Unit-5

Transport Layer

Mr. Atul Pawar , PCCOE, Pune

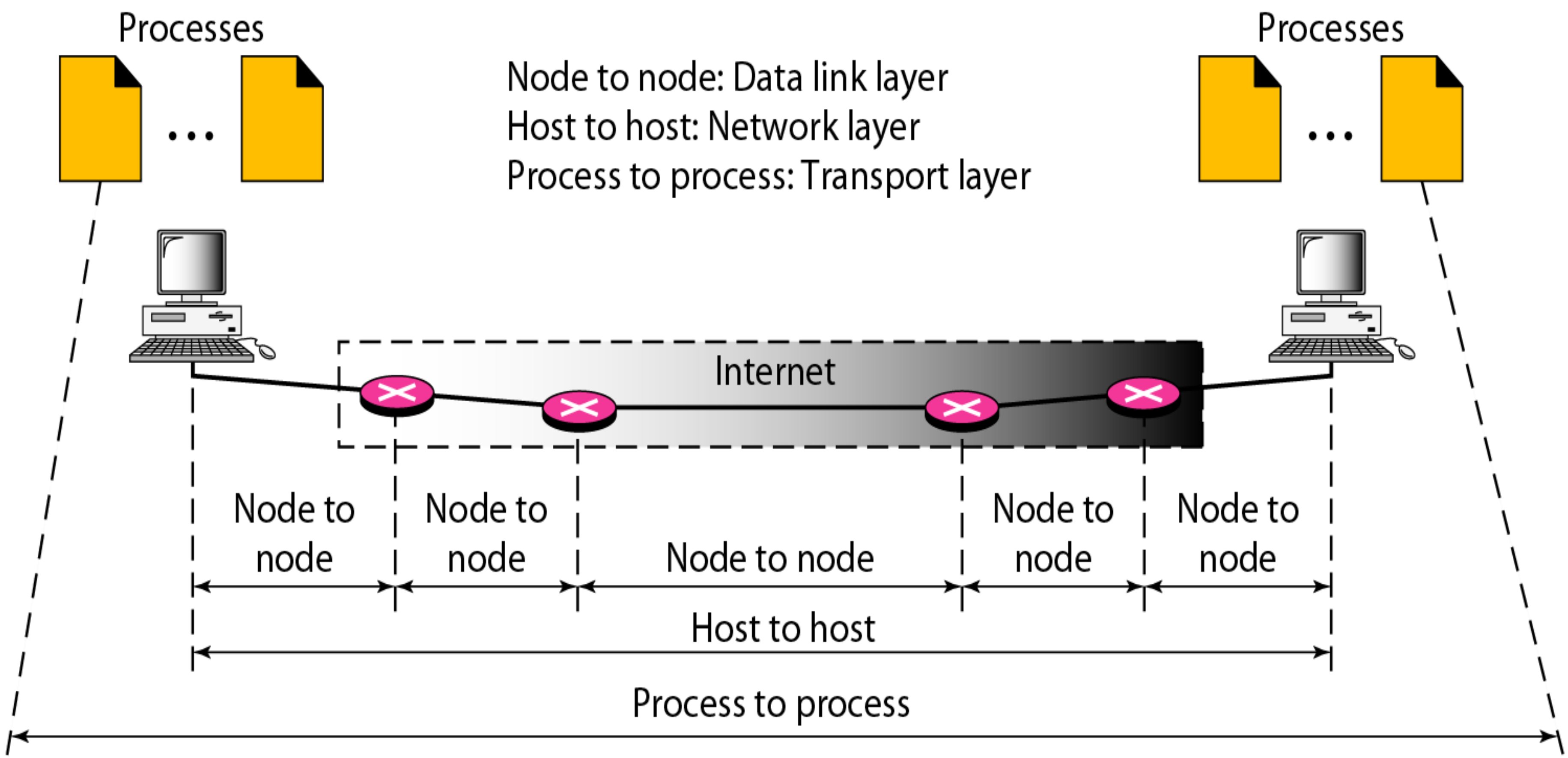
PROCESS-TO-PROCESS DELIVERY

The transport layer is responsible for process-to-process delivery—the delivery of a packet, part of a message, from one process to another. Two processes communicate in a client/server relationship.

Note

The transport layer is responsible for process-to-process delivery.

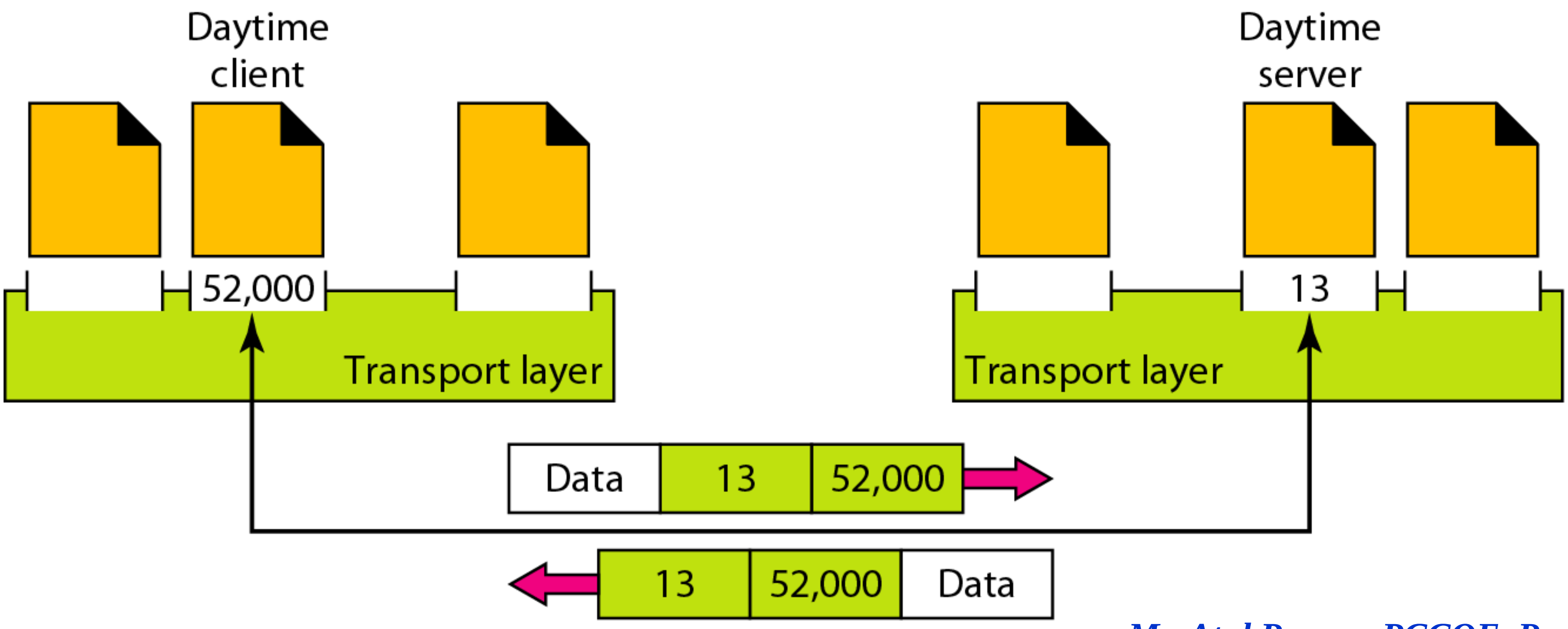
Types of data deliveries



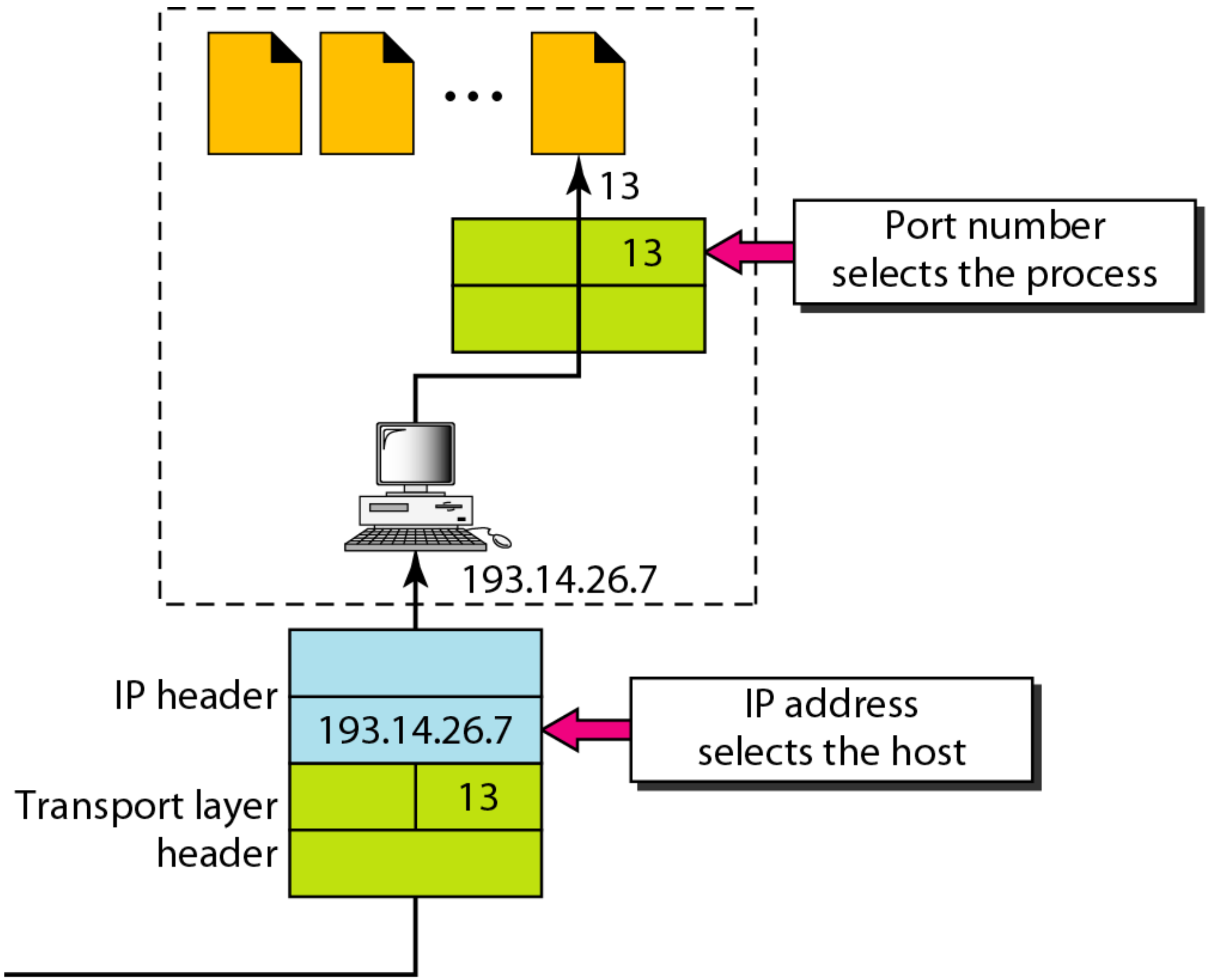
Addressing:

- The Internet has decide to use universal port numbers for servers, these are called **well-known (Permanent) port numbers.**
- The client program defines itself with a port number, chosen Randomly by the transport layer software running on the client host is called **Ephemeral (Temporary) port number.**

Port numbers

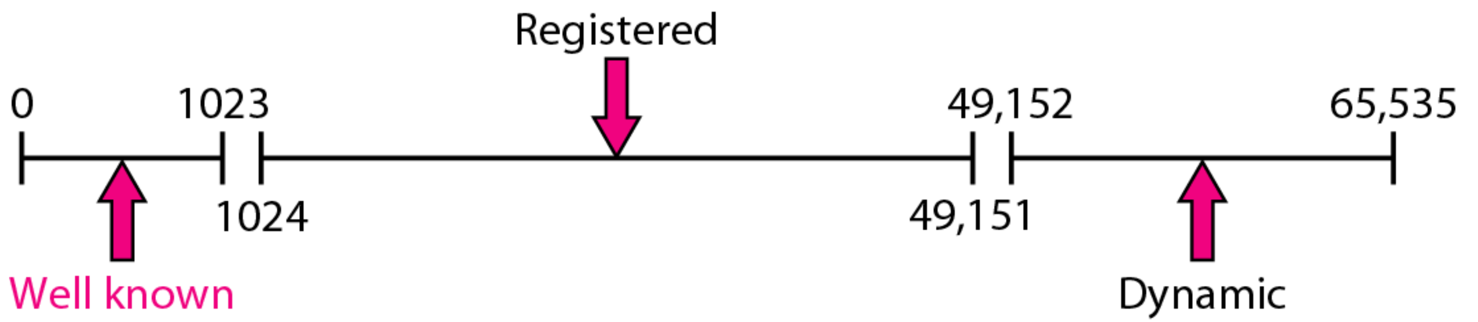


IP addresses versus port numbers



IANA ranges (Internet Assigned Number Authority)

- Port numbers are of 16 bit integers between 0-65,535.
- This port numbers are divided into 3 ranges:



Socket address

- Socket address = IP address + Port number
- Transport layer protocol required Client socket and Server socket.
- The IP header contains the IP addresses; UDP and TCP header contains the Port numbers.

Connectionless Versus Connection-Oriented Service :

A transport layer protocol can either be connectionless or connection-oriented.

Connectionless Service :

In a connectionless service, the packets are sent from one party to another with no need for connection establishment or connection release.

The packets aremay be delayed or lost or may arrive out of sequence.

There is no acknowledgment either.

Eg. UDP, is connectionless.

Connection~Oriented Service :

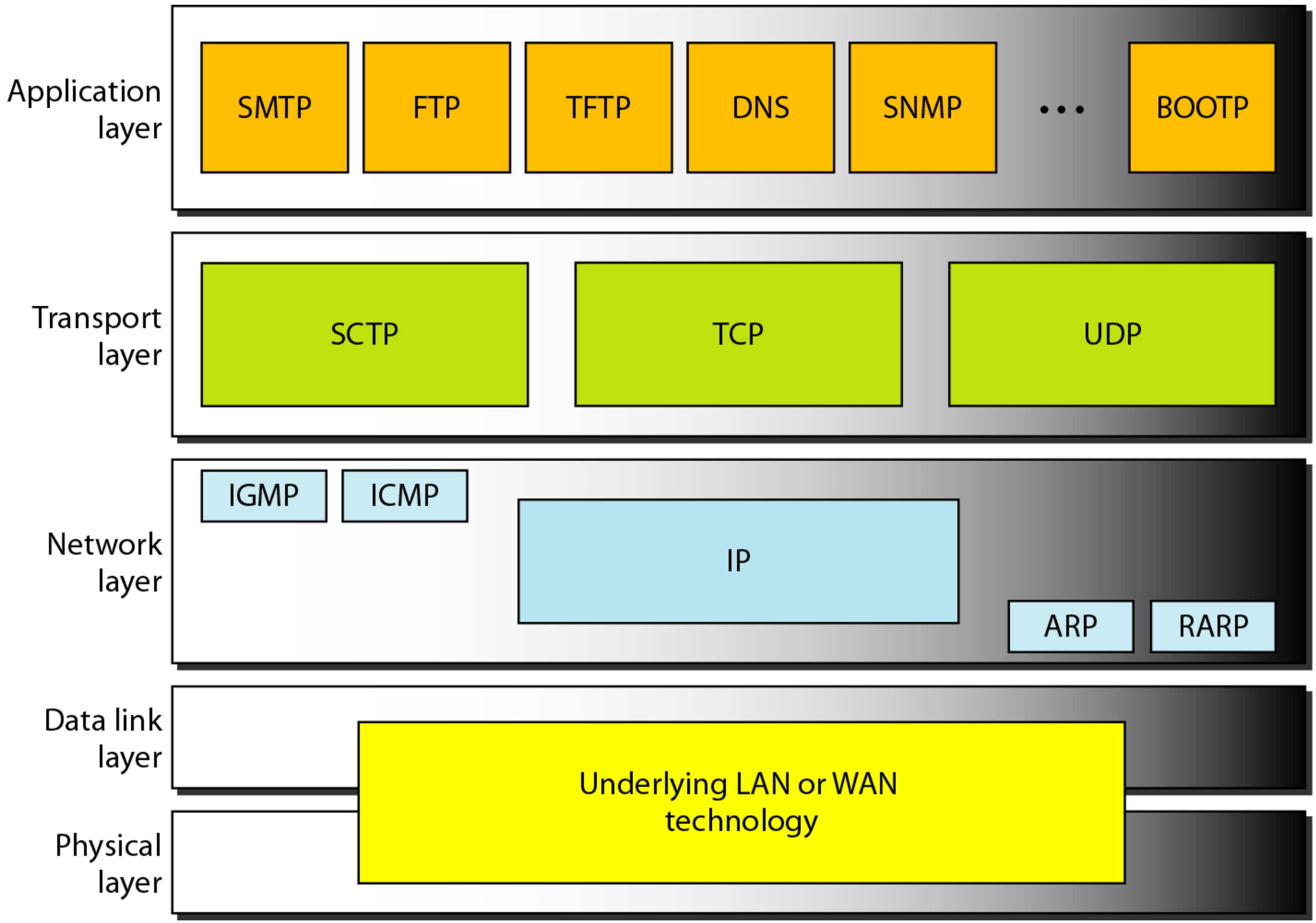
In a connection-oriented service, a connection is first established between the sender and the receiver.

Data are transferred.

At the end, the connection is released.

Eg. TCP and SCTP are connection-oriented protocols.

Position of UDP, TCP, and SCTP in TCP/IP suite



Services Provided to the Upper Layers :

- The ultimate goal of the transport layer is to provide efficient, reliable, and cost-effective data transmission service to its users, normally processes in the application layer.
- To achieve this, the transport layer makes use of the services provided by the network layer.
- The software and/or hardware within the transport layer that does the work is called the transport entity.two types of transport service.
- The connection-oriented transport service is similar to the connection-oriented network service in many ways.
- In both cases, connections have three phases: establishment, data transfer, and release.

Transport Service Primitives

To allow users to access the transport service, the transport layer must provide some operations to application programs, that is, a transport service interface.
Each transport service has its own interface.

Primitive	Packet sent	Meaning
LISTEN	(none)	Block until some process tries to connect
CONNECT	CONNECTION REQ.	Actively attempt to establish a connection
SEND	DATA	Send information
RECEIVE	(none)	Block until a DATA packet arrives
DISCONNECT	DISCONNECTION REQ.	Request a release of the connection

Berkeley Sockets

- Sockets were first released as part of the Berkeley UNIX 4.2BSD software distribution in 1983.
- The primitives are now widely used for Internet programming on many operating systems, especially UNIX -based systems, and there is a socket-style API for Windows called “winsock.”

Primitive	Meaning
SOCKET	Create a new communication endpoint
BIND	Associate a local address with a socket
LISTEN	Announce willingness to accept connections; give queue size
ACCEPT	Passively establish an incoming connection
CONNECT	Actively attempt to establish a connection
SEND	Send some data over the connection
RECEIVE	Receive some data from the connection
CLOSE	Release the connection

Addressing

- When an application process wishes to set up a connection to a remote application process, it must specify which one to connect to.
- Connectionless transport has the same problem: to whom should each message be sent?
- The method normally used is to define transport addresses to which processes can listen for connection requests.
- In the Internet, these endpoints are called ports.
- Which is also called as TSAP (Transport Service Access Point) to mean a specific endpoint in the transport layer.
- In the network layer (i.e., network layer addresses) are also called NSAPs (Network Service Access Points).
- IP addresses are examples of NSAPs.

A possible scenario for a transport connection is as follows:

1. A mail server process attaches itself to TSAP 1522 on host 2 to wait for an incoming call. How a process attaches itself to a TSAP is outside the networking model and depends entirely on the local operating system. A call such as our LISTEN might be used, for example.
2. An application process on host 1 wants to send an email message, so it attaches itself to TSAP 1208 and issues a CONNECT request. The request specifies TSAP 1208 on host 1 as the source and TSAP 1522 on host 2 as the destination. This action ultimately results in a transport connection being established between the application process and the server.
3. The application process sends over the mail message.
4. The mail server responds to say that it will deliver the message.
5. The transport connection is released.

Connection Establishment

- Tomlinson (1975) introduced the three-way handshake.
- This establishment protocol involves one peer checking with the other that the connection request is indeed current.

Case : 1

- Host 1 chooses a sequence number, x, and sends a CONNECTION REQUEST segment containing it to host 2.
- Host 2 replies with an ACK segment acknowledging x and announcing its own initial sequence number, y.
- Finally, host 1 acknowledges host 2’s choice of an initial sequence number in the first data segment that it sends.

Case : 2

- Now let us see how the three-way handshake works in the presence of delayed duplicate control segments.
- The first segment is a delayed duplicate CONNECTION REQUEST from an old connection.
- This segment arrives at host 2 without host 1's knowledge.
- Host 2 reacts to this segment by sending host 1 an ACK segment, in effect asking for verification that host 1 was indeed trying to set up a new connection.
- When host 1 rejects host 2's attempt to establish a connection, host 2 realizes that it was tricked by a delayed duplicate and abandons the connection.
- In this way, a delayed duplicate does no damage.

Connection Release

There are two styles of terminating a connection:

- asymmetric release and symmetric release.

- Asymmetric release is the way the telephone system works: when one party hangs up, the connection is broken.

- Asymmetric release is abrupt and may result in data loss.

- Symmetric release treats the connection as two separate unidirectional connections and requires each one to be released separately.

- Symmetric release, in which each direction is released independently of the other one.

- Here, a host can continue to receive data even after it has sent a DISCONNECT segment.

Four protocol scenarios for releasing a connection.

Case:1

- In this case one of the users sends a DR (DISCONNECTION REQUEST) segment to initiate the connection release.
- When it arrives, the recipient sends back a DR segment and starts a timer, just in case its DR is lost.
- When this DR arrives, the original sender sends back an ACK segment and releases the connection.
- Finally, when the ACK segment arrives, the receiver also releases the connection.
- Releasing a connection means that the transport entity removes the information about the connection from its table of currently open connections and signals the connection’s owner (the transport user) somehow.

Case : 2

- If the final ACK segment is lost, the situation is saved by the timer.
- When the timer expires, the connection is released anyway.
- Now consider the case of the second DR being lost.
- The user initiating the disconnection will not receive the expected response, will time out, and will start all over again.

Case : 3

- Assuming that the second time no segments are lost and all segments are delivered correctly and on time.

Case : 4

- It is the same as case 3 except that now we assume all the repeated attempts to retransmit the DR also fail due to lost segments.
- After N retries, the sender just gives up and releases the connection.
- Meanwhile, the receiver times out and also exits.
- While this protocol usually suffices, in theory it can fail if the initial DR and N retransmissions are all lost.
- The sender will give up and release the connection, while the other side knows nothing at all about the attempts to disconnect and is still fully active.
- This situation results in a half-open connection.

Error Control, Flow Control & Buffering

- Error control is ensuring that the data is delivered with the desired level of reliability, usually that all of the data is delivered without any errors.
- Flow control is keeping a fast transmitter from overrunning a slow receiver.
- The link layer checksum protects a frame while it crosses a single link.
- The transport layer checksum protects a segment while it crosses an entire network path.
- It is an end-to-end check, which is not the same as having a check on every link.
- Saltzer et al. (1984) describe a situation in which packets were corrupted inside a router.
- The link layer checksums protected the packets only while they traveled across a link, not while they were inside the router.
- Thus, packets were delivered incorrectly even though they were correct according to the checks on every link.

- Transport protocols generally use larger sliding windows.
- The buffers are needed at both the sender and the receiver.
- Certainly they are needed at the sender to hold all transmitted but as yet unacknowledged segments.
- They are needed there because these segments may be lost and need to be retransmitted.
- However, since the sender is buffering, the receiver may or may not dedicate specific buffers to specific connections, as it sees fit.
- The receiver may, for example, maintain a single buffer pool shared by all connections.
- When a segment comes in, an attempt is made to dynamically acquire a new buffer.
- If one is available, the segment is accepted; otherwise, it is discarded.

Multiplexing

- Multiplexing, or sharing several conversations over connections, virtual circuits, and physical links plays a role in several layers of the network architecture.
- In the transport layer, the need for multiplexing can arise in a number of ways.
- For example, if only one network address is available on a host, all transport connections on that machine have to use it.
- When a segment comes in, some way is needed to tell which process to give it to. This situation, called **multiplexing.**

- Multiplexing can also be useful in the transport layer for another reason.
- Suppose, for example, that a host has multiple network paths that it can use.
-If a user needs more bandwidth or more reliability than one of the network paths can provide, a way out is to have a connection that distributes the traffic among multiple network paths on a round-robin basis.
- This modus operandi is called inverse multiplexing.
- An example of inverse multiplexing is SCTP (Stream Control Transmission Protocol), which can run a connection using multiple network interfaces.
- In contrast, TCP uses a single network endpoint.

TCP

TCP is a connection-oriented protocol; it creates a virtual connection between two TCPs to send data. In addition, TCP uses flow and error control mechanisms at the transport level.

Topics discussed in this section:

TCP Services
TCP Features
Segment
A TCP Connection
Flow Control
Error Control

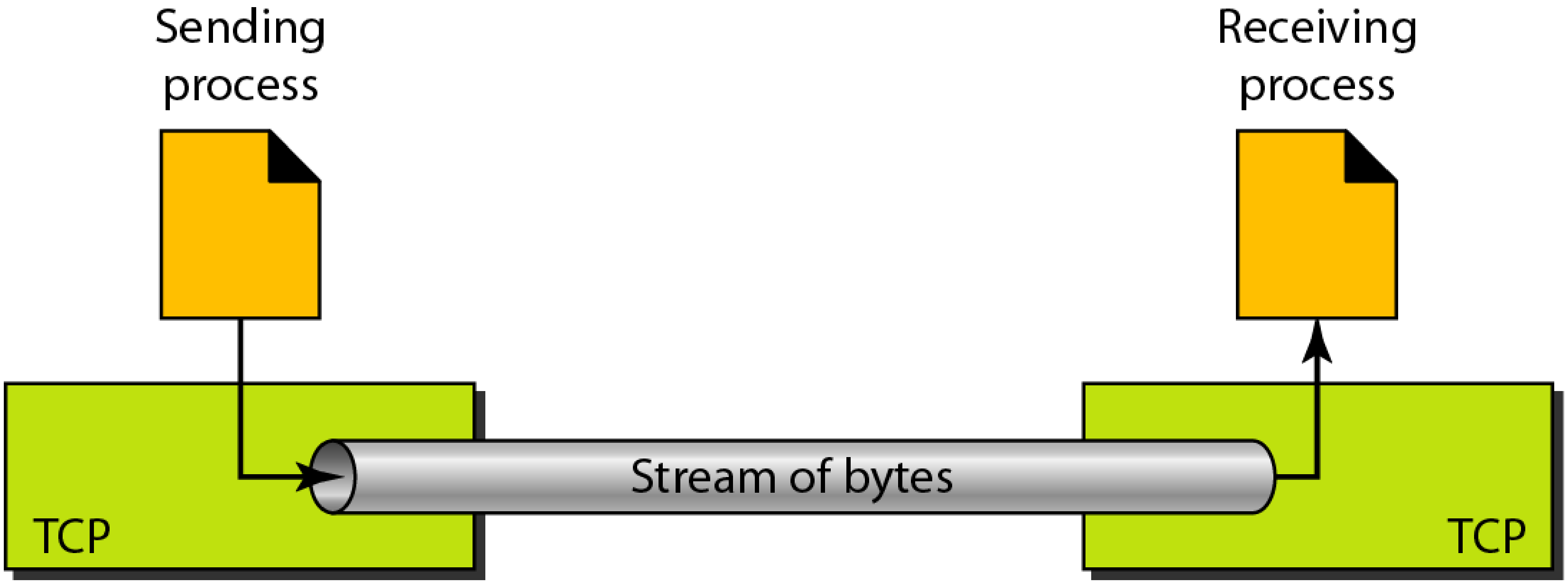
TCP Services:

1.Process to Process communication.

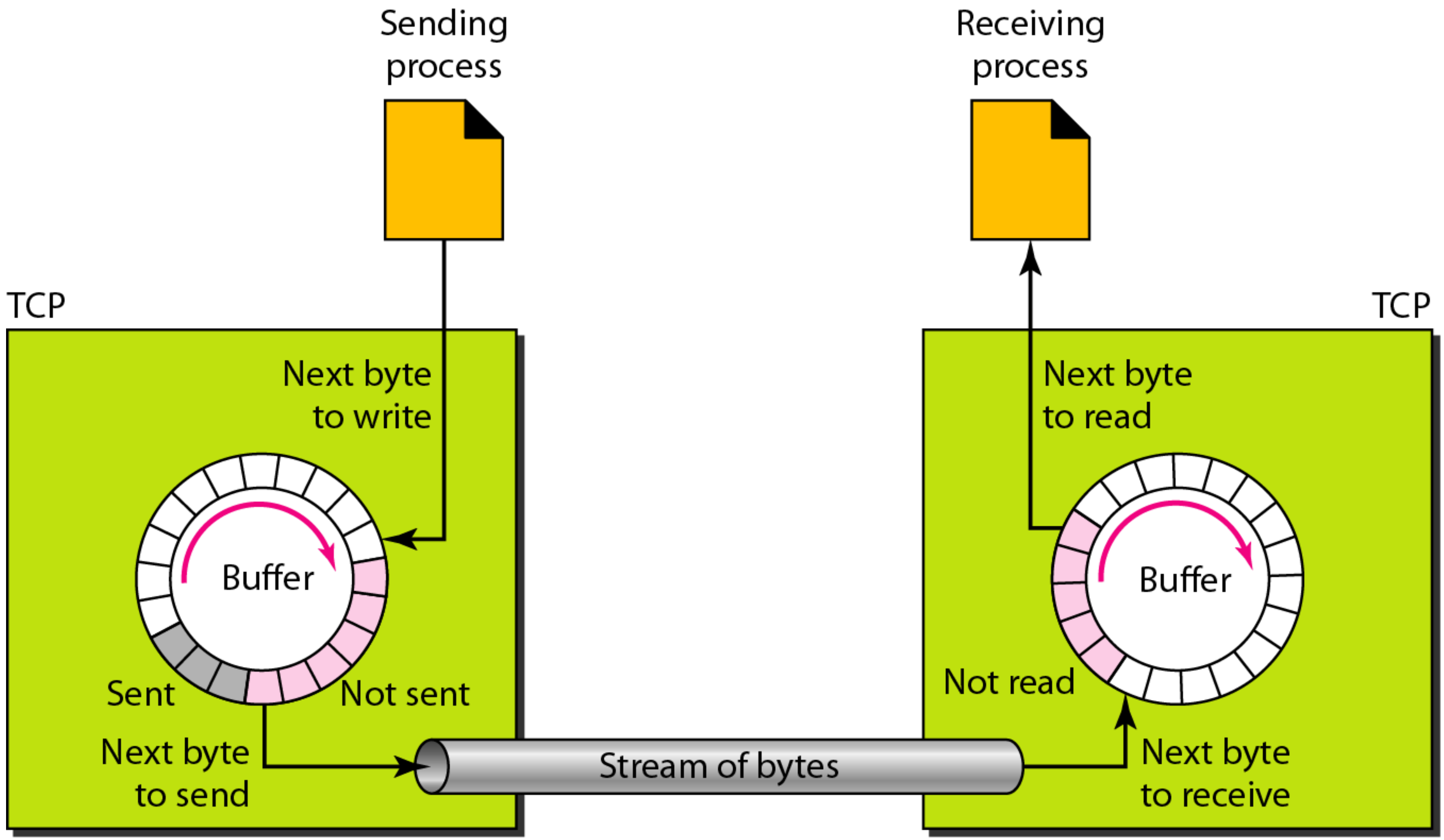
Well-known ports used by TCP

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
20	FTP, Data	File Transfer Protocol (data connection)
21	FTP, Control	File Transfer Protocol (control connection)
23	TELNET	Terminal Network
25	SMTP	Simple Mail Transfer Protocol
53	DNS	Domain Name Server
67	BOOTP	Bootstrap Protocol
79	Finger	Finger
80	HTTP	Hypertext Transfer Protocol
111	RPC	Remote Procedure Call

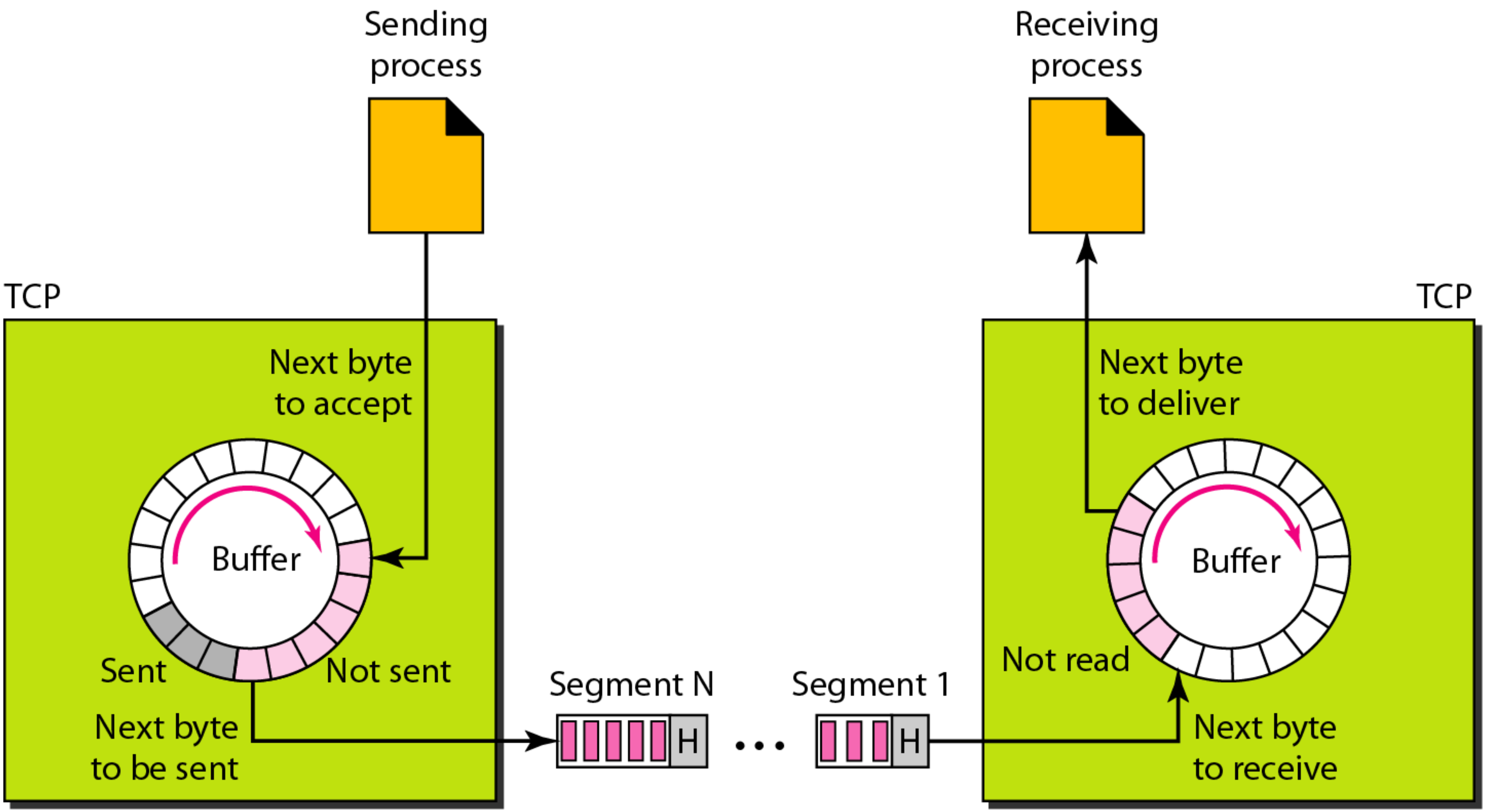
2. Stream delivery Services



3. Sending and receiving buffers



4. TCP segments



5. Full- Duplex Communication.
6. Connection Oriented Services.
7. Reliable Service.

TCP Features:

- **Numbering system.**
- Byte Number.

The bytes of data being transferred in each connection are numbered by TCP.

The numbering starts with a randomly generated number.

- Sequence Number.

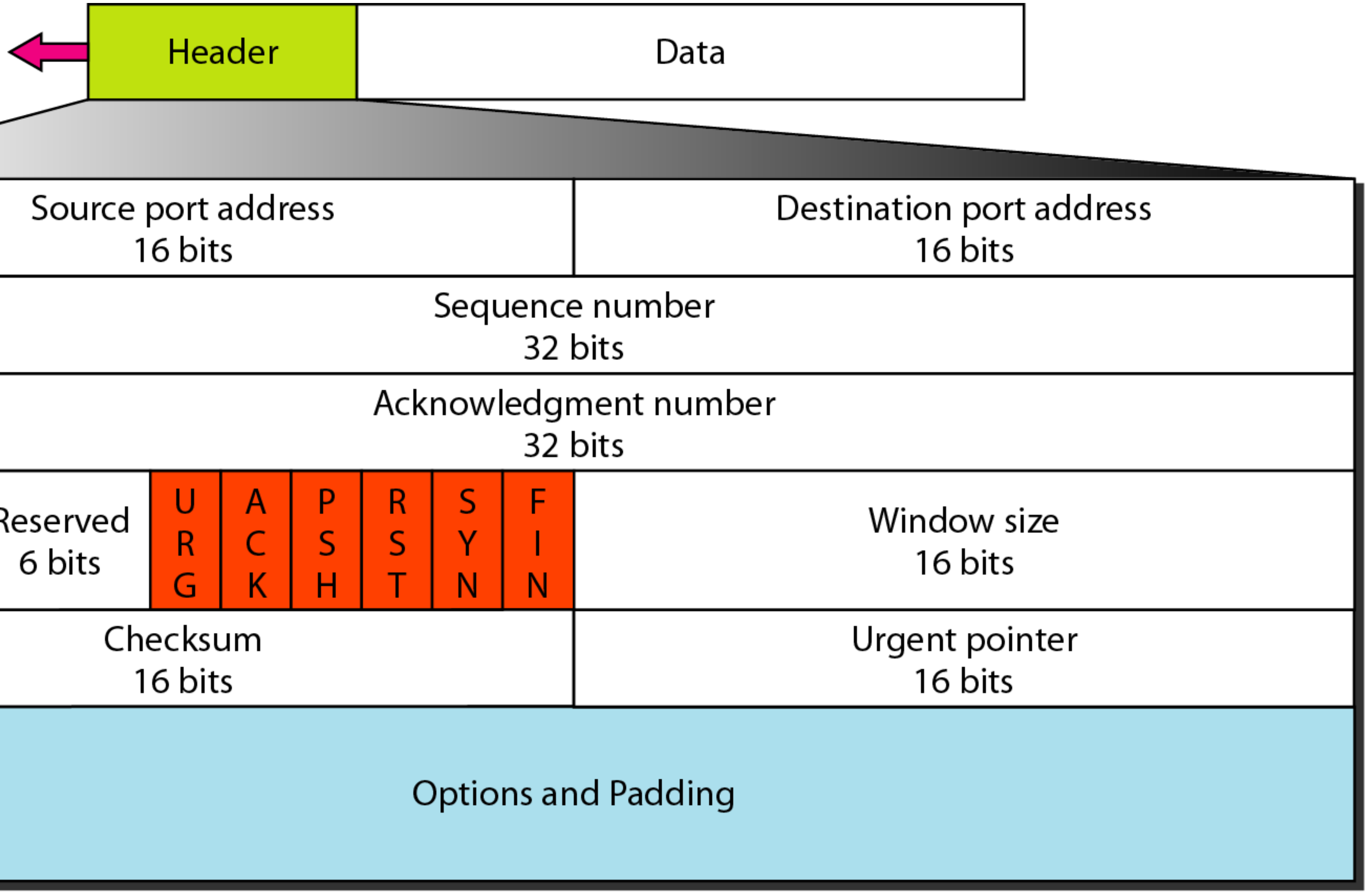
The value in the sequence number field of a segment defines the number of the first data byte contained in that segment.

- Acknowledgment Number.

The value of the acknowledgment field in a segment defines the number of the next byte a party expects to receive.
The acknowledgment number is cumulative.

- **Flow Control.**
- **Error Control.**
- **Congestion Contro**l.

TCP segment format



Control field

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

URG	ACK	PSH	RST	SYN	FIN

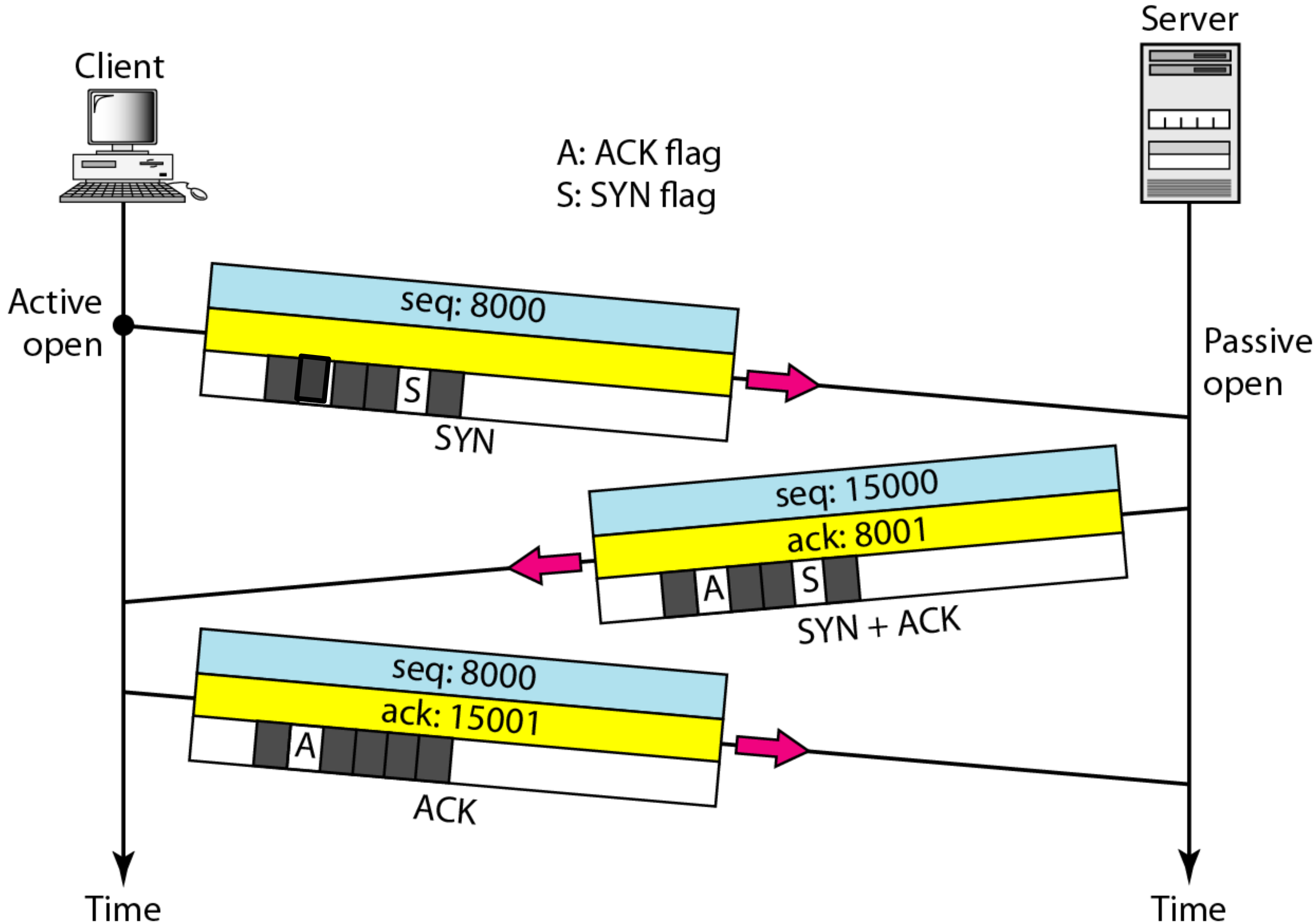
Flag	*Description*
URG	The value of the urgent pointer field is valid.
ACK	The value of the acknowledgment field is valid.
PSH	Push the data.
RST	Reset the connection.
SYN	Synchronize sequence numbers during connection.
FIN	Terminate the connection.

TCP Connection:

It requires 3 phases:

1. Connection Establishment.
2. Data Transfer.
3. Connection Termination.

TCP Connection establishment using three-way handshaking

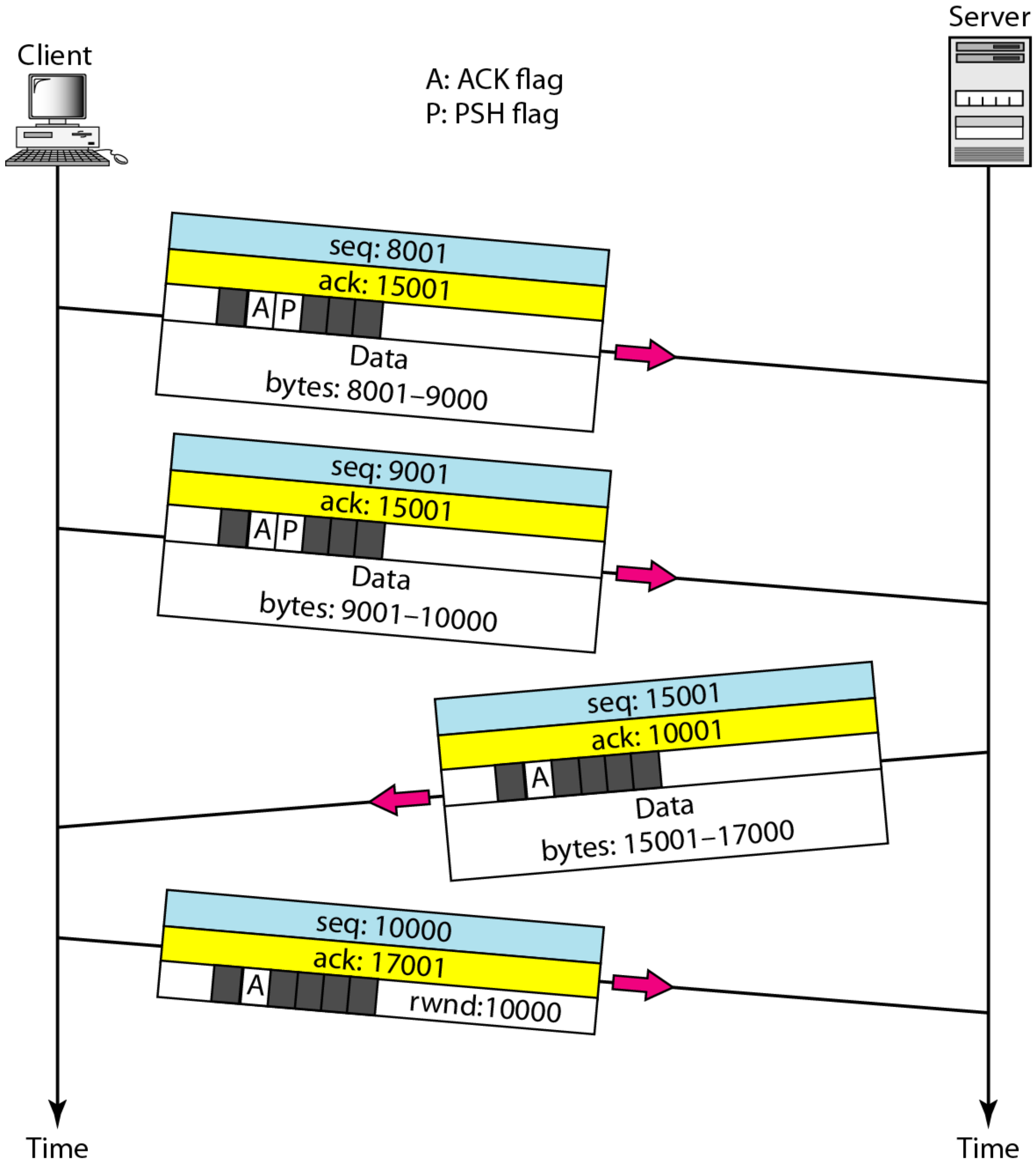


A SYN segment cannot carry data, but it consumes one sequence number.

A SYN + ACK segment cannot carry data, but does consume one sequence number.

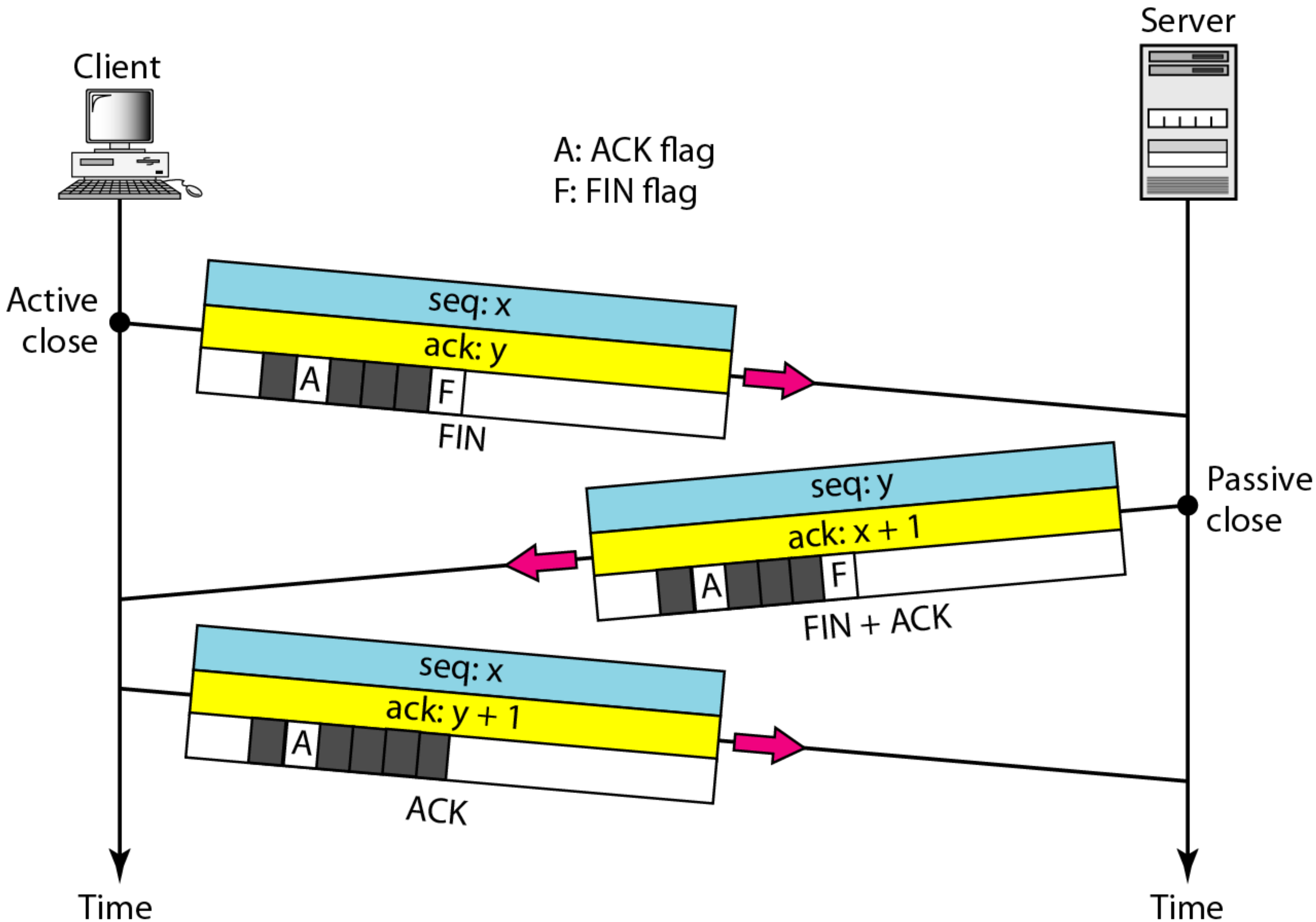
An ACK segment, if carrying no data, consumes no sequence number.

TCP Data transfer



Mr. Atul Pawar , PCCOE, Pune

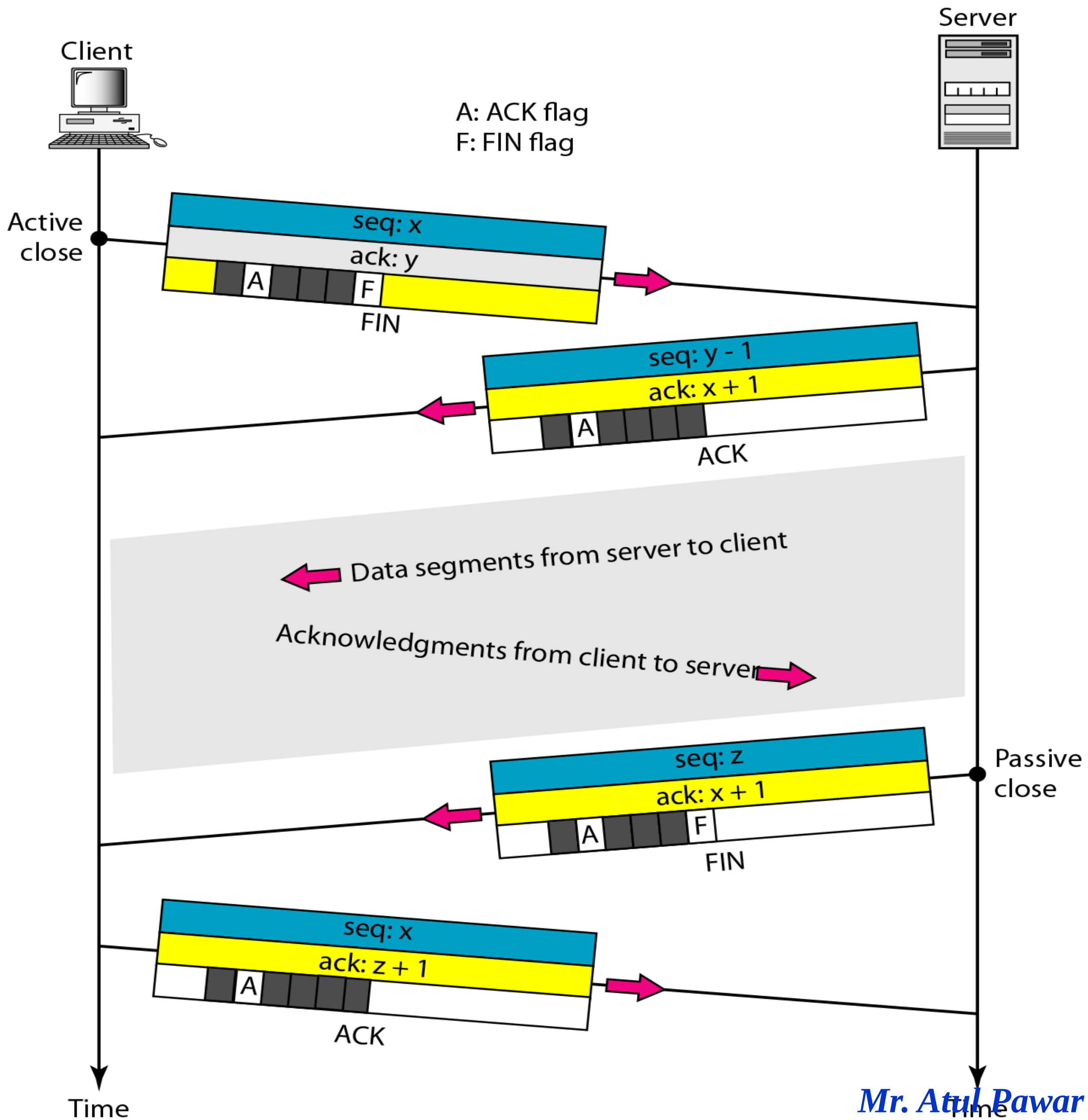
TCP Connection termination using three-way handshaking



The FIN segment consumes one sequence number if it does not carry data.

The FIN + ACK segment consumes one sequence number if it does not carry data.

TCP Four-way handshaking with a Half-close

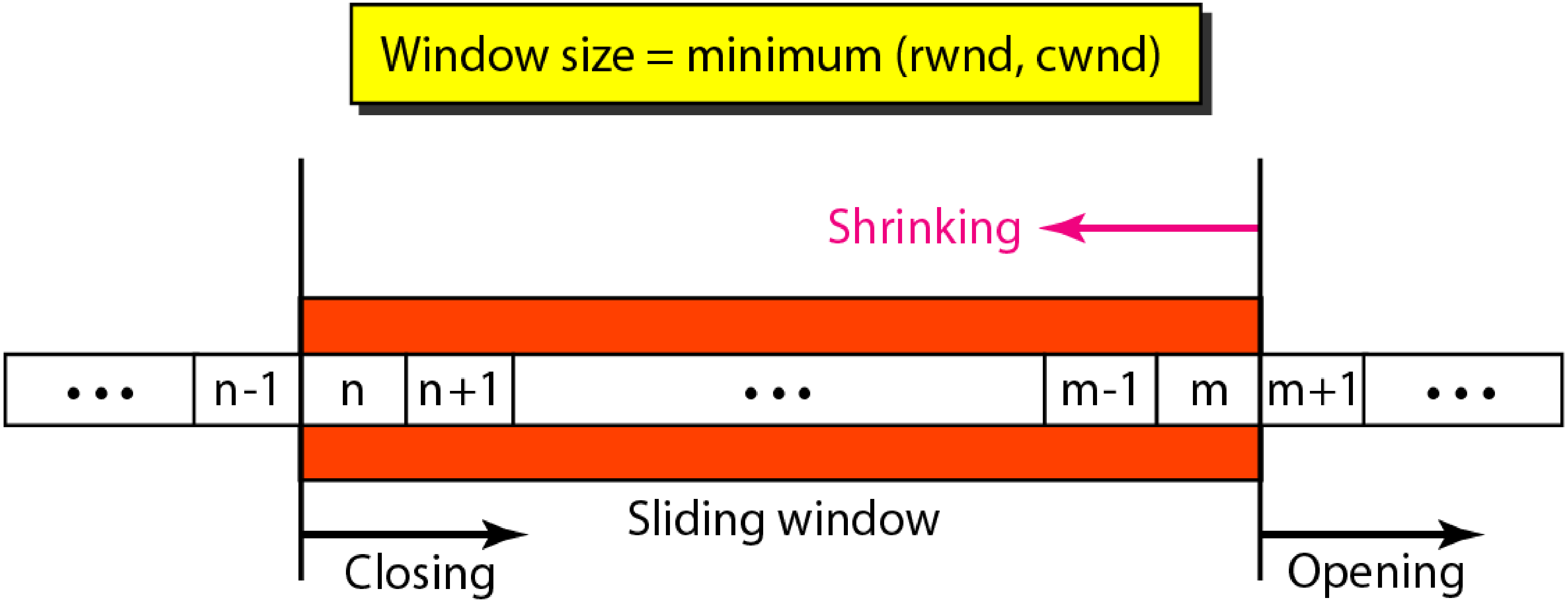


TCP Flow Control:

- TCP uses Sliding Window to handle flow control.
- Sliding window is byte oriented.
- It is of variable size.
- Window is:
  - Open
  - Closed
  - Shrunk

  - Receiver Window.
  - Congestion Window.

TCP Sliding window

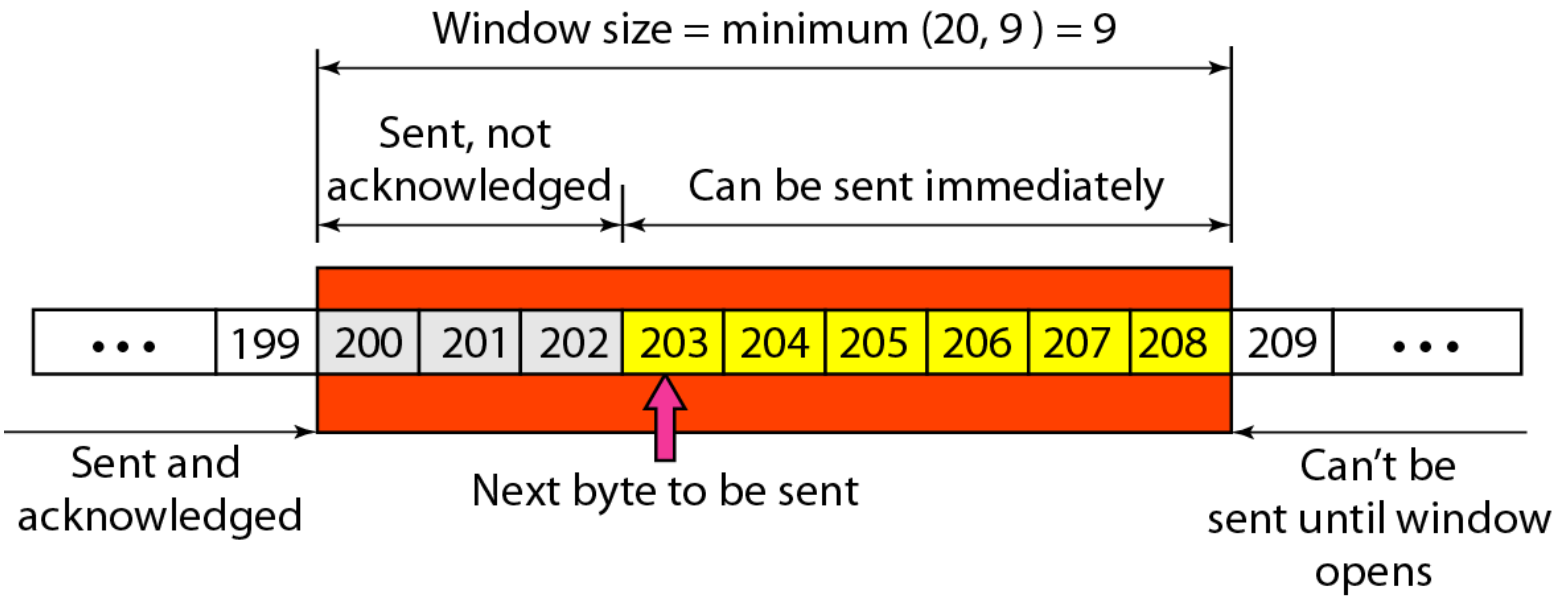


Note

A sliding window is used to make transmission more efficient as well as to control the flow of data so that the destination does not become overwhelmed with data.

TCP sliding windows are byte-oriented.

Example



Window size = minimum (20, 9) = 9
Sent, not acknowledged
Can be sent immediately
• • •
199
200
201
202
203
204
205
206
207
208
209
• • •
Sent and acknowledged
Next byte to be sent
Can't be sent until window opens

Note

Some points about TCP sliding windows:

- The size of the window is the lesser of rwnd and cwnd.
- The source does not have to send a full window’s worth of data.
- The window can be opened or closed by the receiver, but should not be shrunk.
- The destination can send an acknowledgment at any time as long as it does not result in a shrinking window.
- The receiver can temporarily shut down the window; the sender, however, can always send a segment of 1 byte after the window is shut down.

TCP Error Control

- Error control includes mechanisms for detecting corrupted segments, lost segments, out of order segments & duplicated segments.
- Error Detection and Correction in TCP is achieved through:

1. Checksum.
2. Acknowledgment.
3. Retransmission.
   - Retransmission after RTO.
   - Retransmission after 3 duplicates ACK Segments.
4. Out of order Segments.

2. Acknowledgment.

ACK segments do not consume sequence numbers and are not acknowledged.

3. Retransmission.

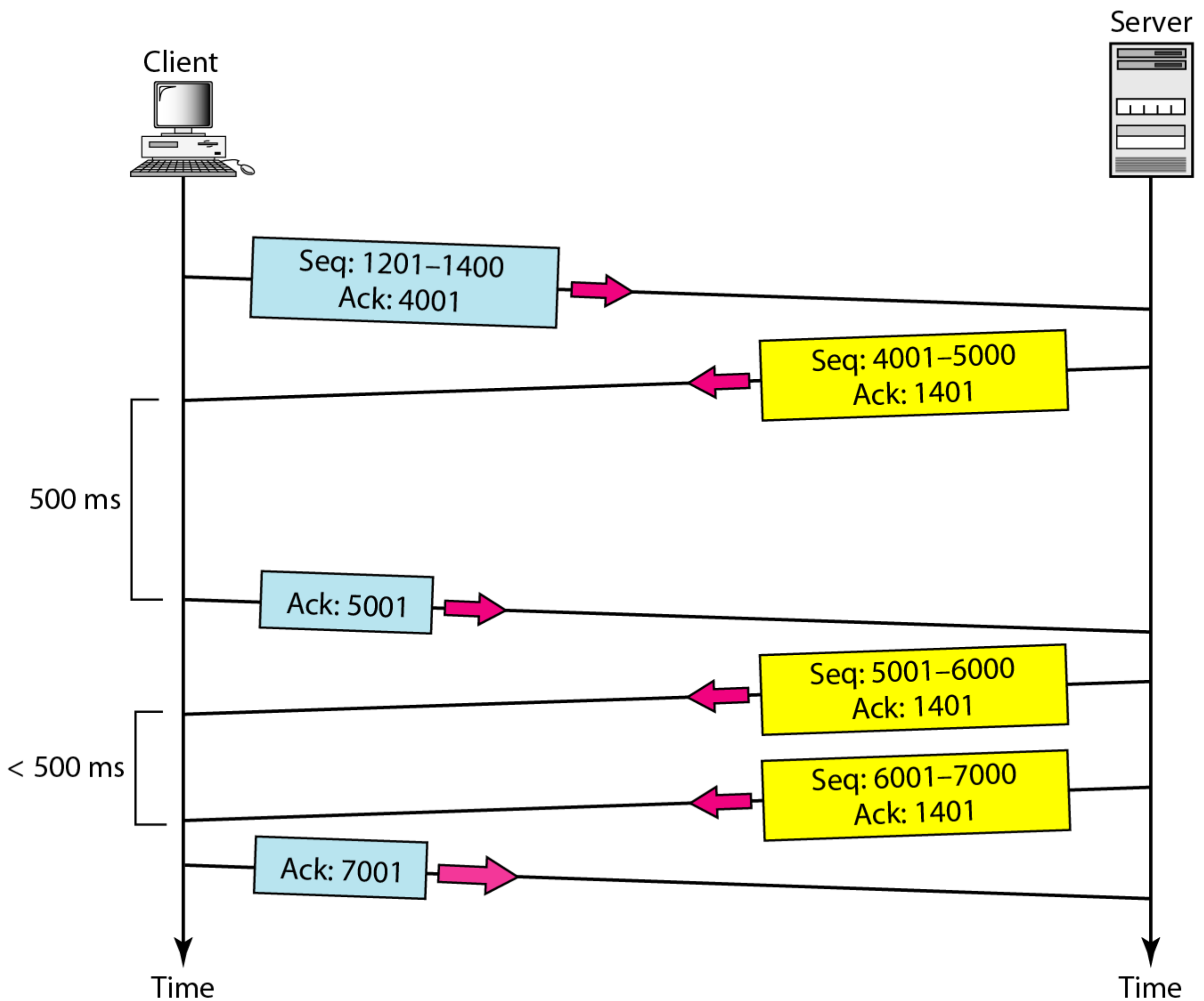
In modern implementations, a retransmission occurs if the retransmission timer expires or three duplicate ACK segments have arrived.

No retransmission timer is set for an ACK segment.

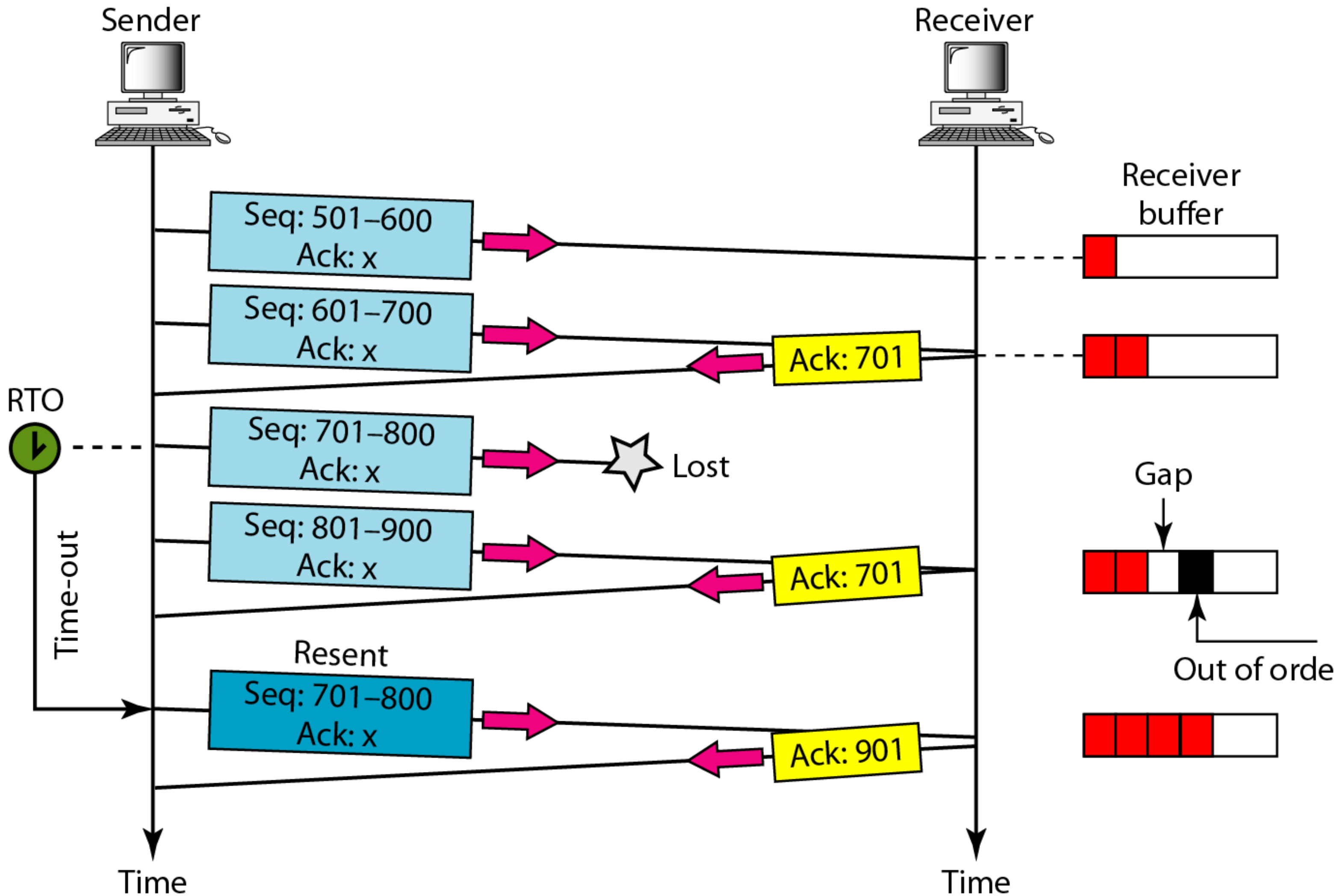
4. Out of order Segments.

Data may arrive out of order and be temporarily stored by the receiving TCP, but TCP guarantees that no out-of-order segment is delivered to the process.

Normal operation



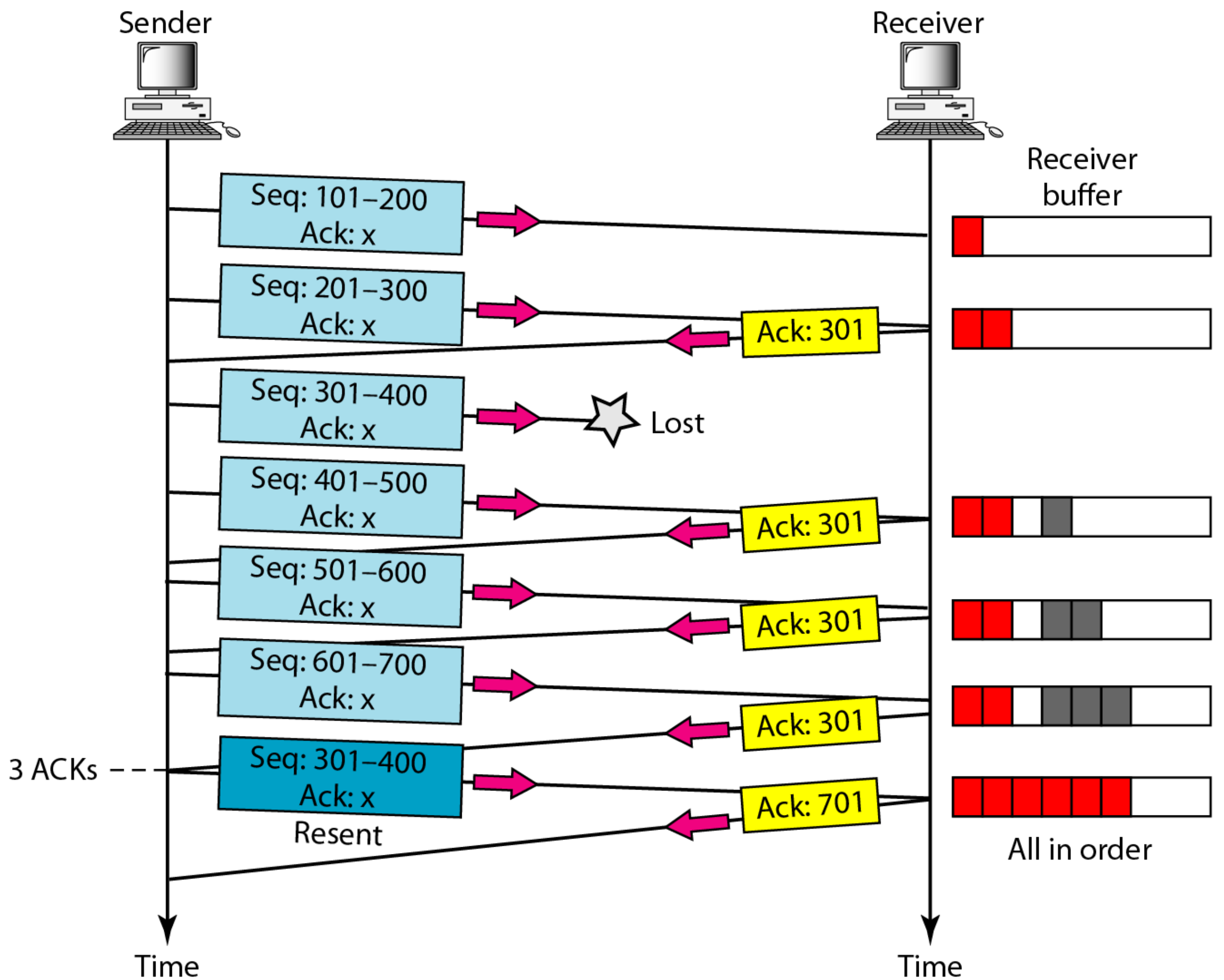
Lost segment



Note

The receiver TCP delivers only ordered data to the process.

Fast retransmission



TCP Congestion Control

- When the load offered to any network is more than it can handle, congestion builds up.
- A control law called **AIMD (Additive Increase, Multiplicative Decrease)** can be used in response to binary congestion signals received from the network.
- According to this law, in response to congestion signals the transport protocol should converge to a fair and efficient bandwidth allocation.
- TCP congestion control is based on implementing this approach using a **window** and with packet loss as the binary signal.
- To do so, TCP maintains a congestion window whose size is the number of bytes the sender may have in the network at any time.
- TCP adjusts the size of the window according to the AIMD rule.

Principle of Congestion Control

- Do not inject new packet into the network untill an old one is delivered.

- TCP tries to do this by dynamically adjusting the window size.

- TCP follows following steps to achieve Congestion Control :

Step 1 : Detect The Congestion :

- Now a days packet loss due to a transmissin errors is very rare due to the Optical fibre cables.

- So most of the loss of packets are due to congestion.

- So all the Internet TCP algorithms assumes that time outs are caused by congestion.

Step 2 : Try to prevent Congestion :

-Suitable window size to be chosen to prevent congestion.

- Sender adjust him self as per reciver buffer size.

Conclusion :

- To prevent congestion, TCP has to deal with 2 problems :

- Reciever capacity & Network Capacity.

Solution :

Actual window size = minimum (rwnd,cwnd)

- rwnd is reciever Window, is called as Flow Control Window.

- cwnd is Congestion Window.

- TCP adjust the size of window as per the AIMD rule.

How to decide Congestion Window Size?

TCP Congestion Control

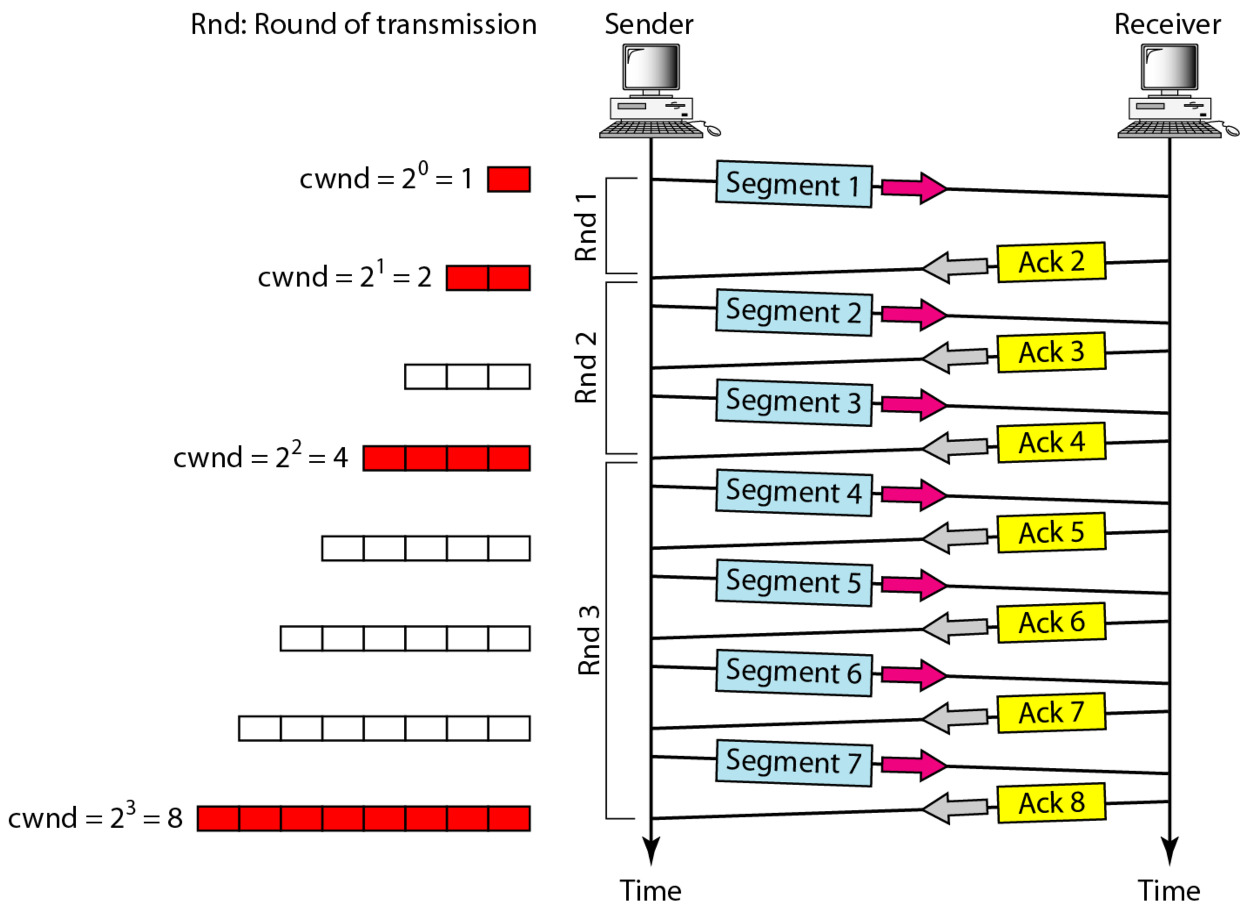
- Congestion Policy
  - TCP general Congestion policy is based on 3 phases:

    Slow Start Algorithm: Exponential Increase

    Congestion Avoidance : Additive Increase

    Congestion Detection : Multiplicative Decrease

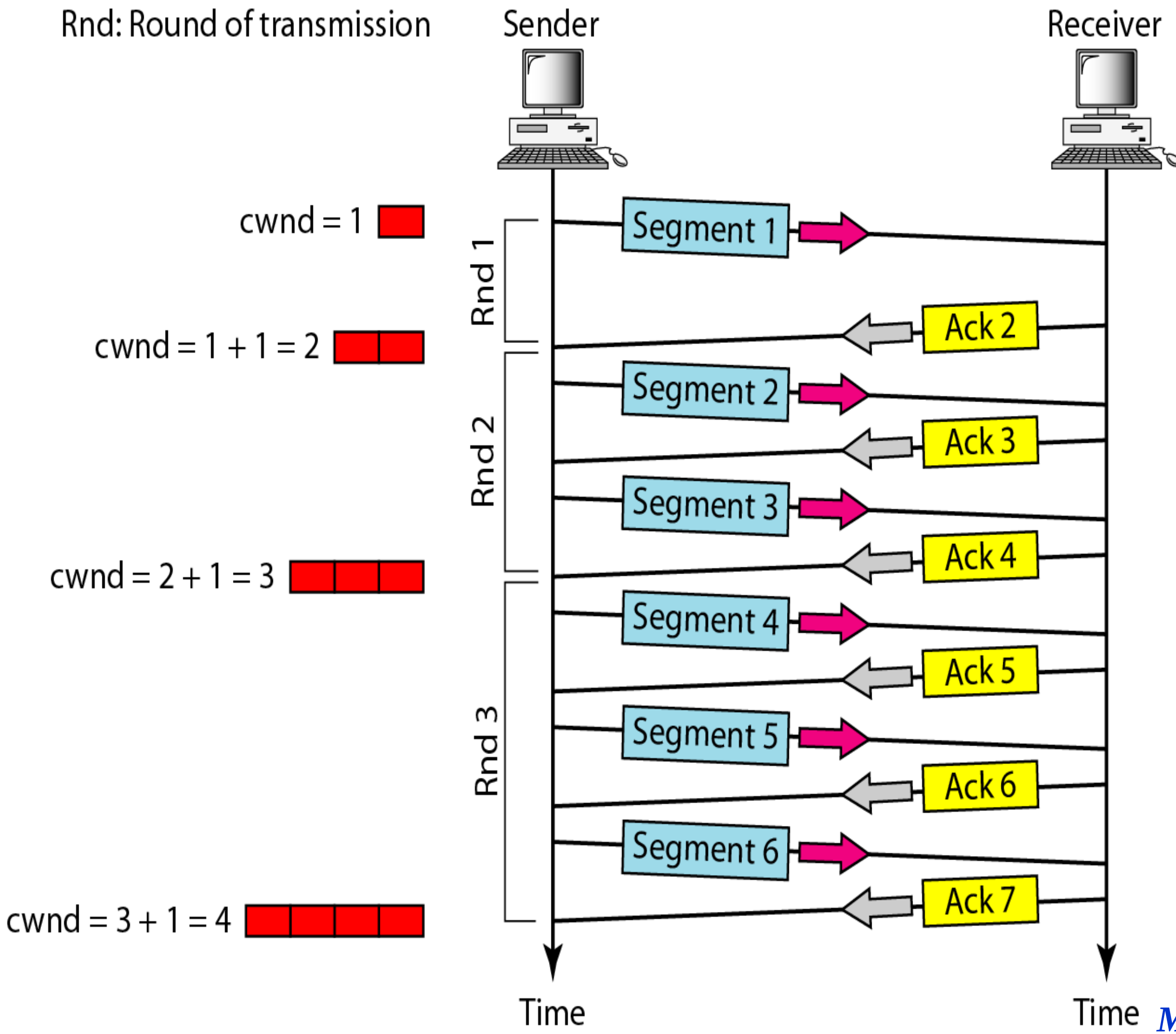
Slow start, exponential increase



Note

In the slow-start algorithm, the size of the congestion window increases exponentially until it reaches a threshold.

Congestion avoidance, additive increase



Mr. Atul Pawar , PCCOE, Pune

Note

In the congestion avoidance algorithm, the size of the congestion window increases additively until congestion is detected.

Congestion Detection : Multiplicative Decrease

- If congestion occurs, the congestion window size must be decreased.

- If retransmission; it means congestion is there in network.

- Retransmission can occur in one of two cases:
    1. when timer times out
    2. when three ACKs are received.

- In both cases, the size of the threshold is dropped to one – half, a **multiplicative decrease.**

Note

An implementation reacts to congestion detection in one of the following ways:

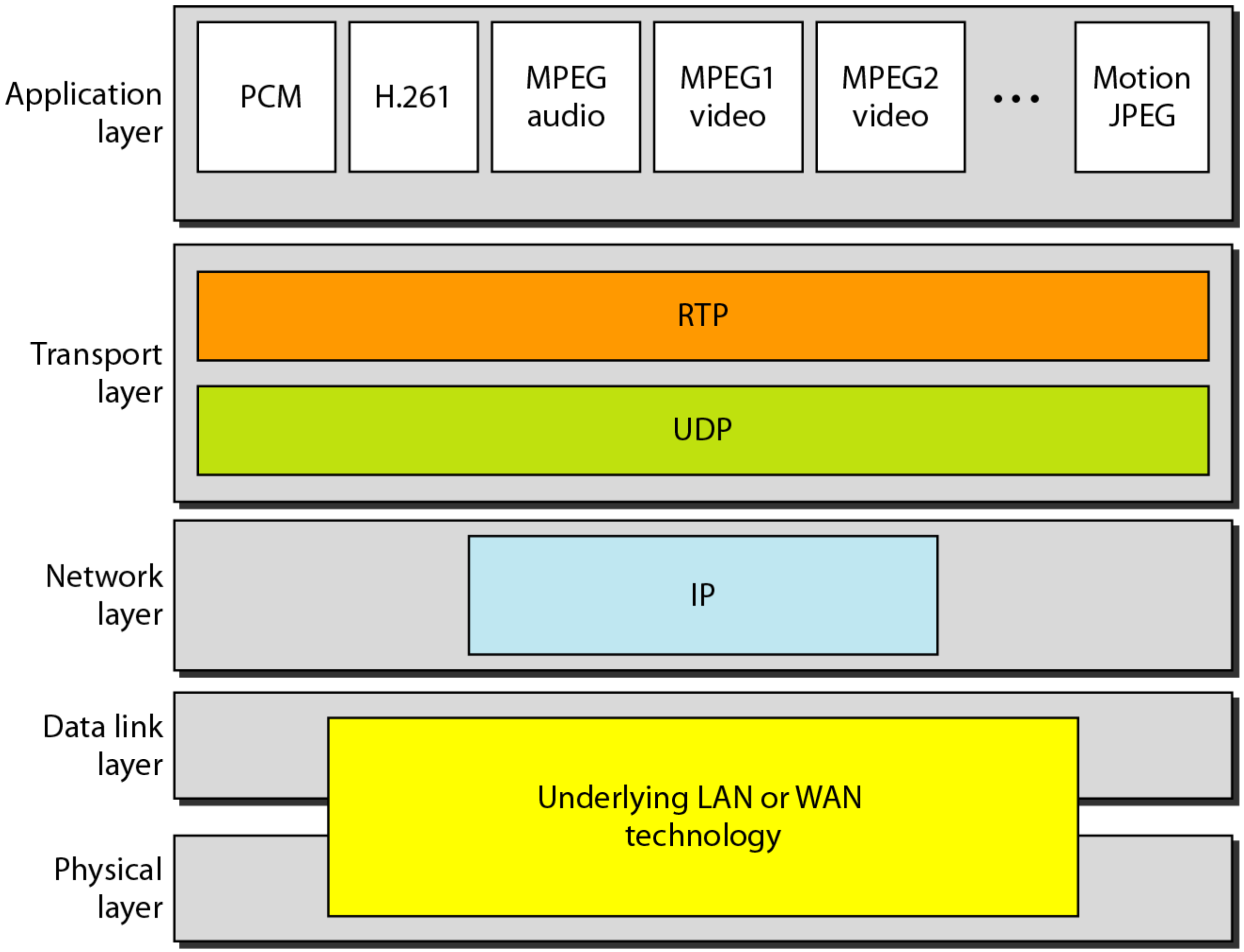
- If detection is by time-out, a new slow start phase starts.
- If detection is by three ACKs, a new congestion avoidance phase starts.

• RTP

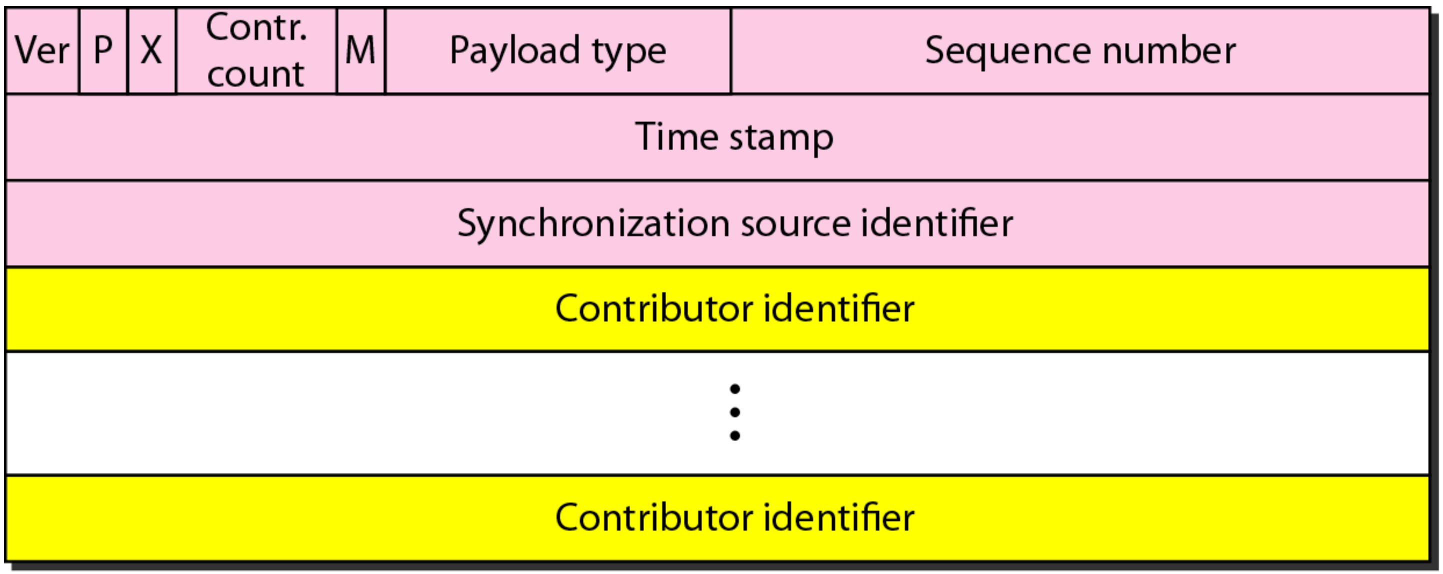
- ***Real-time Transport Protocol (RTP) is the protocol designed to handle real-time traffic on the Internet. RTP does not have a delivery mechanism (multicasting, port numbers, and so on); it must be used with UDP. RTP stands between UDP and the application program. The main contributions of RTP are time-stamping, sequencing, and mixing facilities.***

Eg. : Delivering audio and video over IP networks, communication and entertainment systems that involve streaming media, such as telephony, video teleconference applications, television services and web-based push-to-talk features.

- ***RTP***


Application layer
PCM
H.261
MPEG audio
MPEG1 video
MPEG2 video
• • •
Motion JPEG
Transport layer
RTP
UDP
Network layer
IP
Data link layer
Physical layer
Underlying LAN or WAN technology

- ***RTP packet header format***



- P : (1-bit), if set to 1, indicates the presence of padding at the end of the packet. There is no padding if the value of the P field is 0.

- X : (1-bit), if set to 1, indicates an extra extension header between the basic header and the data.

- Contributor count : (4-bit), this field indicates the number of contributors. Maximum of 15 contributors.

- M : (1-bit) It is a marker, used by the application to indicate the end of its data.

- Payload type : (7-bit) This field indicates the type of the payload.

- ***Payload types***

Type	*Application*	*Type*	*Application*	*Type*	*Application*
0	PCMμ Audio	7	LPC audio	15	G728 audio
1	1016	8	PCMA audio	26	Motion JPEG
2	G721 audio	9	G722 audio	31	H.261
3	GSM audio	10–11	L16 audio	32	MPEG1 video
5–6	DV14 audio	14	MPEG audio	33	MPEG2 video

- Sequence number (16 bits) It is used to number the RTP packets.

- Timestamp : (32-bits) It indicates the time relationship between packets.

- Synchronization source identifier : (32-bit) If there is only one source, this field defines the source.
  However, if there are several sources, the mixer is the synchronization source and the other sources are contributors.

- Contributor identifier : Each of these 32-bit identifiers (a maximum of 15) defines a source.
  When there is more than one source in a session, the mixer is the synchronization source and the remaining sources are the contributors.

- ***Note***

- **RTP uses a temporary even-numbered UDP port.**

- # SCTP

- ***Stream Control Transmission Protocol (SCTP) is a new reliable, message-oriented transport layer protocol. SCTP, however, is mostly designed for Internet applications that have recently been introduced. These new applications need a more sophisticated service than TCP can provide.***

- ***Note***

- **SCTP is a message-oriented, reliable protocol that combines the best features of UDP and TCP.**

SCTP Services

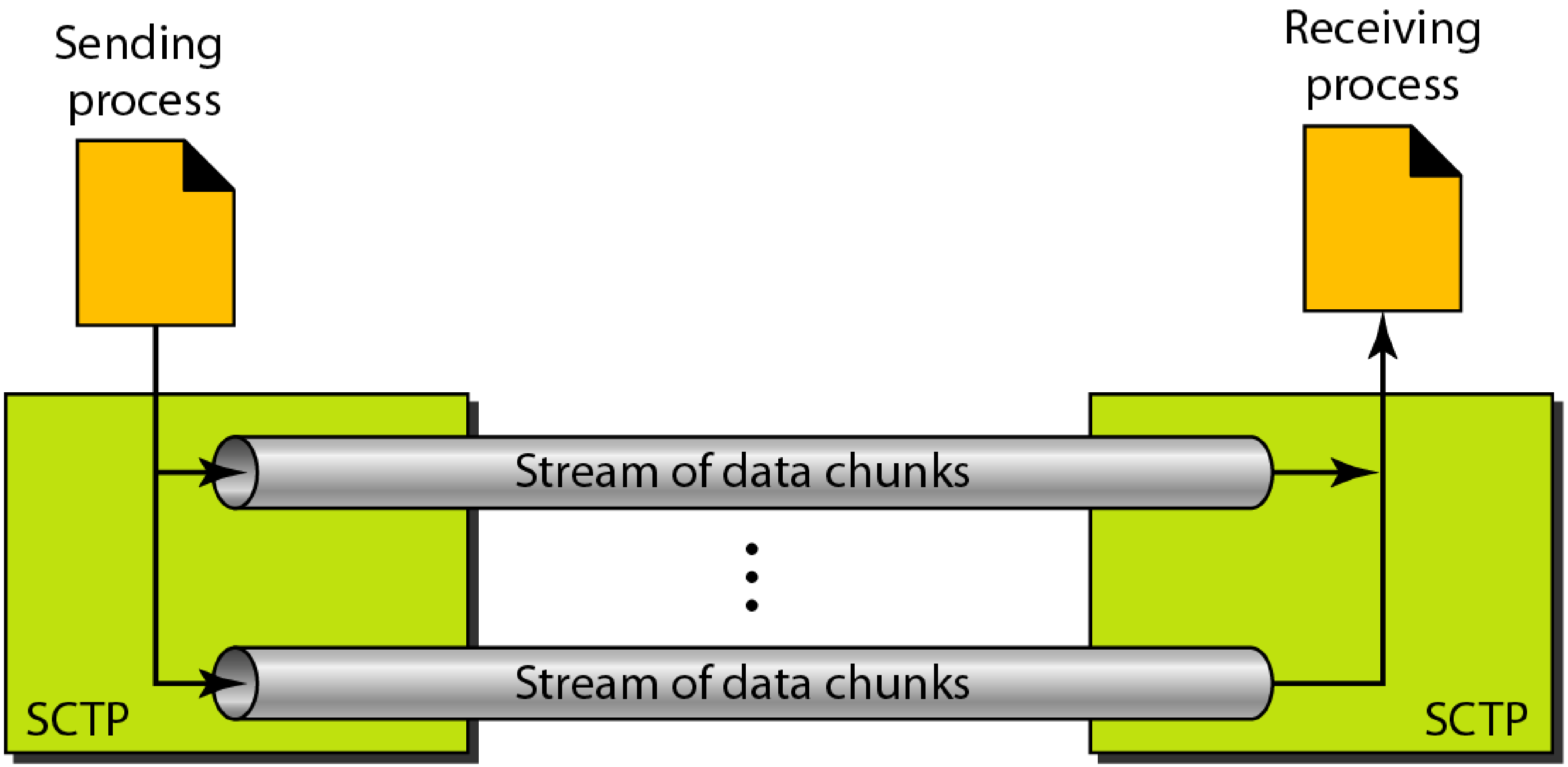
1. Process to Process Communication

SCTP uses all well- known ports in the TCP space.

- ***Some extra port numbers used by SCTP***

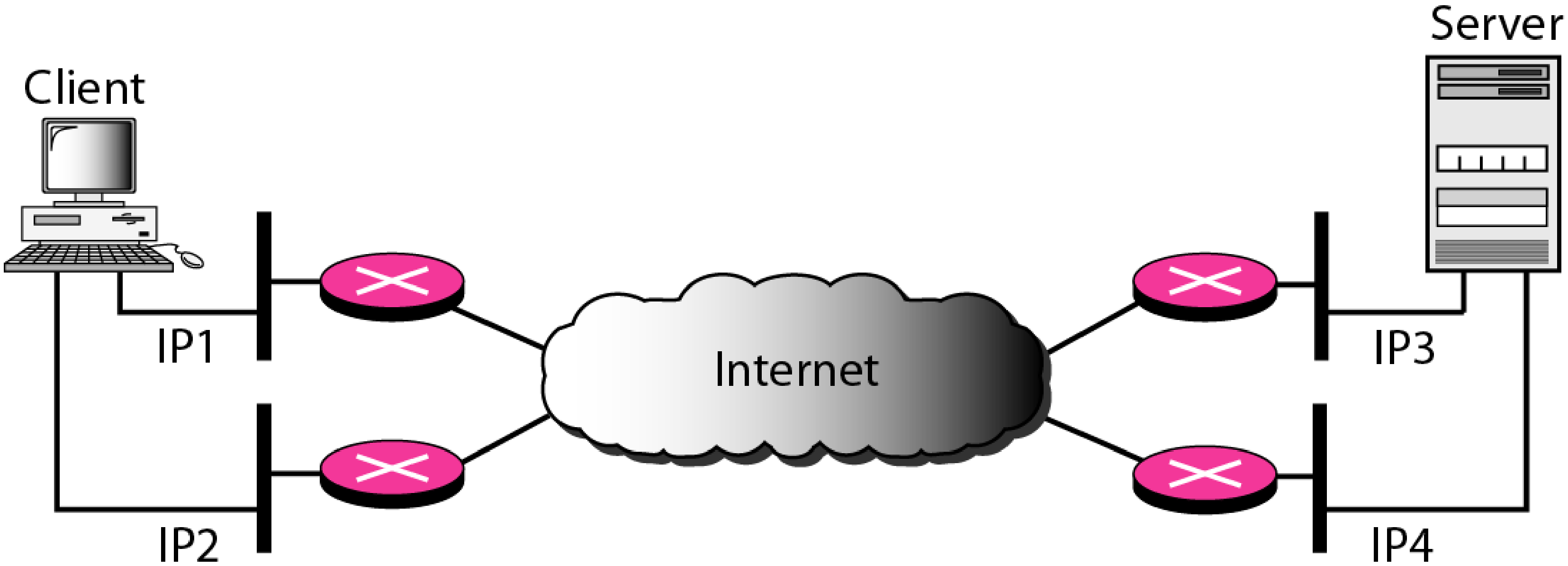
Protocol	*Port Number*	*Description*
IUA	9990	ISDN over IP
M2UA	2904	SS7 telephony signaling
M3UA	2905	SS7 telephony signaling
H.248	2945	Media gateway control
H.323	1718, 1719, 1720, 11720	IP telephony
SIP	5060	IP telephony

• 2. Multiple-streams



- **An association in SCTP can involve multiple streams.**

- **3. Multihoming**



- **SCTP association allows multiple IP addresses for each end.**

4. Full Duplex Communication

- Like TCP, SCTP offers full – duplex service, in which data can flow in both directionas at the same time.
- Each SCTP then has a sendig and receiving buffer.

5. Connection – Oriented Service

- SCTP is a connection oriented protocol.
- In SCTP, a connection is called an association.

6. Reliable Service

- SCTP is reliable transport protocol.
- It uses acknowledgment mechanism.

SCTP Features

1. Transmission Sequence Number (TSN)

- The unit of data in TCP is byte, where in SCTP it is DATA chunk.
- Data transfer in SCTP is controlled by numbering the data chunk.
- SCTP uses a TSN to number the data chunk.
- TSN in SCTP = Seq. Number in TCP.
- TSNs are 32 bits long.

2. Stream Identifier (SI)

- In SCTP, there are several streams.
- Each stream in SCTP needs to be identified by using a Stream Identifier (SI).
- SI mentioned in header with size 16 bit.

SCTP Features

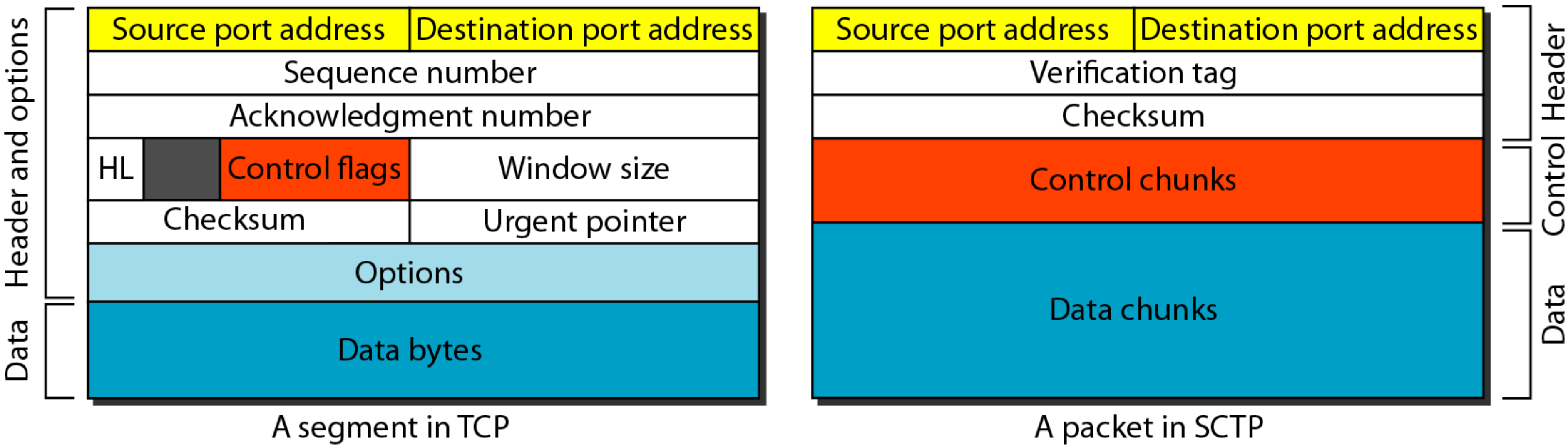
3. Stream Sequence Number (SSN)

- To distinguish between different data chunks belonging to the same stream, SCTP uses SSNs.

4. Packets

- In SCTP data are carried as data chunks, control information is carried as control chunks.
- Several control chunks and data chunks can be packed together in a packet.
- TCP has segments; SCTP has packets.

- ***Comparison between a TCP segment and an SCTP packet***



1. The control information in TCP is part of the header; the control information in SCTP is included in the control chunks. There are several types of control chunks; each is used for a different purpose.

2. The data in a TCP segment treated as one entity; an SCTP packet can carry several data chunks; each can belong to a different stream.

3. The options section, which can be part of a TCP segment, does not exist in an SCTP packet. Options in SCTP are handled by defining new chunk types.

4. The mandatory part of the TCP header is 20 bytes, while the general header in SCTP is only 12 bytes. The SCTP header is shorter due to the following:

   a. An SCTP sequence number (TSN) belongs to each data chunk and hence is located in the chunk's header.

   b. The acknowledgment number and window size are part of each control chunk.

   c. There is no need for a header length field (shown as HL in the TCP segment) because there are no options to make the length of the header variable; the SCTP header length is fixed (12 bytes).

   d. There is no need for an urgent pointer in SCTP.

5. The checksum in TCP is 16 bits; in SCTP, it is 32 bits.

6. The verification tag in SCTP is an association identifier, which does not exist in TCP. In TCP, the combination of IP and port addresses defines a connection; in SCTP we may have multihoming using different IP addresses. A unique verification tag is needed to define each association.

7. TCP includes one sequence number in the header, which defines the number of the first byte in the data section. An SCTP packet can include several different data chunks. TSNs, SIs, and SSNs define each data chunk.

8. Some segments in TCP that carry control information (such as SYN and FIN) need to consume one sequence number; control chunks in SCTP never use a TSN, SI, or SSN. These three identifiers belong only to data chunks, not to the whole packet.

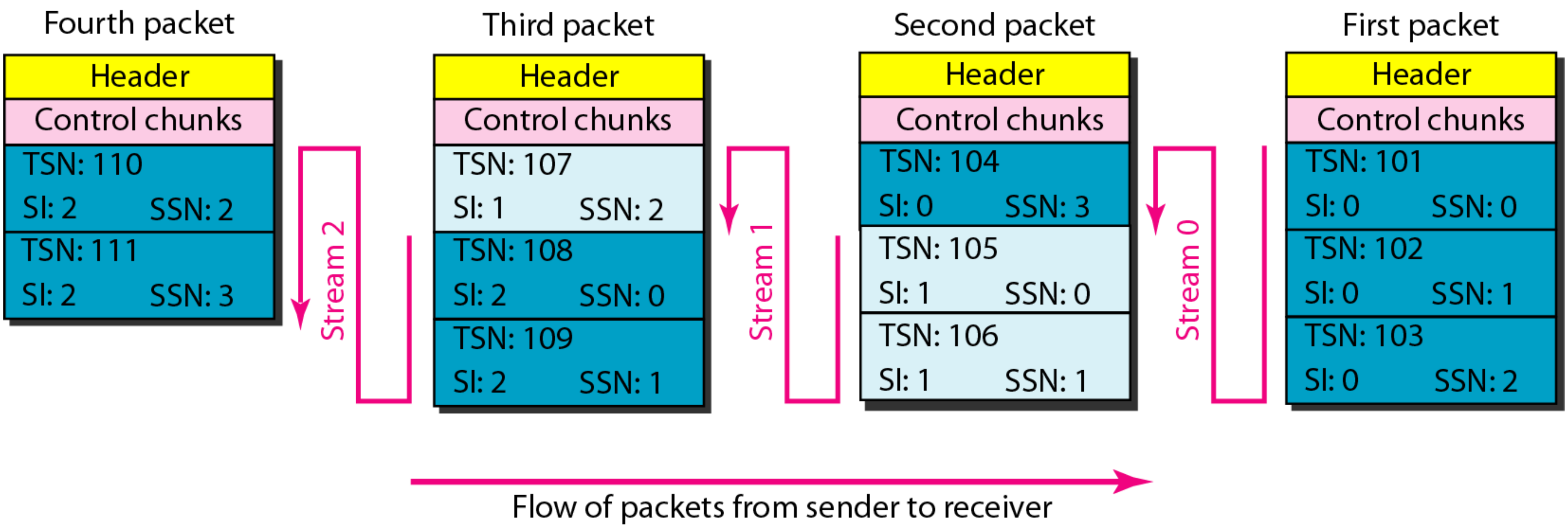
- ***Note***

- **In SCTP, control information and data information are carried in separate chunks.**

- ***Note***

- **Data chunks are identified by three items: TSN, SI, and SSN.**
- **TSN is a cumulative number identifying the association; SI defines the stream; SSN defines the chunk in a stream.**

Packet, data chunks, and streams



5. Acknowledgment

- ***Note***

- In SCTP, acknowledgment numbers are chunk oriented which are used to acknowledge only data chunks;
- control chunks are acknowledged by other control chunks if necessary.

SCTP Features Continue...

6. Flow Control

- Like TCP, SCTP provides flow control.

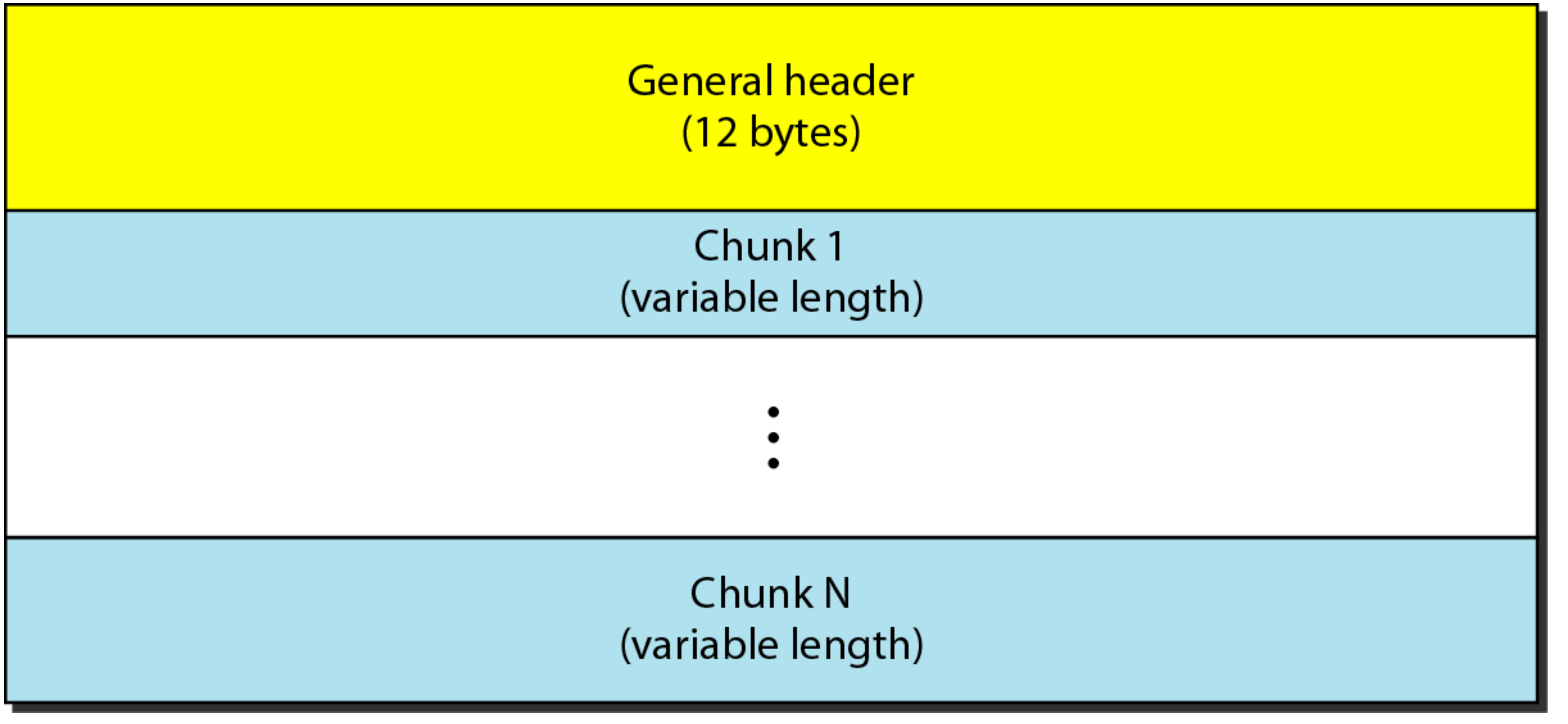
7. Error Control

- TSN numbers and Acknowledge numbers are used for error control.

8. Congestion Control

- Like TCP, SCTP implements congestion control to determine how many data chunks can be injected into the network.

- **SCTP packet format**



- ***Note***

- **In an SCTP packet, control chunks come before data chunks.**

- **General header**

Source port address 16 bits	Destination port address 16 bits
Verification tag 32 bits	
Checksum 32 bits	

Verification tag:

- This is a number that matches a packet to an association.
- This prevents a packet from a previous association from being mistaken as a packet in this association.
- It serves as an identifier for the association; it is repeated in every packet during the association.
- There is a separate verification used for each direction in the association.

- **Chunks**

Type	*Chunk*	*Description*
0	**DATA**	User data
1	**INIT**	Sets up an association
2	**INIT ACK**	Acknowledges INIT chunk
3	**SACK**	Selective acknowledgment
4	**HEARTBEAT**	Probes the peer for liveliness
5	**HEARTBEAT ACK**	Acknowledges HEARTBEAT chunk
6	**ABORT**	Aborts an association
7	**SHUTDOWN**	Terminates an association
8	**SHUTDOWN ACK**	Acknowledges SHUTDOWN chunk
9	**ERROR**	Reports errors without shutting down
10	**COOKIE ECHO**	Third packet in association establishment
11	**COOKIE ACK**	Acknowledges COOKIE ECHO chunk
14	**SHUTDOWN COMPLETE**	Third packet in association termination
192	**FORWARD TSN**	For adjusting cumulative TSN

- ***Note***

- **A connection in SCTP is called an association.**

Features Comparison

Services/Features	SCTP	TCP	UDP
Full-duplex data transmission	yes	yes	yes
Connection-oriented	yes	yes	no
Reliable data transfer	yes	yes	no
Partially reliable data transfer	optional	no	no
Ordered data delivery	yes	yes	no
Unordered data delivery	yes	no	yes
Flow and congestion control	yes	yes	no
Explicit congestion notification support	yes	yes	no
Selective acks	yes	optional	no
Preservation of message boundaries	yes	no	yes
Path maximum transmission unit discovery	yes	yes	no
Application data fragmentation/bundling	yes	yes	no
Multistreaming	yes	no	no
Multihoming	yes	no	no
Protection against SYN flooding attack	yes	no	n/a
Half-closed connections	no	yes	n/a

DATA TRAFFIC

The main focus of congestion control and quality of service is data traffic. In congestion control we try to avoid traffic congestion. In quality of service, we try to create an appropriate environment for the traffic.

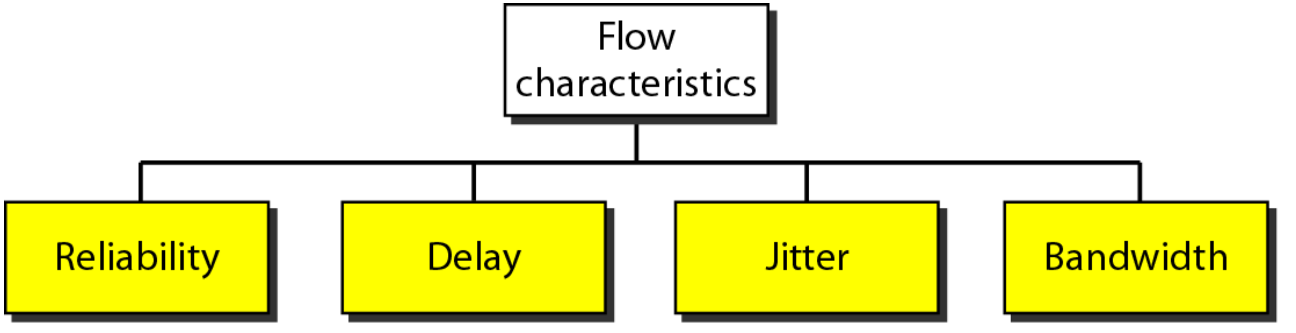
QUALITY OF SERVICE

Quality of service (QoS) is an internetworking issue that has been discussed more than defined. We can informally define quality of service as something a flow seeks to attain.

Topics discussed in this section:

Flow Characteristics

Flow characteristics



TECHNIQUES TO IMPROVE QoS

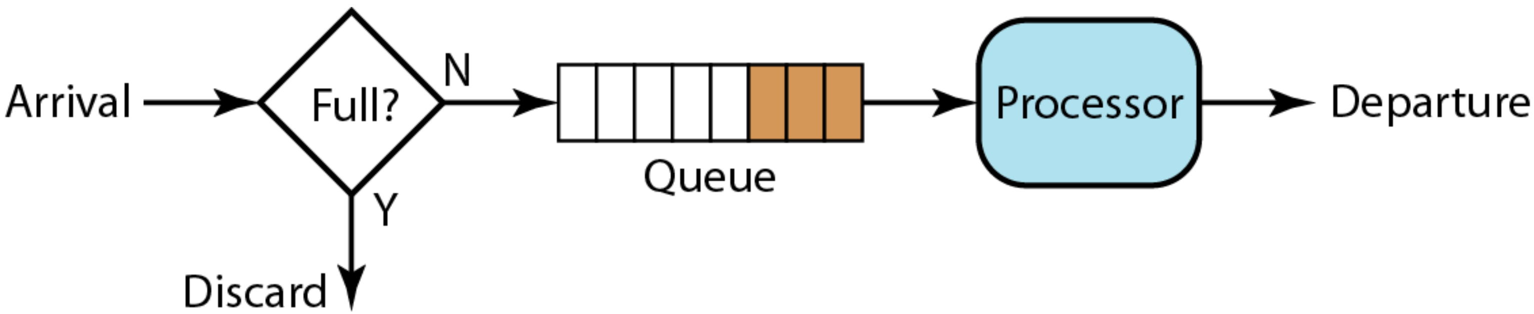
In this section, we discuss some techniques that can be used to improve the quality of service. We briefly discuss four common methods: scheduling, traffic shaping, admission control, and resource reservation.

Topics discussed in this section:

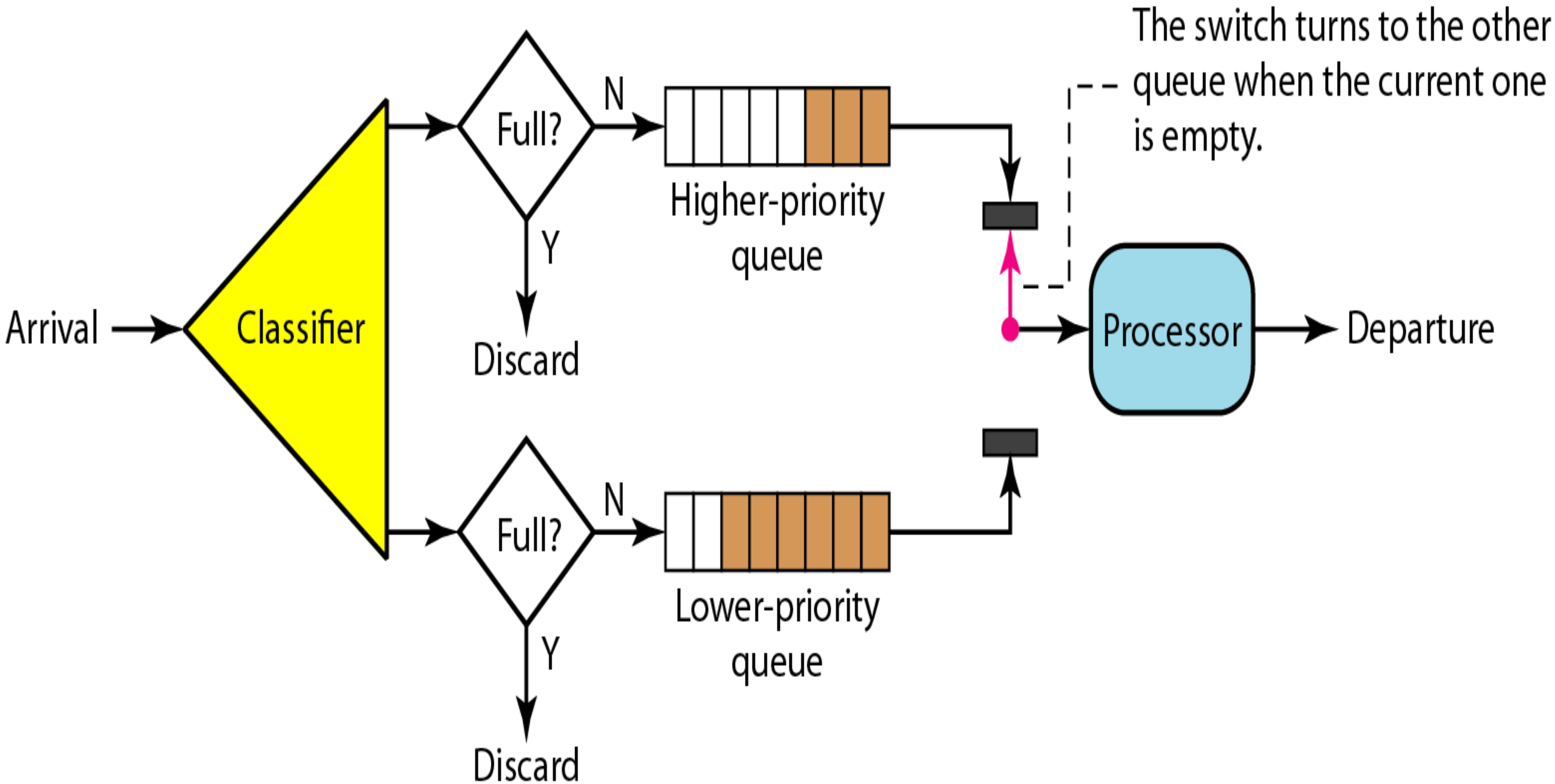
Scheduling
Traffic Shaping
Resource Reservation
Admission Control

Scheduling : 3 techniques.

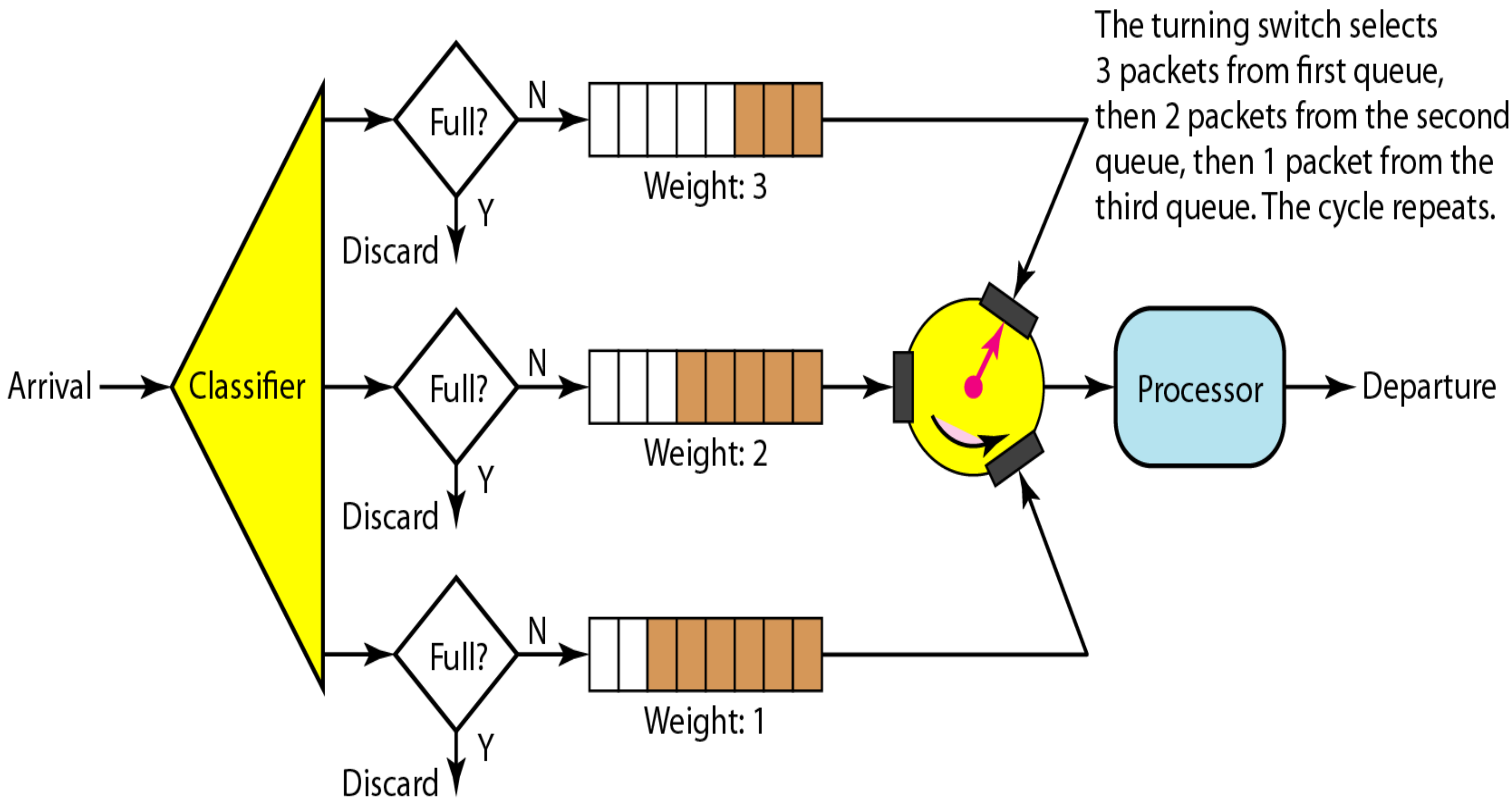
1. FIFO queue



2. Priority queuing

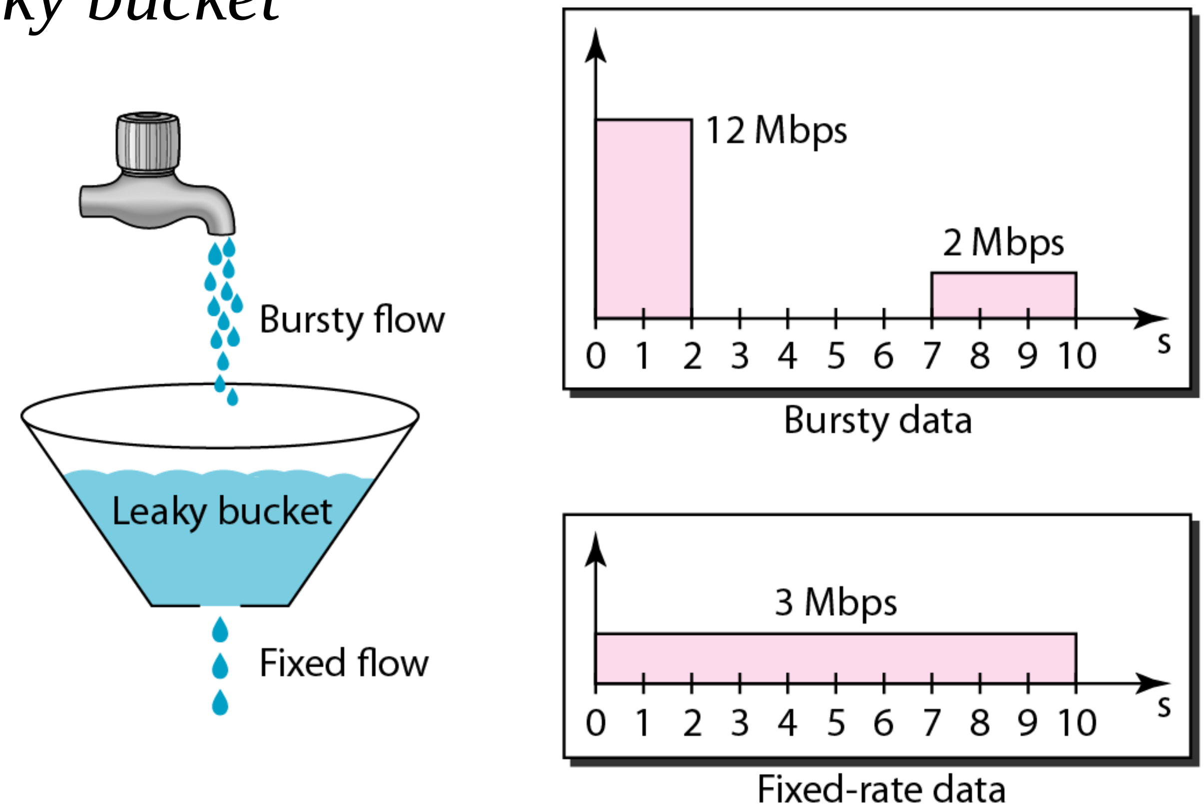


3. Weighted fair queuing

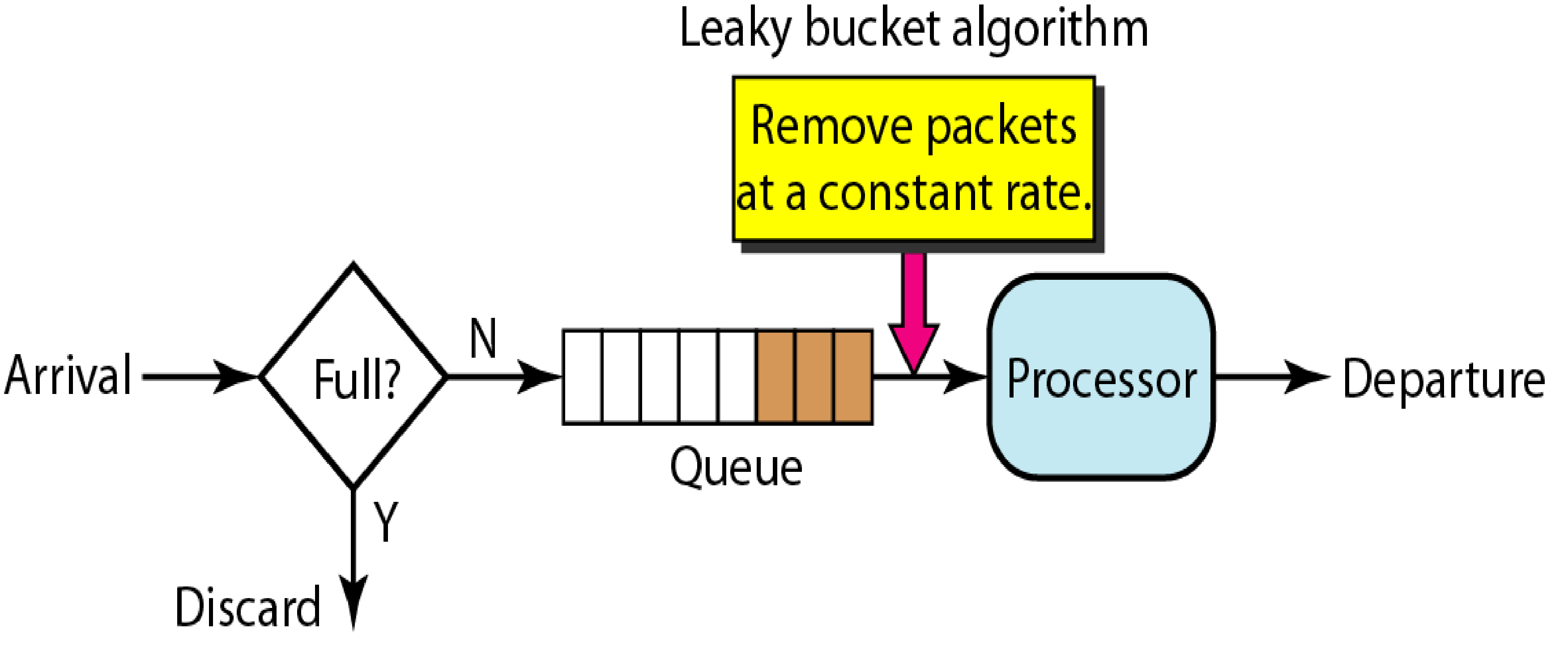


Traffic Shaping : *2 techniques can shape traffic.*

1. Leaky bucket



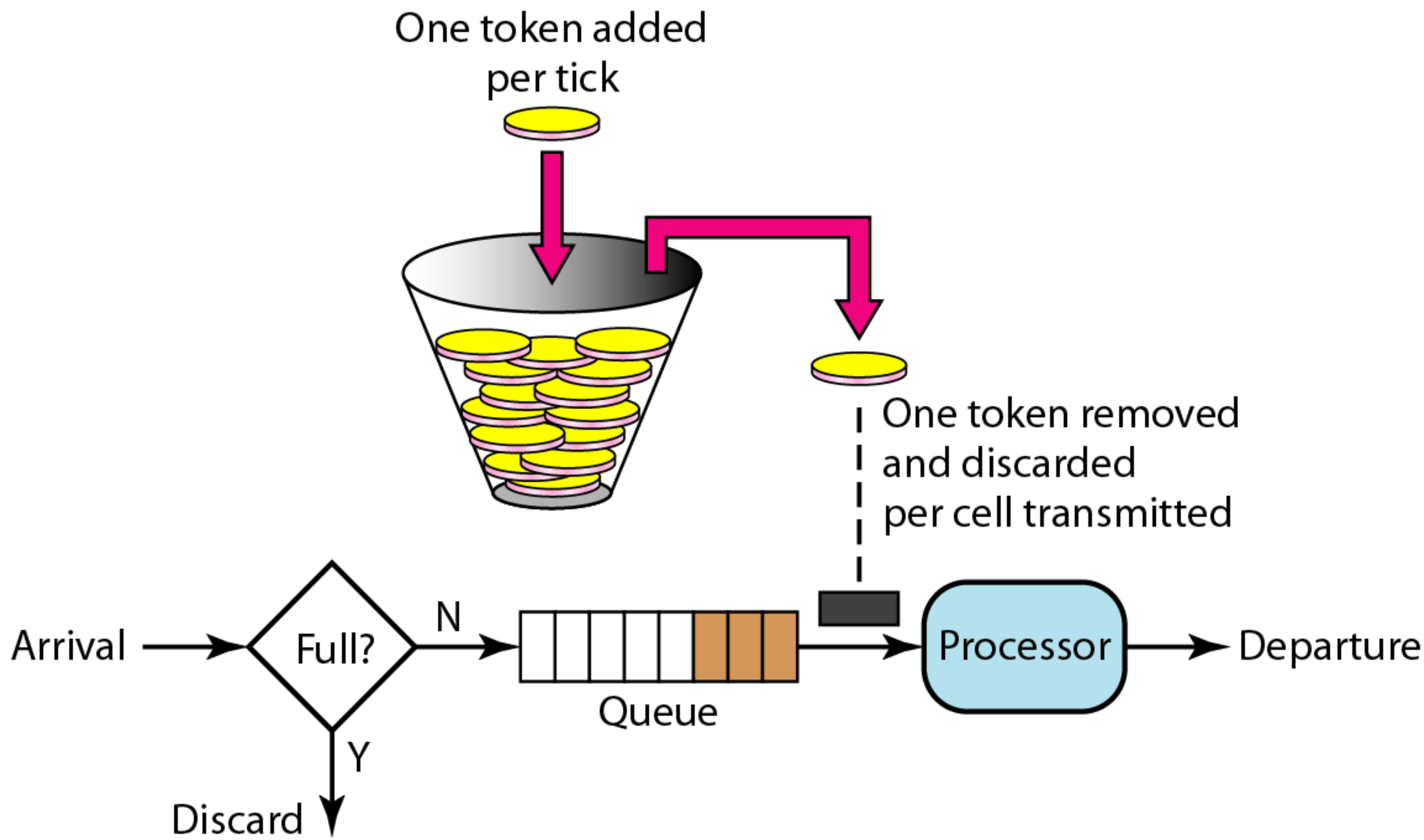
Leaky bucket implementation



Note

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

2. Token bucket



Note

The token bucket allows bursty traffic at a regulated maximum rate.

Resource Reservation :

- A flow of data needs resources such as a buffer, bandwidth, CPU time, and so on.

- The quality of service is improved if these resources are reserved beforehand.

Admission Control :

- Admission control refers to the mechanism used by a router, or a switch, to accept or reject a flow based on predefined parameters called flow specifications.

- Before a router accepts a flow for processing, it checks the flow specifications to see if its capacity (in terms of bandwidth, buffer size, CPU speed, etc.) and its previous commitments to other flows can handle the new flow.

INTEGRATED SERVICES

Two models have been designed to provide quality of service in the Internet: Integrated Services and Differentiated Services. We discuss the first model here.

Note : IntServ model is considered a specific requirement of an applicaion in one particular case regardless of application type.

Note

- Integrated Services is a flow-based QoS model designed for IP (Network layer, connectionless, datagram, packet switching protocol).
- It is also called as IntServ.

- Signaling

- IntServ is flow based model,which means that all accommodations need to be made before a flow can start.
- For that we need connection oriented service at network layer.
- But IP protocol is connectionless protocol, so we need another protocol to be run on top of IP to make a connection oriented protocol before we can use this model.
  - This protocol is called Resource Reservation Protocol (RSVP).

- Flow Spécification
  - When a source makes a reservation, it needs to define a flow specification.
  - A flow spedfication has two parts: Rspec (resource specification) and Tspec (traffic specification).
  - Rspec defines the resource that the flow needs to reserve (buffer, bandwidth, etc.).
  - Tspec defines the traffic characterization of the flow.

Admission

- After a router receives the flow specification from an application, it decides to admit or deny the service.
- The decision is based on the previous commitments of the router and the current availability of the resource.

Service Classes

Guaranteed Service Class

- This type of service is designed for real-time traffic that needs a guaranteed minimum end-to-end delay.
- The end-to-end delay is the sum of the delays in the routers, the propagation delay in the media, and the setup mechanism.

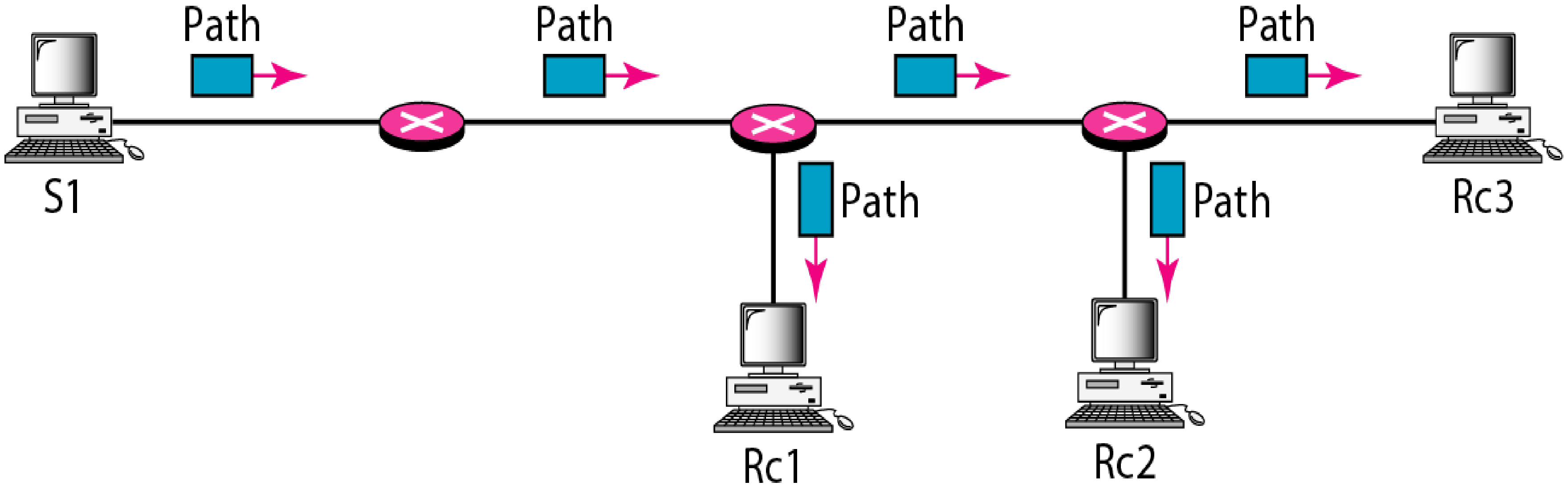
Controlled Load Service Class

- This type of service is designed for applications that can accept some delays, but are sensitive to an overloaded network and to the danger of losing packets.
- Example : file transfer, e-mail, and Internet access.

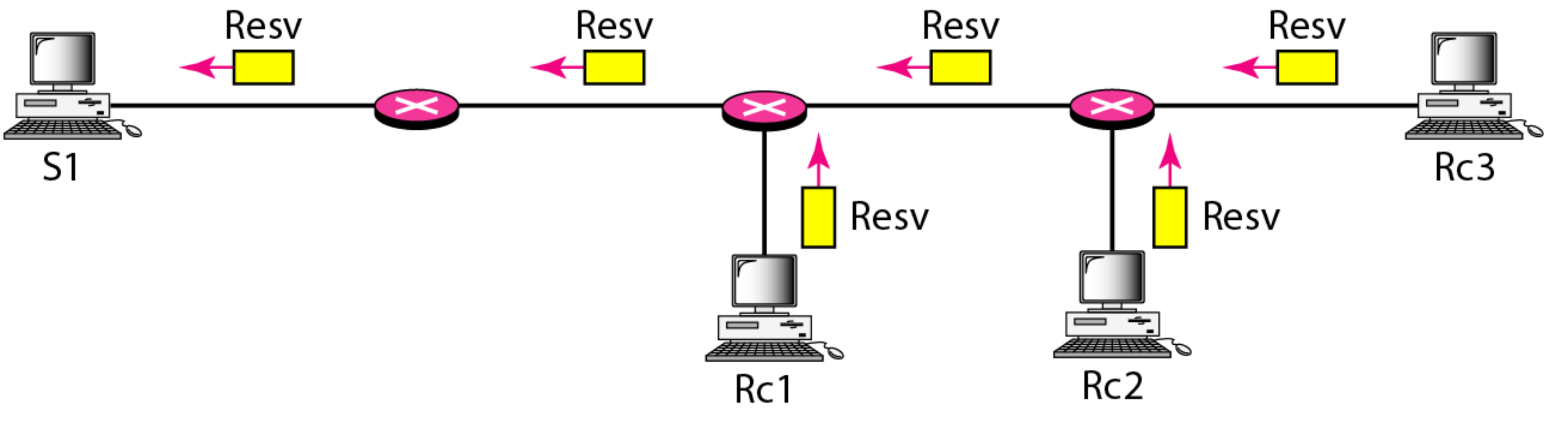
RSVP: (Resource Reservation Protocol)

- Multicast Trees
- Receiver – Based Reservation
- RSVP Messages
  - Path Messages
  - Resv Messages

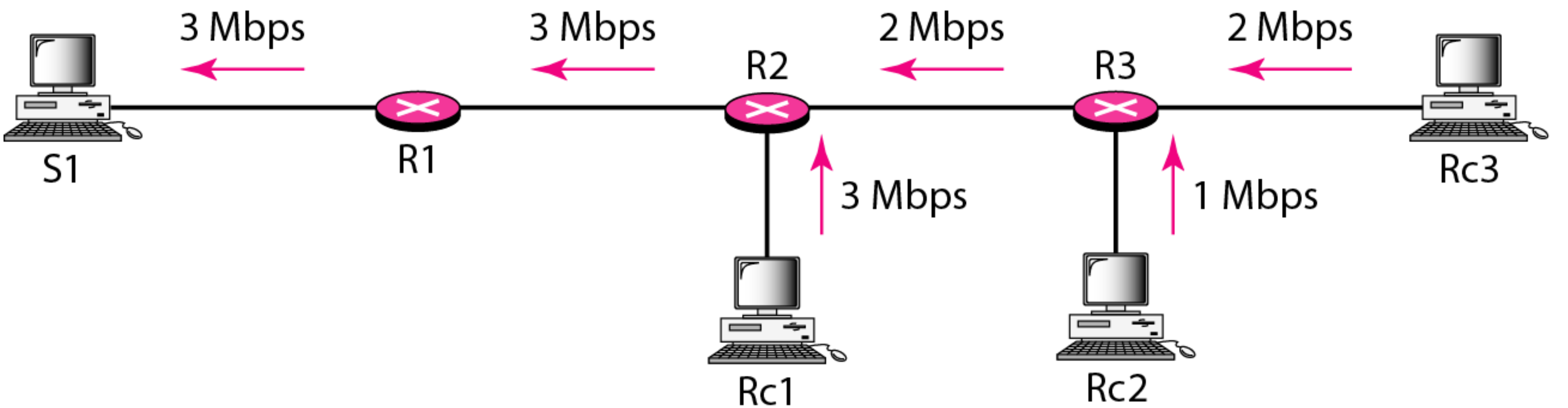
Path messages



Resv messages



Reservation merging



Problems with Integrated Services:

1. Scalability :

- The Integrated Services model requires that each router keep information for each flow.
- As the Internet is growing every day, this is a serious Problem.

2. Service-Type Limitation :

- The Integrated Services model provides only two types of services, guaranteed and control-load.
- Those opposing this model argue that applications may need more than these two types of services.

DIFFERENTIATED SERVICES

Differentiated Services (DS or Diffserv) was introduced by the IETF (Internet Engineering Task Force) to handle the shortcomings of Integrated Services.

Differentiated Services is a class-based QoS model designed for IP.

- In this model, packets are marked by applicatons into classes according to their priorities.
- Routers and Switches, usinf various queuing strategies, route the packets.

Two fundamental changes were made:

1. The main processing was moved from the core of the network to the edge of the network.
- This solves the scalability problem.
- The routers do not have to store information about flows.
- The applications, or hosts, define the type of service they need each time they send a packet.

2. The per-flow service is changed to per-class service.
- The router routes the packet based on the class of service defined in the packet, not the flow.
- This solves the service-type limitation problem.
- We can define different types of classes based on the needs of applications.

DS field

DS field contain 2 subfields:

1. DSCP (Differentiated Services Code Point): 6 bit, defines the Per-Hop Behavior (PHB).
2. CU (currently unused)

<u>Per-Hop Behavior:</u>

PHBs are defined:

1. DE PHB (Default PHB) :
2. EF PHB (Expedited Forwarding PHB):
3. AF PHB (Assured Forwarding PHB):

DE PHB:

- The DE PHB (default PHB) is the same as best-effort delivery, which is compatible with TOS.

EF PHB :

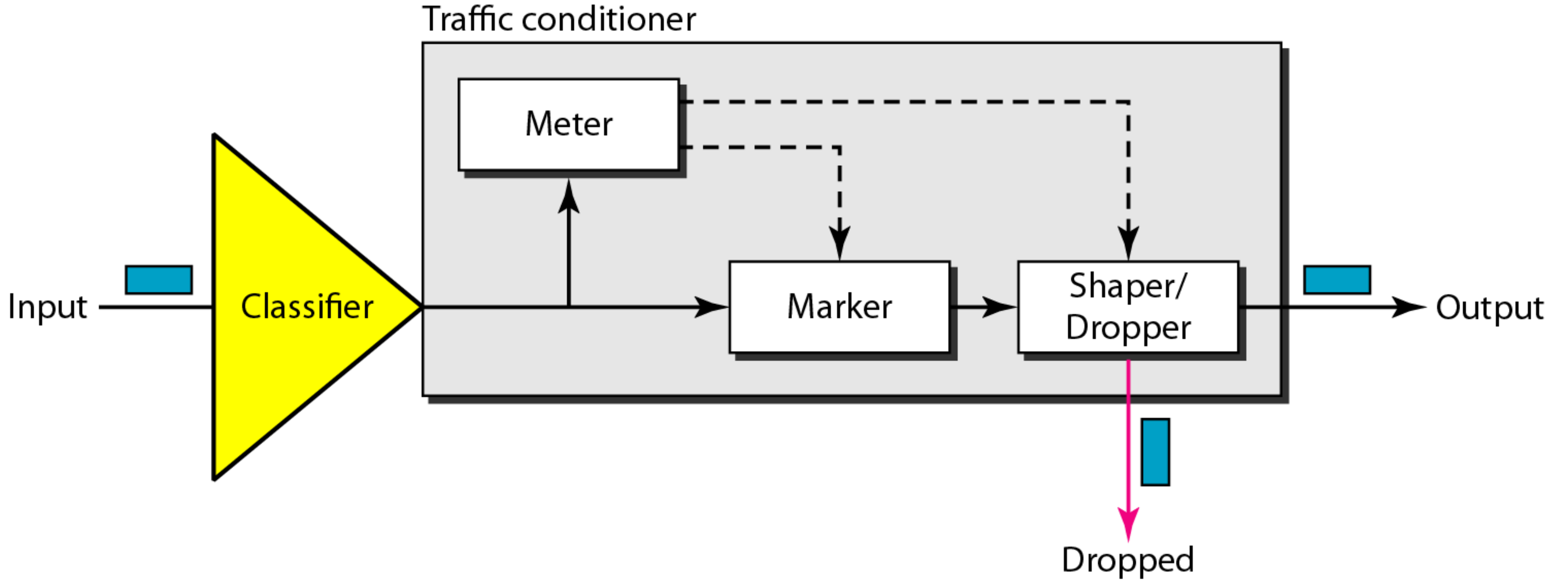
- The EF PHB provides the following services:
  - Low loss
  - Low latency
  - Ensured bandwidth
- This is the same as having a virtual connection between the source and destination.

AF PHB :

- The AF PHB delivers the packet with a high assurance as long as the class traffic does not exceed the traffic profile of the node.
- The users of the network need to be aware that some packetsmay be discarded.

Traffic conditioner

To implement Diffserv, the DS node uses traffic conditioners :



Meter :

- It checks to see if the incoming flow matches the negotiated traffic profile.
- Meter sends this result to other components.
- Meter can use several tools such as a token bucket to check the profile.

Marker :

- It re-mark a packet that is using best effort delivery or down-mark a packet based on information received from the meter.
- It down-mark if the flow does not match the profile..

Shaper:

- It reshape the traffic if it is not compliant with the negotiated profile.

Dropper:

- It discards packets if the flow severely violets the negotiated profile.