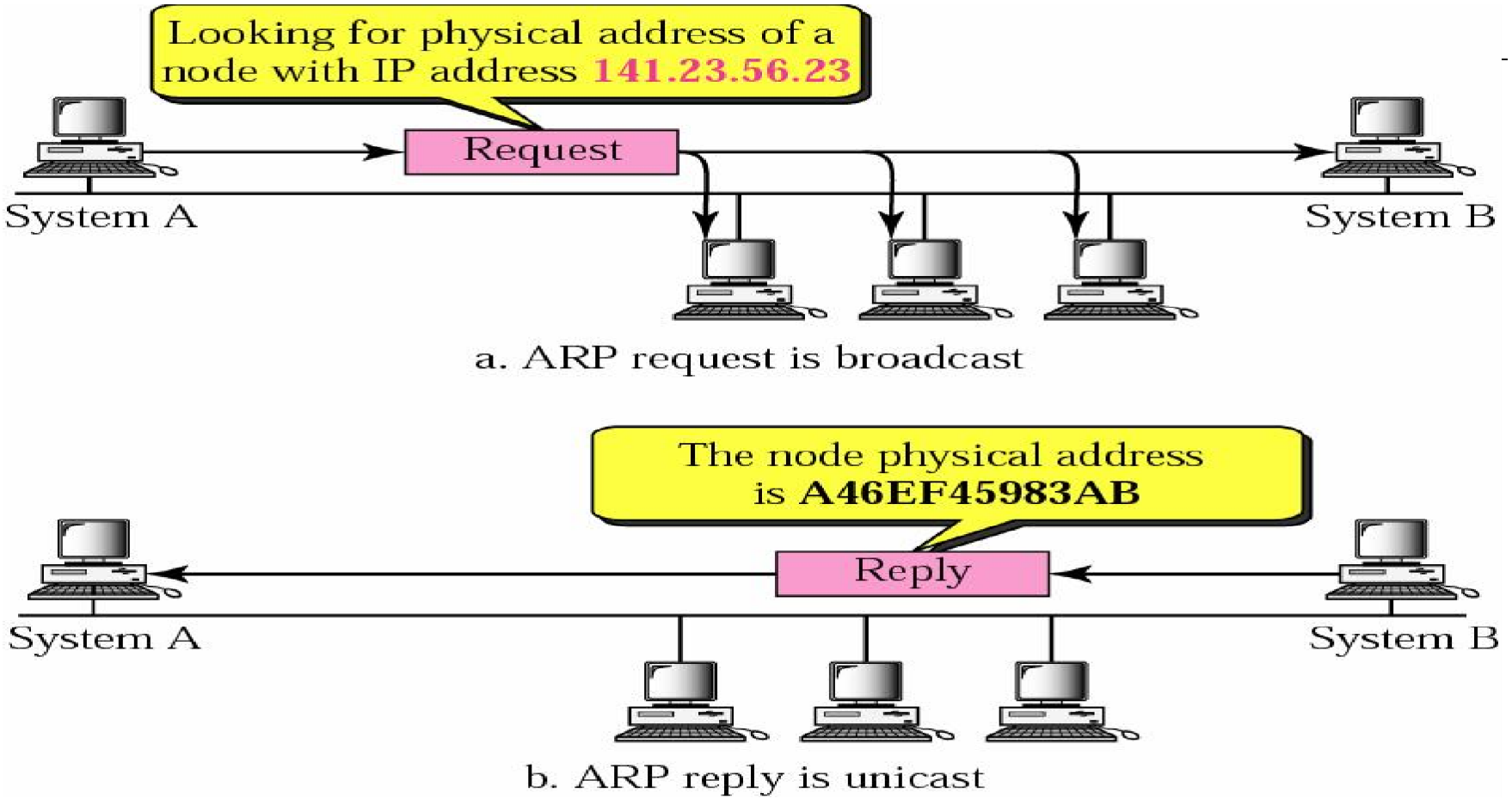
ARP– An Introduction

ARP Operation CONT..

- Then, ARP request message is broadcast by the physical layer
  - For example: in Ethernet, MAC header's destination address is all *1s* (broadcast address)
  - Received by every station on the physical network
- The intended recipient send back an ARP reply message
  - ARP reply message packet is ***unicast***

ARP & RARP

ARP– An Introduction



ARP & RARP

ARP– An Introduction

ARP Packet

Hardware Type		Protocol Type
Hardware length	Protocol length	Operation Request 1, Reply 2
Sender hardware address (For example, 6 bytes for Ethernet)		
Sender protocol address (For example, 4 bytes for IP)		
Target hardware address (For example, 6 bytes for Ethernet) (It is not filled in a request)		
Target protocol address (For example, 4 bytes for IP)		

ARP & RARP

ARP– An Introduction

Packet Format

- HTYPE (Hardware type)
  - 16-bit field defining the underlying type of the network
    - Ethernet is given the type 1
    - ARP can be used on any physical network
- PTYPE (Protocol type)
  - 16-bit field defining the protocol
    - IPv4 is 0800_{16}
    - ARP can be used with any higher-level protocol

ARP & RARP

ARP– An Introduction

Packet Format

- HLEN (Hardware length)
  - 8-bit field defining the length of the physical address in bytes
    - Ethernet has the value of 6
- PLEN (Protocol length)
  - 8-bit field defining the length of the logical address in bytes
    - IPv4 has the value of 4
- OPER (Operation)
  - 16-bit field defining the type of packet
  - (1) = ARP request, (2) = ARP reply

ARP & RARP

ARP– An Introduction

Packet Format

- HLEN (Hardware length)
  - 8-bit field defining the length of the physical address in bytes
    - Ethernet has the value of 6
- PLEN (Protocol length)
  - 8-bit field defining the length of the logical address in bytes
    - IPv4 has the value of 4
- OPER (Operation)
  - 16-bit field defining the type of packet
  - (1) = ARP request, (2) = ARP reply

ARP & RARP

ARP– An Introduction

Packet Format

- SHA (Sender hardware address)
  - A variable-length field defining the physical address of the sender
- SPA (Sender protocol address)
  - A variable-length field defining the logical address of the sender

ARP & RARP

ARP– An Introduction

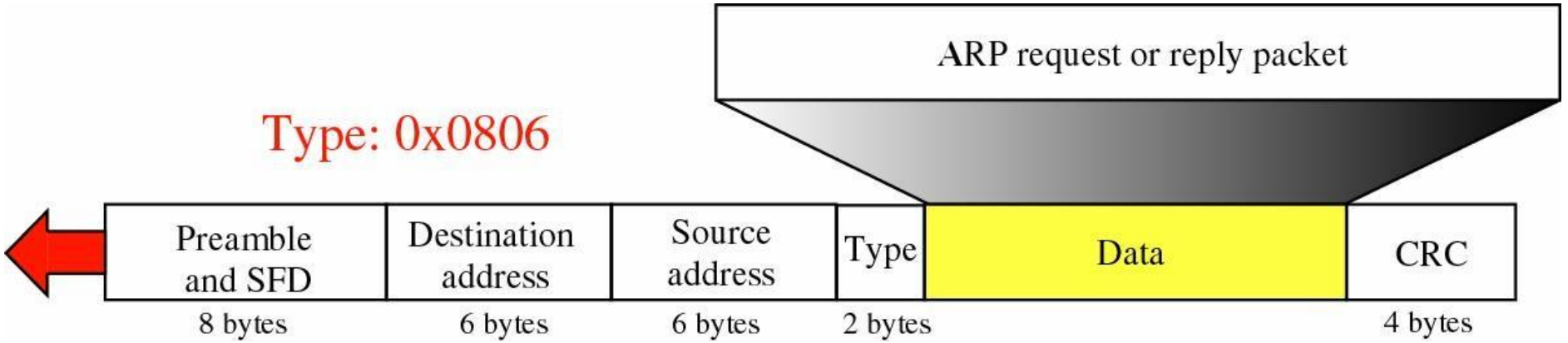
Packet Format

- THA (Target hardware address)
  - A variable-length field defining the physical address of the target
  - For an ARP request operation packet
    - This field is all 0s
- TPA (Target protocol address)
  - A variable-length field defining the logical address of the target

ARP & RARP

ARP– An Introduction

Encapsulation of ARP Packet



- An ARP packet is encapsulated directly into a data link frame
- Type field indicates that the data carried by the frame is an ARP packet

ARP & RARP

ARP– An Introduction

Operations

- The sender knows the target's IP address
- IP asks ARP to create an ARP request message
  - The sender physical address & The sender IP address
  - The target physical address field is filled with 0s
  - The target IP address
- The message is passed to the data link layer to encapsulate in a data link frame
  - Physical destination address is broadcast address

ARP & RARP

ARP– An Introduction

Operations

- Every host or routers receives the frame and since the destination address is broadcast, pass it to the ARP
  - All machines' ARP except the one targeted drop the packet
- The target reply with an ARP reply message that contains its physical address and is unicast
- The sender receives the reply message and knows the target's physical address

ARP & RARP

ARP– An Introduction

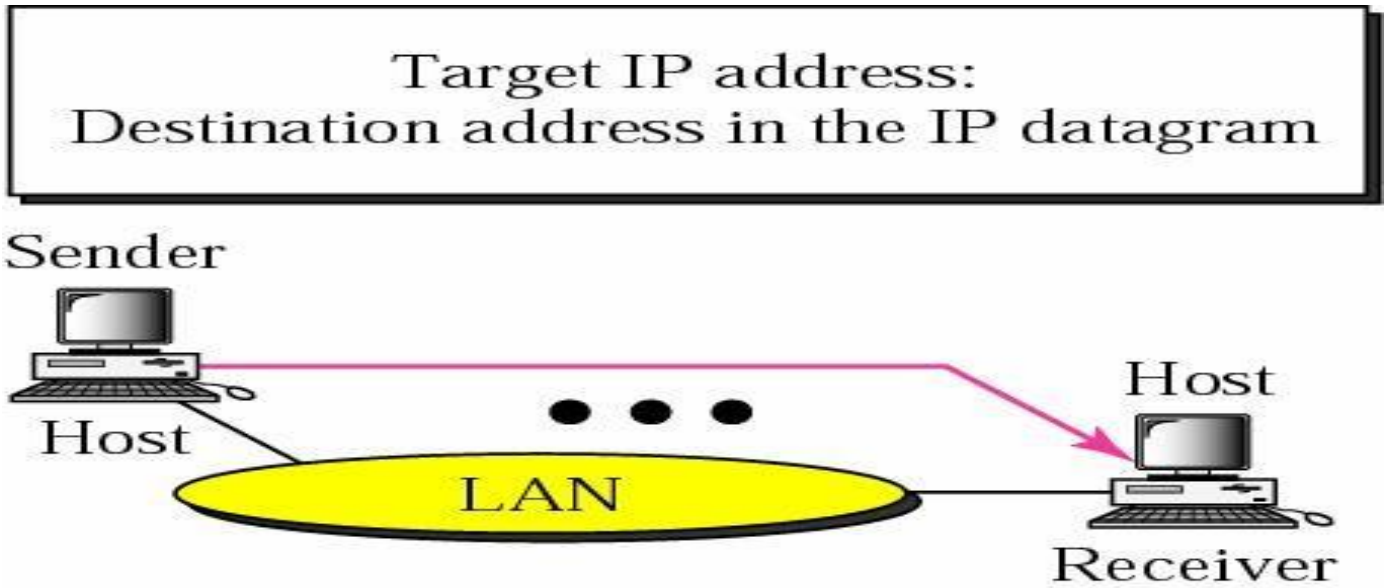
Four Cases to Use ARP

- ***Case 1:*** The sender is a host and wants to send a packet to another host on the same network
  - Use ARP to find another host's physical address
- ***Case 2:*** The sender is a host and wants to send a packet to another host on another network
  - Sender looks at its routing table
  - Find the IP address of the next hop (router) for this destination
  - Use ARP to find the router's physical address

ARP & RARP

ARP– An Introduction

Four Cases Using ARP: Case 1

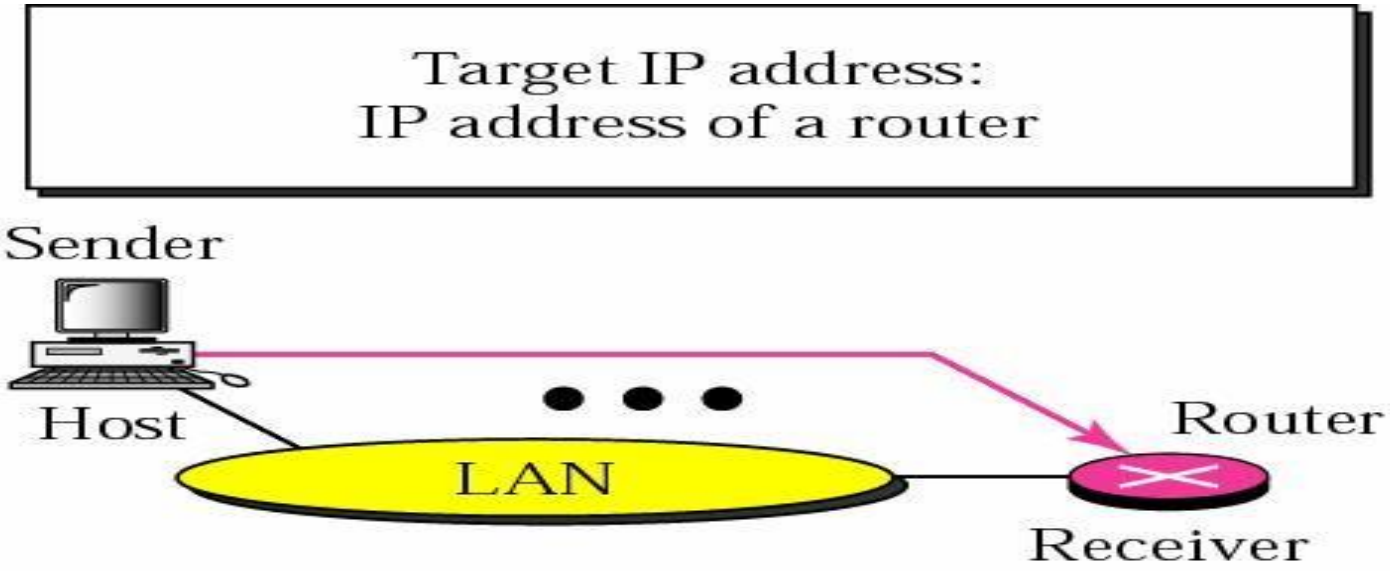


Case 1. A host has a packet to send to another host on the same network.

ARP & RARP

ARP– An Introduction

Four Cases Using ARP: Case 2



Case 2. A host wants to send a packet to another host on another network.
It must first be delivered to a router.

ARP & RARP

ARP– An Introduction

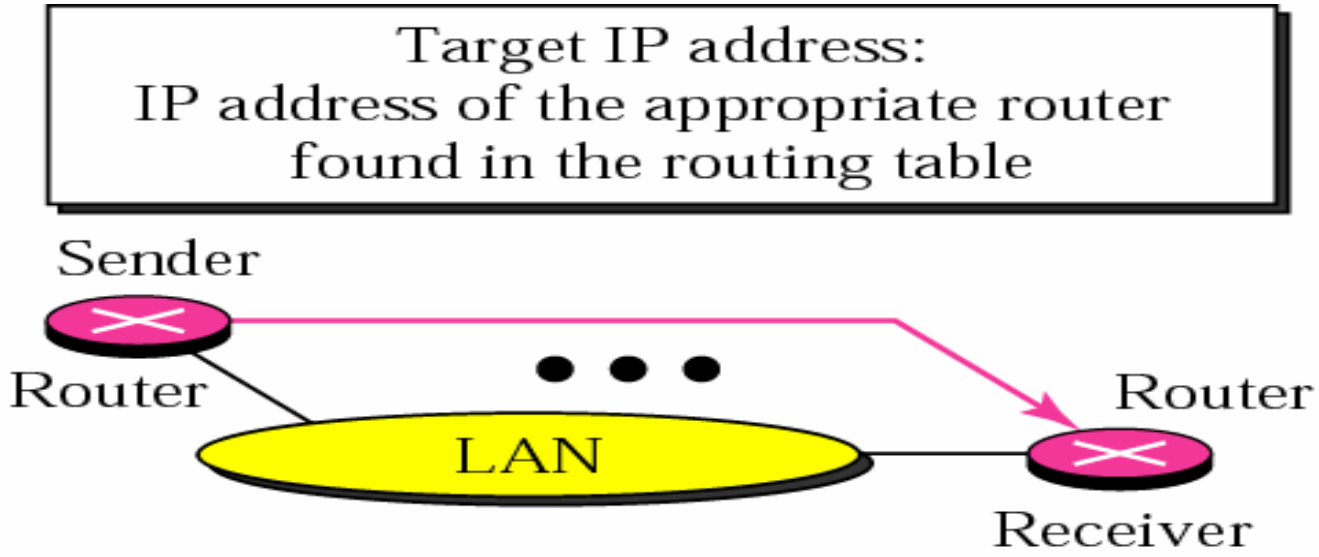
Four Cases to Use ARP

- ***Case 3:*** the sender is a router and received a datagram destined for a host on another network
  - Router check its routing table & find the IP address of the next router
  - Use ARP to find the next router's physical address
- ***Case 4:*** the sender is a router that has received a datagram destined for a host in the same network
  - Use ARP to find this host's physical address

ARP & RARP

ARP– An Introduction

Four Cases Using ARP: Case 3

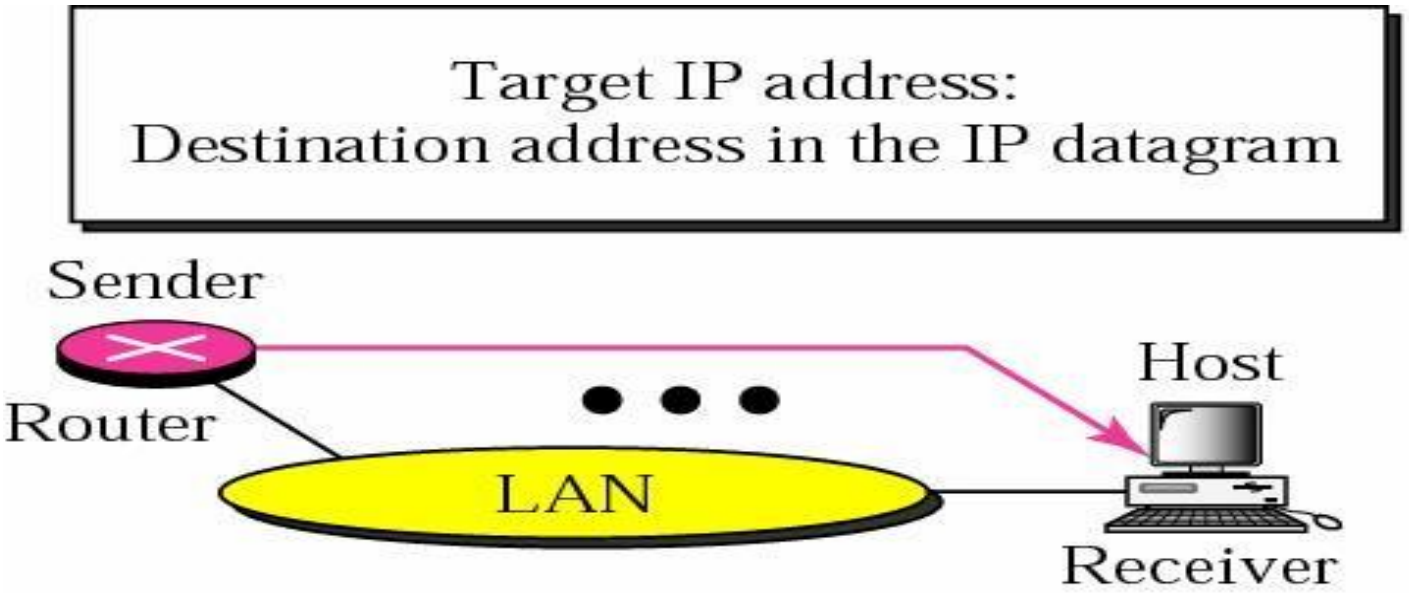


Case 3. A router receives a packet to be sent to a host on another network. It must first be delivered to the appropriate router.

ARP & RARP

ARP– An Introduction

Four Cases Using ARP: Case 4



Case 4. A router receives a packet to be sent to a host on the same network.

ARP & RARP

ARP– An Introduction

An ARP request is broadcast;

an ARP reply is unicast

ARP & RARP

ARP– An Introduction

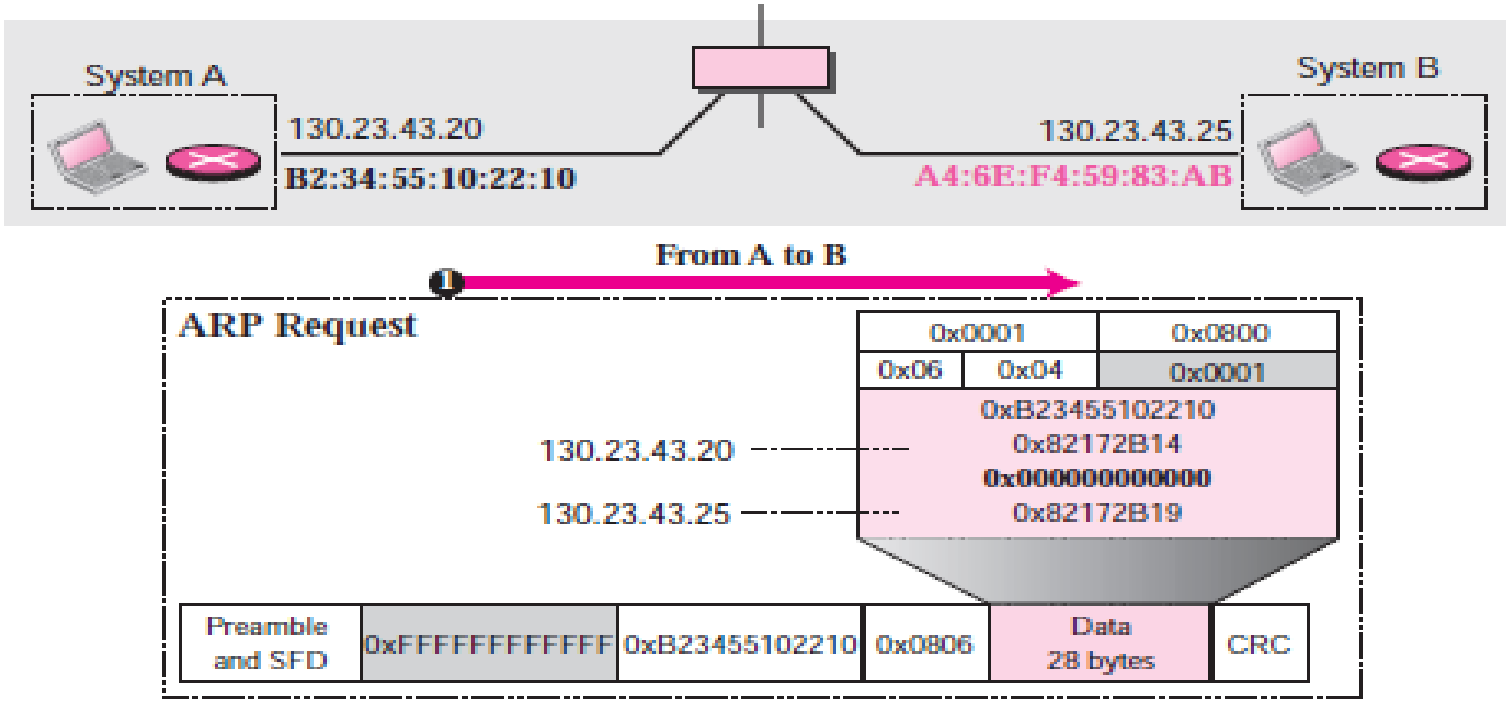
Example 1

- A host with IP address 130.23.43.20 and physical address 0xB23455102210
- Another host with IP address 130.23.43.25 and physical address 0xA46EF45983AB.
- The two hosts are on the same Ethernet network
- Show the ARP request and reply packets encapsulated in Ethernet frames

ARP & RARP

ARP– An Introduction

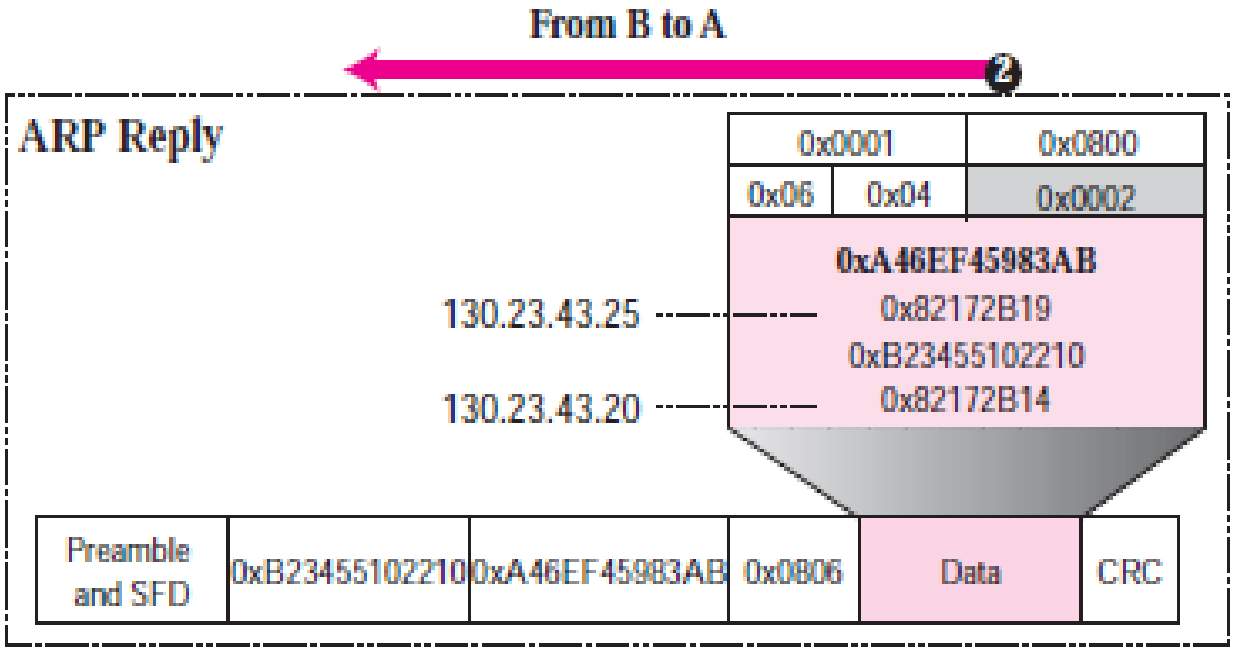
Example 1



ARP & RARP

ARP– An Introduction

Example 1



ARP & RARP

ARP– An Introduction

Proxy ARP

- Used to create a subnetting effect
- A router running a proxy ARP
  - Its ARP acts on behalf of a set of hosts
  - If it receives an ARP request message looking for the address of one of these host
  - Router sends an ARP reply announcing its own hardware (physical) address
  - After the router receives the actual IP packet, It sends the packet to the appropriate host or router

ARP & RARP

ARP– An Introduction

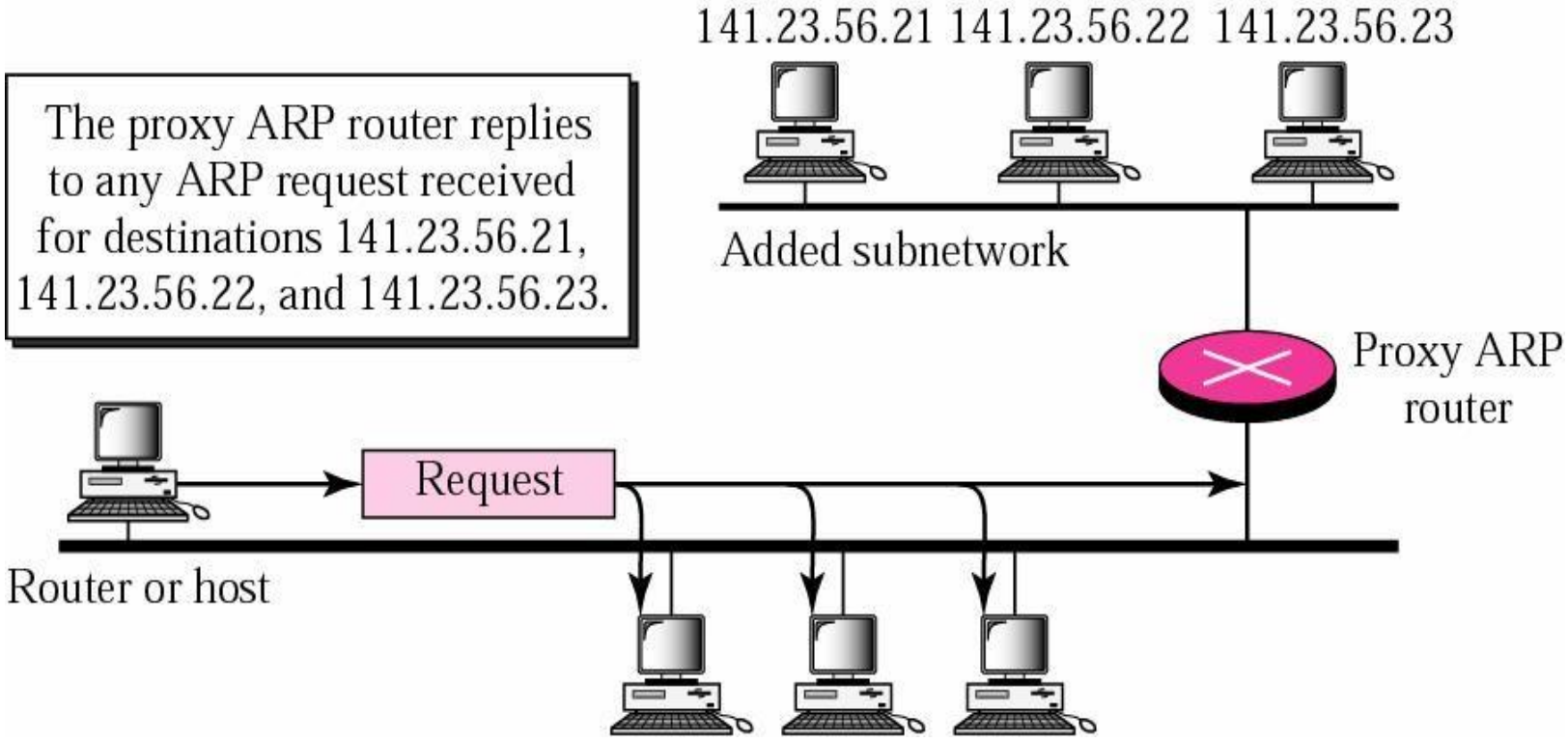
Example

- Administrator need to create a subnet without changing the whole system
- Add a router running a proxy ARP

ARP & RARP

ARP– An Introduction

Proxy ARP



ARP & RARP

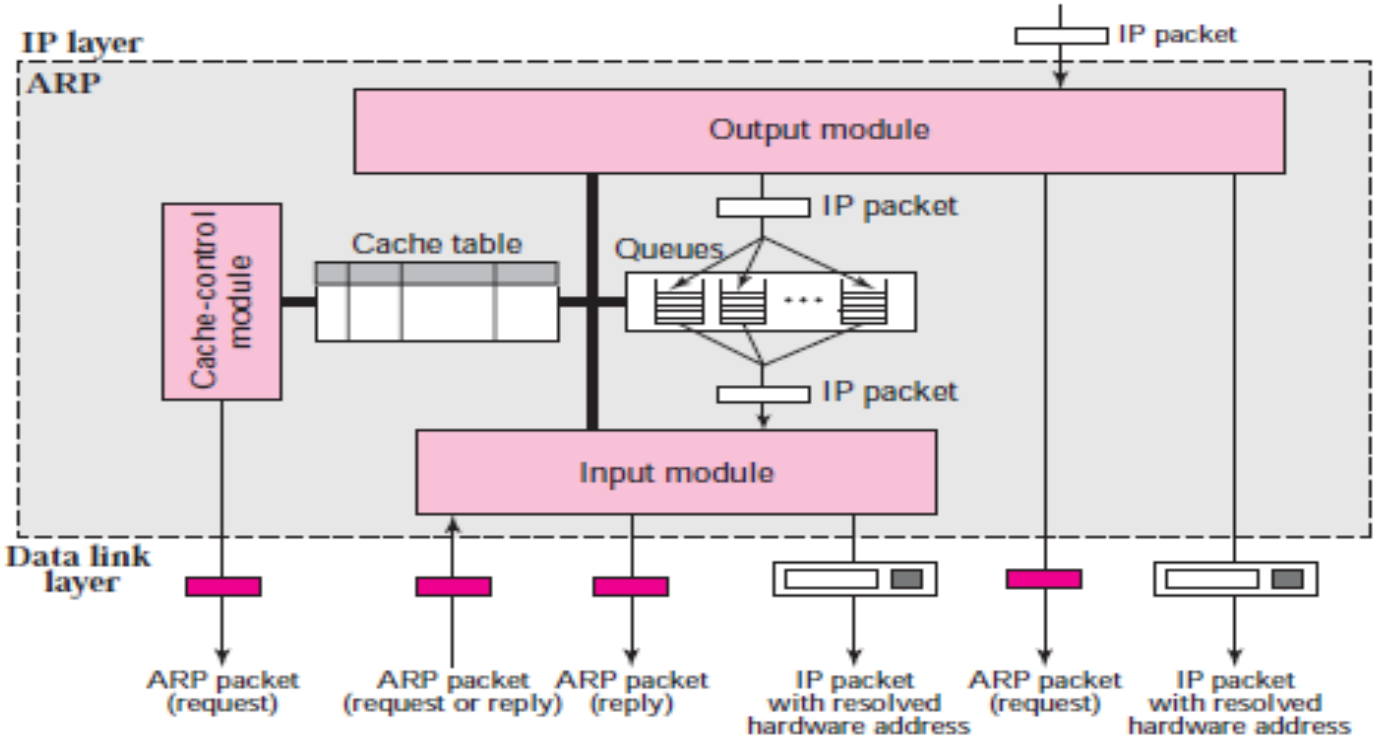
ARP–Package

- Five components in an ARP package
  - A cache table
  - Queues
  - An output module
  - An input module
  - A cache-control module

ARP & RARP

ARP–Package

ARP COMPONENTS



ARP & RARP

ARP–Package

- CACHE TABLE
- Inefficient to use ARP to each datagram destined for the same host or router
  - Introduce the cache table
- Cache table: an array of entries that contains the following's entries

ARP & RARP

ARP–Package

- Content of a Cache Table Entry State:
  - FREE: the lime-to-live for this entry has expired
  - PENDING: a request for this entry has been sent, but the reply has not yet been received
  - RESOLVED: the entry is complete and valid
- Hardware type
- Protocol type
- Hardware length
- Protocol length
  - Above fields are all the same as in the ARP packet

ARP & RARP

ARP–Package

- Content of a Cache Table Entry State:
  - Interface number
  - Queue number: ARP uses numbered queues to enqueuer the packet waiting for address resolution
  - Attempts: the number of times an ARP request is sent out for this entry
  - Time-out: the lifetime of an entry in seconds
  - Hardware address: the destination hardware address
  - Protocol address: the destination IP address

ARP & RARP

ARP–Package

- Content of a Cache Table Entry State:
- Interface number
- Queue number: ARP uses numbered queues to enqueuer the packet waiting for address resolution
- Attempts: the number of times an ARP request is sent out for this entry
- Time-out: the lifetime of an entry in seconds
- Hardware address: the destination hardware address
- Protocol address: the destination IP address

ARP & RARP

ARP–Package

QUEUS

- ARP package maintains a set of queues to hold the IP packets while ARP tries to resolve the hardware address
- Packets for the same destination are usually enqueued in the same queue
- The output module sends unsolved packets into the queue
- The input module removes a packet from the queue and sends it, with the resolved physical address, to data link layer for transmission

ARP & RARP

ARP–Package

Output Module

- Wait until an IP packet from the IP software
- Check the cache table if receiving a IP packet
  - If found and state = RESOLVED
    - Passed to the data link layer for transmission
  - If found and state = PENDING
    - Send packet to this queue and wait
  - If not found
    - Create an entry with state = PENDING
    - Create a queue and enqueue this packet
    - Send an ARP request

ARP & RARP

ARP–Package

Input Module

- Wait until an ARP packet (request or reply) arrives and check the cache table
  - If found state = PENDING
    - Copy the target hardware address in the packet
    - Change the state to RESOLVED
    - Set the value of TIME-OUT for this entry
    - Dequeue the packets from the corresponding queue and set them to the data link layer

ARP & RARP

ARP–Package

Input Module (Conti…)

- If found and state = RESOLVED
  - Copy the target hardware address in the packet
  - Set the value of TIME-OUT for this entry
  - This is because the target hardware address could have been changed
- If not found
  - Create a new entry and adds it to the table

- If the packet is a request
  - Send an ARP reply

ARP & RARP

ARP–Package

Cache Control Module

- Maintain the cache table by periodically check the cache table, entry by entry
- If state is PENDING
  - Increment the value of attempts by 1
  - If (attempts greater than maximum)
    - Change the state to FREE and Destroy the corresponding queue
  - Else
    - Send an ARP request

ARP & RARP

ARP–Package

Cache Control Module

- If state is RESOLVED
  - Decrement the value of time-out by the value of elapsed time
  - If (time-out <= 0)
    - Change the state to FREE
    - Destroy the corresponding queue
- If state is FREE
  - Continue to the next entry

ARP & RARP

Original Cache Table

State	*Queue*	*Attempt*	*Time-out*	*Protocol Addr.*	*Hardware Addr.*
R	5		900	180.3.6.1	ACAE32457342
P	2	2		129.34.4.8	
P	14	5		201.11.56.7	
R	8		450	114.5.7.89	457342ACAE32
P	12	1		220.55.5.7	
F					
R	9		60	19.1.7.82	4573E3242ACA
P	18	3		188.11.8.71	

ARP & RARP

ARP–Package

Example 2

- The ARP output module receives an IP datagram from the IP layer with the destination address 114.5.7.89
- It checks the cache table and finds that an entry exists for this destination with the RESOLVED state
- It extracts the hardware address, which is 457342ACAE32, and sends the packet and the address to the data link layer

ARP & RARP

ARP–Package

Example 3

- Twenty seconds later, the ARP output module receives an IP datagram from the IP layer with the destination address 116.1.7.22.
- It checks the cache table and does not find this destination in the table
- The module adds an entry to the table with the state PENDING and the Attempt value 1
- It also creates a new queue for this destination and enqueues the packet
- It then sends an ARP request to the data link layer for this destination

ARP & RARP

Cache table for Example 3

State	*Queue*	*Attempt*	*Time-out*	*Protocol Addr.*	*Hardware Addr.*
R	5		900	180.3.6.1	ACAE32457342
P	2	2		129.34.4.8	
P	14	5		201.11.56.7	
R	8		450	114.5.7.89	457342ACAE32
P	12	1		220.55.5.7	
P	**23**	**1**		**116.1.7.22**	
R	9		60	19.1.7.82	4573E3242ACA
P	18	3		188.11.8.71	

ARP & RARP

ARP–Package

Example 4

- Fifteen seconds later, the ARP input module receives an ARP packet with target protocol (IP) address 188.11.8.71
- The module checks the table and finds this address
- It changes the state of the entry to RESOLVED and sets the time-out value to 900
- The module then adds the target hardware address (E34573242ACA) to the entry
- Now it accesses queue 18 and sends all the packets in this queue, one by one, to the data link layer

ARP & RARP

Cache table for Example 4

State	*Queue*	*Attempt*	*Time-out*	*Protocol Addr.*	*Hardware Addr.*
R	5		900	180.3.6.1	ACAE32457342
P	2	2		129.34.4.8	
P	14	5		201.11.56.7	
R	8		450	114.5.7.89	457342ACAE32
P	12	1		220.55.5.7	
P	**23**	**1**		**116.1.7.22**	
R	9		60	19.1.7.82	4573E3242ACA
P	18	3		188.11.8.71	

ARP & RARP

ARP–Package

Example 5

- Twenty-five seconds later, the cache-control module waits up
- The time-out values for the first three resolved entries are decremented by 60
- The time-out value for the last resolved entry is decremented by 25
- The state of the next-to-the last entry is changed to FREE because the time-out is zero

ARP & RARP

ARP–Package

Example 5 (Conti…)

- For each of the three pending entries, the value of the attempts field is incremented by one
- Then, the attempts value for one entry (the one with IP protocol address 201.11.56.7) is more than the maximum
  - the state is changed to FREE, the queue is deleted
  - An ICMP message is sent to the original destination

ARP & RARP

Cache table for Example 5

State	*Queue*	*Attempt*	*Time-out*	*Protocol Addr.*	*Hardware Addr.*
R	5		840	180.3.6.1	ACAE32457342
P	2	3		129.34.4.8	
F					
R	8		390	114.5.7.89	457342ACAE32
P	12	2		220.55.5.7	
P	**23**	**2**		**116.1.7.22**	
F					
R	18		875	188.11.8.71	E34573242ACA

ARP & RARP

RARP–Package

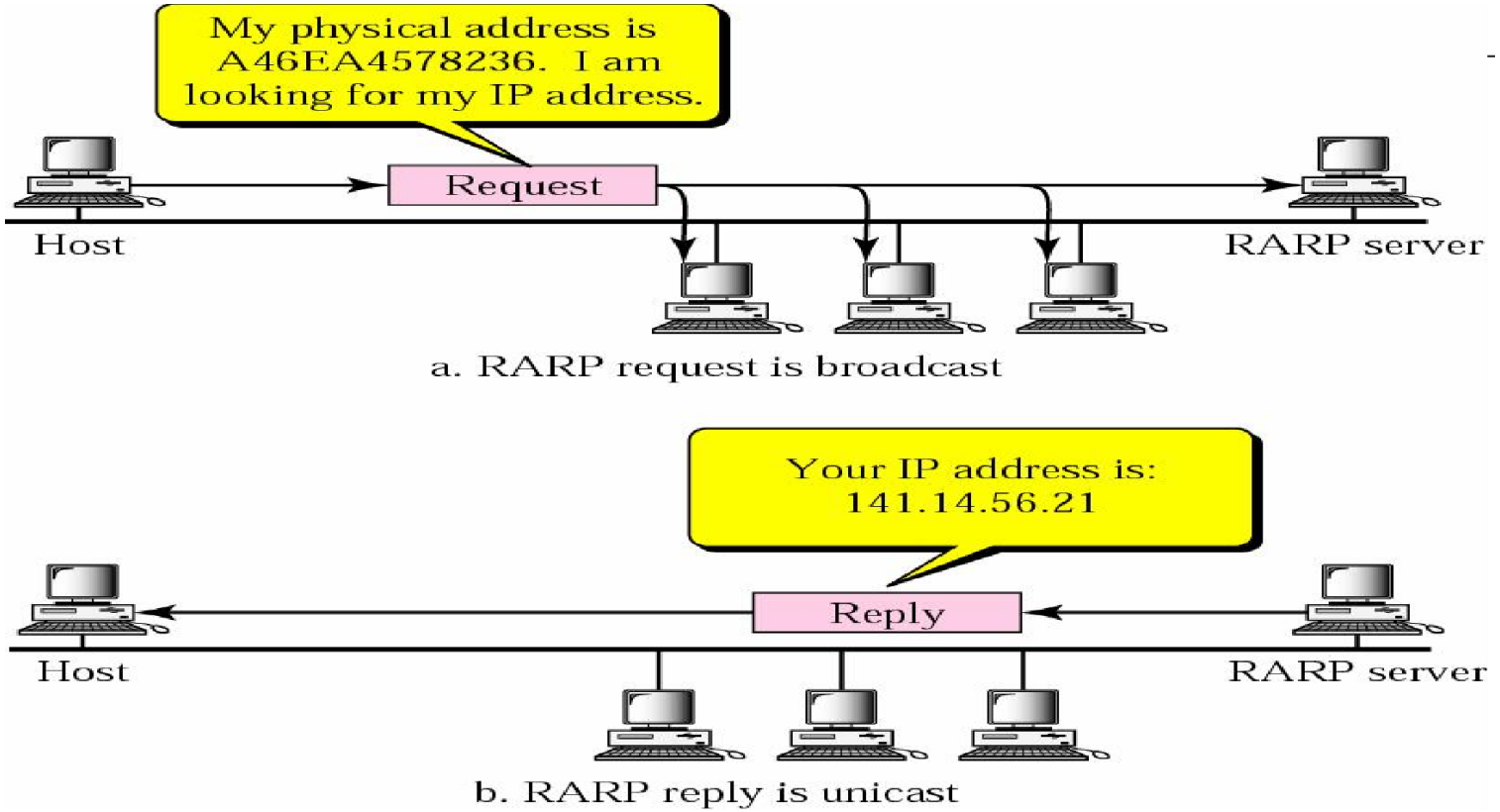
- A diskless machine is usually booted from ROM
- It cannot include the IP address
  - IP address are assigned by the network administrator
- Obtain its logical address by the physical address using the RARP protocol

ARP & RARP

RARP–Package

RARP

RARP Operation



ARP & RARP

RARP–Package

Note

The RARP request packets are broadcast; the RARP reply packets are unicast.

ARP & RARP

RARP–Package

Packet Format

- The format of the RARP packet is the same as the ARP packet
- Except that the operation field is
  - Three for RARP request message
  - Four for RARP reply message

ARP & RARP

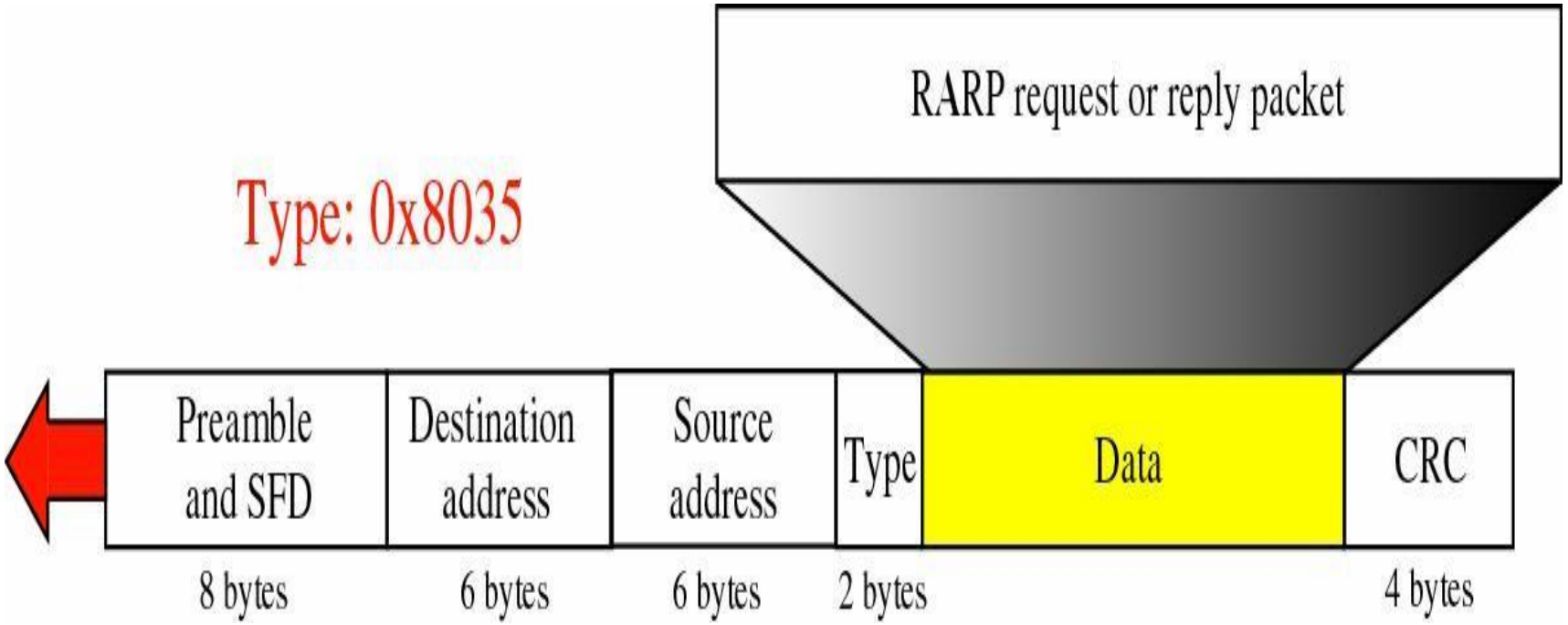
RARP–Package

RARP Packet

Hardware type		Protocol type
Hardware length	Protocol length	Operation Request 3, Reply 4
Sender hardware address (For example, 6 bytes for Ethernet)		
Sender protocol address (For example, 4 bytes for IP) (It is not filled for request)		
Target hardware address (For example, 6 bytes for Ethernet) (It is not filled for request)		
Target protocol address (For example, 4 bytes for IP) (It is not filled for request)		

ARP & RARP

RARP–Package



ARP & RARP

RARP

Alternative Solutions to RARP

- When a diskless computer is booted, it needs more information in addition to its IP address
  - The subnet mask
  - The IP address of a router
  - The IP address of a name server
- RARP cannot provide this extra information
- Two protocols, BOOTP and DHCP, can be used instead of RARP

Internet Control Message Protocol (ICMP)

Internet Control Message Protocol (ICMP)

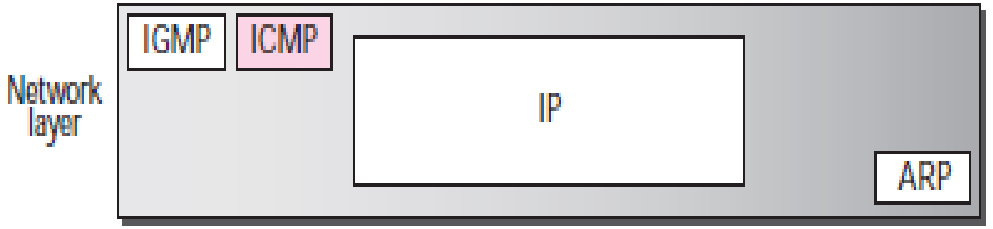
ICMP– An Introduction

- Why ICMP?
  - IP has no error – reporting or error correcting mechanism
- Scenarios in which the error occurs..
  - ✓ What happens if something goes wrong?
  - ✓ What happens if a router must discard a datagram but it cannot find a router to the final destination.
  - ✓ because the time-to-live field has a zero value
  - ✓ What happens if the final destination host must discard all fragments of a datagram due to time limit?

Internet Control Message Protocol (ICMP)

ICMP– An Introduction

- The above are some examples situations where an error has occurred and the IP has no built-in mechanism to notify the original host.
- It depends on Internet Control Message Protocol(ICMP) to provide an error control.
- ICMP –Internet Control Message Protocol
- It is a companion to the IP protocol

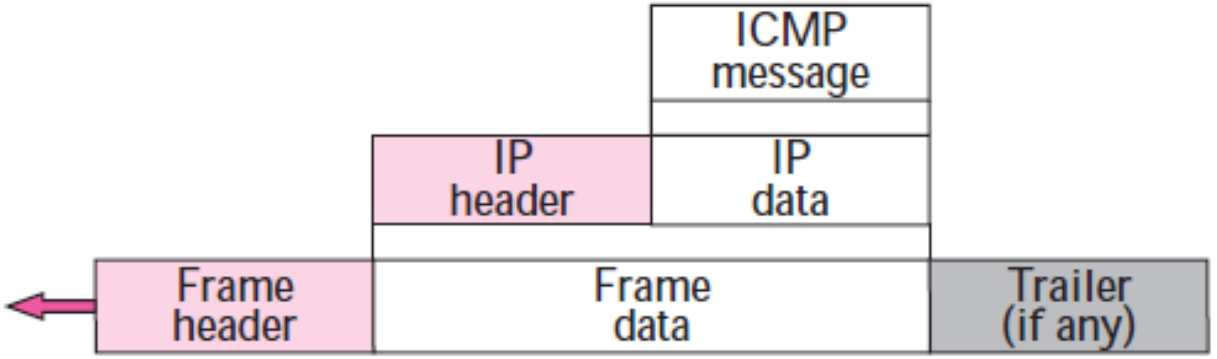


Position of ICMP in the network layer

Internet Control Message Protocol (ICMP)

ICMP– An Introduction

- ICMP -network layer protocol.
- messages are not passed directly to the data link layer.
- The messages encapsulated inside IP datagrams before going to the lower layer .
- The value of the protocol field in the IP datagram is 1 to indicate that the IP data is an ICMP message



ICMP Encapsulation

Internet Control Message Protocol (ICMP)

ICMP– Message

- ICMP message is of Two categories:
  - Error-reporting Messages
    - ✓ This report problems that a router or a host may encounter when it processes an IP packet.
  - The query messages
    - ✓ helps network manager get specific information from a router or another host.
- ✓ For example, nodes can discover their neighbors.

Internet Control Message Protocol (ICMP)

ICMP– Message

Category	*Type*	*Message*
Error-reporting messages	3	Destination unreachable
	4	Source quench
	11	Time exceeded
	12	Parameter problem
	5	Redirection
Query messages	8 or 0	Echo request or reply
	13 or 14	Timestamp request or reply

ICMP Messages

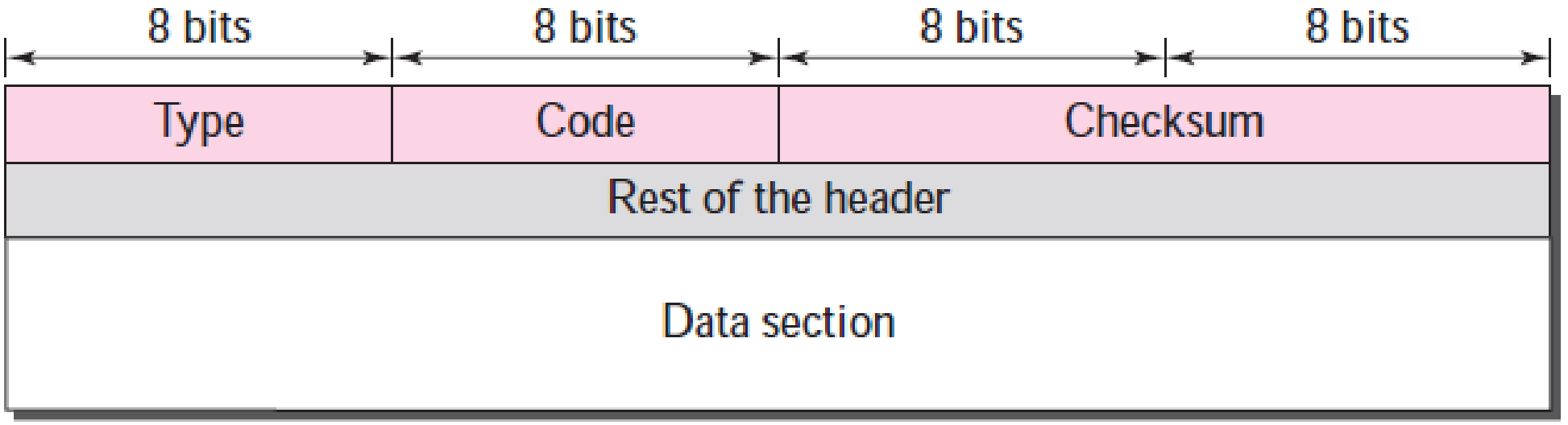
Internet Control Message Protocol (ICMP)

ICMP– Message format

- 8-byte header
- variable-size data section.
- The first field, ICMP type, defines the type of the message.
- The code specifies the reason for the particular message type.
- The checksum field .
- The rest of the header is specific for each message type.
- The data section in error messages carries information for finding the original packet that had the error

Internet Control Message Protocol (ICMP)

ICMP– Message format



General format of ICMP MESSAGE

Internet Control Message Protocol (ICMP)

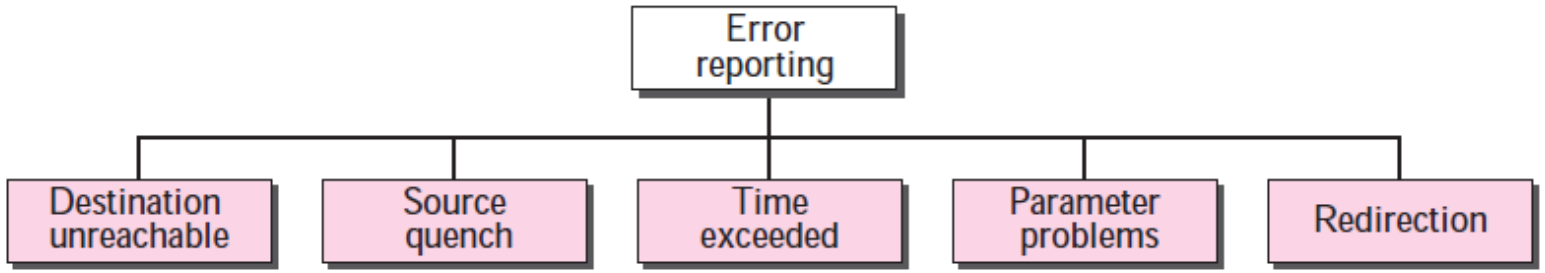
ICMP– MESSAGE FORMAT

Error Reporting Messages

- One of the main responsibilities of ICMP is to report errors.
- IP is an unreliable protocol, error checking and error control are not a concern of IP.
- **ICMP always reports error messages to the original source.**
- Error correction is left to the higher-level protocols.
- Error messages are always sent to the original source

Internet Control Message Protocol (ICMP)

ICMP– MESSAGE FORMAT



Error-reporting messages

Important ICMP error messages:

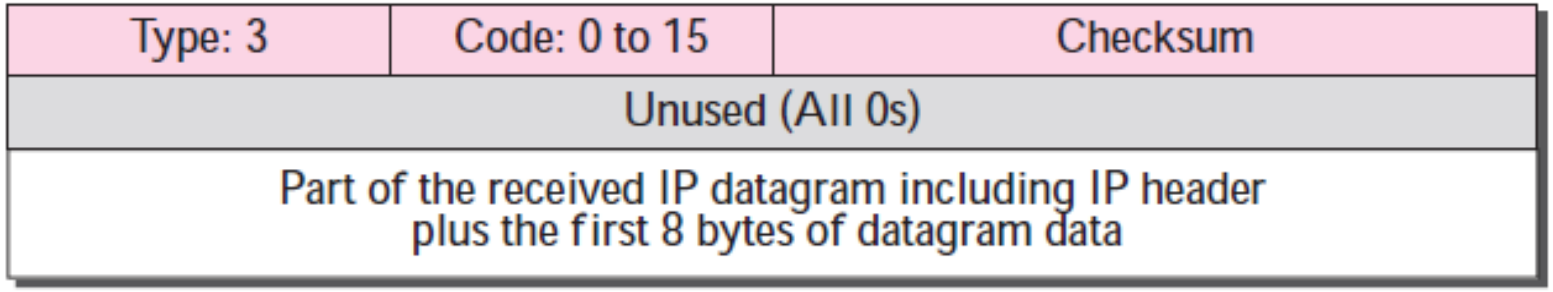
- No ICMP error message will be generated in response to a datagram carrying an ICMP error message.
- No ICMP error message will be generated for a fragmented datagram that is not the first fragment.
- No ICMP error message will be generated for a datagram having a multicast address.
- No ICMP error message will be generated for a datagram having a special address such as 127.0.0.0 or 0.0.0.0.

Internet Control Message Protocol (ICMP)

ICMP– MESSAGE FORMAT

Destination Unreachable

- A router cannot route a datagram or a host cannot deliver a datagram then the datagram is discarded
- The router or the host sends a **destination-unreachable message** back to the source host

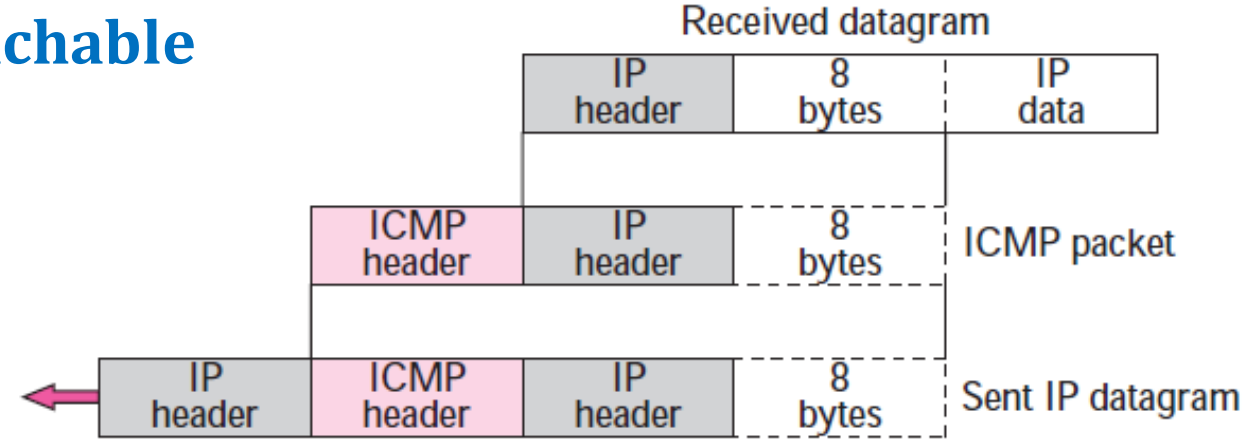


Destination-unreachable format

Internet Control Message Protocol (ICMP)

ICMP– MESSAGE FORMAT

Destination Unreachable



Contents of data field for the error messages

- The code field for this type specifies the reason for discarding the datagram:
  - **Code 0.** The network is unreachable, possibly due to hardware failure
  - **Code 1.** The host is unreachable. This can also be due to hardware failure

Internet Control Message Protocol (ICMP)

ICMP– MESSAGE FORMAT

Destination Unreachable

Code 2. The protocol is unreachable. An IP datagram can carry data belonging to higher-level protocols such as UDP, TCP, and OSPF. If the destination host receives a datagram that must be delivered, for example, to the TCP protocol, but the TCP protocol is not running at the moment, a code 2 message is sent.

Code 3. The port is unreachable. The application program (process) that the datagram is destined for is not running at the moment.

Internet Control Message Protocol (ICMP)

ICMP– MESSAGE FORMAT

Destination Unreachable

Code 4. Fragmentation is required, but the DF (do not fragment) field of the datagram has been set. In other words, the sender of the datagram has specified that the datagram not be fragmented, but routing is impossible without fragmentation.

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

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

Internet Control Message Protocol (ICMP)

ICMP– MESSAGE FORMAT

Destination Unreachable

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

Code 8. The source host is isolated.

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

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

Internet Control Message Protocol (ICMP)

ICMP– MESSAGE FORMAT

Destination Unreachable

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

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

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

Internet Control Message Protocol (ICMP)

ICMP– MESSAGE FORMAT

Destination Unreachable

Code 14. The host is unreachable because the host precedence is violated. The message is sent by a router to indicate that the requested precedence is not permitted for the destination.

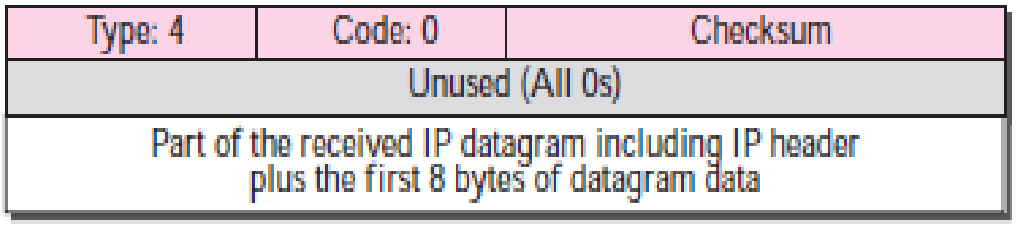
Code 15. The host is unreachable because its precedence was cut off. This message is generated when the network operators have imposed a minimum level of precedence for the operation of the network, but the datagram was sent with a precedence below this level.

Internet Control Message Protocol (ICMP)

ICMP– MESSAGE FORMAT

Source Quench

- There is no flow-control or congestion-control mechanism in the IP protocol.
- A source-quench message informs the source that a datagram has been discarded due to congestion in a router or the destination host.
- The source must slow down the sending of datagrams until the congestion is relieved.



Source Quench Format

Internet Control Message Protocol (ICMP)

ICMP– MESSAGE FORMAT

Time Exceeded

- The time-exceeded message is generated in two forms:

1. Whenever a router decrements a datagram with a time-to-live value to zero, it discards the datagram and sends a time-exceeded message to the original source.

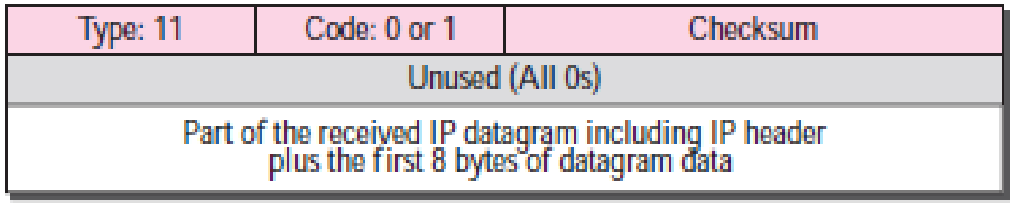
2. When the final destination does not receive all of the fragments in a set time, it discards the received fragments and sends a time-exceeded message to the original Source

Internet Control Message Protocol (ICMP)

ICMP– MESSAGE FORMAT

Time Exceeded

- In a time-exceeded message, code 0 is used only by routers to show that the value of the time-to-live field is zero. Code 1 is used only by the destination host to show that not all of the fragments have arrived within a set time.



Time Exceeded format

Internet Control Message Protocol (ICMP)

ICMP– MESSAGE FORMAT

Type: 12	Code: 0 or 1	Checksum
Pointer	Unused (All 0s)	
Part of the received IP datagram including IP header plus the first 8 bytes of datagram data		

Parameter Problem

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

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

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

Internet Control Message Protocol (ICMP)

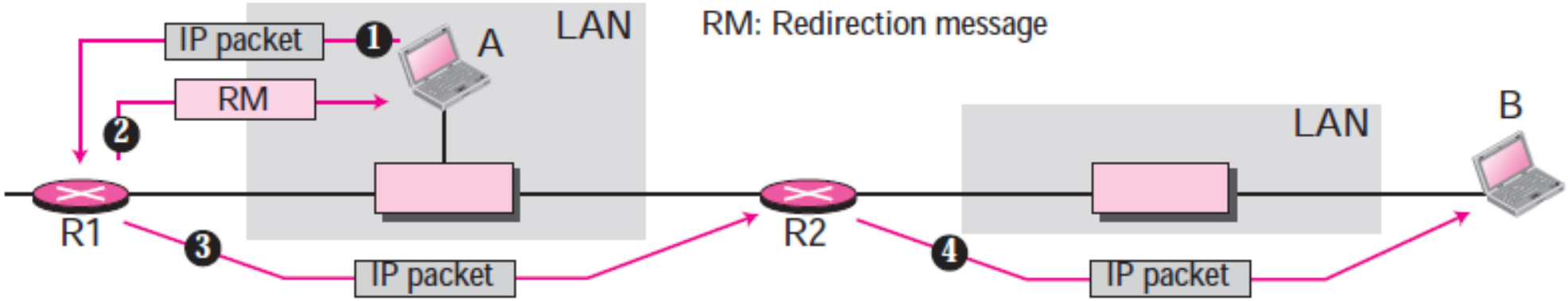
ICMP– MESSAGE FORMAT

A host usually starts with a small routing table that is gradually augmented and updated. One of the tools to accomplish this is the redirection message..

- Code 0. Redirection for a network-specific route.
- Code 1. Redirection for a host-specific route.
- Code 2. Redirection for a network-specific route based on a specified type of service.
- Code 3. Redirection for a host-specific route based on a specified type of service
- An echo-request message can be sent by a host or router. An echo-reply message is sent by the host or router that receives an echo-request message.

Internet Control Message Protocol (ICMP)

ICMP– MESSAGE FORMAT



Type: 5	Code: 0 to 3	Checksum
IP address of the target router		
Part of the received IP datagram including IP header plus the first 8 bytes of datagram data		

Internet Control Message Protocol (ICMP)

Echo Request And Reply

- An echo-request message can be sent by a host or router. An echo-reply message is sent by the host or router that receives an echo-request message.
- Echo-request and echo-reply messages can be used by network managers to check the operation of the IP protocol.
- Echo-request and echo-reply messages can test the reachability of a host. This is usually done by invoking the ping command.

Internet Control Message Protocol (ICMP)

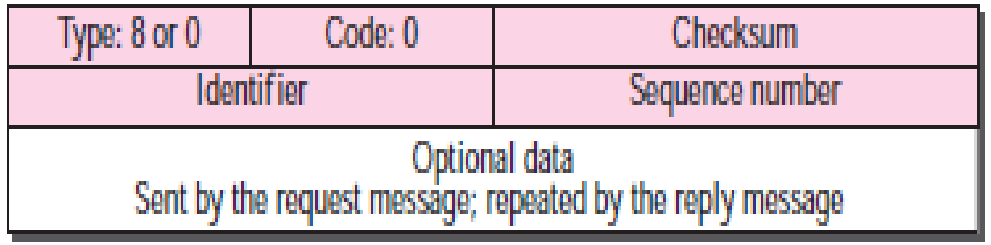
Timestamp Request and Reply

- The **timestamp-request** and **timestamp-reply messages** to determine the round-trip time needed for an IP datagram to travel between them

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

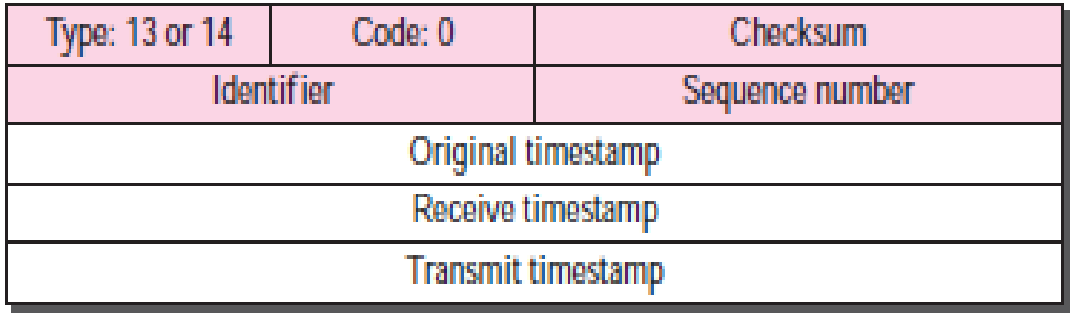
- Timestamp-request and timestamp-reply messages can be used to calculate the round-trip time between a source and a destination machine even if their clocks are not synchronized.
- The timestamp-request and timestamp-reply messages can be used to synchronize two clocks in two machines if the exact one-way time duration is known.

Type 8: Echo request
Type 0: Echo reply



Echo-request and echo-reply messages

Type 13: request
Type 14: reply



Timestamp-request and timestamp-reply message format

Internet Control Message Protocol (ICMP)

Timestamp Request and Reply

- original timestamp: 46 receive timestamp: 59
- transmit timestamp: 60 return time: 67
- sending time = 59 – 46 = 13 milliseconds
- receiving time = 67 – 60 = 7 milliseconds
- round-trip time = 13 + 7 = 20 milliseconds
- Time difference = receive timestamp – (original timestamp field + one-way time duration)
- Time difference = 59 – (46 + 10) = 3

Internet Control Message Protocol (ICMP)

ICMP– MESSAGE FORMAT

Checksum Calculation

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

1. The checksum field is set to zero.

2. The sum of all the 16-bit words (header and data) is calculated.

3. The sum is complemented to get the checksum.

4. The checksum is stored in the checksum field.

Internet Control Message Protocol (ICMP)

ICMP– MESSAGE FORMAT

Checksum Testing

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

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

2. The sum is complemented.

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

Internet Control Message Protocol (ICMP)

ICMP– DEBUGGING TOOL

- To check whether host or router is alive and running
- To trace the route of a packet.
- Two tools that use ICMP for debugging: *ping* and *traceroute*

Ping

- The **ping** program to find if a host is alive and responding.
- ***Command : ping the ip of the host***.(ping 152.18.1.3)
- The source host sends ICMP echo request messages (type: 8, code: 0);
- The destination, if alive, responds with ICMP echo reply messages.

Internet Control Message Protocol (ICMP)

ICMP– DEBUGGING TOOL

Ping cont…

- Starts the sequence number from 0; this number is incremented by one each time a new message is sent.
- *ping* can calculate the round-trip time.
- Inserts the sending time in the data section of the message.
- When packet arrives it subtracts the arrival time from the departure time to get the **Round-Trip Time** (RTT).
- The TTL (time to live) field is encapsulates an ICMP message as 62, which means the packet cannot travel more than 62 hops

Internet Control Message Protocol (ICMP)

ICMP– DEBUGGING TOOL

Example : $ ping fhda.edu

```
PING fhda.edu (153.18.8.1) 56 (84) bytes of data.
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=0 ttl=62 time=1.91 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=1 ttl=62 time=2.04 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=2 ttl=62 time=1.90 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=3 ttl=62 time=1.97 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=4 ttl=62 time=1.93 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=5 ttl=62 time=2.00 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=6 ttl=62 time=1.94 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=7 ttl=62 time=1.94 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=8 ttl=62 time=1.97 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=9 ttl=62 time=1.89 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=10 ttl=62 time=1.98 ms
--- fhda.edu ping statistics ---
11 packets transmitted, 11 received, 0% packet loss, time 10103 ms
rtt min/avg/max = 1.899/1.955/2.041  ms
```

Internet Control Message Protocol (ICMP)

ICMP– DEBUGGING TOOL

Ping cont…

- Ping data bytes as 56 and the total number of bytes as 84.
- 8 bytes ICMP header+ 20 bytes of IP header to 56 =84
- *ping* defines the number of bytes as 64(56 + 8).
- Interrupts message ctrl+c.

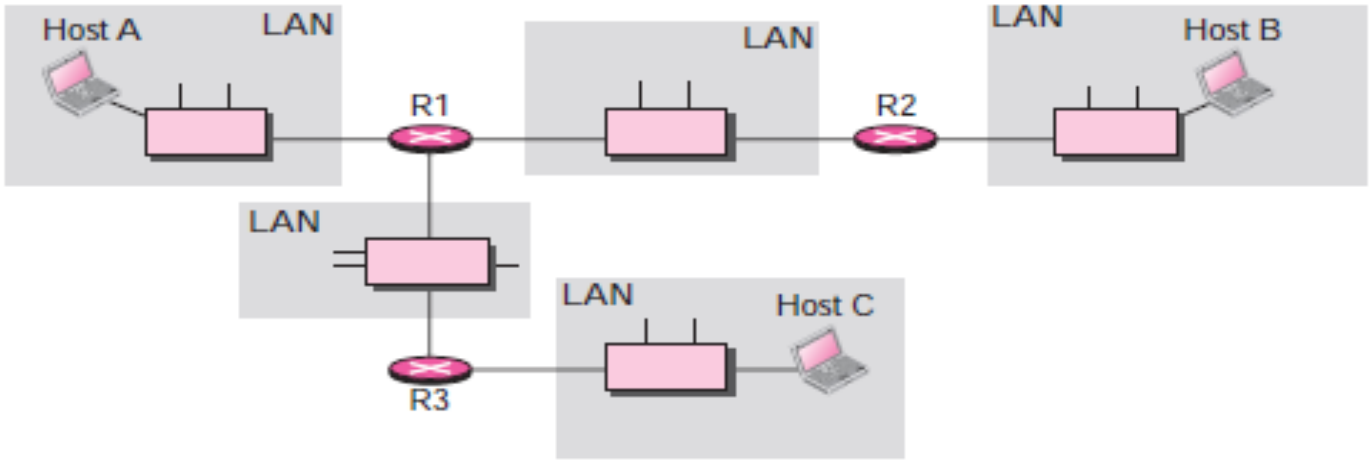
- it prints the statistics of the probes
  - ✓ number of packets sent,
  - ✓ the number of packets received.
  - ✓ the total time
  - ✓ the RTT minimum, maximum, and average.

Internet Control Message Protocol (ICMP)

ICMP– DEBUGGING TOOL

TRACE ROUTE

- The ***traceroute*** program in UNIX or ***tracert*** in Windows.
- It is used to route the packets from source to destination.
- Example Scenario



The Traceroute Program Operation

Internet Control Message Protocol (ICMP)

ICMP– DEBUGGING TOOL

Trace route cont....

- *In above example* Given the topology, A packet from host A to host B travels through routers R1 and R2.
- The traceroute program find the address of router R1 & RTT between host A and router R1.
- The program repeats steps a to c three times to get a better average round-trip time.

  a. Host A sends a packet to destination B using UDP the message is encapsulated in an IP packet with a TTL value of 1. The program notes the time the packet is sent

Internet Control Message Protocol (ICMP)

ICMP– DEBUGGING TOOL

Trace route cont.....

b. Router R1 receives the packet and decrements the value of TTL to 0. It then discards the packet (because TTL is 0).

c. In receiver the ICMP messages uses the source address of the IP packet to find the IP address of router R1 and also makes note of the time the packet has arrived.

- The traceroute program repeats the previous steps to find the address of router R2 and the round-trip time between host A and router R2.
- The round-trip time between host A and host B.

Internet Control Message Protocol (ICMP)

ICMP– DEBUGGING TOOL

Trace route cont.....

Example: The traceroute program to find the route from the computer voyager.deanza.edu to the server fhda.edu. The following shows the result.

```
$ traceroute fhda.edu
traceroute to fhda.edu (153.18.8.1), 30 hops max, 38 byte packets
1 Dcore.fhda.edu (153.18.31.25) 0.995 ms 0.899 ms 0.878 ms
2 Dbackup.fhda.edu (153.18.251.4) 1.039 ms 1.064 ms 1.083 ms
3 tiptoe.fhda.edu (153.18.8.1) 1.797 ms 1.642 ms 1.757 ms
```

- In the above example the destination is 153.18.8.1.
- TTL value is 30 hops.
- The packet contains 38 bytes: 20 bytes of IP header, 8 bytes of UDP header, and 10 bytes of application data.

Internet Control Message Protocol (ICMP)

ICMP– DEBUGGING TOOL

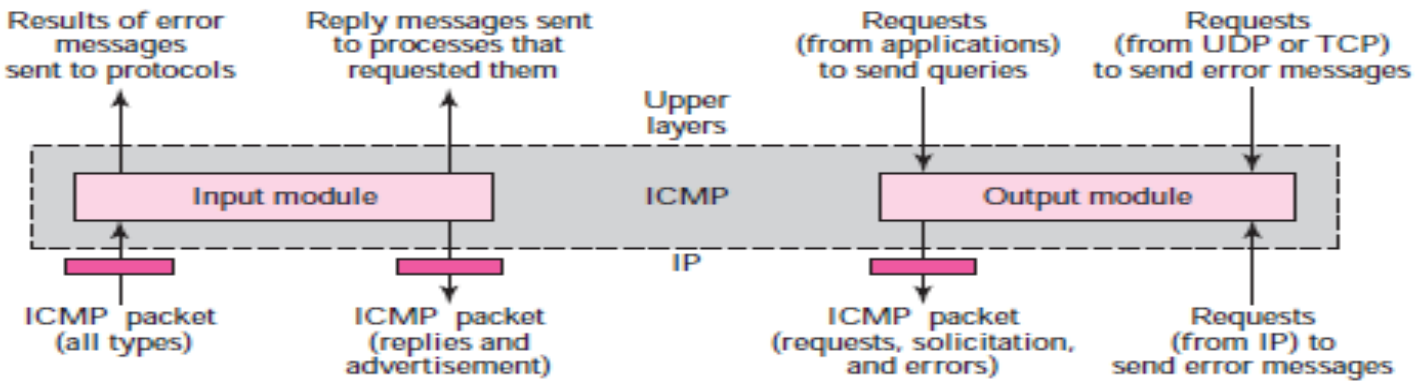
Trace route cont.....

- The router is named Dcore.fhda.edu with IP address 153.18.31.254.
- The Round Trip Time
    - 1. 0.995 milliseconds,
    - 2. 0.899 milliseconds
    - 3. 0.878 milliseconds
- The router is named Dbackup.fhda.edu with IP address 153.18.251.4.
- The third line shows the destination host.

Internet Control Message Protocol (ICMP)

ICMP– Package

- To handle the ICMP sending and receiving messages
- ICMP package made of two modules:
  - input module
  - output module



ICMP package

Internet Control Message Protocol (ICMP)

ICMP– Package

Package cont..

Input Module

- Handles all received ICMP messages.
- Invoked when an ICMP packet is delivered from the IP layer.
- If the received packet is a
  - ✓ request - the module creates a reply and sends it out.
  - ✓ redirection message - Uses the information to update the routing table.
  - ✓ error message - It informs the protocol about the situation that caused the error.

Input Module Pseudo code

```
ICMP_Input_module (ICMP_Packet)
 {
      If (the type is a request)
      {
            Create a reply
            Send the reply
      }
      If (the type defines a redirection)
      {
      Modify the routing table
}
 If (the type defines other error messages)
      {
      Inform the appropriate source protocol
       }
 Return
}
```

Internet Control Message Protocol (ICMP)

ICMP– Package

Package cont..

Output Module

- Responsible for creating request, solicitation, or error messages requested by a higher level or the IP protocol.
- receives a demand from IP,UDP, or TCP to send one of the ICMP error messages.
- IP request is first allowed
- An ICMP message cannot be created for four situations:
  - ✓ an IP packet carrying an ICMP error message.
  - ✓ a fragmented IP packet.
  - ✓ A multicast IP packet.
  - ✓ an IP packet having IP address 0.0.0.0 or 127.X.Y. Z.

Output Module Pseudo code

```
ICMP_Output_Module (demand)
{
    If (the demand defines an error message)
    {
        If (demand comes from IP AND is forbidden)
        {
            Return
        }
    If (demand is a valid redirection message)
    {
        Return
    }
Create an error message
If (demand defines a request)
    {
        Create a request message
    }
Send the message
 Return
}
```


UDP
User Datagram
Protocol

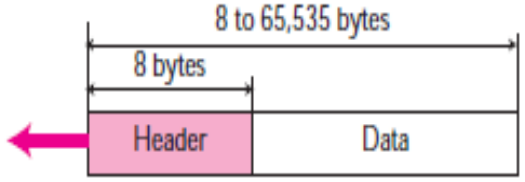
User Datagram Protocol (UDP)

UDP – An Introduction

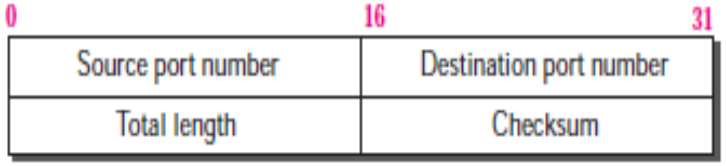
- Connectionless service
- Unreliable transport protocol.
- No flow control / No Acknowledgement
- Process to Process communication
- Powerless
- It uses minimum of over heads
- No reliability is obtained using UDP
- Less interaction between sender and receiver

User Datagram Protocol (UDP)

USER DATAGRAM



a. UDP user datagram



b. Header format

User datagram hearde format

- Above picture depicts user data gram format
- UDP packets are called user datagrams(messages)
- It has fixed size header of 8 bytes

User Datagram Protocol (UDP)

UDP Datagram various fields

- Source Port Number:
  - 16 bits long – port ranges from 0-65535.
  - This port number will be used by the source host for identification.
- Destination Port Number:
  - 16 bits long.
  - Used by the process running on the Destination machine.
  - Application level service on end machine

User Datagram Protocol (UDP)

UDP Datagram various fields

- Length:

UDP length=IP length – IP headers Length

  - Length field specifies the entire length of UDP packet (including header).
  - It is 16-bits field and minimum value is 8-byte, i.e. the size of UDP header itself.
  - A user datagram is encapsulated in an IP datagram.

- Checksum:

  - This field is used to detect errors over the entire user datagram (header plus data)

User Datagram Protocol (UDP)

UDP Services

- Process to Process communication
- UDP provides process-to-process communication using sockets, a combination of IP addresses and port numbers. Several port numbers used by UDP.

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

User Datagram Protocol (UDP)

UDP Services cont…

- Connectionless services:
  - Connectionless service.
  - Each datagram is independent even it comes from same source and delivered in same destination.
  - The datagrams are not numbered.
  - No connection establishments is done
  - cannot send a stream of data to UDP, It will chop them into differentrelated user datagrams.
- Flow control:
  - No Flow Control and no window mechanism.
  - The receiver may overflow with incoming messages.
  - UDP should provide for this service, if needed

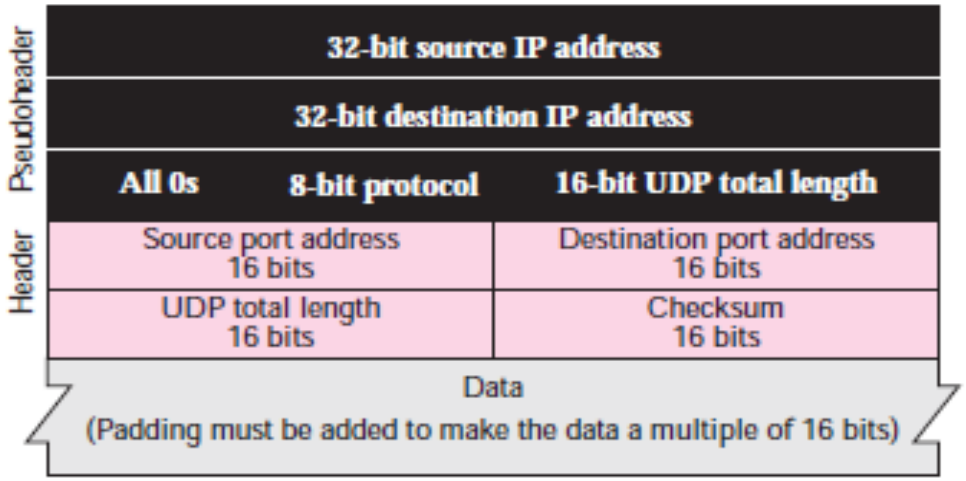
User Datagram Protocol (UDP)

UDP Services cont…

- Error Control:
  - No Error control mechanism except for checksum
  - Sender is unknown about the loss and duplication.
  - When error detects in receiver the datagram is discarded

Checksum

- Three sections: a pseudo header, the UDP header, and the data coming from the application layer.



Pseudoheader for checksum calculation

User Datagram Protocol (UDP)

UDP Services cont...

153.18.8.105		
171.2.14.10		
All 0s	17	15
1087		13
15		All 0s
T	E	S	T
I	N	G	Pad

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

Check Sum Calculation

User Datagram Protocol (UDP)

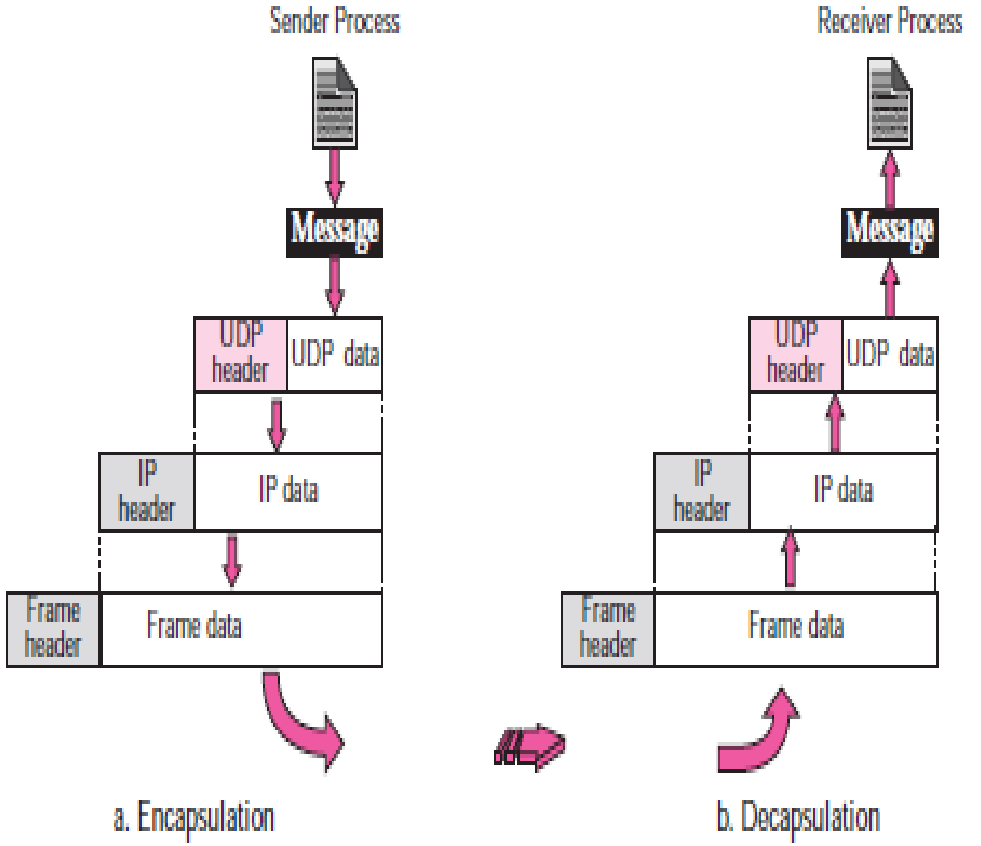
UDP Services cont...

- Congestion Control
  - It is a connectionless protocol so congestion control not provided.
- Encapsulation and Decapsulation
  - Encapsulation
    - A process send message through UDP
    - Along pair of socket address and the length of data.
    - UDP receives data and add UDP header then pass to IP with socket.
    - IP add its own header using value 17 in protocol filed.
    - Indicating UDP protocol.
    - The IP datagram is then passed to the data link layer.
    - The data link layer receives the IP datagram and passes it to the physical layer.
    - The physical layer encodes the bits into electrical or optical signals and sends it to the remote machine.

User Datagram Protocol (UDP)

UDP Services cont...

- ***Decapsulation:***
  - At the destination host:
    - the physical layer decodes the signals pass to data link layer.
    - The data link layer uses the header (and the trailer) to check the data.
  - If there is no error
    - the header and trailer are dropped , datagram is passed to IP.



Encapsulation and Decapsulation

User Datagram Protocol (UDP)

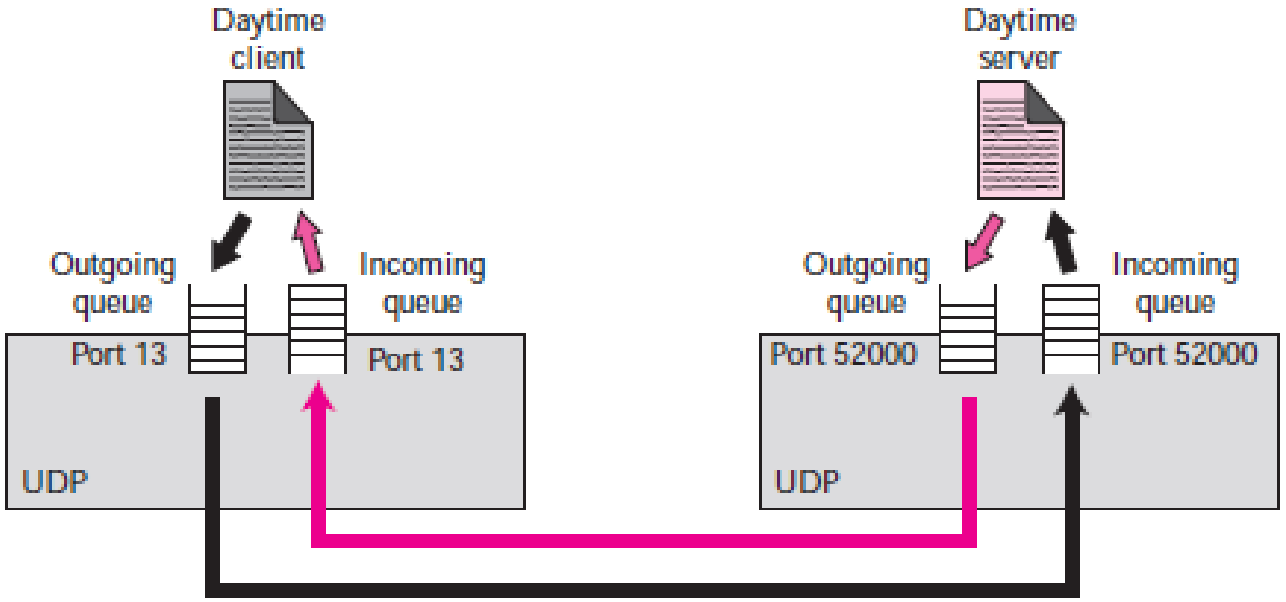
UDP Services cont...

- The header is dropped and the user datagram is passed to UDP with the sender and receiver IP addresses.
- Checksum is to check the entire user datagram.
- the header is dropped and the application data along with the sender socket address is passed to the process.
- The sender socket address is passed to the process in case it needs to respond to the message received.

User Datagram Protocol (UDP)

UDP Services cont...

Queuing in UDP



User Datagram Protocol (UDP)

UDP Services cont...

Queuing in UDP

- At the client site, when a process starts, it requests a port number from the operating system. Some implementations create both an incoming and an outgoing queue associated with each process. Other implementations create only an incoming queue associated with each process
- process wants to communicate with multiple processes, it obtains only one port number and eventually one outgoing and one incoming queue. The queues opened by the client are, in most cases, identified by ephemeral port numbers. The queues function as long as the process is running. When the process terminates, the queues are destroyed.
- The client process can send messages to the outgoing queue by using the source port number specified in the request

User Datagram Protocol (UDP)

UDP Services cont...

Queuing in UDP

- The client process can send messages to the outgoing queue by using the source port number specified in the request
- UDP removes the messages one by one and, after adding the UDP header, delivers them to IP. An outgoing queue can overflow
- It happens the operating system can ask the client process to wait before sending any more messages

User Datagram Protocol (UDP)

UDP Services cont...

Applications of UDP:

Used for simple request response communication when size of data is less hence there is lesser concern about flow and error control.

It is suitable protocol for multicasting as UDP supports packet switching.

Following implementations uses UDP as a transport layer protocol:

- NTP (Network Time Protocol)
- DNS (Domain Name Service)
- BOOTP, DHCP.
- NNP (Network News Protocol)
- Quote of the day protocol
- TFTP, RTSP, RIP, OSPF.

UDP is null protocol if you remove checksum field.

User Datagram Protocol (UDP)

UDP Services cont...

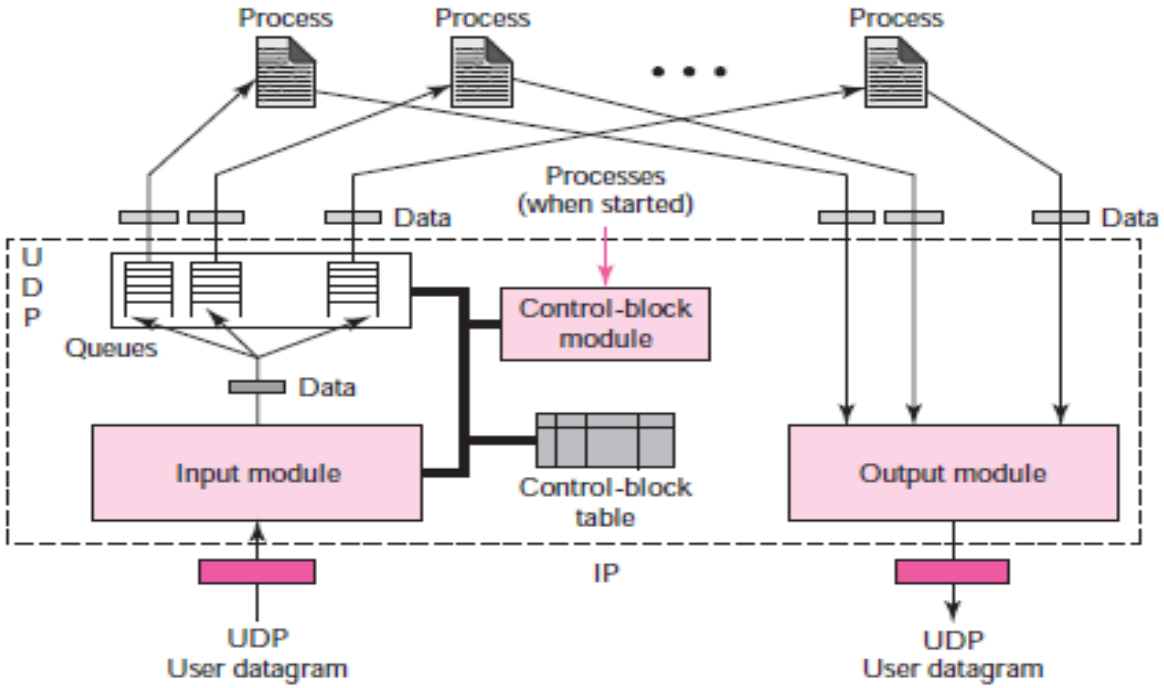
When to use UDP? Reduce the requirement of computer resources.

When using the Multicast or Broadcast to transfer.

The transmission of Real-time packets, mainly in multimedia applications

User Datagram Protocol (UDP)

UDP Package



UDP design

User Datagram Protocol (UDP)

UDP Package

- The UDP package involves five components:
- **Control-Block Table**
  - UDP has a control-block table to keep track of the open ports.
  - Each entry in this table has a minimum of four fields: the state, which can be FREE or IN-USE, the process ID, the port number, and the corresponding queue number.
- **Input Queues**
  - Our UDP package uses a set of input queues, one for each process. In this design, we do not use output queues.

User Datagram Protocol (UDP)

UDP Package

- **Control-Block Module**
  - The control-block module is responsible for the management of the control-block table.
  - When a process starts, it asks for a port number from the operating system.
  - The operating system assigns well-known port numbers to servers and ephemeral port numbers to clients.
  - The process passes the process ID and the port number to the control-block module to create an entry in the table for the process.

```
UDP_Control_Block_Module (process ID, port number)
{
Search the table for a FREE entry.
if (not found)
Delete one entry using a predefined strategy.
Create a new entry with the state IN-USE
Enter the process ID and the port number.
 Return.
} // End module
```

User Datagram Protocol (UDP)

UDP Package

- **Input Module**
- The input modulereceives a user datagram from the IP. It searches the control-block table to find an entry having the same port number as this user datagram.
- If the entry is found, the module uses the information in the entry to enqueue the data.If the entry is not found, it generates an ICMP message..

```
UDP_INPUT_Module (user_datagram)
{
Look for the entry in the control_block table
if (found)
{
Check to see if a queue is allocated
If (queue is not allocated)
 allocate a queue
else
enqueue the data
} //end if
else
{
Ask ICMP to send an "unreachable port" message
Discard the user datagram
} //end else
Return.
} // end module
```

User Datagram Protocol (UDP)

UDP Package

Output Module

- The output module is responsible for creating and sending user datagrams.

```
UDP_OUTPUT_MODULE (Data)
{
    Create a user datagram
    Send the user datagram
    Return.
}
```

TCP Header

Transmission Control Protocol (TCP)

TCP Segment

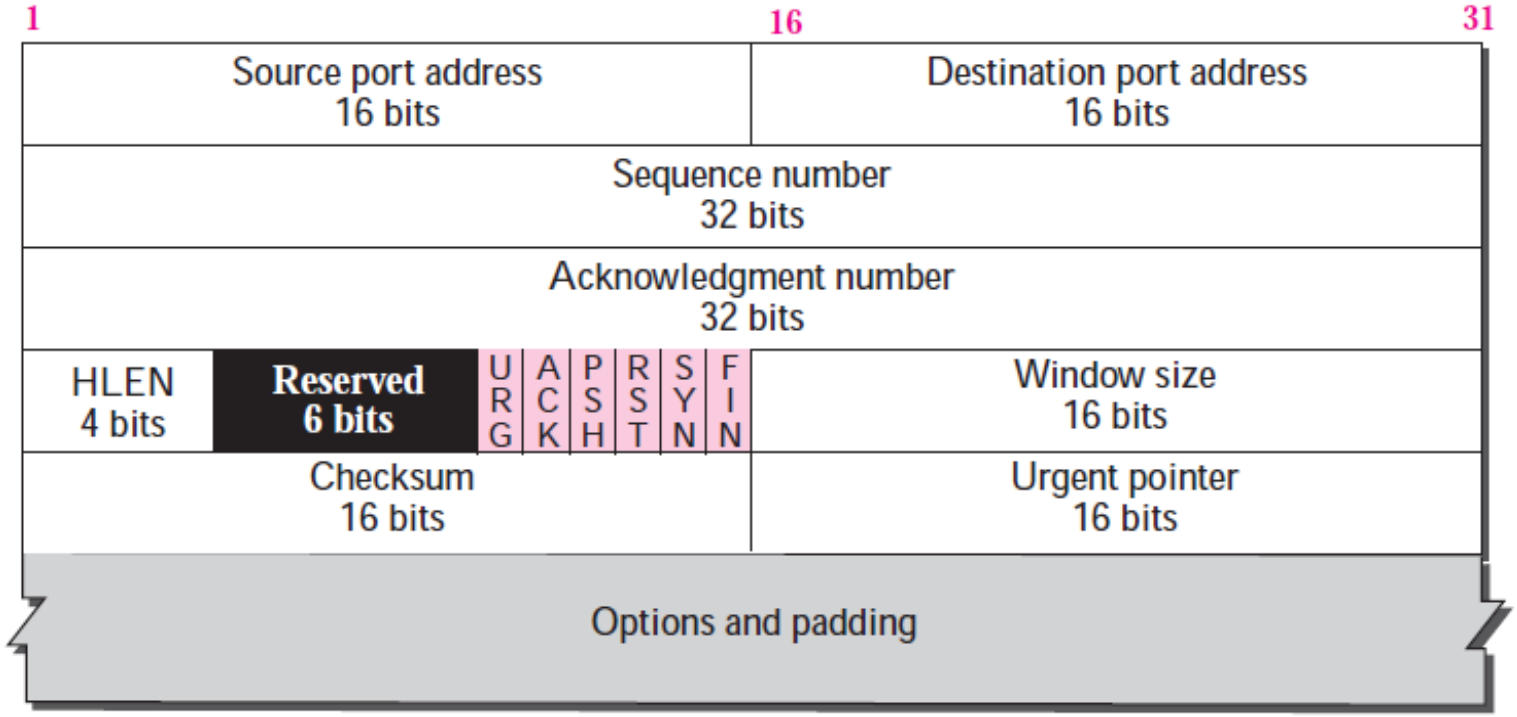
- A packet in TCP is called a ***Segment***
- ***Segment – Format***
  - Header - 20 to 60 Bytes
    - In case of no options Header is of size 20 bytes
    - When options are used Header size extends up to 60 Bytes
  - Data from application program

Format of Segment

Transmission Control Protocol (TCP)

TCP Header

- Size of TCP Header varies from 20 to 60 bytes



Format of TCP Header

Source: Behrouz A. Forouzan, "TCP IP Protocol Suite " 4th edition, 2010, McGraw-Hill ISBN: 0073376043

Transmission Control Protocol (TCP)

TCP Header Contd…

- The various fields of TCP header are as follows:

i. Source Port Address
ii. Destination Port Address
iii. Sequence Number
iv. Acknowledgement Number
v. Header Length
vi. Reserved

vii. Control
viii. Window Size
ix. Urgent Pointer
x. Options
xi. Checksum

Transmission Control Protocol (TCP)

TCP Header Contd...

i. Source Port Address

- 16 bit Field
- Defines port number of application program in host sending the segment

ii. Destination Port Address

- 16 bit Field
- Defines port number of application program in host receiving the segment

iii. Sequence Number

- 32 bit field
- Defines number of first byte of data contained in Segment
- Each byte is numbered for connectivity reasons
- Sequence number provides information regarding the first byte of segment to destination host

Transmission Control Protocol (TCP)

TCP Header Contd...

iv. Acknowledgement Number

- 32 bit Field
- Defines the byte number the receiver expects to receive from senders
- If 'n' bytes are received from sender 'n+1' is sent as acknowledgement number
- Acknowledgment and Data go hand in hand

v. Header Length

- 4 bit Field
- Indicates number of 4 byte words in Header
- Value of header length varies between 5 (5 x 4 = 20) and 20 (20 x 4 = 60)

vi. Reserved

- 6 bit field
- Reserved for future use

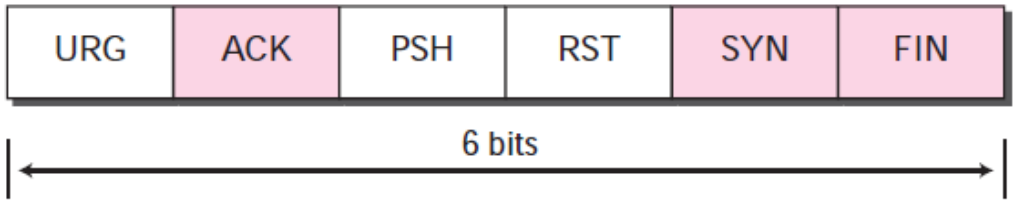
Transmission Control Protocol (TCP)

TCP Header Contd…

vii. Control

- 6 bit Field
- Defines 6 unique control bits / flags
- Can be set one / more at a time
- Enables
  - Flow Control
  - Connection Establishment, Termination and Abortion
  - Mode of TCP Data transfer

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



Control Field of TCP Header

Source: Behrouz A. Forouzan, "TCP IP Protocol Suite " 4th edition, 2010, McGraw-Hill ISBN: 0073376043

Transmission Control Protocol (TCP)

TCP Header Contd…

viii. Window Size

- 16 bit Field
- Maximum Window Size: 65, 535 bytes
- Defines Window size of sending TCP
- Determined by Receiver
- Referred as (*rwnd)*
- Sender must adhere to the window size fixed by the receiver

ix. Urgent Pointer

- 16 bit field
- Used only if urgent data is part of segment
- Valid only when urgent flag is set

x. Options

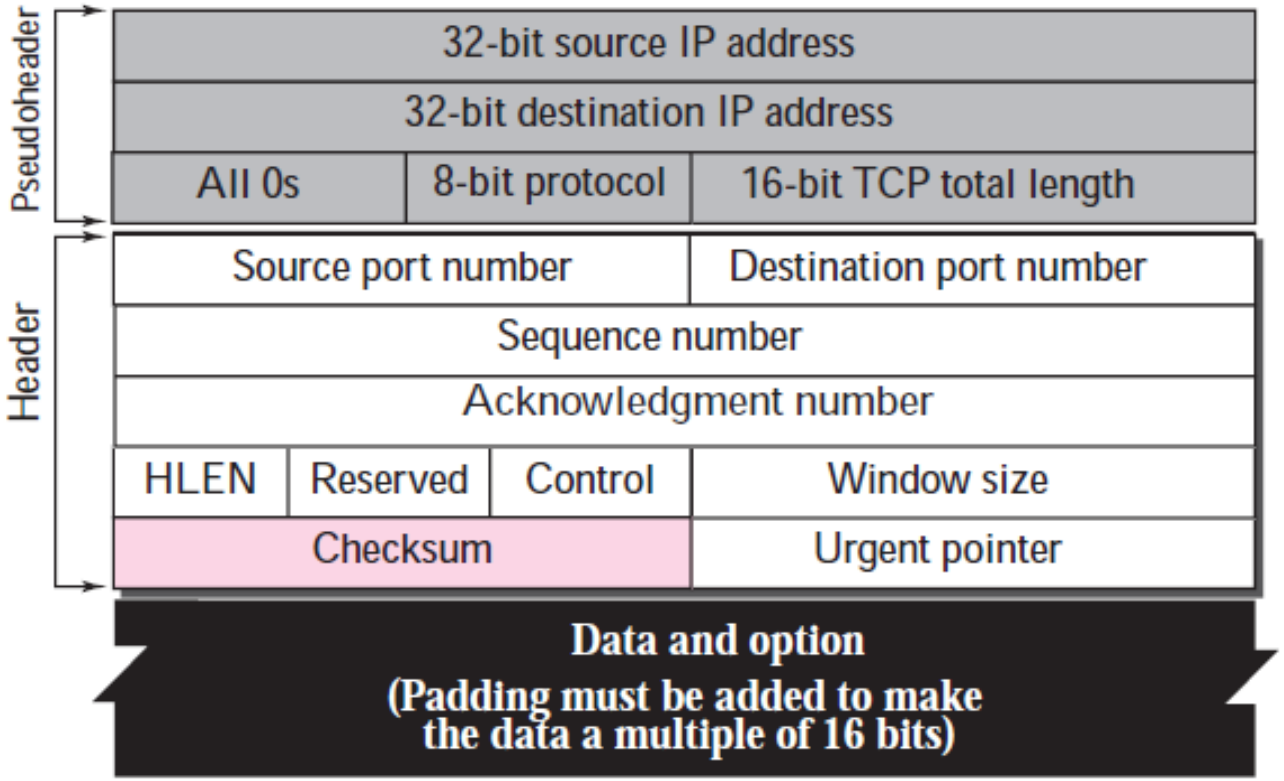
- Can accommodate up to 40 bytes of optional information

Transmission Control Protocol (TCP)

TCP Header Contd…

xi. Checksum

- 16 bit field
- Mandatory in TCP
- Detects error over entire TCP
- Pseudoheader added to segment



Pseudoheader Added to the TCP Datagram

Source: Behrouz A. Forouzan, "TCP IP Protocol Suite " 4th edition, 2010, McGraw-Hill ISBN: 0073376043

A TCP Connection

Transmission Control Protocol (TCP)

A TCP Connection

- Connection Oriented Protocol
- Virtual path established between source and destination
- TCP Segments sent and received over Virtual path
  - Enables retransmission of Lost / Damaged segments
  - Waits for segments arriving out of order
- Phases of TCP Connection
  i. Connection Establishment
  ii. Data Transfer
  iii. Connection Termination

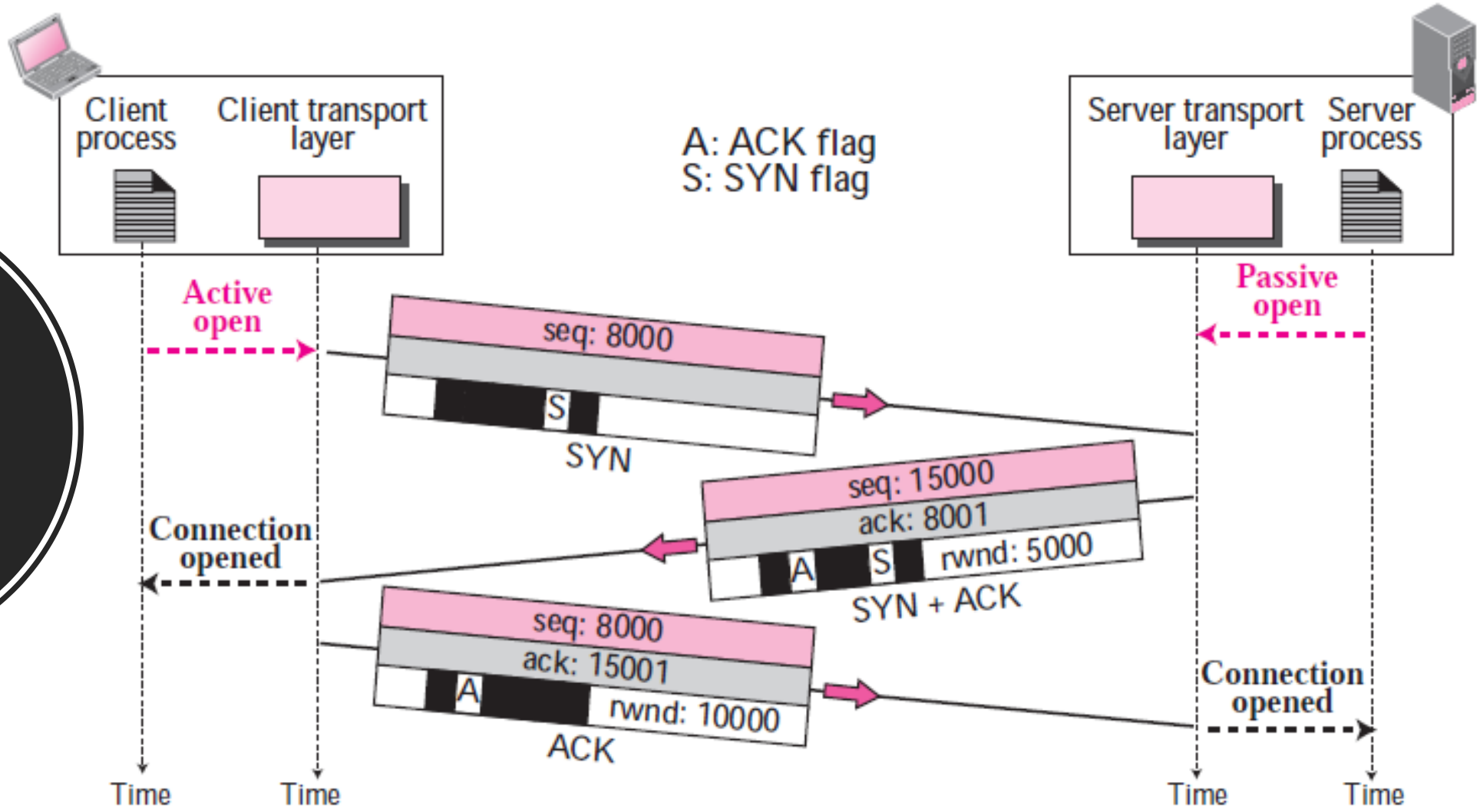
Transmission Control Protocol (TCP)

A TCP Connection Contd…

i. Connection Establishment

- Transmission mode in TCP: ***Full-Duplex***
- Simultaneous transfer of data between Sender and Receiver
- Mutual approval of Communication between sender and receiver mandatory before data transfer
- Connection establishment performed through ***3-Way Handshaking***
  - ***Passive Open*** – Server program informs TCP that it is ready to accept connection
  - ***Active Open*** – Client program intending to connect to a Server informs TCP

Connection Establishment using 3-way Handshaking



Source: Behrouz A. Forouzan, "TCP IP Protocol Suite " 4th edition, 2010, McGraw-Hill ISBN: 0073376043

Transmission Control Protocol (TCP)

A TCP Connection Contd…

i. Connection Establishment Contd…

- ***Step 1***
  - Client is in active open mode
  - Client sends first SYN segment and sets flag to SYN
  - ***Initial Sequence Number (ISN)*** : 8000 (Random Number) sent to Server
  - SYN segment (Control Segment) carries no data
  - SYN segment consumes one Sequence number
  - ***Note:*** No ACK / Window Size sent along with ISN

Transmission Control Protocol (TCP)

A TCP Connection Contd…

i. Connection Establishment Contd…

- ***Step 2***
  - Server sends second segment (SYN + ACK) and sets flag to SYN & ACK
    - **ACK** - Server acknowledges receipt of first segment from Client
    - **SYN** – Segment for communication from Server to Client
      - New Sequence Number : ***15000*** (Random Number)
      - Acknowledgement Number for Segment from client: ***8001*** (Inc. by 1)
      - Window Size (***rwnd***): Set to ***5000*** by Server (to be used by Client)
    - (SYN+ACK) segment consumes one Sequence number

Transmission Control Protocol (TCP)

A TCP Connection Contd…

i. Connection Establishment Contd…

- ***Step 3***
  - Client sends third segment (ACK) and sets ACK flag
    - **ACK** – Client acknowledges receipt of second segment from Server
      - Sequence Number : ***8000*** (Used in first Segment)
      - Acknowledgement Number for Segment from Server: ***15001*** (Inc. by 1)
      - Window Size (***rwnd***) : Set to ***10000*** by Client (to be used by Server)
    - (ACK) segment consumes no Sequence number

Transmission Control Protocol (TCP)

A TCP Connection Contd…

i. Connection Establishment Contd…

- ***Synchronous Flooding Attack***
  - Type of Denial of Service Attack (DoS)
  - Malicious attackers send large count of SYN segments to a Server
  - SYN segments pretend to come from various Clients using different IP Address
  - Server assumes that Clients are in active open mode and allocates resources
  - Server Sends SYN + ACK segments to fake clients and waits for response
  - Server runs out of resources and is unable to accept connection requests from legitimate clients

Transmission Control Protocol (TCP)

A TCP Connection Contd…

i. Connection Establishment Contd…

- ***Simultaneous Open***
  - Client and Server issues active open mode (Rare Phenomenon)
  - SYN + ACK segment sent from Client to Server and vice versa

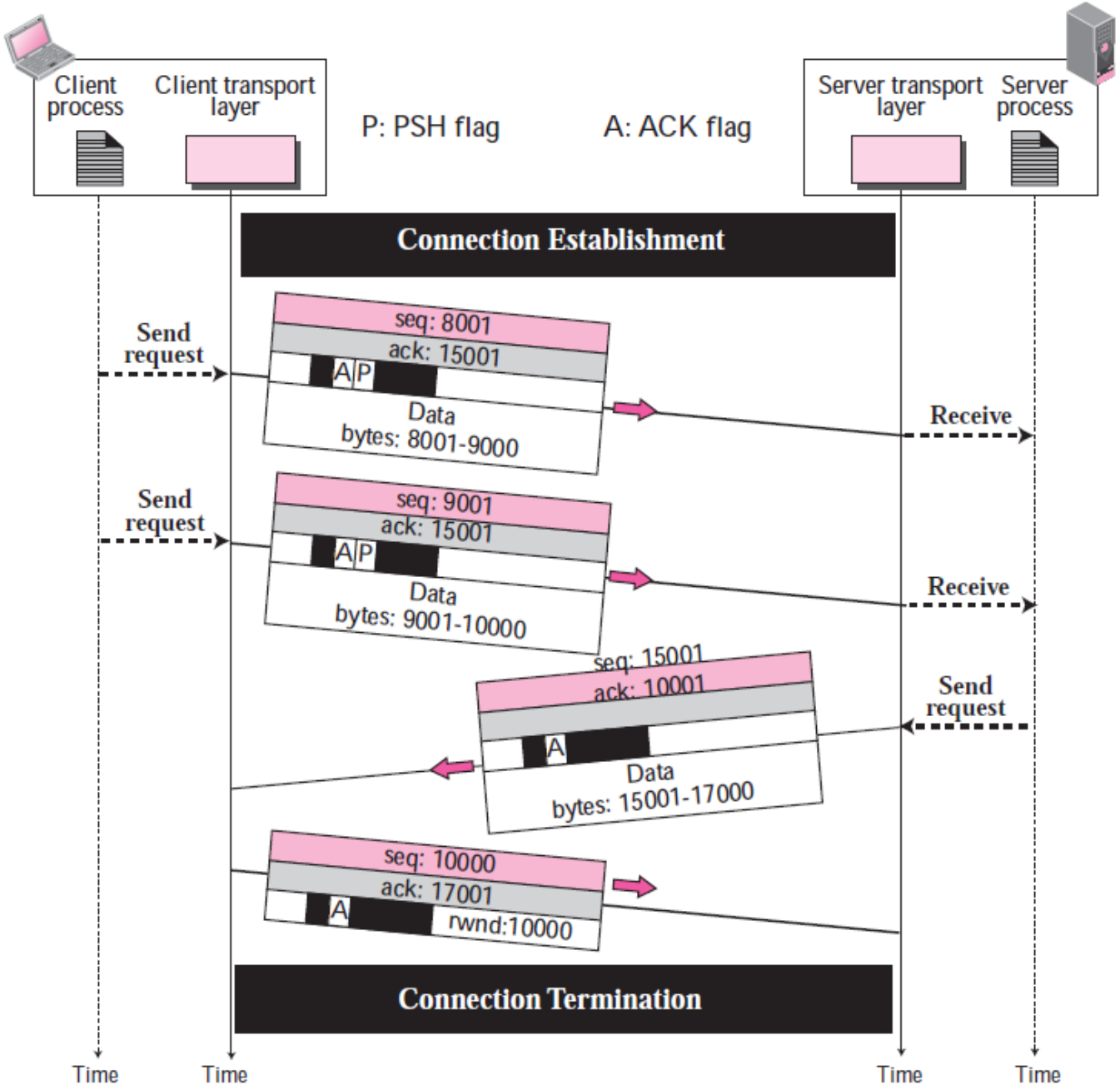
Transmission Control Protocol (TCP)

A TCP Connection Contd…

ii. TCP Data Transfer

- Data transfer is bidirectional after connection is established
- Data and acknowledgements can be sent between client and server bidirectionally
- ***Example (After Connection Establishment)***
  - Client transmits 2000 bytes of data in 1st and 2nd segment to the Server
    - Push flag (PSH) & ACK flags set for Data Segments sent by client
    - ***Segment 1:*** seq(8001), ack(15001), A & P flags set, Data bytes (8001-9000)
    - ***Segment 2:*** seq (9001), ack (15001), A & P flags set, Data bytes (9001 – 10000)

- Server sends 2000 bytes of data in the 3rd segment
- ***Segment 3:*** seq (15001), ack (10001), A flag set, Data bytes (15001 – 17000)
  - ACK flag set & SYN flag not set for data segments sent by server
- ***Segment 4:*** seq (10001), ack (17001), A flag set, rwnd (10000)



Data Transfer in TCP

Source: Behrouz A. Forouzan, "TCP IP Protocol Suite " 4th edition, 2010, McGraw-Hill ISBN: 0073376043

Transmission Control Protocol (TCP)

A TCP Connection Contd…

ii. TCP Data Transfer Contd…

- ***Flexibility of TCP***
  - Sending TCP uses buffer to store incoming data stream from applications
  - Receiving TCP uses buffer to store arriving data and send to applications
  - Disadvantage: Delayed delivery of data
- ***Pushing Data***
  - Sending TCP creates an segment and transfers immediately by setting PSH bit
  - PSH bit indicates that receiving TCP must deliver the received data segment to application program immediately

Transmission Control Protocol (TCP)

A TCP Connection Contd…

ii. TCP Data Transfer Contd…

- Data presented from application program to TCP as stream of bytes
  - Consecutive positions assigned to each byte of data
- **Exception:** Application program needs to send Urgent bytes that needs to be specially treated (With top priority)

- ***Solution: Urgent Data***
  - Send a data segment by setting the URG bit
  - Urgent data inserted at segment beginning by the sending TCP
  - End of the urgent data in segment indicated by urgent pointer field in header

Transmission Control Protocol (TCP)

A TCP Connection Contd…

ii. TCP Data Transfer Contd…

- Segment with URG bit is received by the receiving TCP
- Receiving TCP informs receiving application of the segment with URG bit

Transmission Control Protocol (TCP)

A TCP Connection Contd…

iii. TCP Connection Termination

- Connection termination initiated by the Client by default
- However, Server can also choose to close the TCP connection with client
- Options for connection termination
  a) 3-way Handshaking
  b) 4-way Handshaking with Half-Close Option

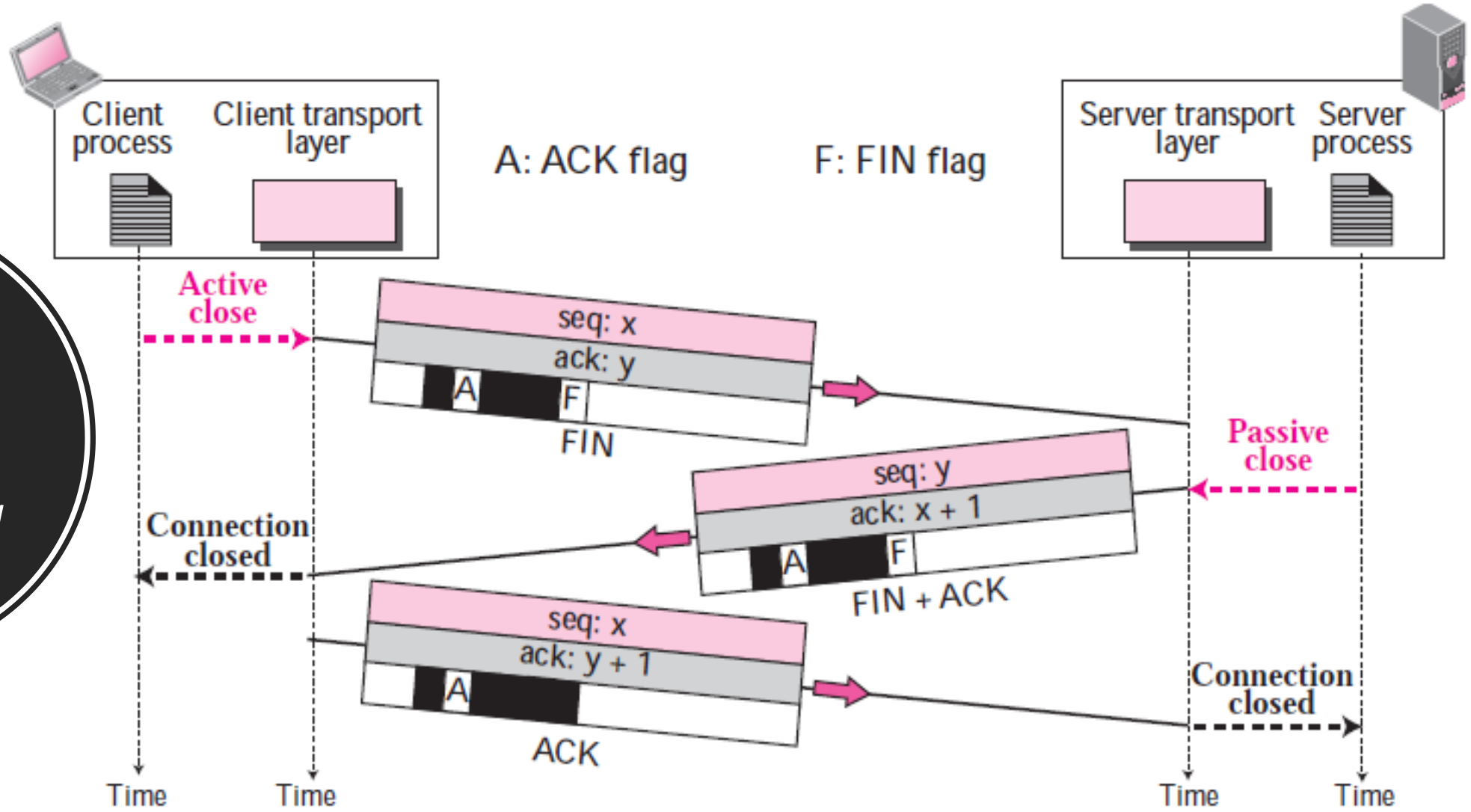
Transmission Control Protocol (TCP)

A TCP Connection Contd…

a) Connection Termination using 3-way Handshaking

- ***Passive Close*** – Server program informs TCP that it is ready to close connection
- ***Active Close*** – Client program intending to close connection to a Server informs TCP
- ***Step 1***
  - Client process sends close command to Client TCP
  - Client TCP sends 1st Segment (FIN) with FIN flag and ACK flag set
  - FIN Segment consumes one sequence number (if it carries no data)
  - Note: FIN segment may carry last chunk of data also

Connection Termination using 3-way Handshaking



Source: Behrouz A. Forouzan, "TCP IP Protocol Suite " 4th edition, 2010, McGraw-Hill ISBN: 0073376043

Transmission Control Protocol (TCP)

d) A TCP Connection Contd…

a) Connection Termination using 3-way Handshaking

- ***Step 2***
  - Server TCP sends 2nd segment (FIN + ACK) to confirm receipt of FIN segment
  - Server announces connection closing from its side
  - 2nd Segment may contain last chunk of data
- ***Step 3***
  - Client TCP sends 3rd segment (ACK) that confirms FIN receipt from Server
  - Acknowledgment number – Sequence number + 1
  - 3rd Segment cannot carry data

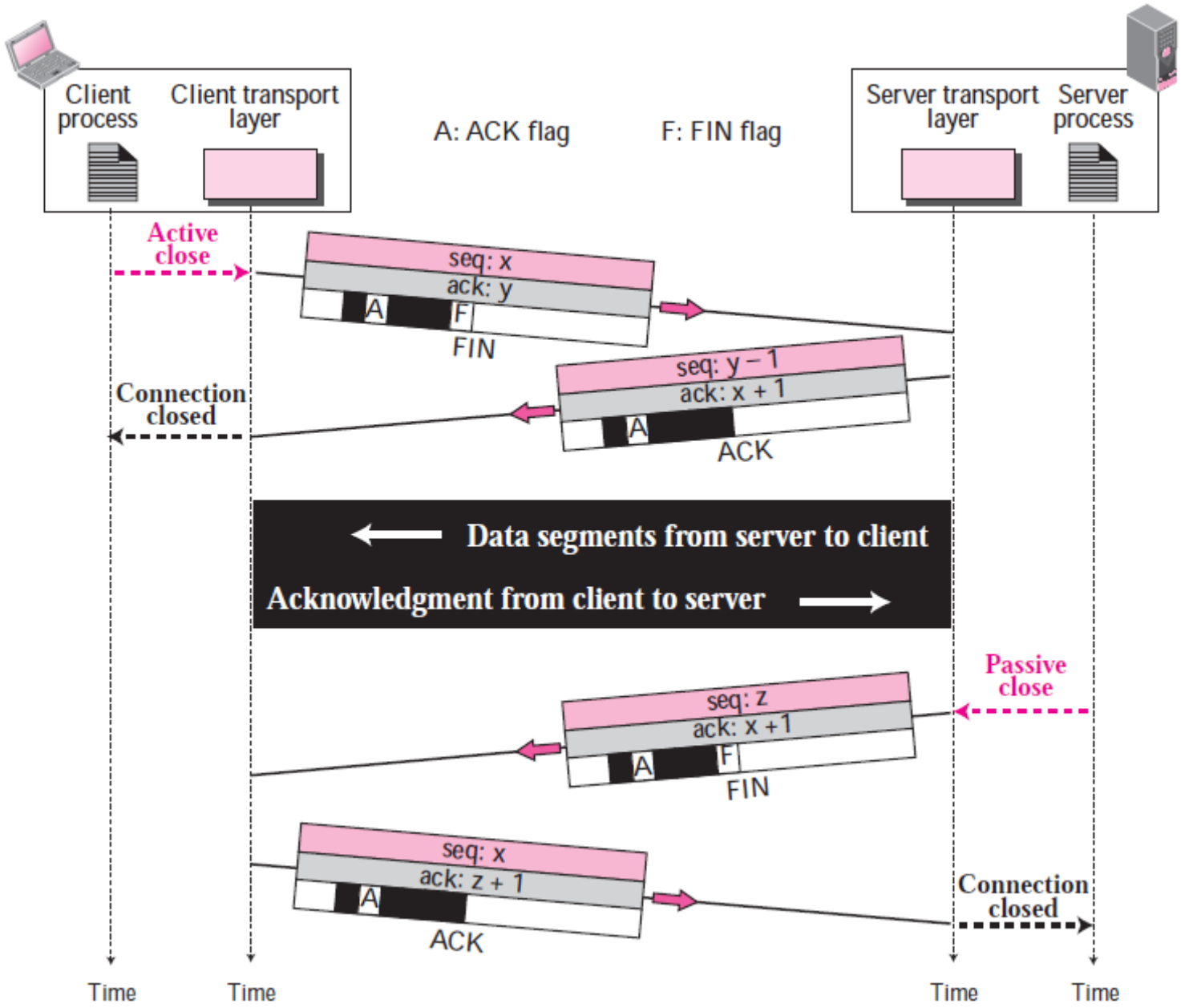
Transmission Control Protocol (TCP)

A TCP Connection Contd…

b) Half Close Connection Termination

- ***Half Close –*** In client / Server communication if one can stop sending data while the other can send data it is called an Half-Close
  - The Client or Server can issue a Half-Close request
- Example – Sorting
  - Client sends data for sorting to Server
    - Client closes connection in Client-Server direction
  - Server receives data from client & keeps connection open
    - Till Sorting is completed & result sent back to Client

Half-Close



Source: Behrouz A. Forouzan, "TCP IP Protocol Suite " 4th edition, 2010, McGraw-Hill ISBN: 0073376043

Transmission Control Protocol (TCP)

A TCP Connection Contd…

iv. TCP Connection Reset

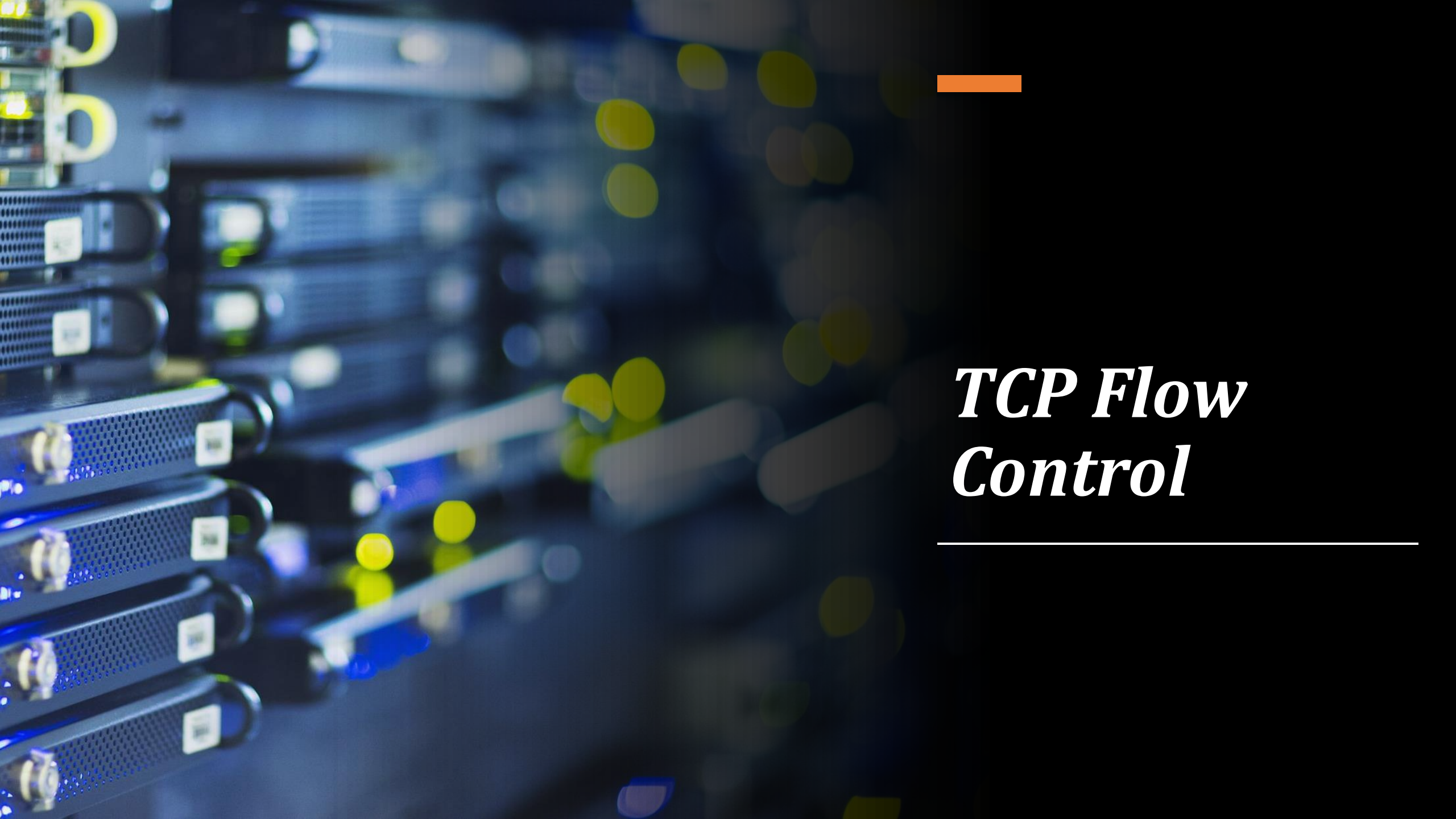
- Reset Flag (RST) - Denies connection / Aborts connection / Terminates existing idle connection
  - ***Denying Connection***
    - Client requests to a non-extent server port
    - Server sends segment with RST flag set and denies request
  - ***Aborting Connection***
    - Client / Server TCP aborts existing connection by sending RST segment to abort connection

Transmission Control Protocol (TCP)

A TCP Connection Contd…

iv. TCP Connection Reset Contd…

- ***Termination Idle Connection***
  - Client finds server idle or vice versa
  - Client / Server sends RST segment to terminate connection
  - Similar to abort

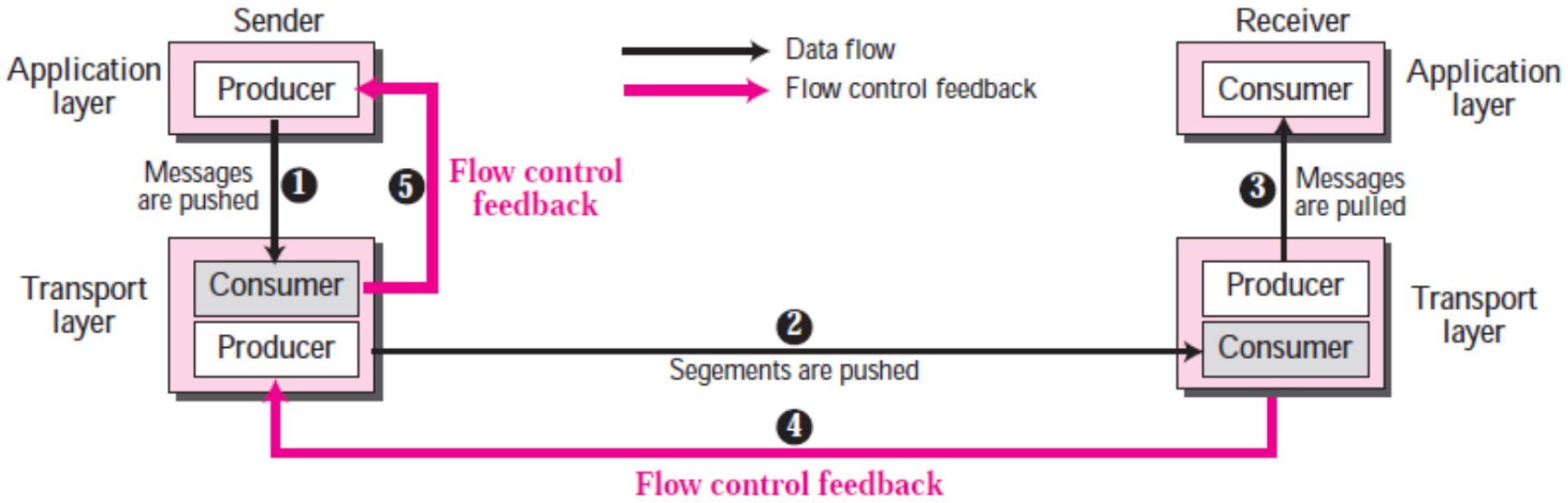


TCP Flow Control

Transmission Control Protocol (TCP)

TCP Flow Control

- Creates a balance between rate of data production and the rate of data consumption
- ***Assumption:*** Channel between sender & receiver is error-free



Data Flow and Flow Control Feedbacks in TCP

Source: Behrouz A. Forouzan, "TCP IP Protocol Suite " 4th edition, 2010, McGraw-Hill ISBN: 0073376043

Transmission Control Protocol (TCP)

TCP Flow Control Contd…

1) Messages are pushed from the Sending application to TCP Client
2) Message segment from TCP Client is pushed to TCP Server
3) Messages are pulled by receiving application from TCP Server
4) Flow control feedback is sent from TCP server to TCP client
5) TCP client forwards the flow control feedback to sending application

- ***Opening and Closing Windows***
  - Buffer size of sender & receiver is fixed during connection establishment
  - Window sizes of Sender / Receiver is controlled and adjusted by TCP Server
  - Opening / Closing / Shrinking of client window is controlled by receiver

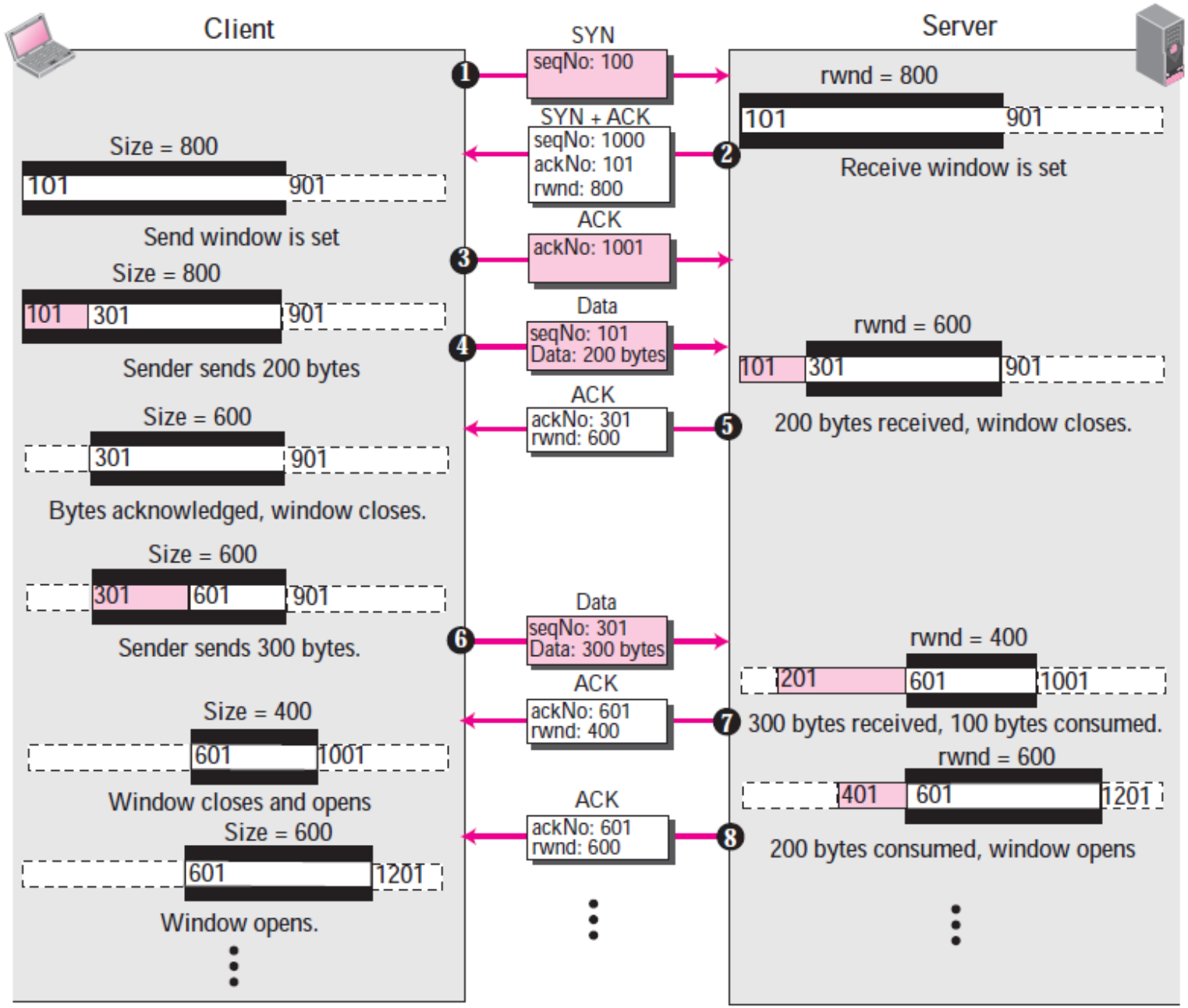
Transmission Control Protocol (TCP)

TCP Flow Control Contd…

Scenario – TCP Flow Control

- ***Segment 1***
  - Connection request – (SYN) segment from client to server
  - Initial Sequence Number (ISN): 100 (Next byte to Arrive : 101)
  - Server allots buffer size = 800 (Assumption)
  - Server allots Window size (rwnd) = 800
- ***Segment 2***
  - (ACK + SYN) segment from Server to Client
  - Set ACK no = 101 & Buffer Size = 800 bytes

Flow Control – A Scenario



Source: Behrouz A. Forouzan, "TCP IP Protocol Suite " 4th edition, 2010, McGraw-Hill ISBN: 0073376043

Transmission Control Protocol (TCP)

TCP Flow Control Contd…

Scenario – TCP Flow Control

- ***Segment 3***
  - ACK segment sent from client to server
- ***Segment 4***
  - Client sets window size to 800 (since rwnd from server is 800)
  - Client process pushes 200 bytes of data to TCP client
  - TCP client creates data segment with bytes (101-300) and sends to Server
  - Client window adjusted
    - Shows 200 bytes as sent but no ACK received from Server

Transmission Control Protocol (TCP)

TCP Flow Control Contd…

Scenario – TCP Flow Control

- Shows 200 bytes as sent but no ACK received from Server
- Server stores 200 bytes in buffer & closes receive window
- Server indicates the next expected byte as 301

- ***Segment 5***
  - Server acknowledges receipt of 200 bytes from client & reduces rwnd to 600
  - Client receives acknowledgement and resizes window size to 600
  - Client closes the window (101-300) from left to right
  - Client indicates the next byte to send as 301

Transmission Control Protocol (TCP)

TCP Flow Control Contd…

Scenario – TCP Flow Control

- ***Segment 6***
  - Client pushes 300 more bytes to the server (Seq. No = 301 & Data = 300 bytes)
  - Server stores 300 bytes in buffer
  - 100 bytes of data are pulled by the Client process
  - So, Window size is reduced by 100 bytes to the left & opened by 100 bytes to the right
  - Overall, TCP client window size is reduced by 200 bytes
  - Now, Receiver window size (rwnd) = 400

Transmission Control Protocol (TCP)

TCP Flow Control Contd…

Scenario – TCP Flow Control

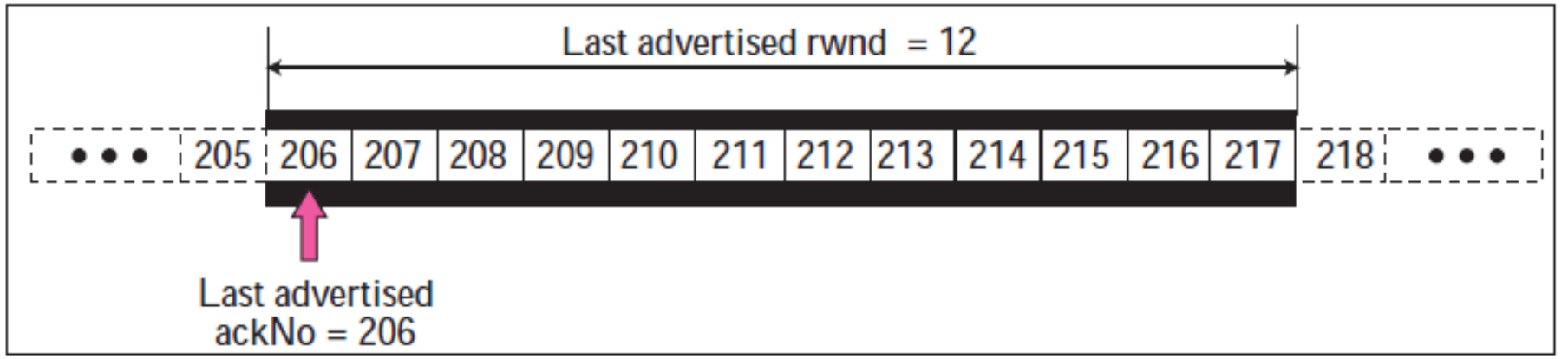
- ***Segment 7***
  - TCP Server acknowledges receipt of 300 bytes and sets window size (rwnd) = 400 & TCP Client reduces window size to 400
  - Sender windows closes by 300 bytes from the left and opens by 100 bytes to the right
- This process continues until all the data segments are sent from server to client and connection gets closed

Transmission Control Protocol (TCP)

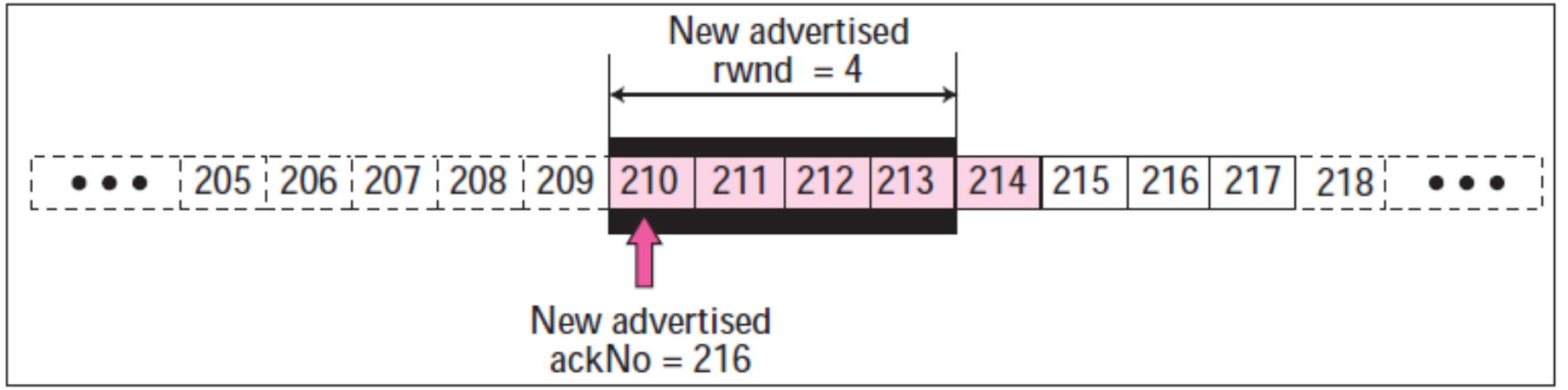
TCP Flow Control Contd…

- ***Shrinking of Windows***
  - Shrinking : Decreasing window size i.e., right wall moves towards the left
  - Sender window may shrink based on rwnd value defined by receiver
  - Receiver window cannot shrink
  - ***Illustration***
    - Upto 205 bytes of data received and acknowledged by sender
    - Last advertised rwnd = 12 (Window size) & last advertised ACK No = 206
      - Data can be sent from byte 206 to byte 217 (Since rwnd = 12)
    - New advertised rwnd = 12 (Window size) & new advertised ACK No = 210

The window after the new advertisement; Window has shrunk



a. The window after the last advertisement



Source: Behrouz A. Forouzan, "TCP IP Protocol Suite " 4th edition, 2010, McGraw-Hill ISBN: 0073376043

Transmission Control Protocol (TCP)

TCP Flow Control Contd…

- ***Shrinking of Windows Contd…***
  - Shrinking of window has occurred from byte 217 to byte 213 (Window had moved from right to left
  - Shrinking can be prevented by the relation given below:

    New ACK number + new rwnd > = Lask ACK Number + Last rwnd

  - To prevent shrinking,
    - Wait until enough buffer locations are available in its window

Transmission Control Protocol (TCP)

TCP Flow Control Contd…

- ***Silly Window Syndrome***
  - Occurs when either the sending process creates data slowly (or) the receiving process consumes data slowly (or) both
  - Silly window syndrome sends / receives data in small segments thus resulting in poor efficiency
  - ***Example***
    - A 42 byte TCP datagram is needed to send Segment with 2 bytes of data
    - Overhead : 42 / 2 ⇒ Network capacity used inefficiently

Transmission Control Protocol (TCP)

TCP Flow Control Contd…

- ***Silly Window Syndrome Created by Sender***
  - Sending TCP creates syndrome since sending application program creates data slowly i.e., 1 byte at an instance
  - ***Suggestions***
    - Prevent sending TCP from transmitting data byte by byte
    - Sending TCP made to wait & collect data to send data in larger block
    - ***Disadvantage:*** Waiting too long delays the process
  - ***Solution***
    - Nagle's Algorithm

Transmission Control Protocol (TCP)

TCP Flow Control Contd…

- ***Nagle's Algorithm for Silly Window Syndrome Created by Sender***
  i. First segment of data sent by sending TCP irrespective of the size of segment
  ii. Data is accumulated in buffer by sending TCP until acknowledgment is received from receiving TCP (or) enough data accumulated to send a segment
  iii. Repeat step 2 until transmission completes
- Nagle's algorithm works based on speed of application program and the network speed
  - Faster the application program larger the segment size
- ***Advantage:*** Simple to implement

Transmission Control Protocol (TCP)

TCP Flow Control Contd…

- ***Silly Window Syndrome Created by Receiver***
  - Syndrome created when serving application consumes slowly
  - ***Example***
    - 1 KB data blocks created by sending application
    - 1 byte of consumed at a time by receiving application
    - Once sending window buffer is full, window size (rwnd) becomes 0
  - ***Solution 1:*** Clarks's Solution
    - Send ACK as data segment arrives
    - Set rwnd = 0 iff receive buffer is half empty (or) there is enough space

Transmission Control Protocol (TCP)

TCP Flow Control Contd…

- ***Silly Window Syndrome Created by Receiver Contd…***
  - ***Solution 2:*** Delayed Acknowledgment
    - Acknowledgement is withheld when segment arrives
    - Receiver waits for space in incoming buffer before acknowledging
    - Delayed ACK prevents sending TCP from sliding
    - Delayed ACK reduces network traffic
    - Note: ACK not to be delayed by more than 500 ms

TCP Error Control

Transmission Control Protocol (TCP)

TCP Error Control

- TCP – Reliable Transport layer Protocol
- Entire data stream to be delivered without error / loss / duplication
- Reliability is provided by TCP using Error Control
- Error Control includes:
  - Finding and resending corrupted segments
  - Resending lost segments
  - Storing out-of-order segments till missed segments arrive
  - Discarding duplicate segments
- Error control achieved by: Checksum, Acknowledgement and Time-out

Transmission Control Protocol (TCP)

TCP Error Control Contd…

a) Checksum

- TCP uses 16-bit mandatory checksum field
- Checksum field associated with each segment
  - Checks for corrupted segment
- Invalid Checksum: Segment discarded by receiving TCP & considered lost

Transmission Control Protocol (TCP)

TCP Error Control Contd…

b) Acknowledgement

- Acknowledgement segments (ACK) carry no data & confirms data segment receipt
- Types : Cumulative and Selective Acknowledgment
- ***Cumulative Acknowledgment (ACK)***
  - 32-bit ACK field used
  - Acknowledges segments cumulatively (sets ACK flag to 1)
  - No feedback provided for discarded, lost or duplicate segments
- ***Selective Acknowledgement (SACK)***
  - Reports out of order & duplicate segments

Transmission Control Protocol (TCP)

TCP Error Control Contd…

b) Acknowledgement Contd…

- No provision for SACK in TCP header
- SACK included as part of options field in the TCP header

- ***Rules for Generating Acknowledgments***

1) When data is sent from sender to receiver, ACK provides the next Seq. No expected to be received
   - This results in less traffic and less segments between sender and receiver
2) In case of one in-order segment remaining, receiver needs to delay sending ACK segment. Network traffic is thus reduced.

Transmission Control Protocol (TCP)

TCP Error Control Contd…

b) Acknowledgement Contd…

3) At no point of time there should be more than two in-order segments unacknowledged. (Thwarts unnecessary retransmission)
4) Receiver acknowledges (ACK) an higher out-of-order sequence number immediately leading to fast retransmission of next segment
5) Receiver sends ACK when a missing segment arrives. Segments reported as missing are thus informed to the receiver
6) Receiver discards duplicate segment & sends ACK indicating the next in-order segment. Lost ACK segment problems are thus solved.

Transmission Control Protocol (TCP)

TCP Error Control Contd…

c) Retransmission of Segments

- When retransmission occurs?
  - Expiry of retransmission timer (or)
  - Sender receives more than 2 duplicate ACK's for 1st segment
- ***Retransmission after RTO***
  - One retransmission time-out (RTO) maintained by sending TCP for each connection
  - In case of time-out, timer is restarted by TCP & first segment of Queue is sent
  - This version of TCP is called ***Tahoe***

Transmission Control Protocol (TCP)

TCP Error Control Contd…

c) Retransmission of Segments Contd…

- ***Retransmission after 3 Duplicate Segments***
  - Also called as Fast Retransmission & followed by most implementations
  - TCP version called as ***Reno***
  - If 3 identical duplicate ACK's along with the original ACK are received for a segment, the next segment is retransmitted
  - Note: Retransmission does not wait for time-out in this case

Transmission Control Protocol (TCP)

TCP Error Control Contd…

d) Out-of-Order Segments

- Out-of-Order segments are not discarded by TCP
- TCP flags such segments as out-of-order and store them temporarily until missing segments arrive
- TCP makes sure that data segments are delivered in sequence to the process

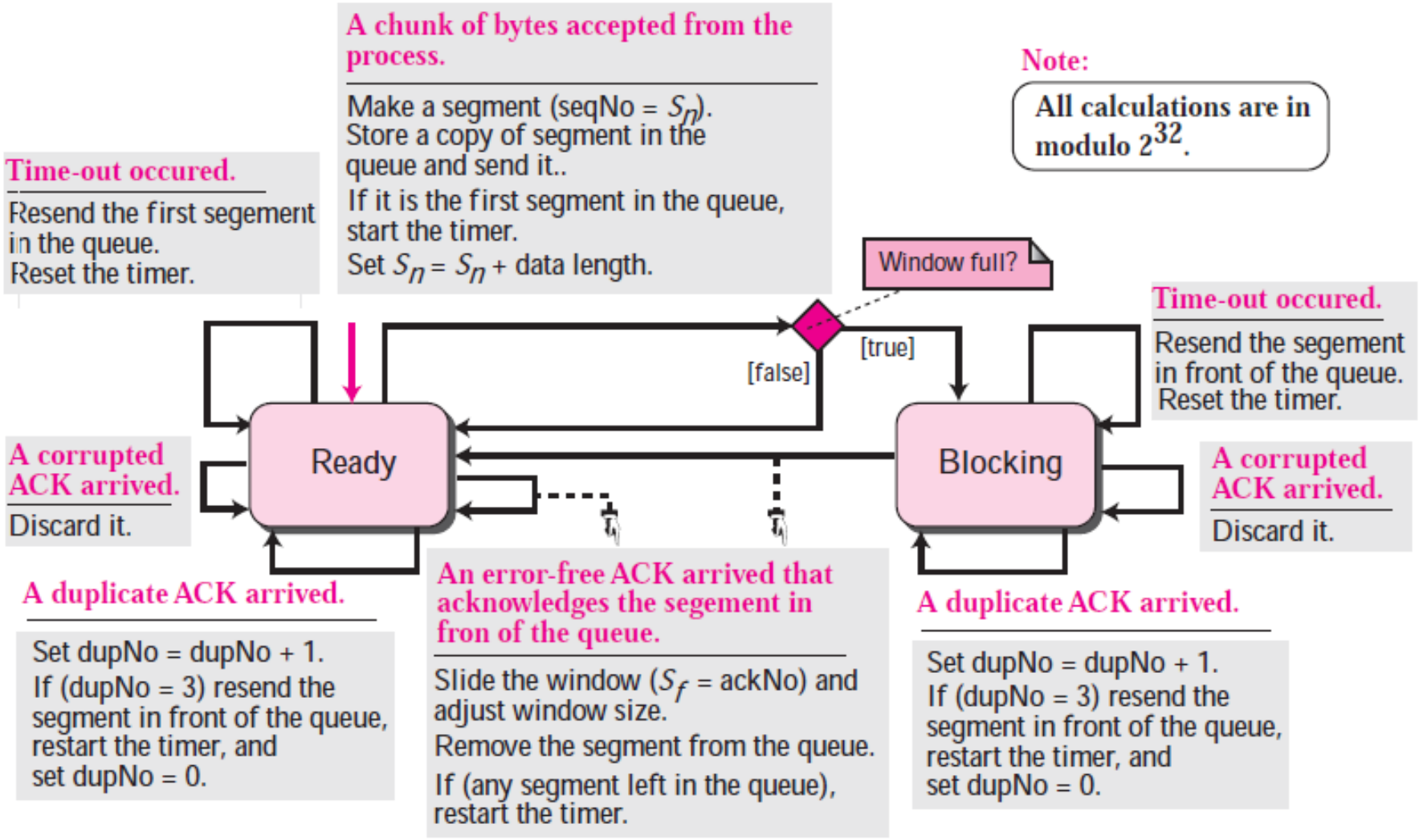
Transmission Control Protocol (TCP)

TCP Error Control Contd…

e) FSM for Data Transfer in TCP

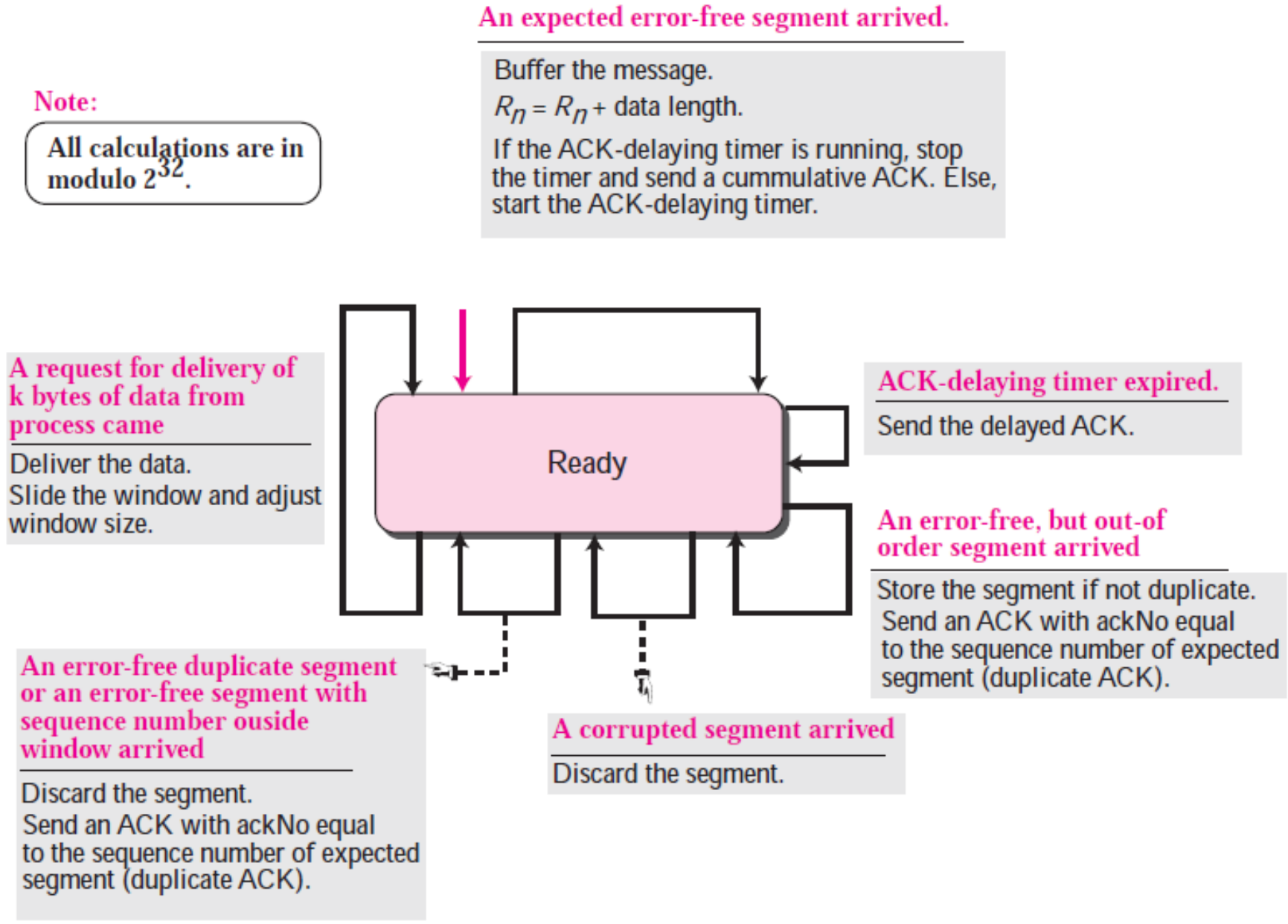
- FSM – Finite State Machine
- Similar to Selective repeat and Go Back-N protocol
- ***Sender-side & Receiver Side FSM***
  - ***Assumption :*** Unidirectional communication
  - ***Ignored Parameters:*** Selective ACK and Congestion Control
  - Nagle's algorithm / Windows shutdown not included in FSM
  - ***Advantage:*** Fast transmission policy using 3 duplicate ACK segments
  - ***Bi-directional FSM :*** Complex and more practical

Simplified FSM for TCP Sender Side



Source: Behrouz A. Forouzan, "TCP IP Protocol Suite " 4th edition, 2010, McGraw-Hill ISBN: 0073376043

Simplified FSM for TCP Receiver Side



Source: Behrouz A. Forouzan, "TCP IP Protocol Suite " 4th edition, 2010, McGraw-Hill ISBN: 0073376043

TCP Congestion Control

Transmission Control Protocol (TCP)

TCP Congestion Control

- Congestion window and congestion policy handles TCP congestion

a) Congestion Window

- Client window size (rwnd) decided by the available buffer space of Server
- Ignored entity in deciding window size : Network Congestion
- Sender window size determined by,
  - rwnd (receiver advertised window size) &
  - cwnd (Congestion window size)

Actual window size = min (rwnd, cwnd)

Transmission Control Protocol (TCP)

TCP Congestion Control Contd…

TERMINOLOGIES & ABBREVIATIONS USED

- ***rwnd –*** Sender Window Size
- ***cwnd –*** Congestion Window Size
- ***ssthresh –*** Slow Start Threshold
- ***MSS –*** Maximum Segment Size
- ***ACK –*** Acknowledgement
- ***RTT –*** Round Trip Time
- ***RTO –*** Retransmission Time-out

Transmission Control Protocol (TCP)

TCP Congestion Control Contd…

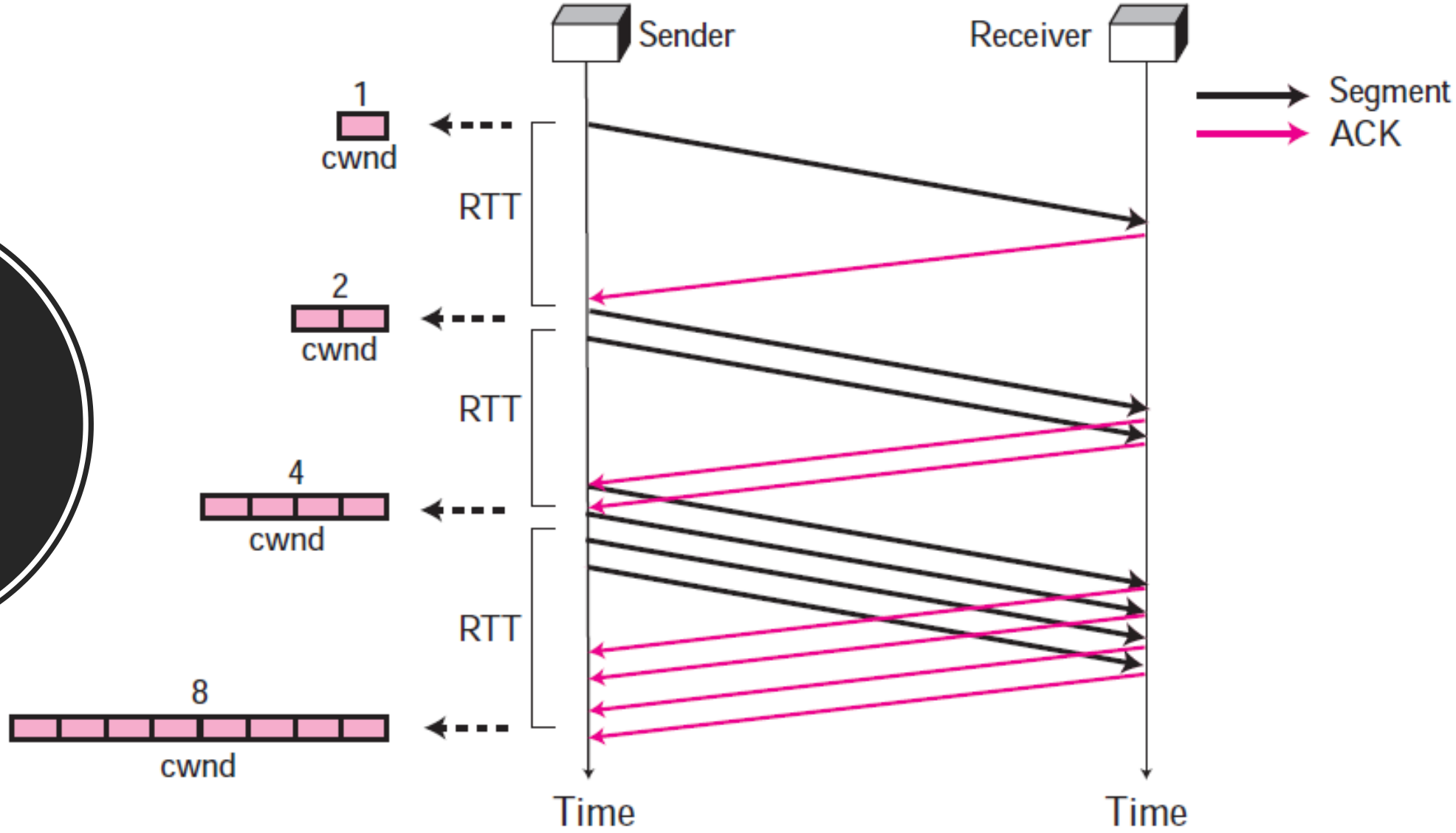
b) Congestion Policy

- Three phases : Slow Start, Congestion avoidance & Congestion detection

i. Slow Start (Exponential Increase)

- ***Assumption:*** rwnd > cwnd
- cwnd initialized to one Maximum window size (MSS)
- On arrival of each ACK, cwnd increases by 1
- Algorithm starts slowly & grows exponentially
- Delayed ACK policy is ignored
- ***Note:*** Consider each segment is individually acknowledged

Slow Start, Exponential Increase



Source: Behrouz A. Forouzan, "TCP IP Protocol Suite " 4th edition, 2010, McGraw-Hill ISBN: 0073376043

Transmission Control Protocol (TCP)

TCP Congestion Control Contd…

b) Congestion Policy

i. Slow Start (Exponential Increase) contd…

- Initial value of cwnd = 1 MSS

No of MSS sent	No of Segments Acknowledged	RTT	cwnd in MSS
Nil	Nil	-	1
1	1	1	$1 \times 2 = 2 \Rightarrow 2^1$
2	2	2	$2 \times 2 = 4 \Rightarrow 2^2$
4	4	3	$4 \times 2 = 8 \Rightarrow 2^3$
8	8	4	$8 \times 2 = 16 \Rightarrow 2^4$

Transmission Control Protocol (TCP)

TCP Congestion Control Contd…

b) Congestion Policy

i. Slow Start (Exponential Increase) contd…

- For Delayed Acknowledgements
  - If multiple segments are acknowledged accumulatively, cwnd increases by 1
  - ***Example:*** if ACK = 4 them cwnd = 1
  - Growth is exponential but not to the power of 2
  - Slow start stops with a threshold value ***ssthresh***
    - It stops when ***window size = = ssthresh***

Transmission Control Protocol (TCP)

TCP Congestion Control Contd…

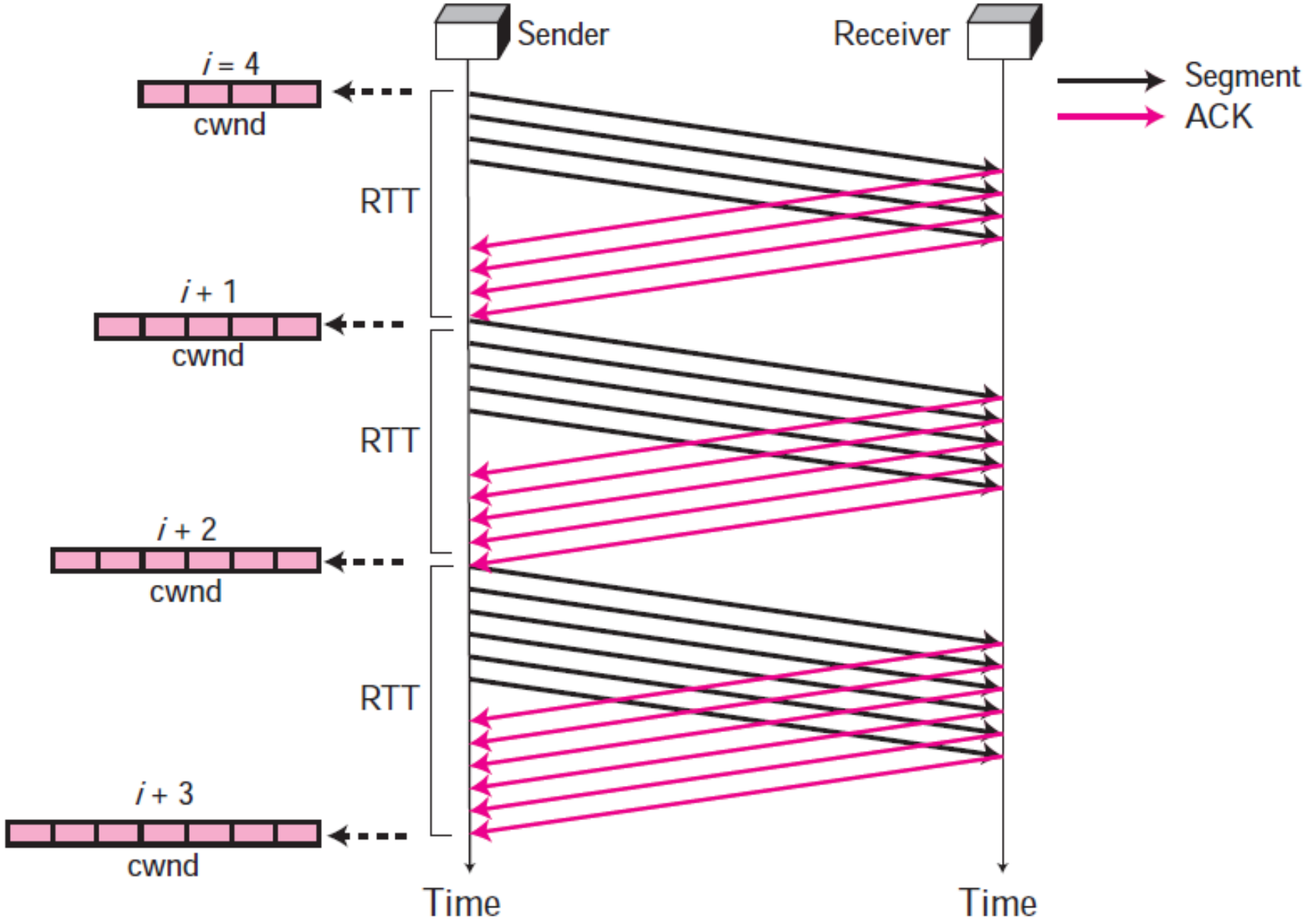
b) Congestion Policy

ii. Congestion Avoidance : Additive Increase

- Slow start increases congestion window size (cwnd) exponentially
- Congestion avoidance increases cwnd additively
- Additive phase begins when slow start reaches ssthresh i.e. cwnd = I
- Increase in cwnd is based on RTT & not on number of ACK's

RTT	cwnd in MSS
1	i
2	i + 1

Congestive avoidance, Additive Increase



Source: Behrouz A. Forouzan, "TCP IP Protocol Suite " 4th edition, 2010, McGraw-Hill ISBN: 0073376043

Transmission Control Protocol (TCP)

TCP Congestion Control Contd…

b) Congestion Policy

iii. Congestion Detection : Multiplicative Decrease

- Size of Cwnd must be decreased in case of congestion
- Retransmission occurs during missing segments / lost segments
- Retransmission helps identify whether congestion has occurred or not
- Retransmission occurs when
  - There is RTO Time-out
  - On receipt of three duplicate ACK's
- Note: In both cases, ssthresh is reduced by half (Multiplicative decrease)

Transmission Control Protocol (TCP)

TCP Congestion Control Contd...

b) Congestion Policy

iii. Congestion Detection : Multiplicative Decrease Contd...

a) Time-out increases possibility of congestion. TCP reacts as follows:

- Ssthresh set to half the value of rwnd
- Cwnd initialized to 1
- Slow start phase is initiated again

b) Three duplicate ACK's indicates a weaker possibility of Congestion. Also called as fast transmission & fast recovery. TCP reacts as follows:

- Ssthresh set to half the value of rwnd

Transmission Control Protocol (TCP)

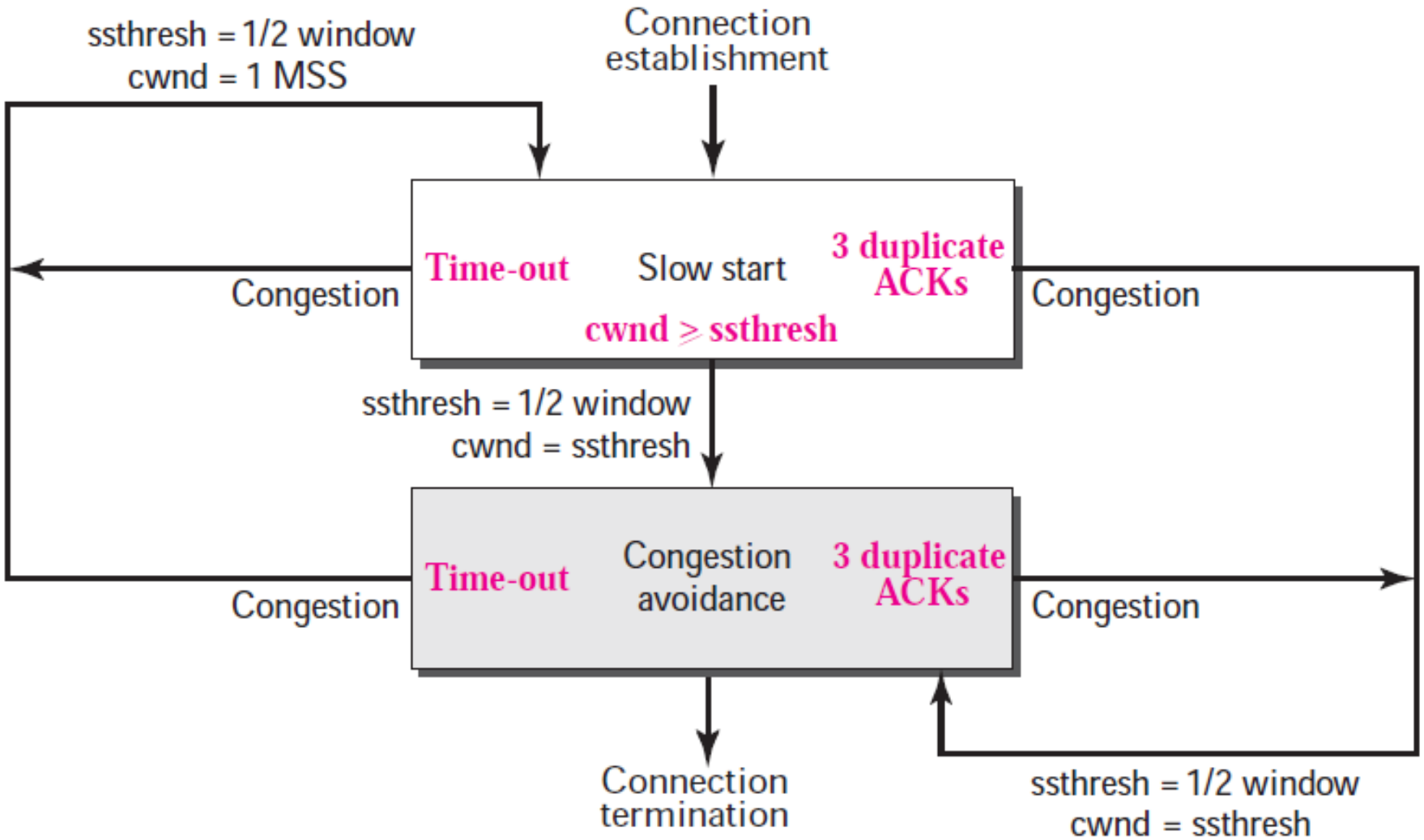
TCP Congestion Control Contd…

b) Congestion Policy

iii. Congestion Detection : Multiplicative Decrease Contd…

- Cwnd = ssthresh
- Congestion avoidance phase is initiated again

TCP Congestion Policy : A Summary



Source: Behrouz A. Forouzan, "TCP IP Protocol Suite " 4th edition, 2010, McGraw-Hill ISBN: 0073376043

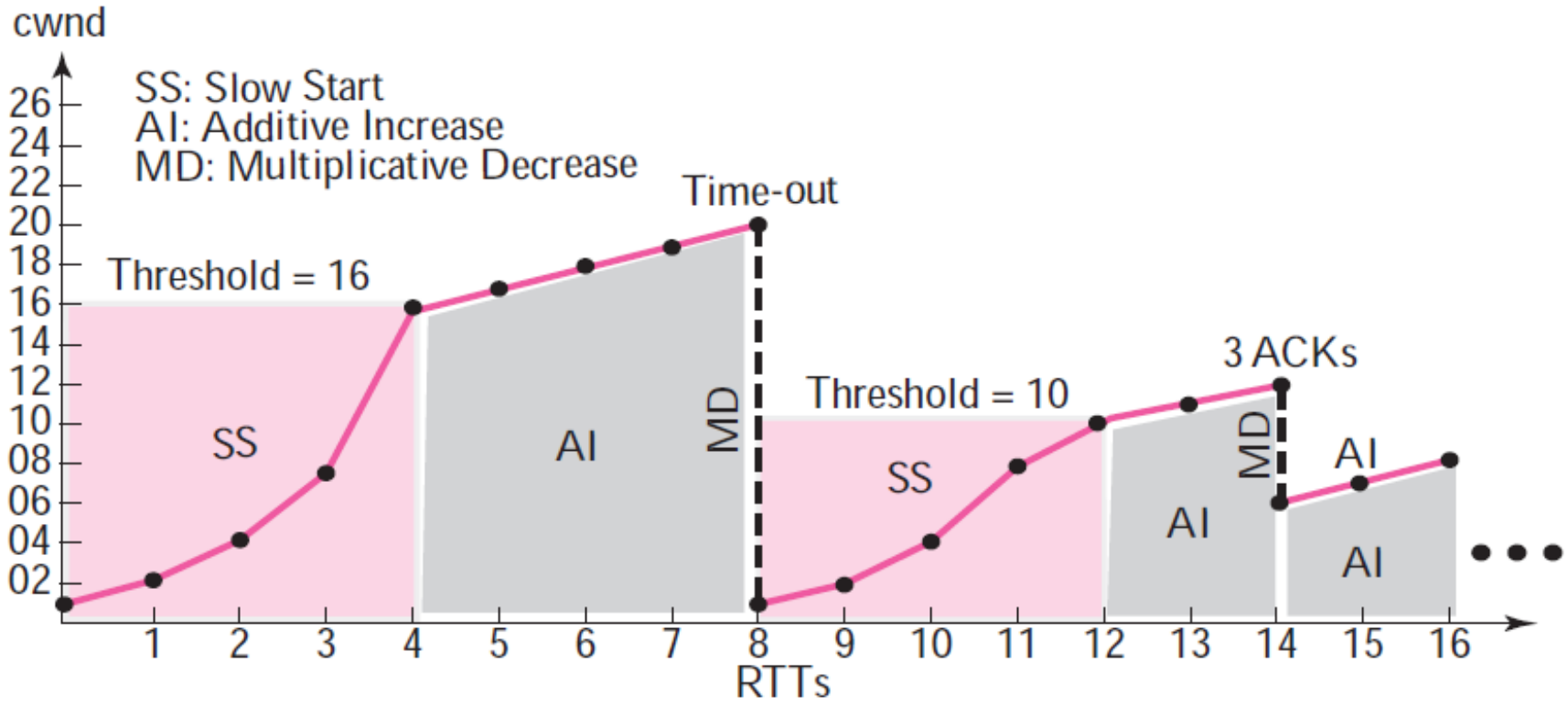
Transmission Control Protocol (TCP)

TCP Congestion Control Contd…

Summarization with Example

- Assumptions
  - Maximum window Size (MSS) = 32
  - Threshold (ssthresh) = 16
- TCP moves to slow start
  - rwnd starts from 1 and and grows exponentially till it reaches ssthresh (16)
- Additive increase increases rwnd from 16 to 20 (one by one)
  - When rwnd = 20, time-out occurs
- Multiplicative Decrease: ssthresh reduced to 10 (half the window size)

Congestion Example



Source: Behrouz A. Forouzan, "TCP IP Protocol Suite " 4th edition, 2010, McGraw-Hill ISBN: 0073376043

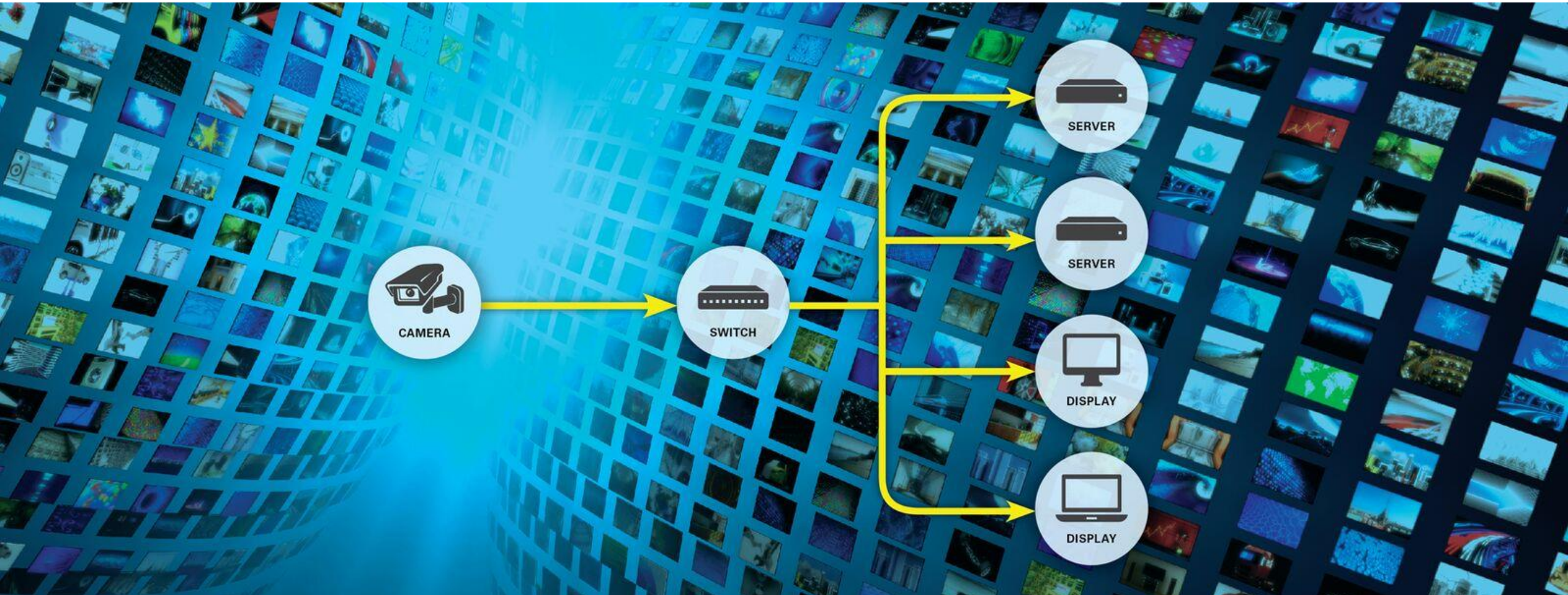
Transmission Control Protocol (TCP)

TCP Congestion Control Contd…

Summarization with Example Contd…

- New ssthresh = 10
- TCP moves to Slow start again
  - rwnd starts from 1 and and grows exponentially till it reaches new ssthresh (10)
- Additive increase increases rwnd from 10 to 12 (one by one)
- 2 duplicate ACK's are received by the sender
- Multiplicative Decrease: ssthresh reduced to 6 (half the window size)

Multicast & Multicast Routing Protocols



Multicast & Multicast Routing Protocols

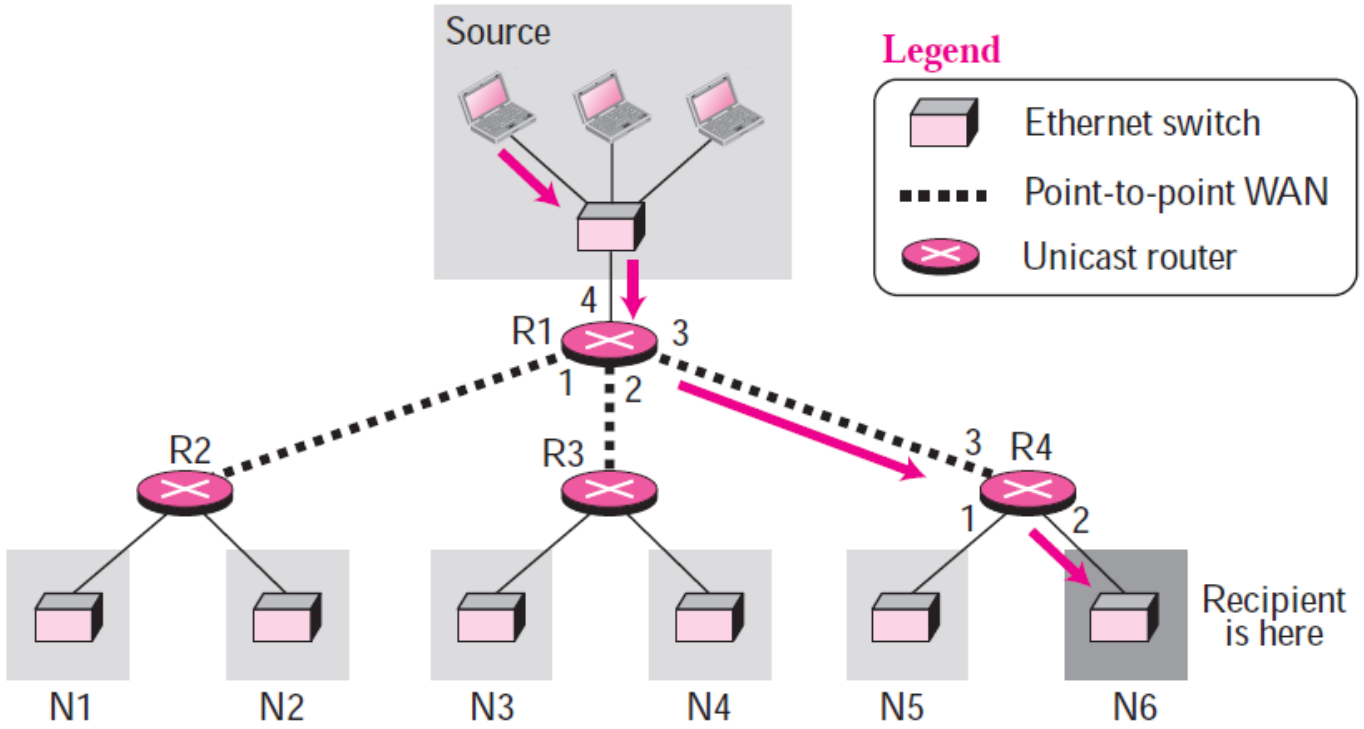
Introduction to Multicasting

a) Unicasting

- One Source and one Destination network
- Relationship between Source and Destination : One to One
- What is Unicasting?
  - Each router in the datagram path forwards packets to only one interface
- ***Example***
  - ***Problem:*** Delivery of packet from Source to Destination (N6)
  - ***Routers used:*** R1, R2 & R3
  - No of Ethernet Switches: 1

➢ *Solution*

- R1 forwards packets using interface 3
- R4 forwards packets using interface 2
- Delivery from Source to destination N6 is network's responsibility
- Mode of Delivery : Broadcast to all Hosts / Only to Destination (N6)



Unicasting – Example Scenario

Source: Behrouz A. Forouzan, "TCP IP Protocol Suite " 4th edition, 2010, McGraw-Hill ISBN: 0073376043

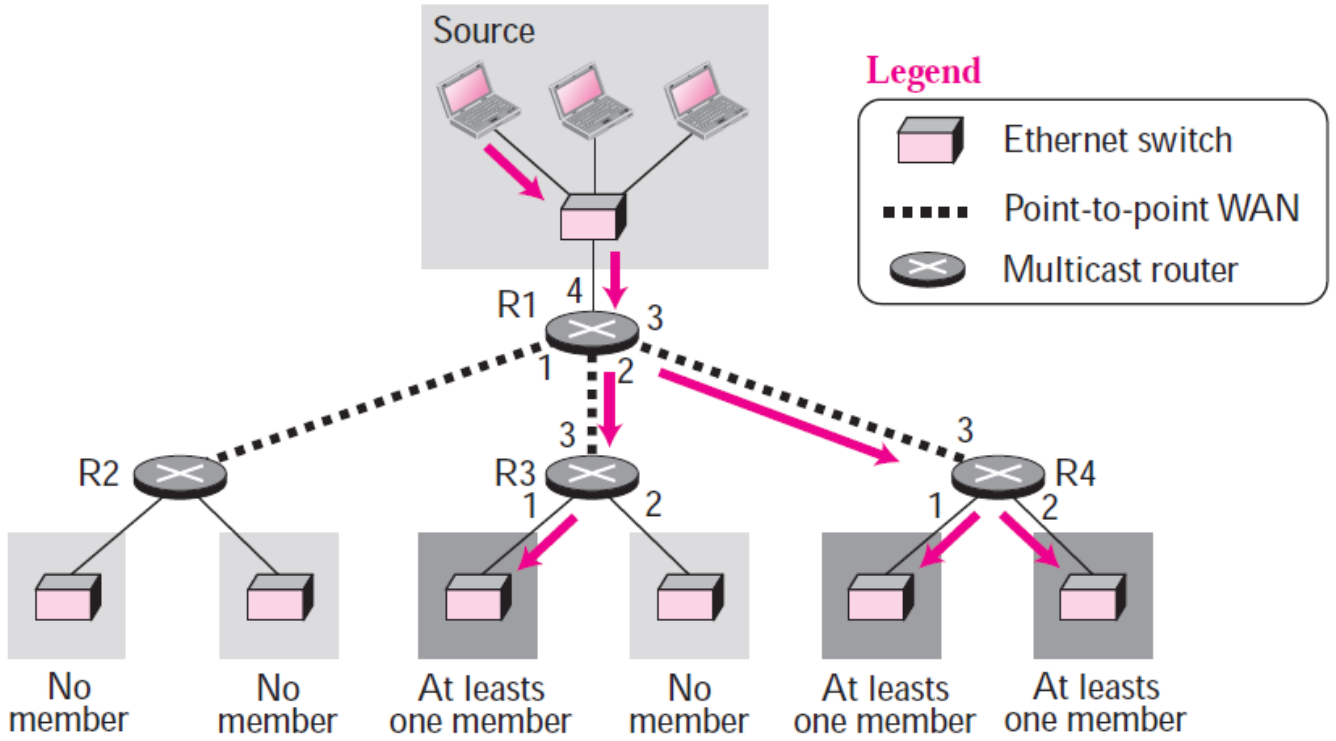
Multicast & Multicast Routing Protocols

Introduction to Multicasting Contd…

b) Multicasting

- One Source and one group of Destinations
- Relationship between Source and Destination : One to Many
- ***Source address:*** Unicast Address
- ***Destination address:*** Group of one / more Destination networks
  - At least one member interested in receiving the Multicast datagram
- Group address defines the group members
- ***Example***
  - Note: Unicast router replaced by Multicast router

- ***Routers used:*** R1, R2 & R3
- No of Ethernet Switches: 1
- ***Solution***
  - R1 forwards packets using interface 2 & 3
  - R4 forwards packets using interface 1 & 2
  - R3 forwards packet through interface 1



Multicasting – Example Scenario

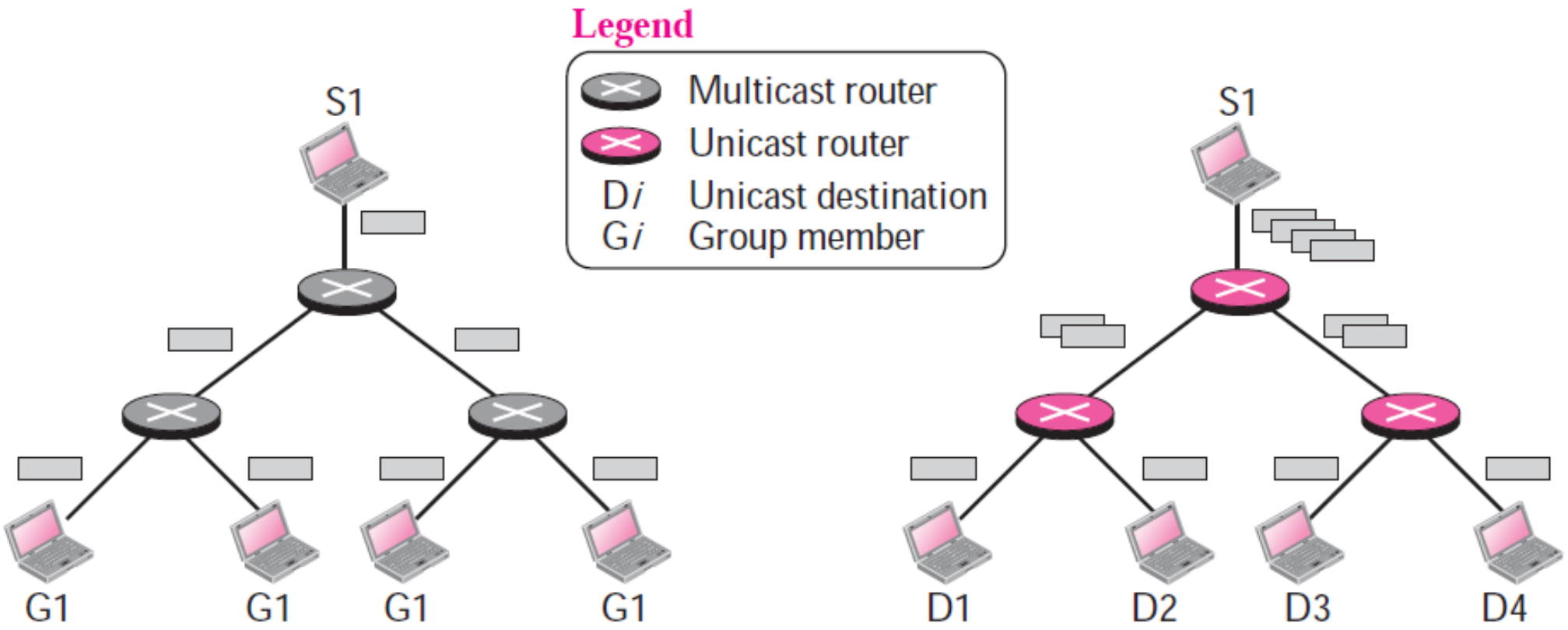
Source: Behrouz A. Forouzan, "TCP IP Protocol Suite " 4th edition, 2010, McGraw-Hill ISBN: 0073376043

Multicast & Multicast Routing Protocols

Introduction to Multicasting Contd...

c) Multicasting vs Multiple Unicasting

Multicasting	Multiple Unicasting
Single packet starts from the Source & duplicated by Routers	Multiple packets start from the Source and may have duplicates
Destination address in each packet remains the same for duplicates	Each packet has a different Unicast destination address
A single packet travels between two routers	There may be multiple copies between two routers
Example:	Example: Group E-mail



Multicasting ***Multiple Unicasting***

Multicasting vs Multiple Unicasting

Source: Behrouz A. Forouzan, "TCP IP Protocol Suite " 4th edition, 2010, McGraw-Hill ISBN: 0073376043

Multicast & Multicast Routing Protocols

Introduction to Multicasting Contd…

d) Applications of Multicasting

i. Access to distributed databases

- ❑ Information stored in more than one location

ii. Information Dissemination

- ❑ Sending same information to multiple customer

iii. Dissemination of News

iv. Teleconferencing (Permanent or Temporary group)

v. Distance Learning

- ❑ Virtual Online classes

Multicast & Multicast Routing Protocols

Introduction to Multicasting Contd…

e) Why Multicasting cannot be emulated with Unicasting?

- Multicasting is efficient than unicasting
- Requires less bandwidth than unicasting
- There is no delay in Multicasting

f) Broadcasting

- One Source and all hosts are Destinations
- Relationship between Source and Destination : One to All
- Internet does not support broadcasting explicitly
  - Due to Huge traffic & Bandwidth issues

Multicast Addresses

Multicast & Multicast Routing Protocols

Multicast Addresses in IPv4

- Destination address for a group of hosts that are part of a multicast group
- If there are no filtering mechanisms, all the recipients will receive the message broadcast through multicast
- ***Multicast Addresses in IPv4***
  - Block assigned for multicasting is 224.0.0.0/4
  - i.e., block has 2^{28} = 268,435,456 addresses
  - 224.0.0.0 to 239.255.255.255 is the address space for Multicast
  - CIDR need not be assigned to every designated range

Multicast & Multicast Routing Protocols

CIDR	*Range*	*Assignment*
224.0.0.0/24	224.0.0.0 → 224.0.0.255	Local Network Control Block
224.0.1.0/24	224.0.1.0 → 224.0.1.255	Internetwork Control Block
	224.0.2.0 → 224.0.255.255	AD HOC Block
224.1.0.0/16	224.1.0.0 → 224.1.255.255	ST Multicast Group Block
224.2.0.0/16	224.2.0.0 → 224.2.255.255	SDP/SAP Block
	224.3.0.0 → 231.255.255.255	Reserved
232.0.0.0/8	232.0.0.0 → 224.255.255.255	Source Specific Multicast (SSM)
233.0.0.0/8	233.0.0.0 → 233.255.255.255	GLOP Block
	234.0.0.0 → 238.255.255.255	Reserved
239.0.0.0/8	239.0.0.0 → 239.255.255.255	Administratively Scoped Block

Multicast Address Ranges

Source: Behrouz A. Forouzan, "TCP IP Protocol Suite " 4th edition, 2010, McGraw-Hill ISBN: 0073376043

Multicast & Multicast Routing Protocols

Multicast Addresses in IPv4 Contd…

a) Local Network Control Block (224.0.0.1/24)

- Used for Protocol control traffic & not used for general Multicast communication
- Multicast / Multicast related protocols use this block of address
- Routers are not allowed to forward IP packets with TTL set to 1
- Packet remains in the network

b) Internetwork Control Block (224.0.1/24)

- Used for Protocol control traffic
- IP packets within this address block can be forwarded over the internet
- ***Example:*** 224.0.1.1 is used by NTP protocol

Addresses in Network Control Block

Address	*Assignment*
224.0.0.0	Base address (reserved)
224.0.0.1	All systems (hosts or routers) on this network
224.0.0.2	All routers on this network
224.0.0.4	DMVRP routers
224.0.0.5	OSPF routers
224.0.0.7	ST (stream) routers
224.0.0.8	ST (stream) hosts
224.0.0.9	RIP2 routers
224.0.0.10	IGRP routers
224.0.0.11	Mobile Agents
224.0.0.12	DHCP servers
224.0.0.13	PIM routers
224.0.0.14	RSVP encapsulation
224.0.0.15	CBT routers
224.0.0.22	IGMPv3

Source: Behrouz A. Forouzan, "TCP IP Protocol Suite " 4th edition, 2010, McGraw-Hill ISBN: 0073376043

Multicast & Multicast Routing Protocols

Multicast Addresses in IPv4 Contd…

c) AD-HOC Block

- ***Range:*** 224.0.2.0 to 224.0.255.0
- Assigned to applications that does not fit in 1st or 2nd block

d) Stream Multicast Group Block

- 224.1.0.0/16
- Allocated for stream multimedia

e) SAP / SDP Block

- 224.2.0.0/16
- Used for Session Announcement protocol & Session Directory protocol

Multicast & Multicast Routing Protocols

Multicast Addresses in IPv4 Contd…

f) SSM Block

- 232.0.0.0/8
- Used for Source Specific Multicasting

g) GLOP Block

- 233..0.0.0/8
- Defines a range of globally assigned addresses
- Used inside an autonomous system (AS) and assigned to a 16 bit number
- AS number inserted as the 2-middle Octet in the block to create range of 256 multicast addresses

Multicast & Multicast Routing Protocols

Multicast Addresses in IPv4 Contd…

h) Administratively Scoped Block

- 239.0.0.0/8
- Used in a particular area in internet
- Address in this block restricted to an organization

Internet Group Management Protocol (IGMP)

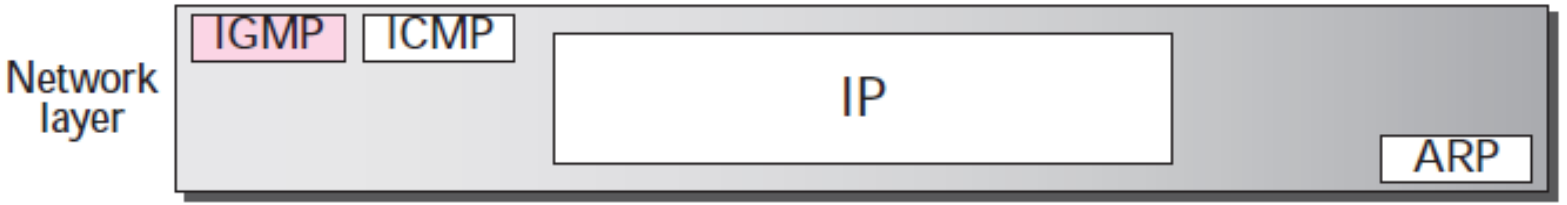
Multicast & Multicast Routing Protocols

Internet Group Management Protocol (IGMP)

- Multicast Communication: Message sent by sender to recipients of same group
  - One copy of message is copied and forwarded by multicast routers
- Multicast routers must know list of groups & minimum one loyal member related to each interface
- Information about members to be shared between multicast routers
- Information collected at two levels:
  - Locally (collected by IGMP)
  - Globally (propagated to other routers)

Multicast & Multicast Routing Protocols

Internet Group Management Protocol (IGMP) contd...



Position of IGMP in Network Layer

- IGMP collects & interprets information about group members in a network locally
- ***Note:*** IGMP is designed at the IP layer for the above said purpose

Source: Behrouz A. Forouzan, "TCP IP Protocol Suite " 4th edition, 2010, McGraw-Hill ISBN: 0073376043

Multicast & Multicast Routing Protocols

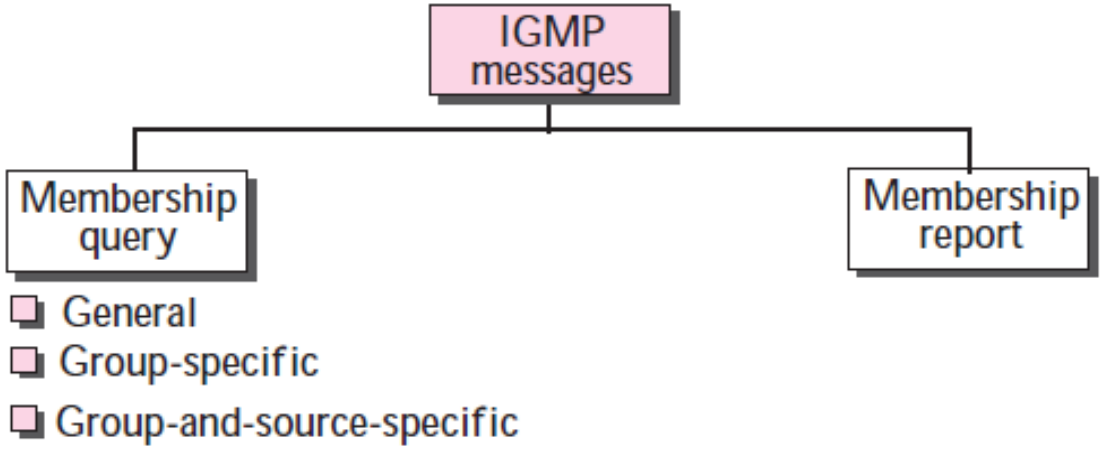
Internet Group Management Protocol (IGMP)

- ***Group Management***
- IGMP manages group membership
  - Provides information to the multicast routers about membership status of routers connected to a network
  - Maintains a list of groups in the network which has at least one loyal member
  - Without IGMP, traffic increases & more bandwidth is consumed
- ***Versions of IGMP:*** 1,2 & 3
  - Version 1 & 2 provides Any Source Multicast (ASM)
  - Version 3 provides Source Specific Multicast (SSM)

Multicast & Multicast Routing Protocols

Internet Group Management Protocol (IGMP)

- ***IGMPv3 Messages***
- Two types of messages
  a) Membership Query message
     i. General
     ii. Group Specific
     iii. Group and Source Specific
  b) Membership Report Message



IGMP Messages

Source: Behrouz A. Forouzan, "TCP IP Protocol Suite " 4th edition, 2010, McGraw-Hill ISBN: 0073376043

Multicast & Multicast Routing Protocols

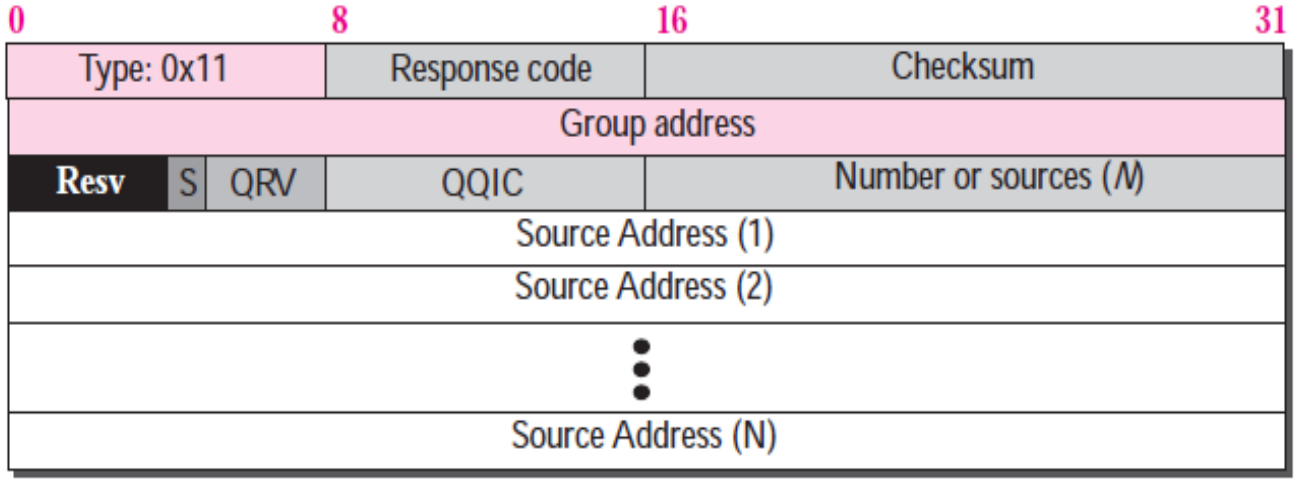
Internet Group Management Protocol (IGMP)

- ***IGMPv3 Messages Contd…***

a) Membership Query message

i. General Query Message

- Router probes each neighbor to report the whole group membership list



Membership Query Message Format

ii. Group Specific Query Message

- Router probes each neighbor to report if it is still interested in a specific group
- ***Multicast group address*** defined as *x.y.z.t* in group address field

Multicast & Multicast Routing Protocols

Internet Group Management Protocol (IGMP)

- ***IGMPv3 Messages Contd…***

iii. Group and Source Specific Query message

- Router probes each neighbor to report it is still in a specific multicast group

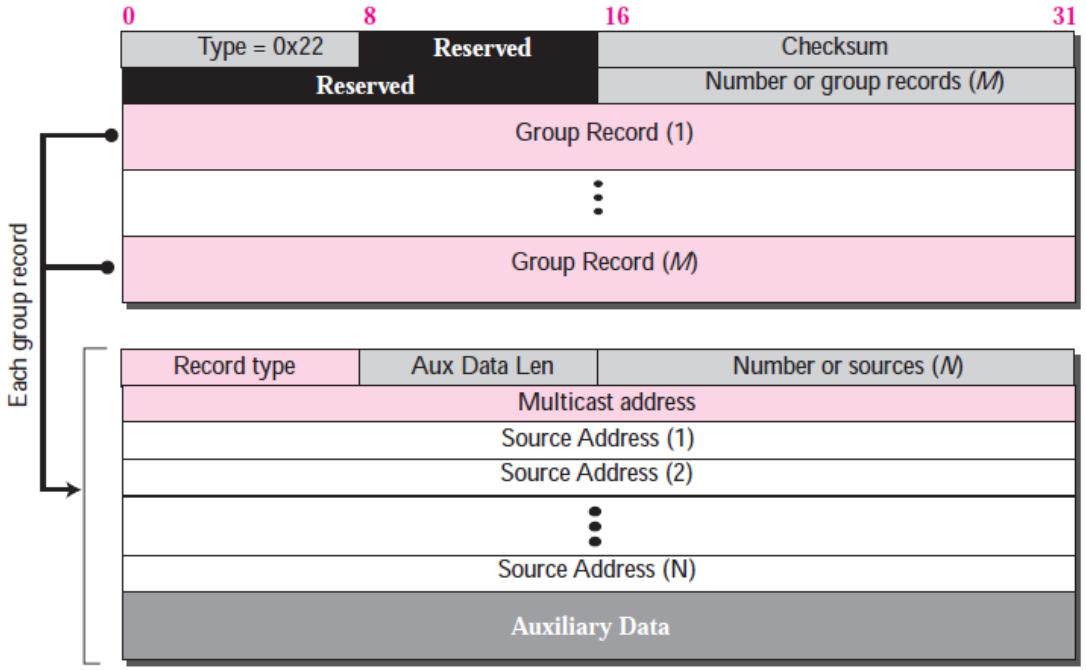
b) Membership Query message format

- Important fields
  - Checksum
  - Number of Sources
  - Aux data
  - Aux Data Len

Multicast & Multicast Routing Protocols

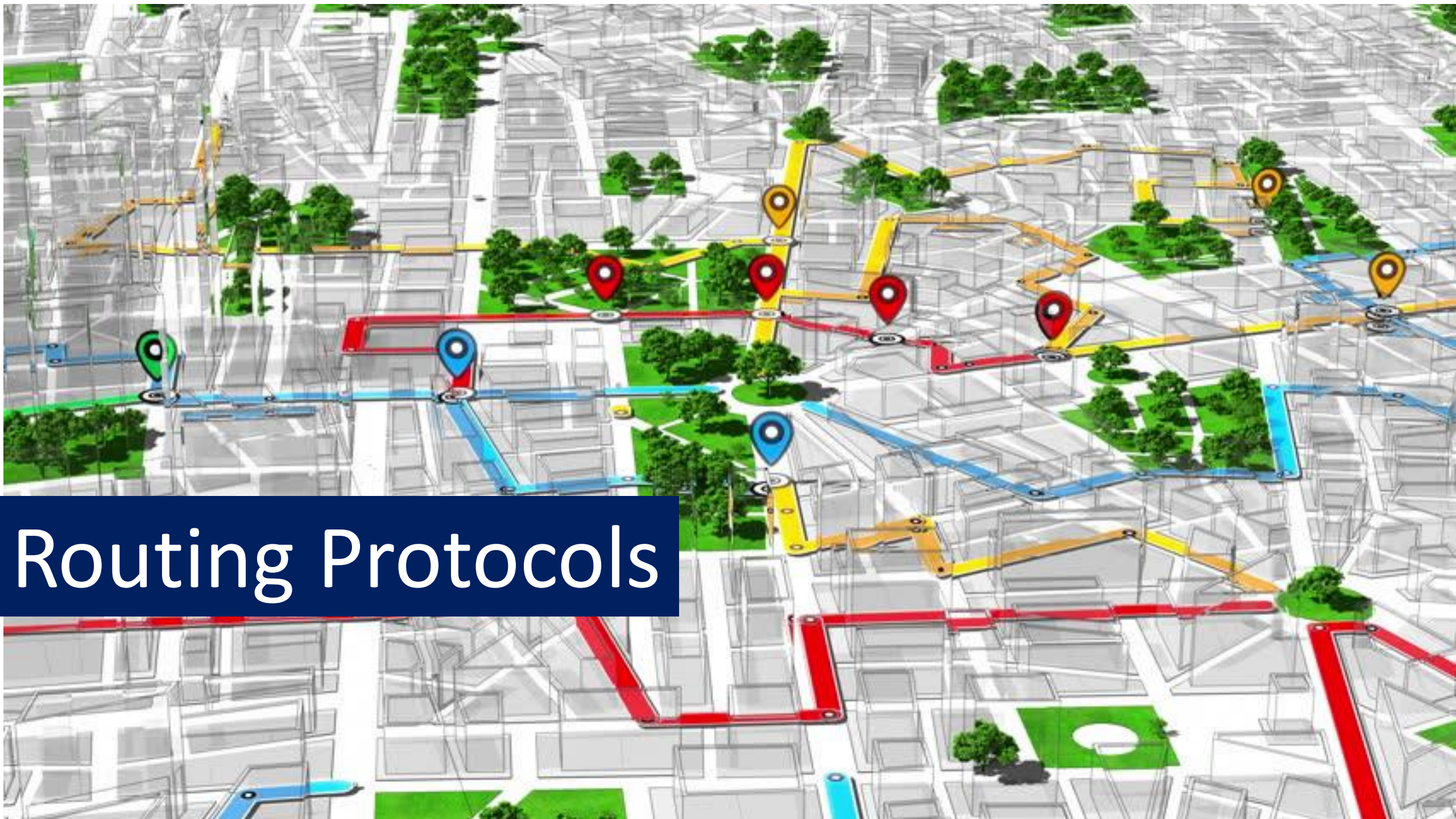
Internet Group Management Protocol (IGMP)

➢ IGMPv3 Messages Contd…



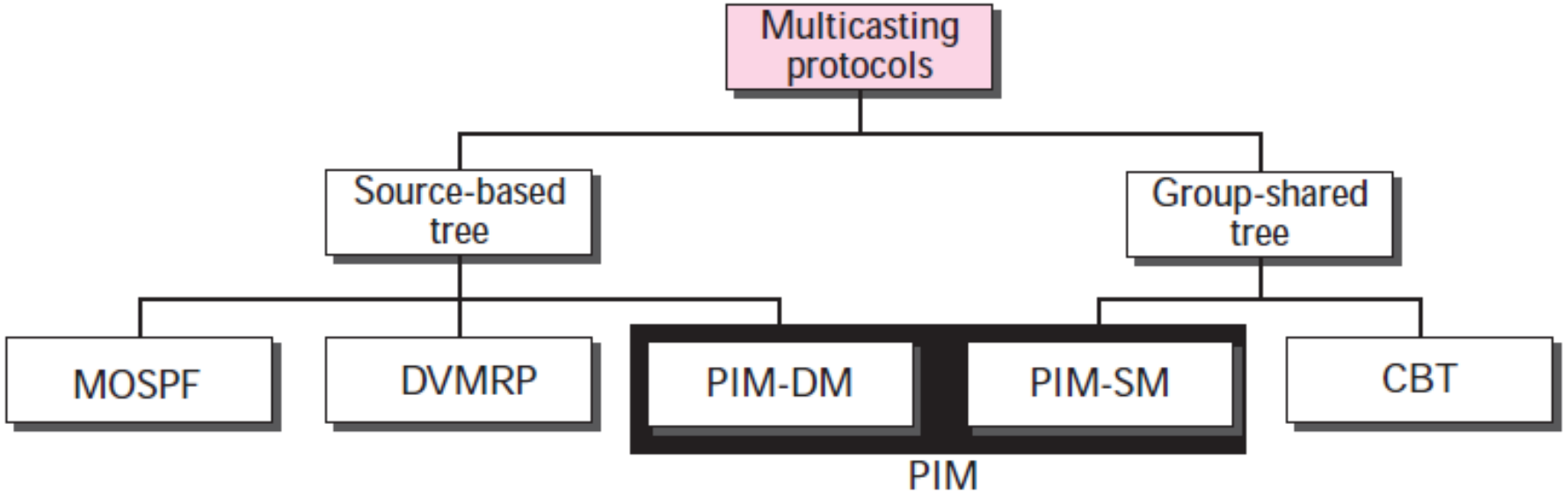
Membership Report Message Format

Source: Behrouz A. Forouzan, "TCP IP Protocol Suite " 4th edition, 2010, McGraw-Hill ISBN: 0073376043


Routing Protocols

Multicast & Multicast Routing Protocols

Routing Protocols



Taxonomy of Common Multicast Protocols

Source: Behrouz A. Forouzan, "TCP IP Protocol Suite " 4th edition, 2010, McGraw-Hill ISBN: 0073376043

Multicast & Multicast Routing Protocols

Routing Protocols Contd…

Multicast Link state routing

- Extension of unicast routing
- Uses source based tree approach
- Node advertises every group that has any loyal member on the link
- Group information comes form IGMP
- Router creates ‘n’ groups based on Link state Packets (LSP’s)
- Note: Each router has a routing table that has ‘n’ shortest path trees
- Disadvantage: More Time & Space needed to create & Store shortest path trees

Multicast & Multicast Routing Protocols

Routing Protocols Contd…

a) Multicast Open Shortest path first : MOSPF

- Data driven protocol & Extension of OSPF
- Uses multicast link state routing to create source based trees
- Group membership LSA includes hosts in the tree that's belongs to a group
- When MOSPF router encounters a datagram with a given source and group address
  - Router constructs the Dijkstra shortest path tree

Multicast & Multicast Routing Protocols

Routing Protocols Contd…

b) Multicast Distance Vector Routing Protocol

- Uses source-based trees but router does not make a routing table
- Router receives multicast packet and forwards
- After forwarding the packet the table is destroyed
- MDV uses a process based on 4 decision making strategies
  i. Flooding
  ii. Reverse path forwarding (RPF)
  iii. Reverse Path Broadcasting (RPB)
  iv. Reverse path Multicasting (RPM)

Multicast & Multicast Routing Protocols

Routing Protocols Contd…

b) Multicast Distance Vector Routing Protocol Contd…

i. Flooding

- Router forwards received packets and sends to every interface except itself
- Destination group address is not checked
- Every network with active members receives the packet
- Networks without active members also receive the packet
- Flooding supports broadcast not unicast
- ***Disadvantage:*** Creation of loops
  - Overcome by Reverse Path Forwarding (RPF)

Multicast & Multicast Routing Protocols

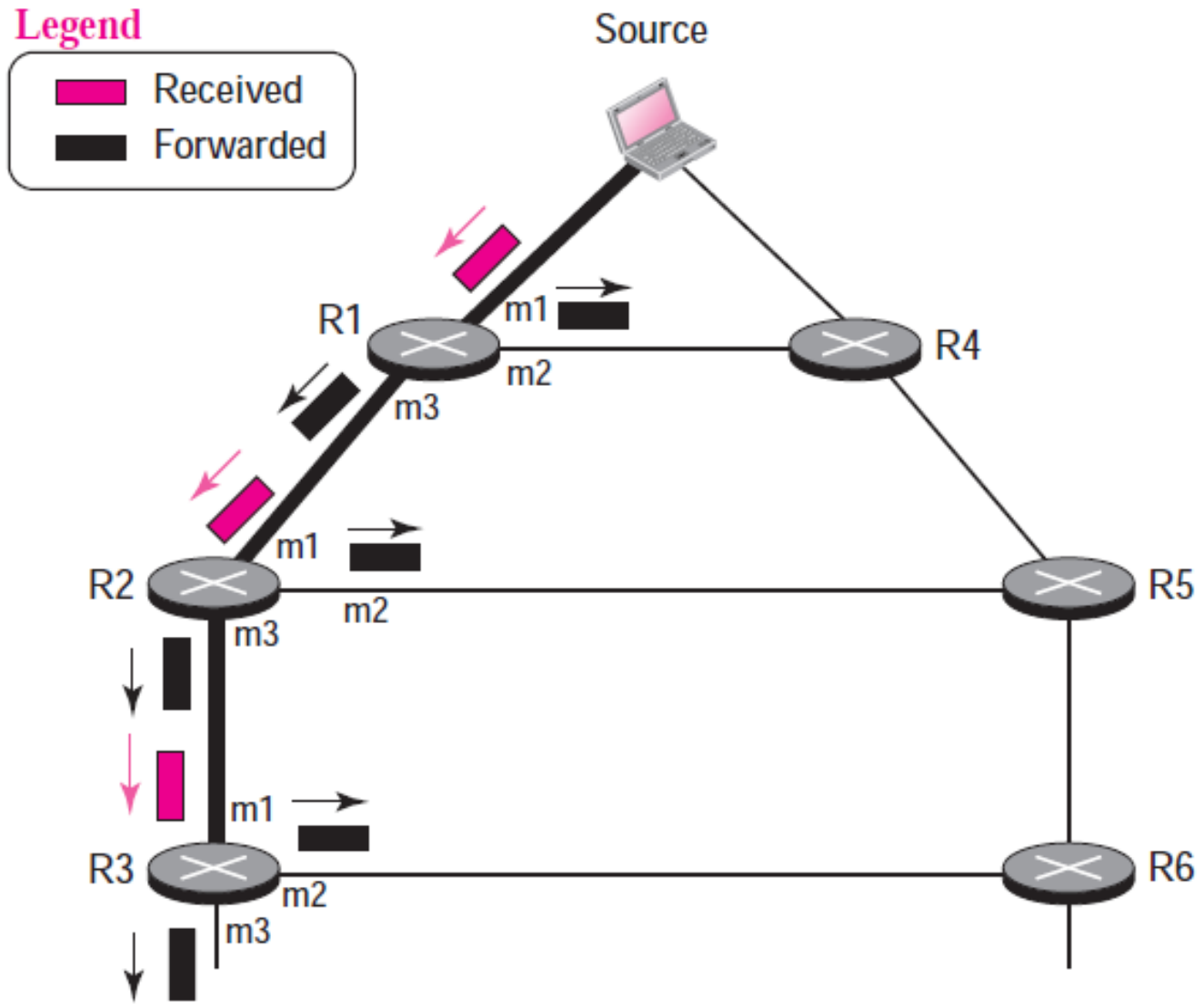
Routing Protocols Contd…

b) Multicast Distance Vector Routing Protocol Contd…

ii. Reverse Path Forwarding (RPF)

- Modified flooding strategy: Only one copy is forwarded to avoid flooding
  - Router forwards the copy that has travelled the shortest path from Source to router
- RPF uses Unicast routing table to find this packet
- Looping is prevented, since there is only one shortest path from Source to Router
- ***Disadvantage:*** No guarantee that each network receives only one copy; Network may receive multiple copies

- Routers: R1, R2, R3, R4, R5 & R6
- Shortest Path : R1, R2 & R2
- R1 receives packet from source through interface m1
  - Packet forwarded
- Packet forwarded to R2 and R3 similarly
- Note: Upstream routers towards source always discards packet not gone through the shortest path



Reverse Path Forwarding (RPF)

Source: Behrouz A. Forouzan, "TCP IP Protocol Suite " 4th edition, 2010, McGraw-Hill ISBN: 0073376043

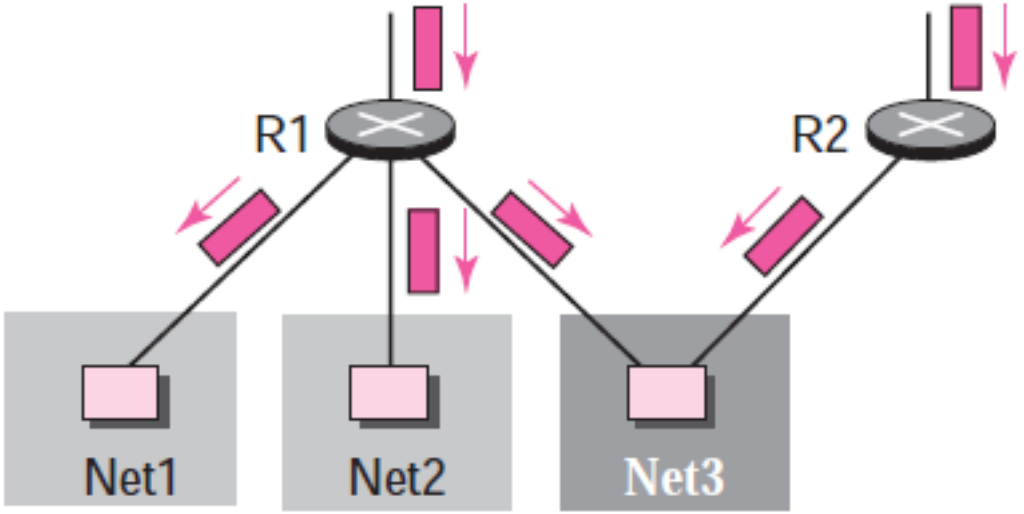
Multicast & Multicast Routing Protocols

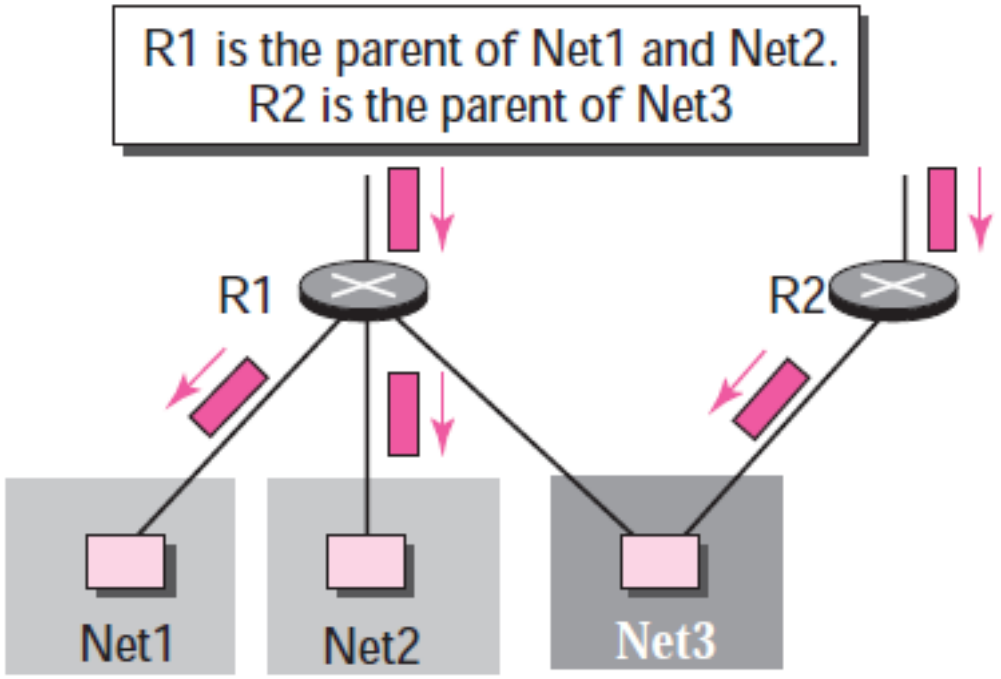
Routing Protocols Contd…

b) Multicast Distance Vector Routing Protocol Contd…

iii. Reverse Path Broadcasting (RPB)

- Problem in RPF is overcome by eliminating duplication
- Restriction used in RPB: Only one parent router defined for each network
- Policy: For each source, router sends packet only out of interfaces which is the designated parent
- RPB guarantees the elimination of duplicate copies
- ***Designated Parent:*** Router with the shortest path to the source
- ***Disadvantage:*** RPB broadcasts, not multicasts





Reverse Path Forwarding (RPF)

Reverse Path Broadcasting (RPB)

Difference between RPF and RPB

Source: Behrouz A. Forouzan, "TCP IP Protocol Suite " 4th edition, 2010, McGraw-Hill ISBN: 0073376043

Multicast & Multicast Routing Protocols

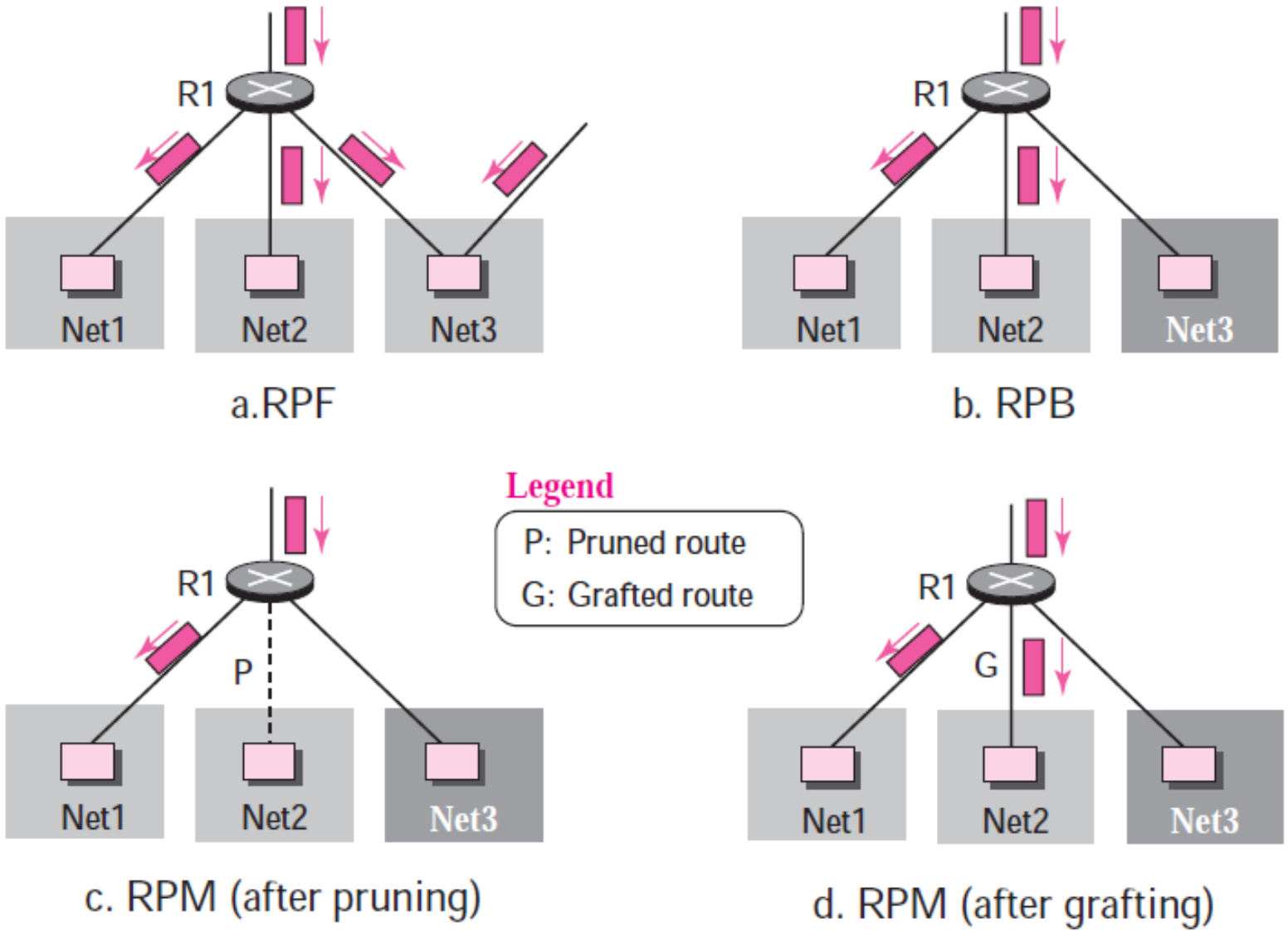
Routing Protocols Contd…

b) Multicast Distance Vector Routing Protocol Contd…

iv. Reverse Path Multicasting (RPM)

- Multicast packet must reach networks with active members for the particular group only
- To convert broadcast to multicast: use Pruning & Grafting
- ***Pruning:*** Prune message sent to upstream router to prune corresponding interface
- ***Grafting:*** Graft message forces upstream router to resume sending multicast images

RPF, RPB & RPM



Source: Behrouz A. Forouzan, "TCP IP Protocol Suite " 4th edition, 2010, McGraw-Hill ISBN: 0073376043

Multicast & Multicast Routing Protocols

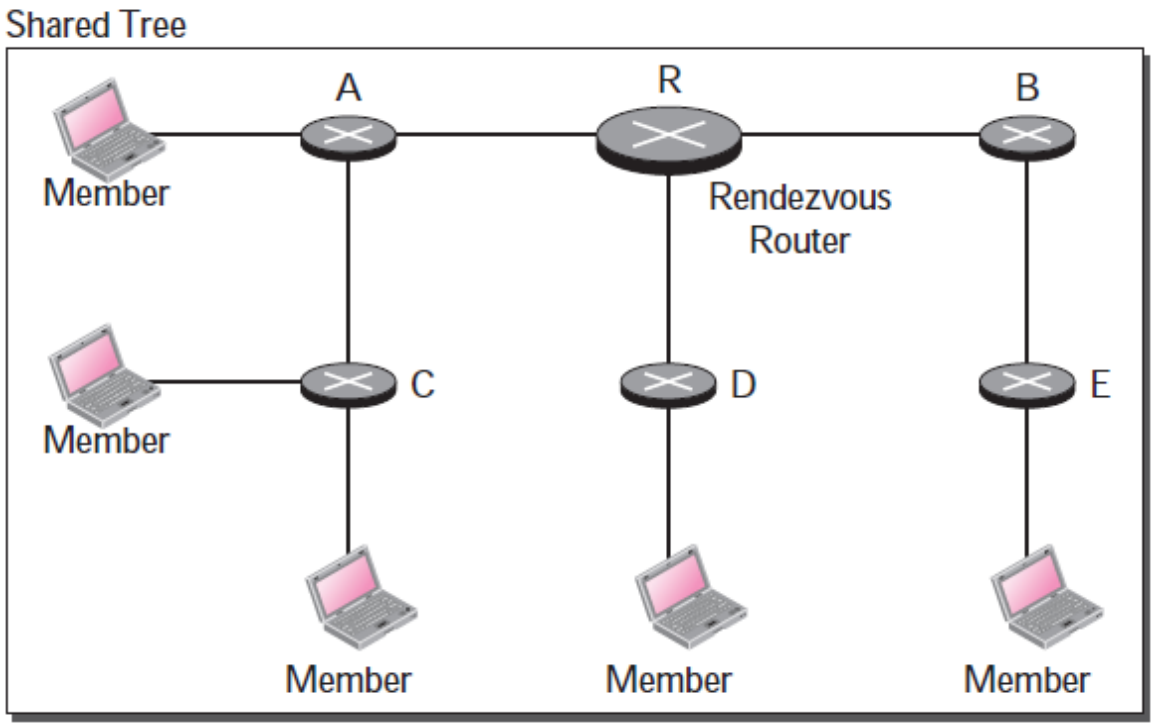
Routing Protocols Contd…

c) Core-Based Tree Protocol

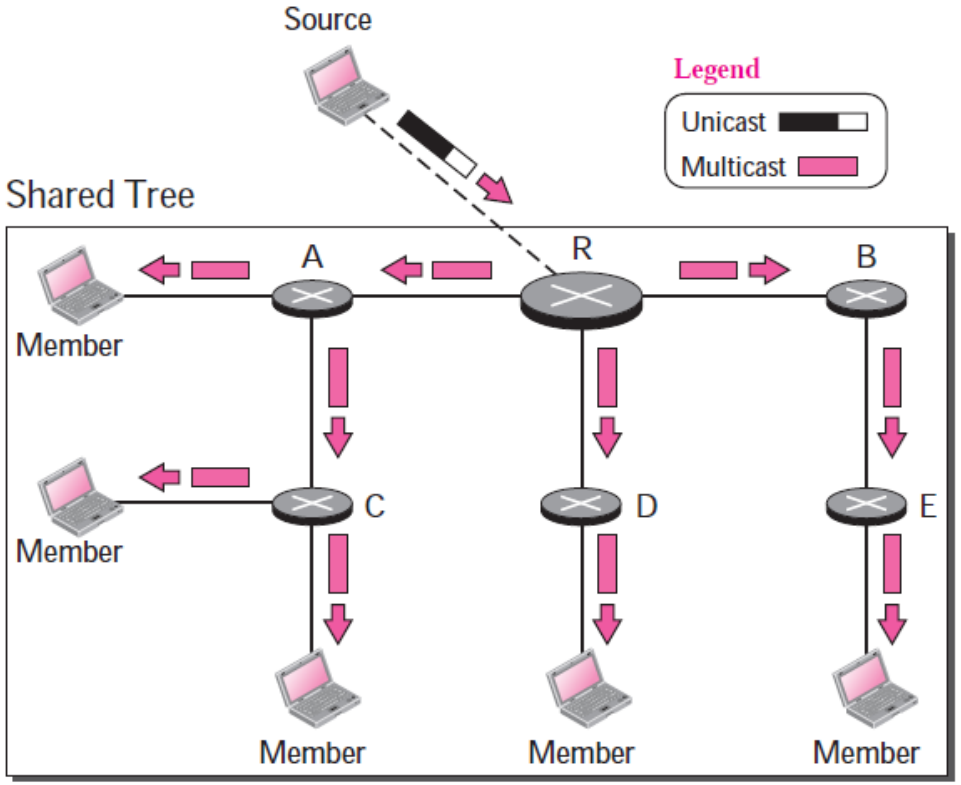
- Group-shared tree, center-based protocol; Uses one tree per group
- One of the routers in the tree is called the core
- ***Procedure for sending Packet from Source to Group members:***

1) Source encapsulates the multicast packet inside a unicast packet with the unicast destination address of the core and sends it to the core
   - This part of delivery is done using a unicast address
   - The only recipient is the core router
2) Core decapsulates the unicast packet and forwards it to all interested interfaces

3) Each router that receives the multicast packet, in turn, forwards it to all interested interfaces



Group-shared Tree with Rendezvous Router



Sending a Multicast Packet to the Router

Source: Behrouz A. Forouzan, "TCP IP Protocol Suite " 4th edition, 2010, McGraw-Hill ISBN: 0073376043

Multicast & Multicast Routing Protocols

Routing Protocols Contd…

d) Protocol Independent Multicast (PIM)

- Consists of two protocols

i. Protocol Independent Multicast, Dense Mode (PIM-DM)

- Source-based tree routing protocol
  - Uses RPF & Pruning / Grafting strategies for Multicasting
- Used when each router in involved in Multicasting
- Used in dense environment : LAN
- PIM-DM can be either be a Distance Vector Protocol (RIP) (or) Link State Protocol (OSPF)

Multicast & Multicast Routing Protocols

Routing Protocols Contd…

d) Protocol Independent Multicast (PIM) Contd…

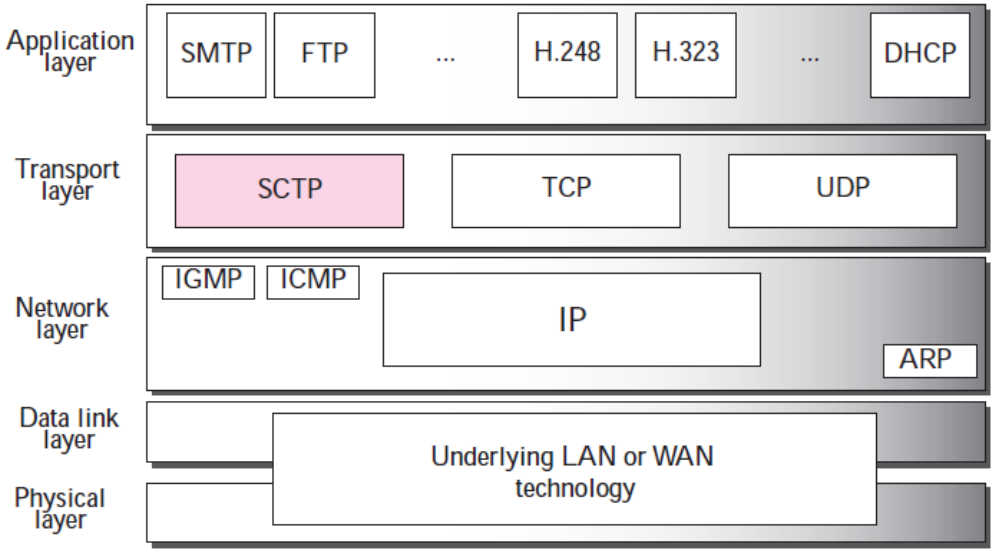
ii. Protocol Independent Multicast, Sparse Mode (PIM-SM)

- Group-shared tree routing protocol; Similar to CBT
- Simple to implement
- Rendezvous point at the Source of Tree
- No Acknowledgement required from a join message
- ***Characteristic:*** Can switch between Group shared (or) Source based Strategy

Stream Control Transmission Protocol

Introduction to SCTP

- Reliable, Message-oriented general purpose transport layer protocol
- Capable of handling Multimedia and Stream traffic
- Designed for Internet applications
- Lies between application & network layer
- Servers a a middleman between application programs & network operations
- Combines best features of UDP & TCP

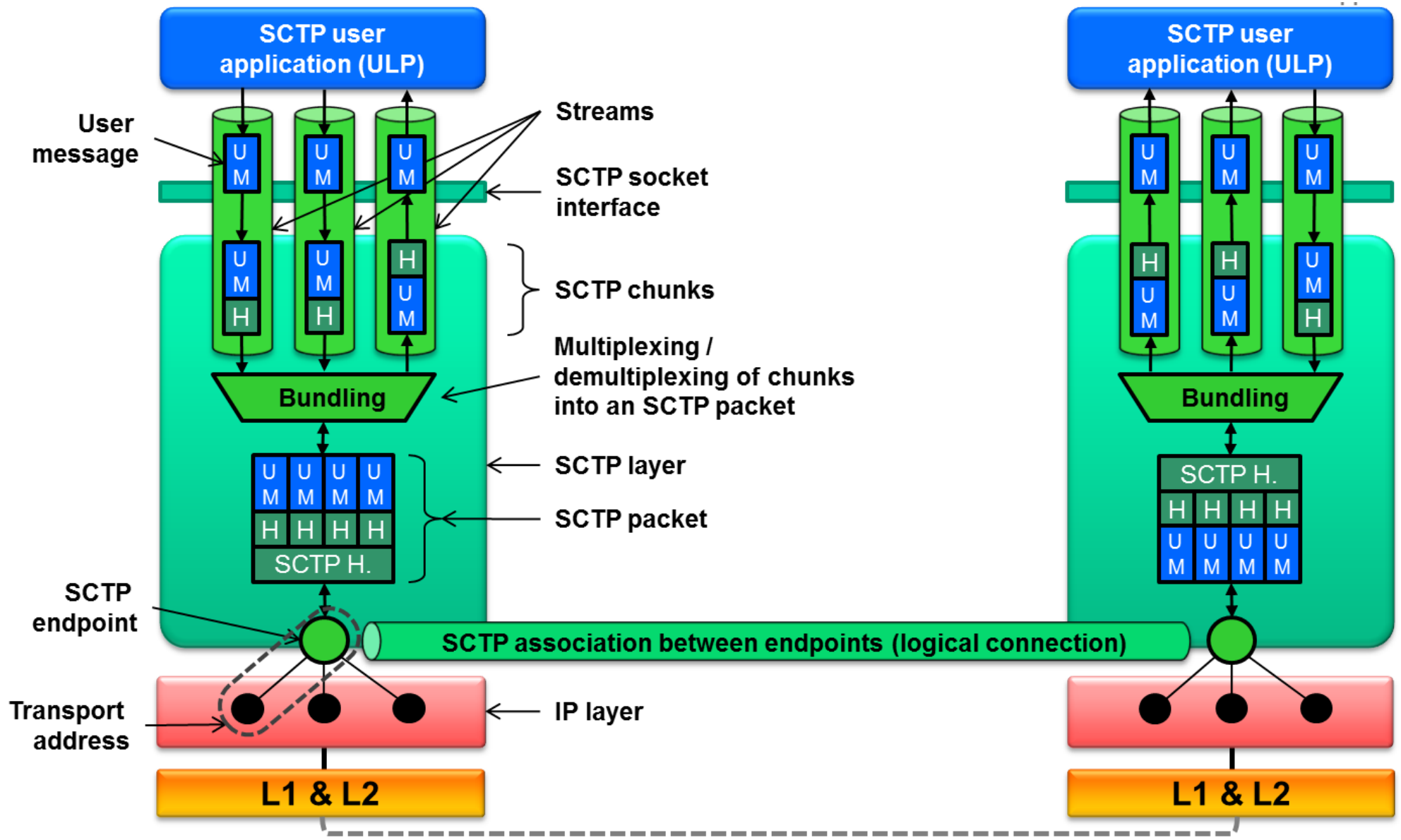


Relationship of SCTP to other Protocols in TCP/IP Suite

Stream Control Transmission Protocol

Introduction to SCTP Contd...

- SCTP preserves message boundaries
- Detects lost data, duplicate data and out-of-order data
- Congestion control and flow control mechanisms available in SCTP
- ***Characteristics of SCTP***
  a) Confirmed Transmission
  b) Data Fragmentation
  c) Sequenced Delivery
  d) Bunding
  e) Fault tolerance at Network Level



Conceptual Model of SCTP

Source: https://indigoo.com/petersblog/?p=185

Stream Control Transmission Protocol

Introduction to SCTP Contd…

- ***Key features of SCTP***

  a) No head-of-line blocking

  b) Message-based data transfer

  c) Multihoming

  d) Protection against connection flooding

Source: https://indigoo.com/petersblog/?p=185

Stream Control Transmission Protocol

Difference between TCP, UDP & SCTP

	TCP	UDP	SCTP
Reliability	trustworthy	Unreliable	Trustworthy
Connection type	Connection-oriented	Connectionless	Connection-oriented
Transmission type	Byte-oriented	News-oriented	News-oriented
Transfer sequence	Strictly ordered	Disordered	Partially ordered
Overload control	Yes	No	Yes
Error tolerance	No	No	Yes

Source: https://www.ionos.com/digitalguide/server/know-how/sctp/

References

Learning Resources Used for Unit I

1) Behrouz A. Forouzan, “TCP IP Protocol Suite ” 4th edition, 2010, McGraw-HillISBN: 0073376043
2) https://indigoo.com/petersblog/?p=185
3) https://www.ionos.com/digitalguide/server/know-how/sctp/

Thank You