SNS COLLEGE OF TECHNOLOGY

Coimbatore-35
An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with ‘A+’ Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

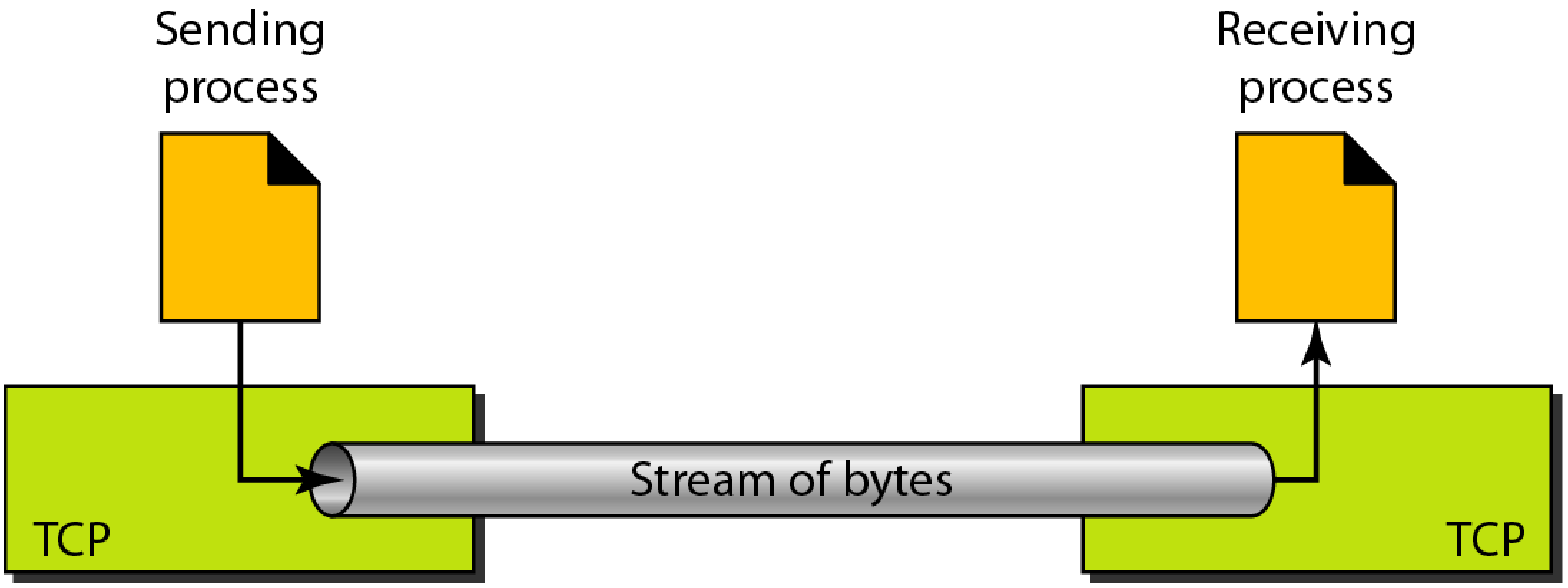
DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

23AMB302- COMPUTER NETWORKS & SECURITY

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.

Stream Delivery Service

- TCP is a stream-oriented protocol.
- TCP allows the sending process to deliver data as a stream of bytes and allows the receiving process to obtain data as a stream of bytes.
- TCP creates an environment in which the two processes seem to be connected by an imaginary “tube” that carries their bytes across the Internet.
- The sending process produces (writes to) the stream and the receiving process consumes (reads from) it.


Sending process
Receiving process
Stream of bytes
TCP
TCP

Full-Duplex Communication

- TCP offers full-duplex service, where data can flow in both directions at the same time.
- Each TCP endpoint then has its own sending and receiving buffer, and segments move in both directions.

Multiplexing and Demultiplexing

- TCP performs multiplexing at the sender and demultiplexing at the receiver.

Connection-Oriented Service

- TCP is a connection-oriented protocol.
- A connection needs to be established for each pair of processes.
- When a process at site A wants to send to and receive data from another process at site B, the following three phases occur:

1. The two TCP's establish a logical connection between them.

2. Data are exchanged in both directions.
3. The connection is terminated.

Reliable Service

- TCP is a reliable transport protocol.
- It uses an acknowledgment mechanism to check the safe and sound arrival of data.

TCP SEGMENT

A packet in TCP is called a segment.

Data unit exchanged between TCP peers are called ***segments*.**

A TCP segment encapsulates the data received from the application layer.

The TCP segment is encapsulated in an IP datagram, which in turn is encapsulated in a frame at the data-link layer.

TCP is a byte-oriented protocol, which means that the sender writes bytes into a TCP connection and the receiver reads bytes out of the TCP connection.

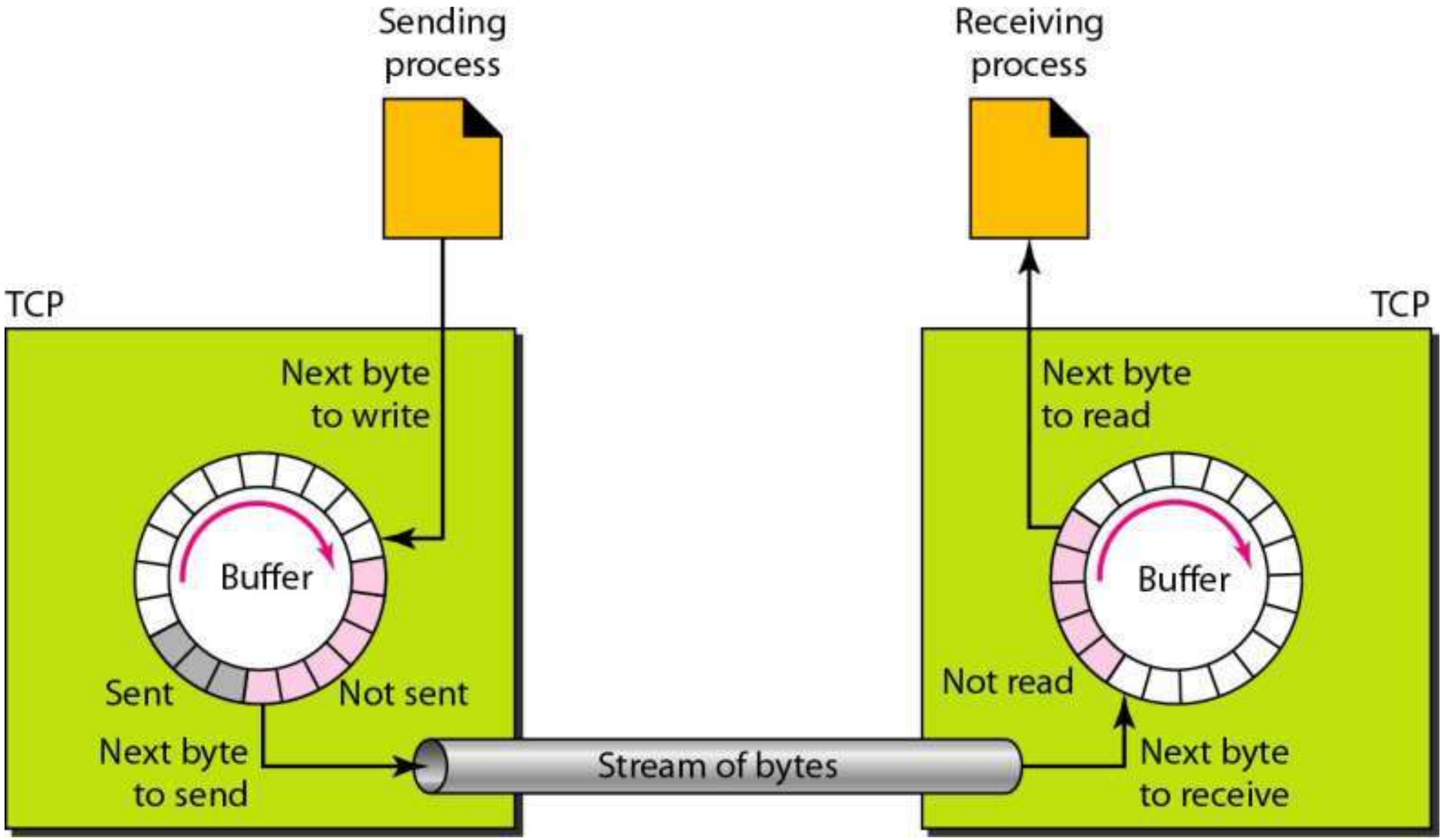
TCP on the source host buffers enough bytes from the sending process to fill a reasonably sized packet and then sends this packet to its peer on the destination host.

TCP on the destination host then empties the contents of the packet into a receive buffer, and the receiving process reads from this buffer at its leisure.

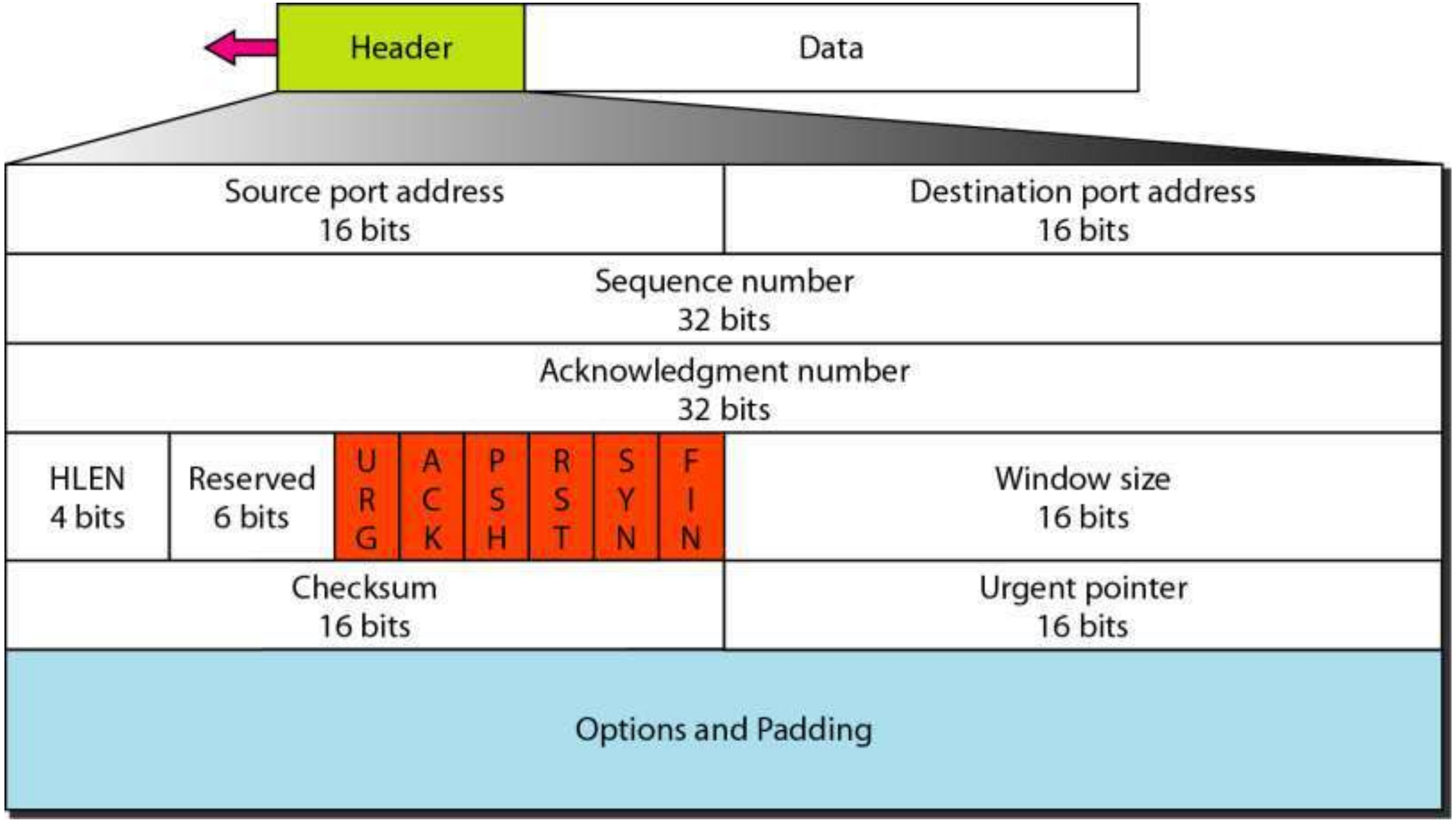
TCP connection supports byte streams flowing in both directions.

The packets exchanged between TCP peers are called segments, since each one carries a segment of the byte stream.

Sending and receiving buffers



TCP Segment Format



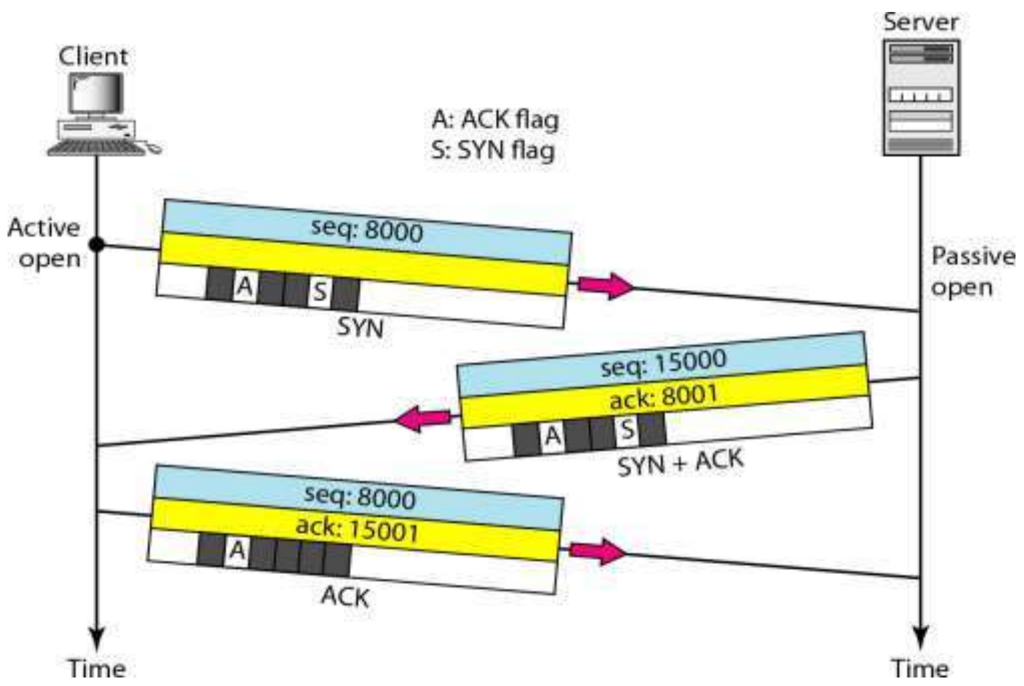
Flag	*Description*
URG	The value of the urgent pointer field is valid.
ACK	The value of the acknowledgment field is valid.
PSH	Push the data.
RST	Reset the connection.
SYN	Synchronize sequence numbers during connection.
FIN	Terminate the connection.

- **Advertised Window**—defines receiver’s window size and acts as flow control.
- **Checksum**—It is computed over TCP header, Data, and pseudo header containing IP fields (Length, SourceAddr & DestinationAddr).
- **UrgPtr** — used when the segment contains urgent data. It defines a value that must be added to the sequence number.
  Options - There can be up to 40 bytes of optional information in the TCP header.

TCP CONNECTION MANAGEMENT

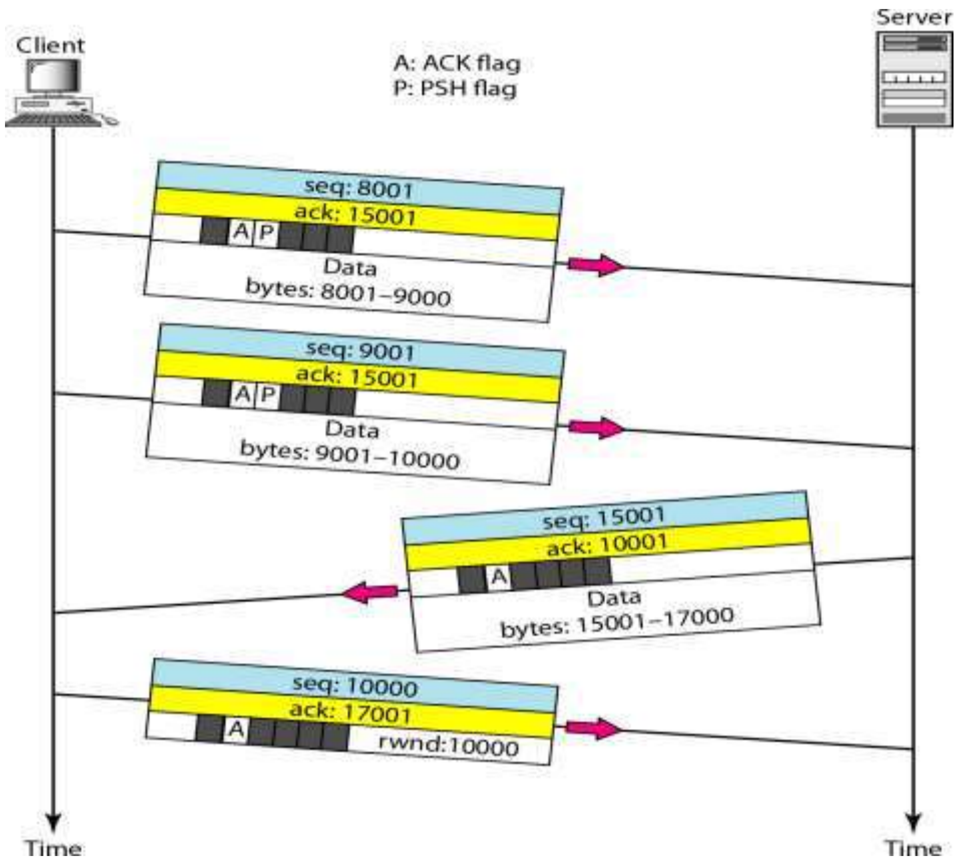
- TCP is connection-oriented.
- A connection-oriented transport protocol establishes a logical path between the source and destination.
- All of the segments belonging to a message are then sent over this logical path.
- In TCP, connection-oriented transmission requires three phases:
- **Connection Establishment, Data Transfer and Connection Termination.**

Connection establishment using three-way handshaking



1. Client sends a SYN segment to the server containing its initial sequence number
2. Server responds with a segment that acknowledges client's segment and specifies its initial sequence number
3. Finally, client responds with a segment that acknowledges server's sequence number

Data Transfer



- ⍰ After connection is established, bidirectional data transfer can take place.
- ⍰ The client and server can send data and acknowledgments in both directions.
- ⍰ The data traveling in the same direction as an acknowledgment are carried on the same segment.
- ⍰ The acknowledgment is piggybacked with the data.

Connection Termination

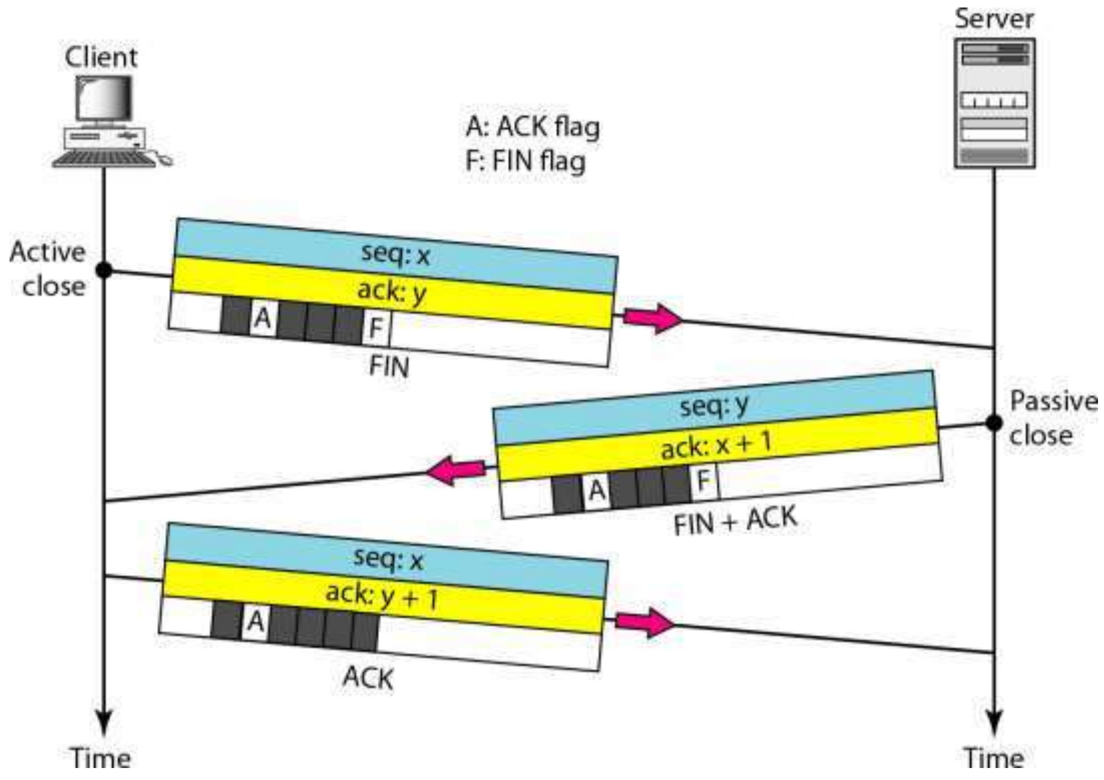
Connection termination or teardown can be done in two ways :

- **Three-way Close**
- **Half-Close**

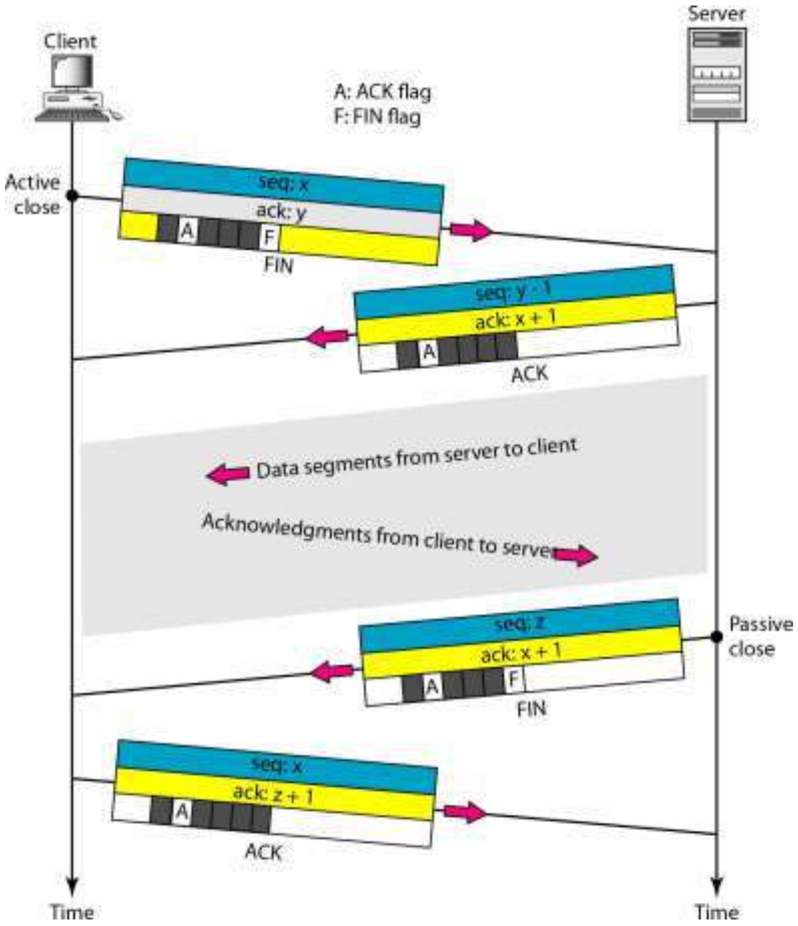
Three-way Close

Both client and server close *simultaneously.*

- Client sends a **FIN(Terminates the connection)** segment.
- The FIN segment can include last chunk of data.
- Server responds with FIN + ACK segment to inform its closing.
- Finally, client sends an ACK segment

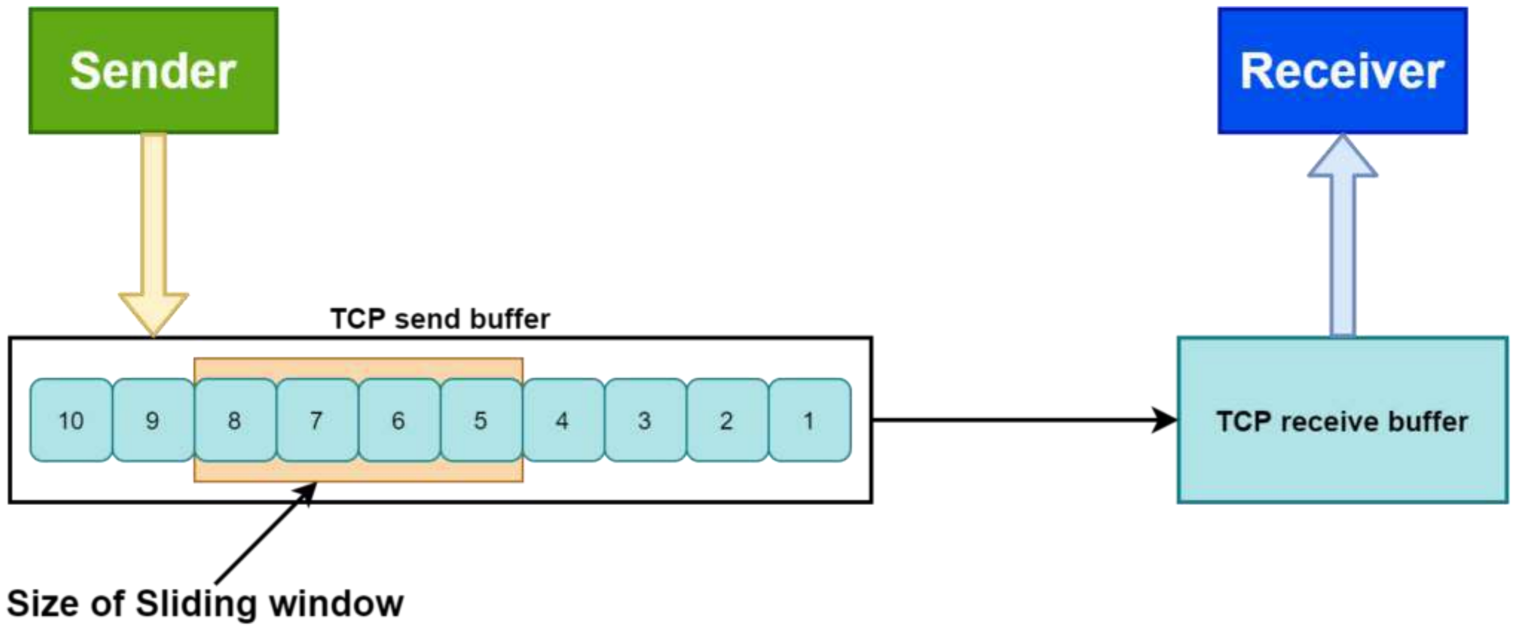


Half close



- Client half-closes the connection by sending a FIN segment.
- Server sends an ACK segment.
- Data transfer from client to the server *stops*.
- After sending all data, server sends FIN segment to client, which is acknowledged by the client.

TCP FLOW CONTROL

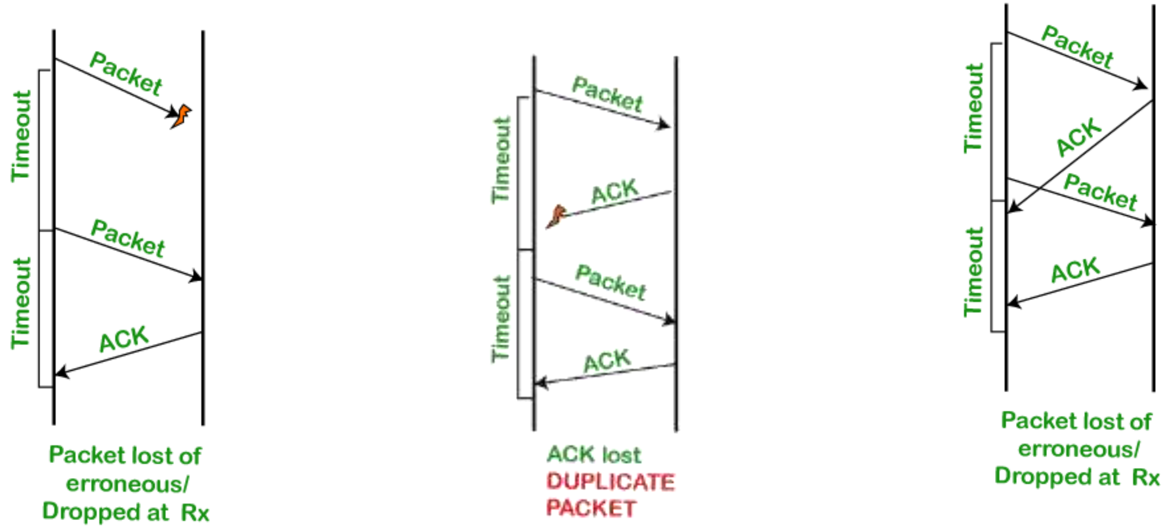


A sliding window is used to make transmission more efficient as well as to control the flow of data so that the destination does not become overwhelmed with data.TCP sliding windows are byte oriented

- The TCP stores the data that needs to be sent in the send buffer and the data to be received in the receive buffer.
- Flow control makes sure that no more packets are sent by the sender once the receiver's buffer is full as the messages will be dropped and the receiver won't be able to handle them.
- In order to control the amount of data sent by the TCP, the receiver will create a buffer which is also known as Receive Window.

Retransmission

- The TCP retransmission means resending the packets over the network that have been either lost or damaged.



Data packet is lost | Acknowledgement is lost | Early timeout

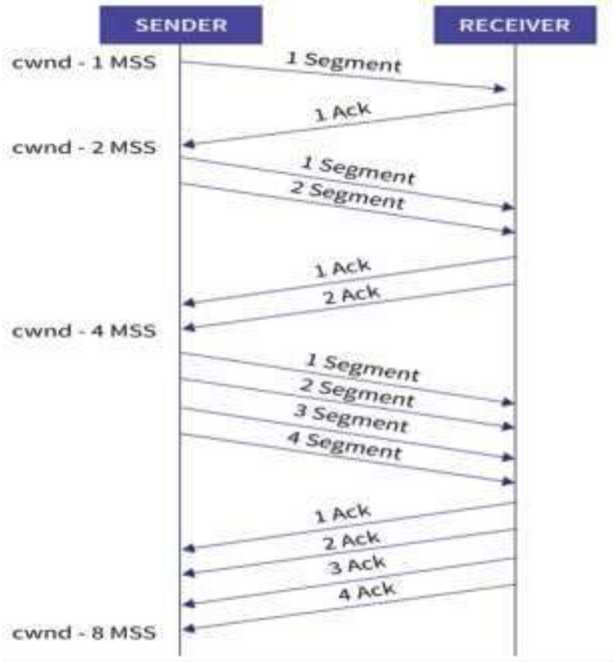
TCP Congestion Control

Congestion occurs if load (number of packets sent) is greater than capacity of the network (number of packets a network can handle).

- When too many packets are contending for the same link
- When load exceeds capacity, queues become full and the routers discard some packets and throughput declines sharply.
- ○ The queue overflows ○ Packets get dropped ○ Network is congested

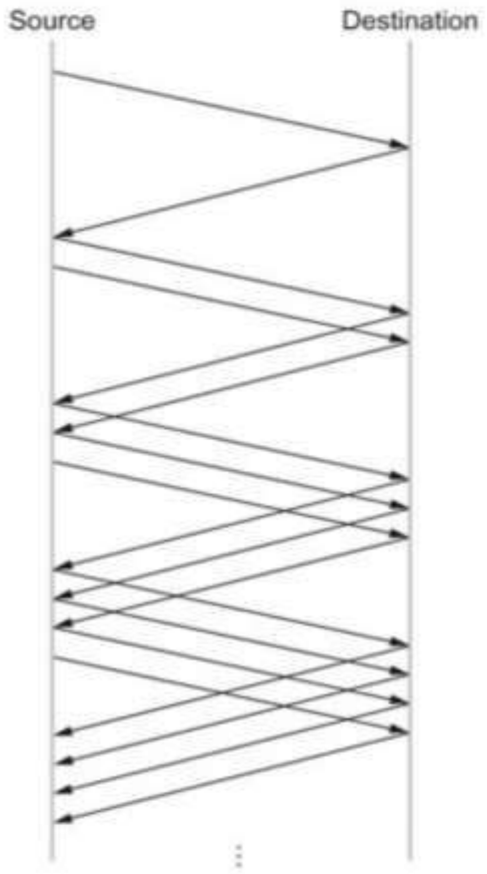
Slow Start Phase

- The sender sets congestion window size = maximum segment size (1 MSS) at the initial stage. The size of the congestion window increases **exponentially** in this phase.
- Slow start is repeated until CongestionWindow reaches **CongestionThreshold** and thereafter 1 packet per RTT.

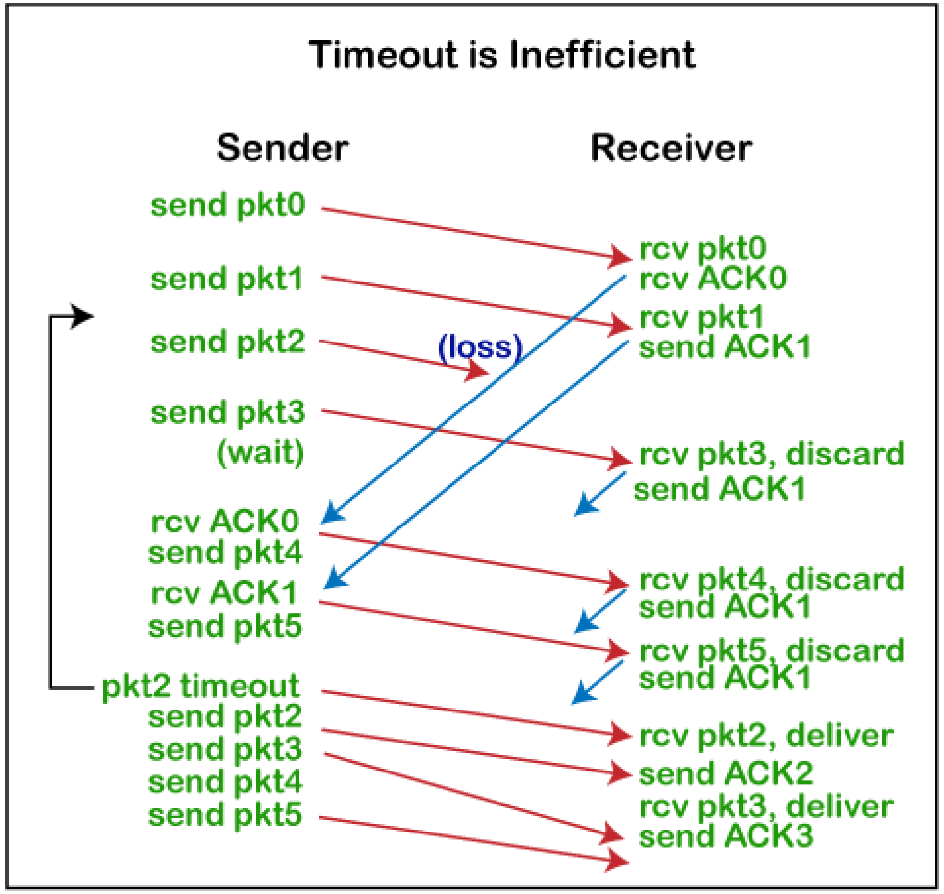


Additive Increase / Multiplicative Decrease (AIMD)

- TCP source ***initializes*** CongestionWindow based on congestion level in the network.
- Source ***increases*** CongestionWindow when level of congestion goes down and ***decreases*** the same when level of congestion goes up.
- TCP interprets ***timeouts*** as a sign of congestion and reduces the rate of transmission.
- On timeout, source reduces its CongestionWindow by **half,** i.e., *multiplicative decrease*. For example, if CongestionWindow = 16 packets, after timeout it is 8.
- When ACK arrives CongestionWindow is *incremented* marginally, i.e., ***additive increase***.



Retransmit And Fast Recovery



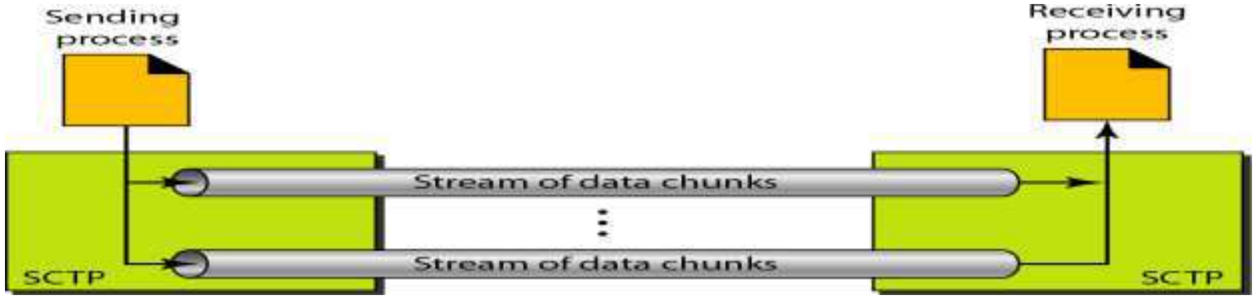
- When a packet arrives **out of order**, receiving TCP resends the same acknowledgment (***duplicate ACK***) it sent last time.
- When ***three duplicate*** ACK arrives at the sender, it infers that corresponding packet may be lost due to congestion and retransmits that packet. This is called ***fast retransmit*** before regular timeout.

SCTP (Stream Control Transmission Protocol)

- Stream Control Transmission Protocol (SCTP) is a reliable, message-oriented transport layer protocol.
- SCTP has mixed features of TCP and UDP.
- SCTP provides the Congestion control as well as Flow control.
- SCTP is especially designed for internet applications as well as multimedia communication.

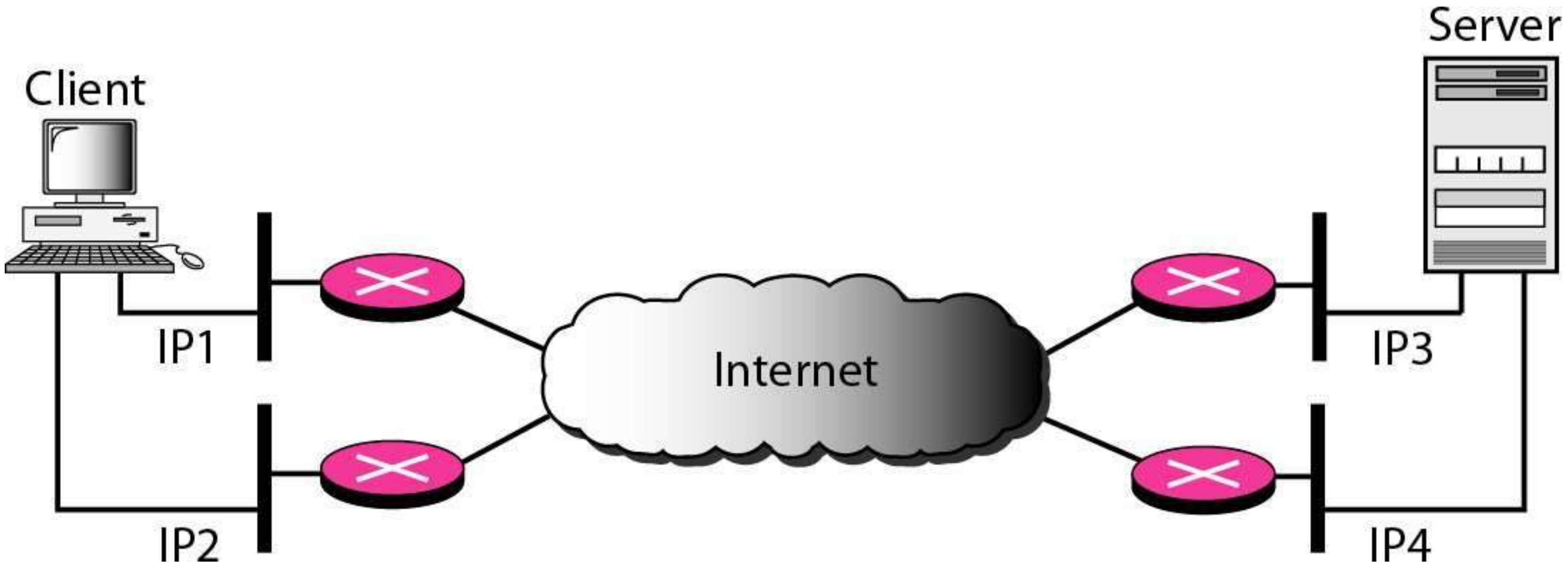
SCTP SERVICES

- **Process-to-Process Communication**
- SCTP provides process-to-process communication.
- **Multiple Streams**
- SCTP allows multistream service in each connection, which is called ***association*** in SCTP terminology.
- If one of the streams is blocked, the other streams can still deliver their data.



Multihoming

1. An SCTP association supports multihoming service.
2. The sending and receiving host can define multiple IP addresses in each end for an association.
3. In this fault-tolerant approach, when one path fails, another interface can be used for data delivery without interruption.



Full-Duplex Communication

- SCTP offers full-duplex service, where data can flow in both directions at the same time. Each SCTP then has a sending and receiving buffer and packets are sent in both directions.

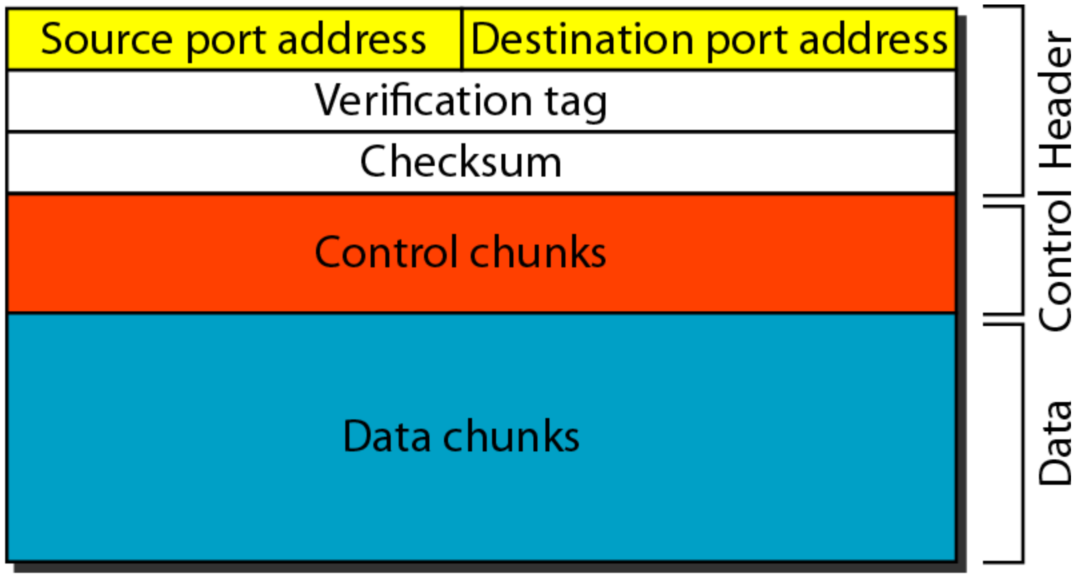
Connection-Oriented Service

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

Reliable Service

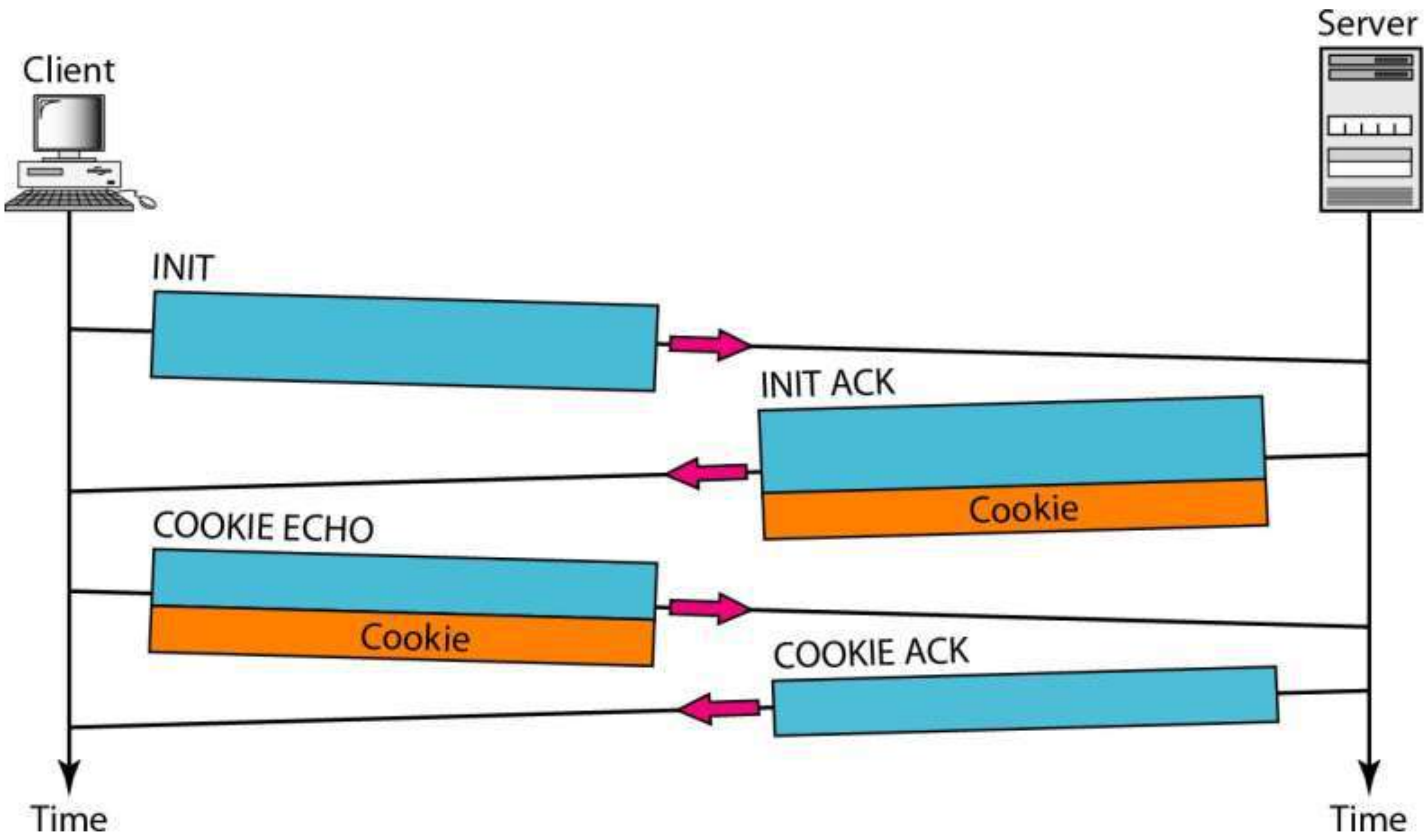
- SCTP is a reliable transport protocol.
- It uses an acknowledgment mechanism to check the safe and sound arrival of data.

SCTP Packet format

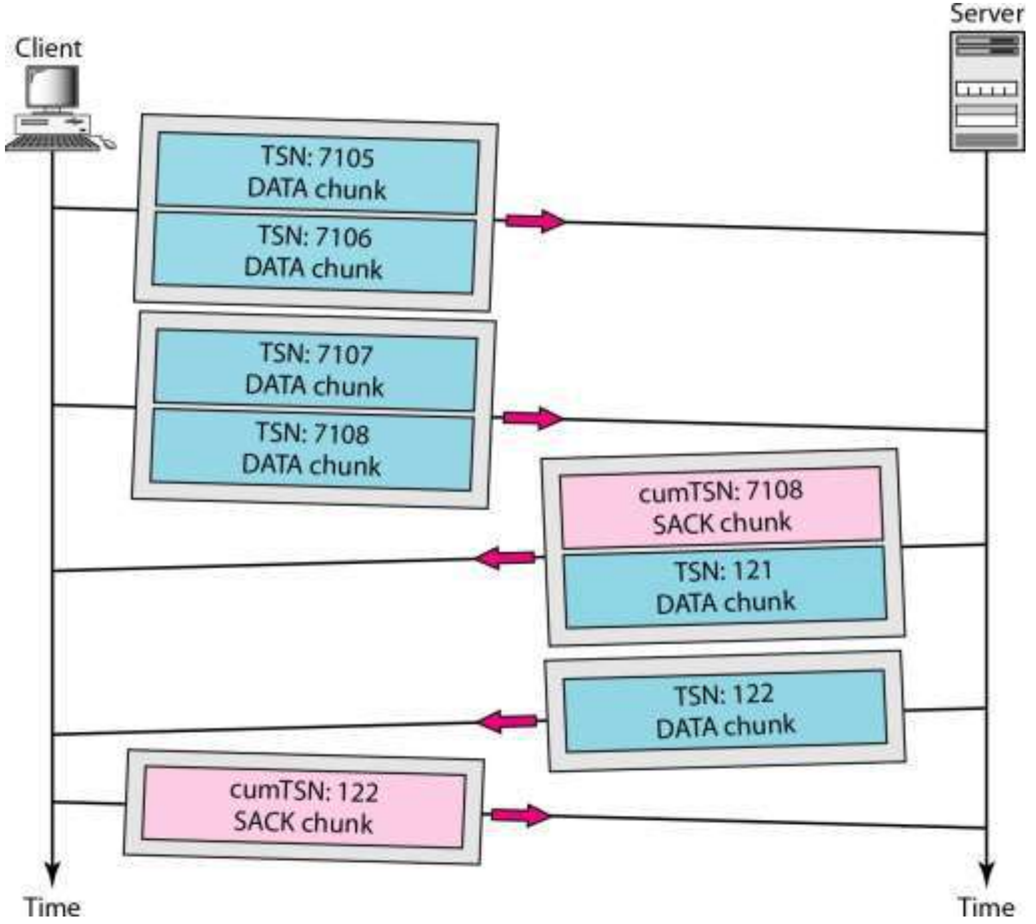


A packet in SCTP

Four -way Handshaking

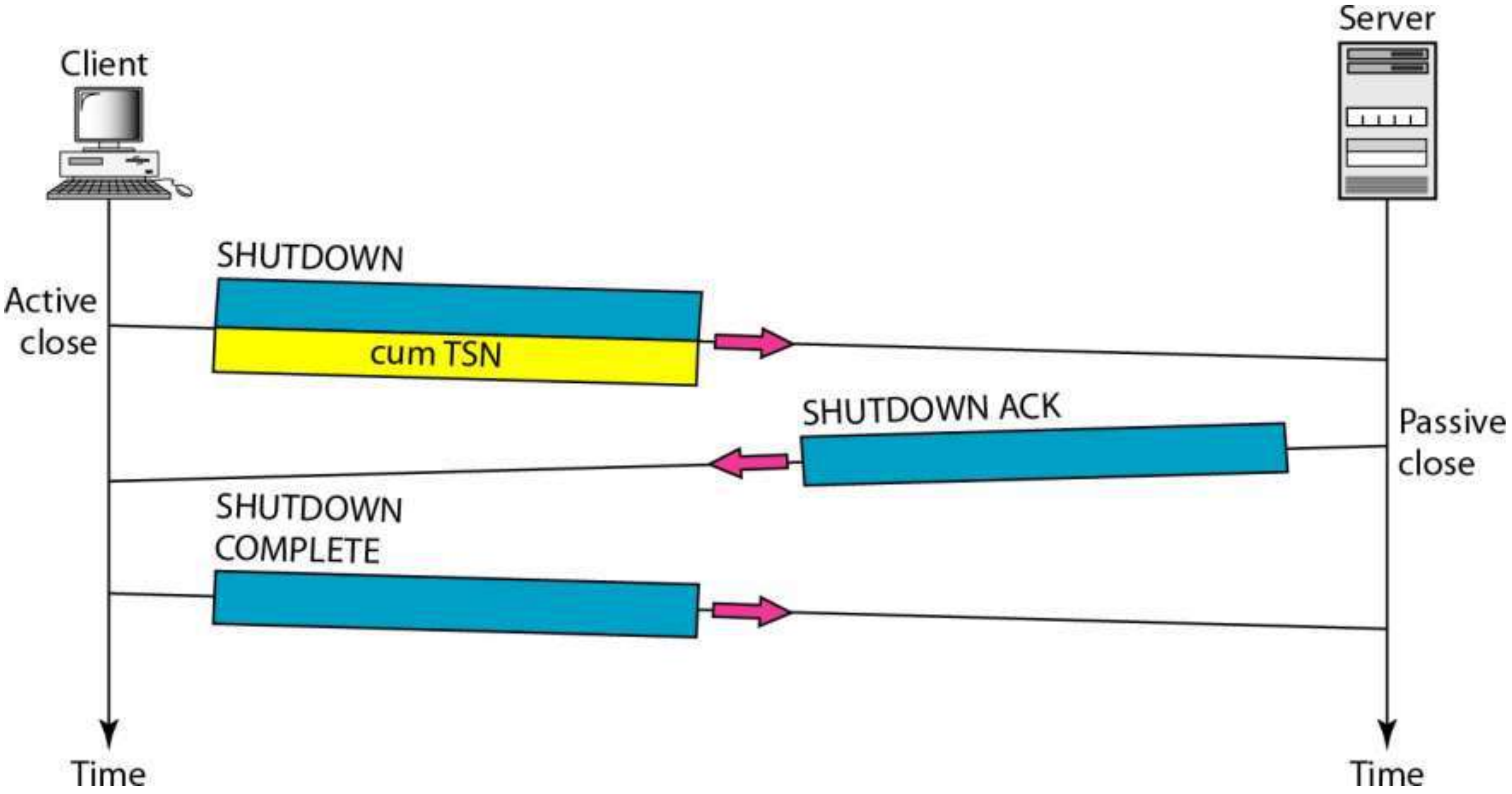


Data Transfer

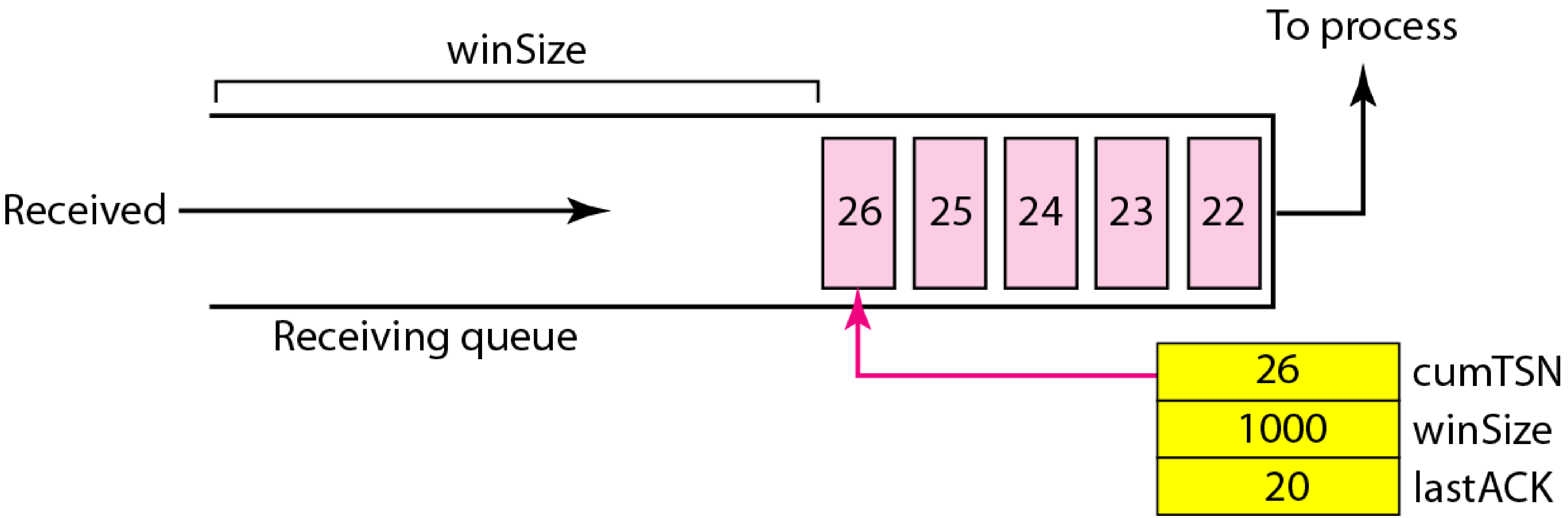


The acknowledgment in SCTP defines the cumulative TSN, the TSN of the last data chunk received in order.

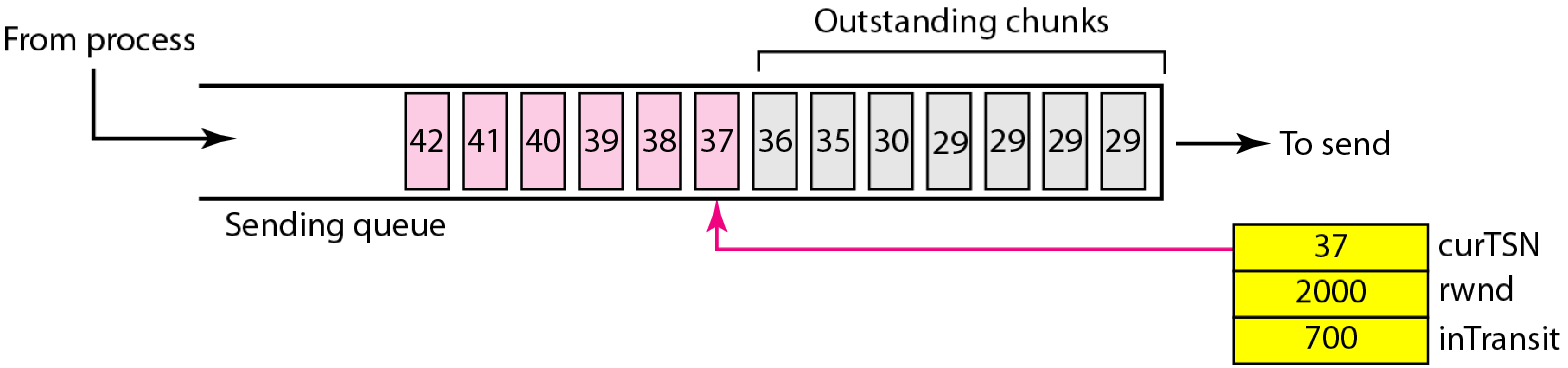
Association termination



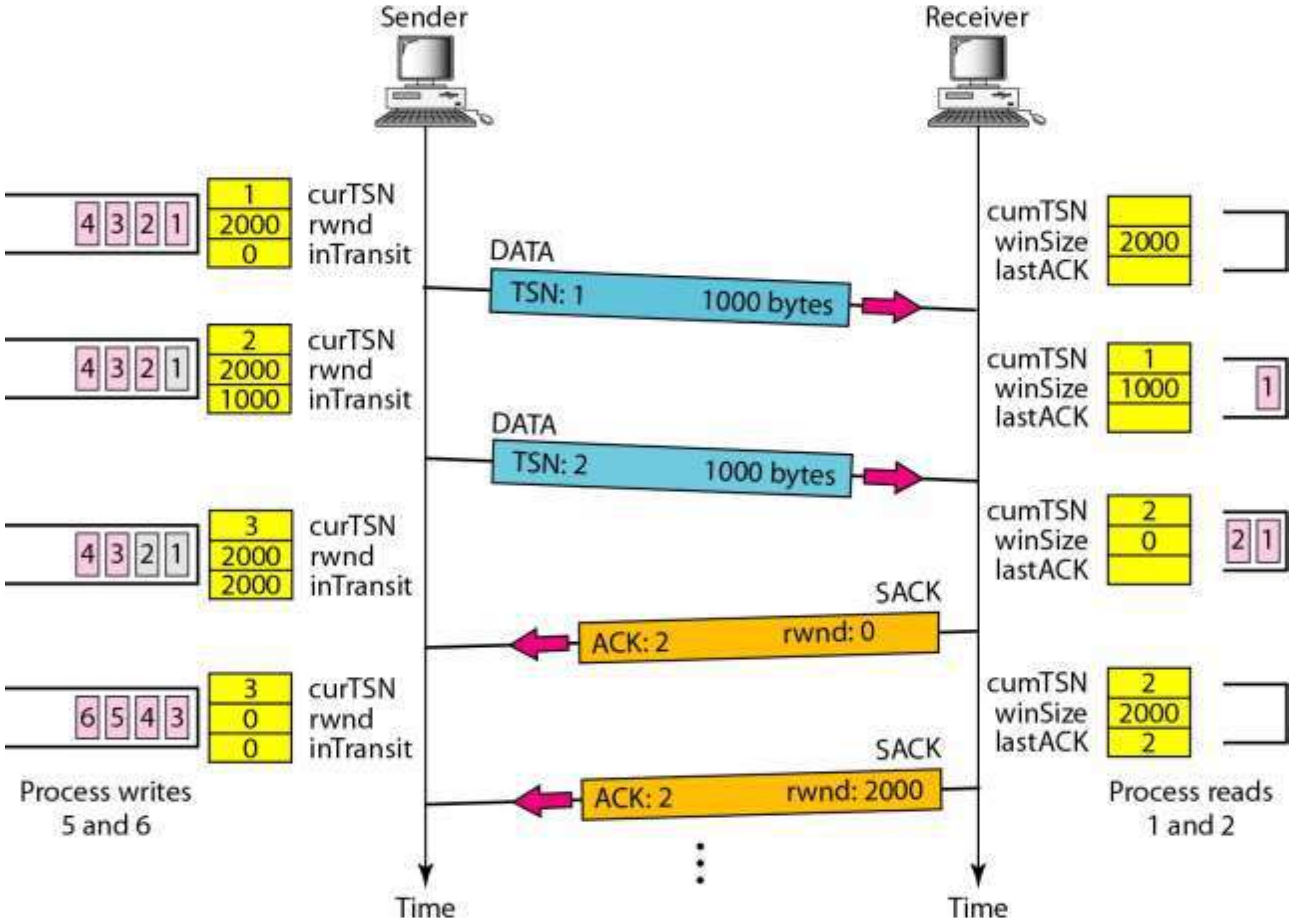
Flow control , Receiver site



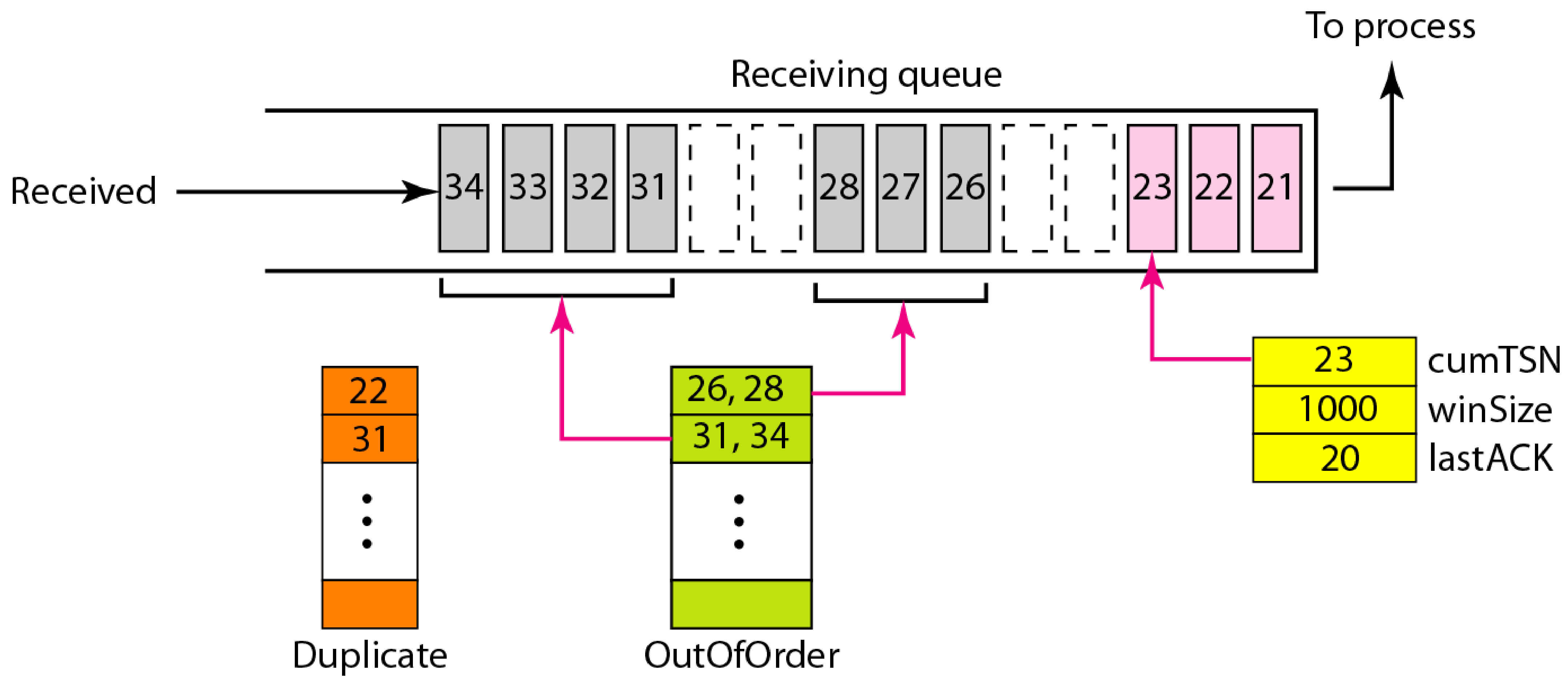
Flow control,Sender site



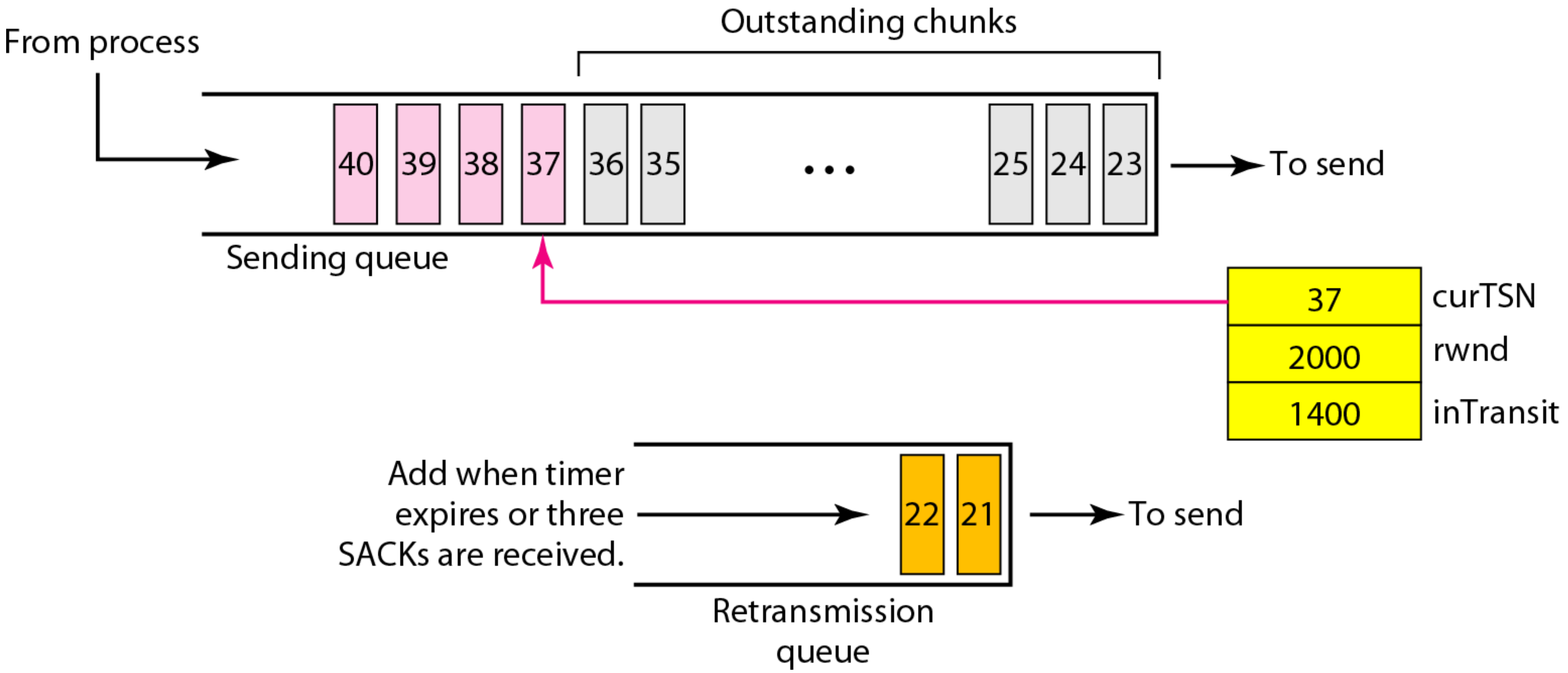
Flow control scenario



Error Control, Receiver site



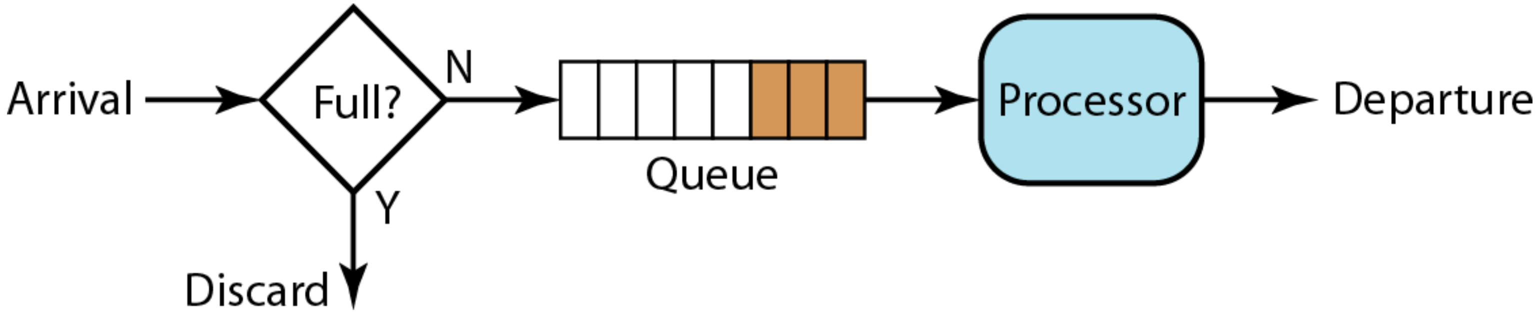
Error control, sender site



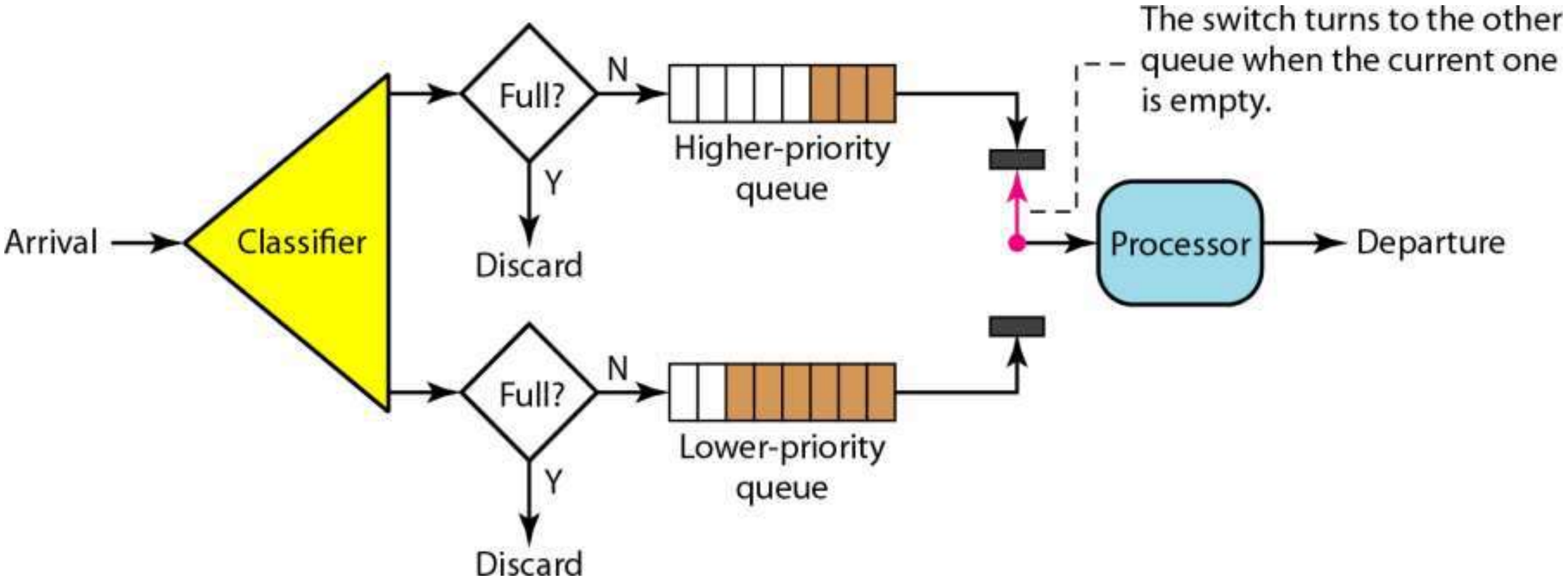
QOS

Quality of Service (QOS) determines a network's capability to support predictable service over various technologies

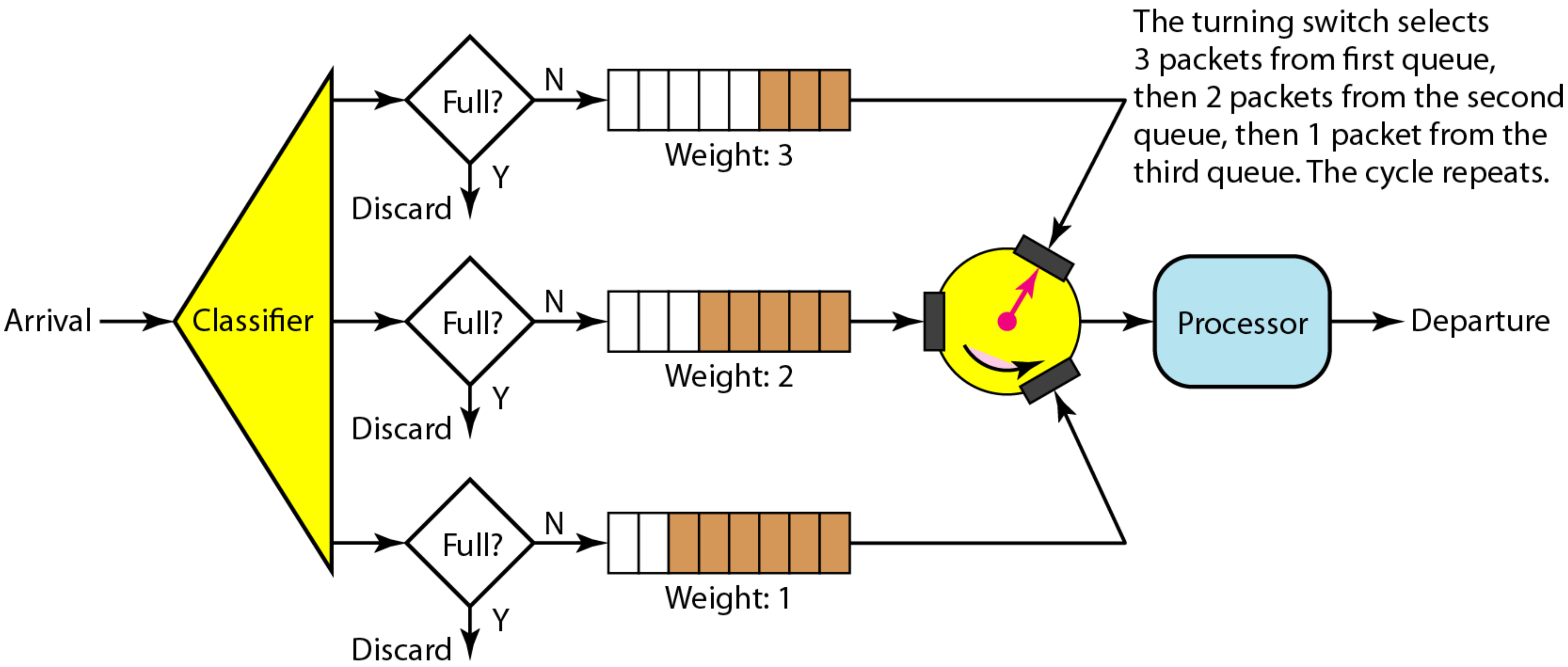
FIFO Queue



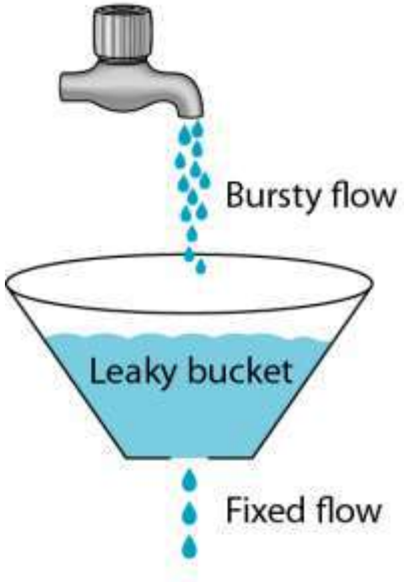
Priority Queuing

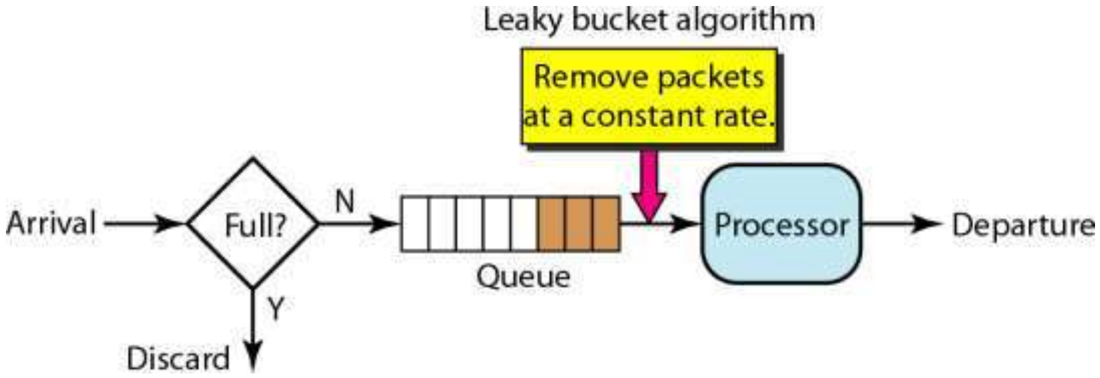


Weighted Fair Queuing



Leaky Bucket





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.