CS321: Computer Networks

Introduction to Transport Layer

Dr. Manas Khatua
Assistant Professor
Dept. of CSE
IIT Jodhpur
E-mail: manaskhatua@iitj.ac.in

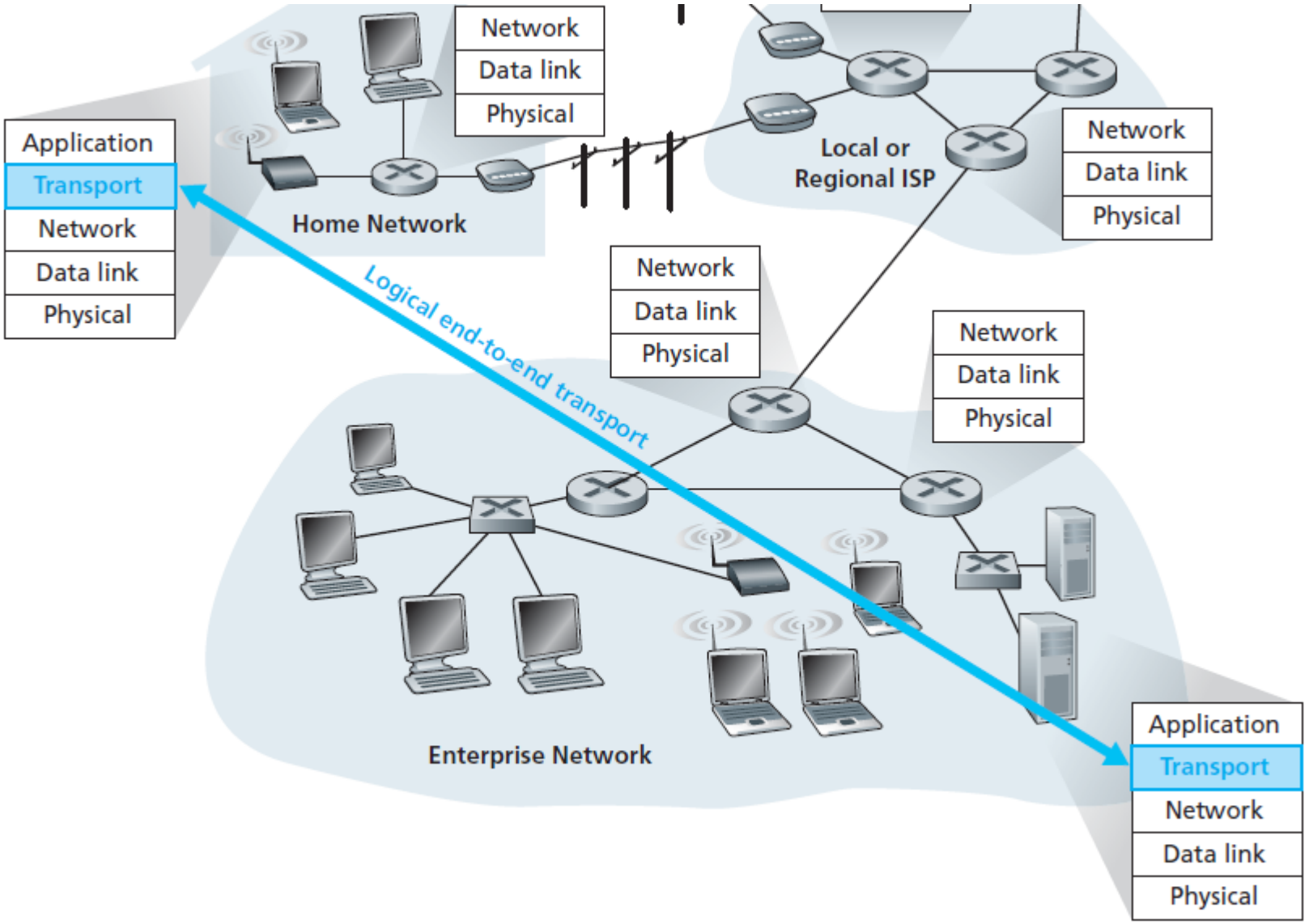
Introduction

- In TCP/IP suite, it provides services to the application layer and receives services from the network layer.
- General Services
  - process-to-process connection
  - addressing
  - multiplexing and de-multiplexing
  - error, flow, and congestion control

- Transport-Layer Protocol strategies
  - Simple Protocol
  - Stop-and-Wait
  - Go-Back-N
  - Selective-Repeat

- Transport-Layer Protocols for the Internet
  - Connection less protocol: UDP
  - Connection oriented protocol : TCP

Logical end-to-end trasport



Network vs Transport Layer

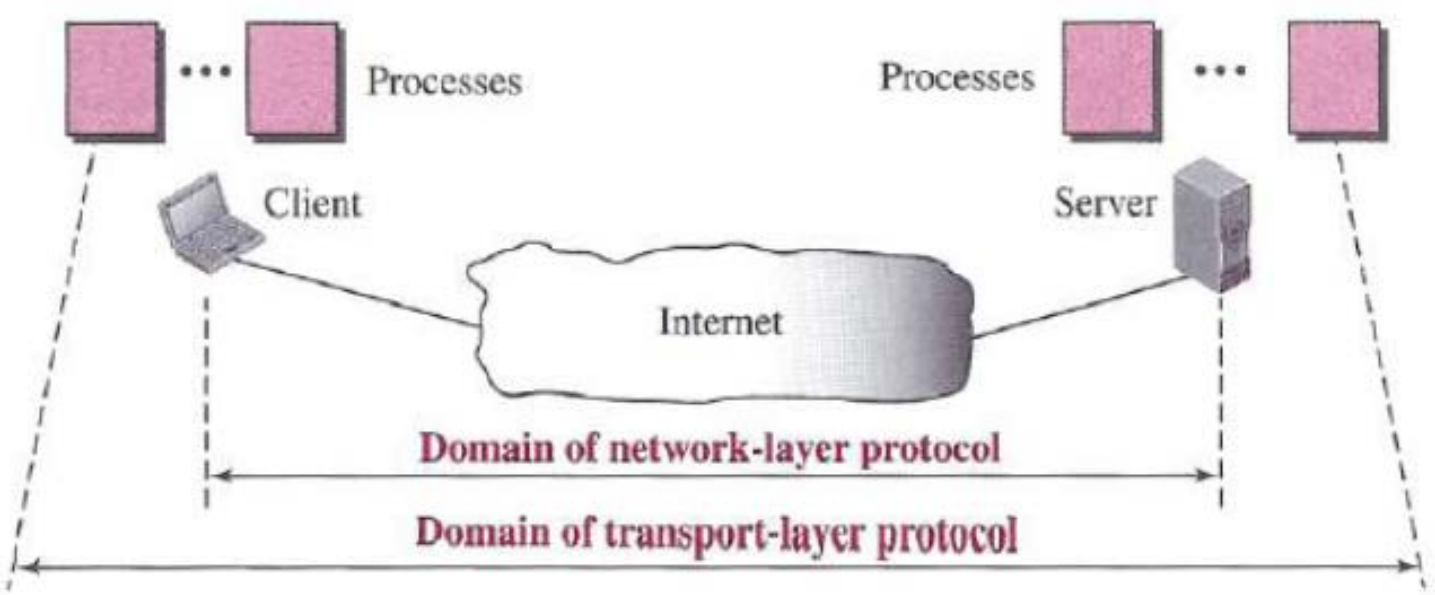
- transport-layer protocols are implemented in the end systems but not in network routers

- IP provides communication between hosts
- TCP/UDP provides communication between processes

- transport-layer packets: segments
- network-layer packet: datagram

- IP service model: best-effort delivery but unreliable service
- TCP/UDP service model: reliable data transfer

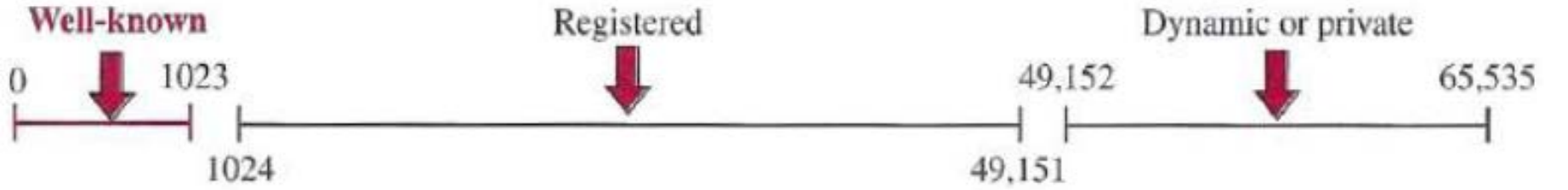
Process-to-Process Communication



- One method for this: client-server approach
- Host is identified by IP address
- Process is identified by port number
- In TCP/IP: port numbers 0-65535 (16 bits)
  - Client uses: ephemeral ports (short-lived ports, >1023)
  - Server Uses: well-known ports

ICANN Ranges

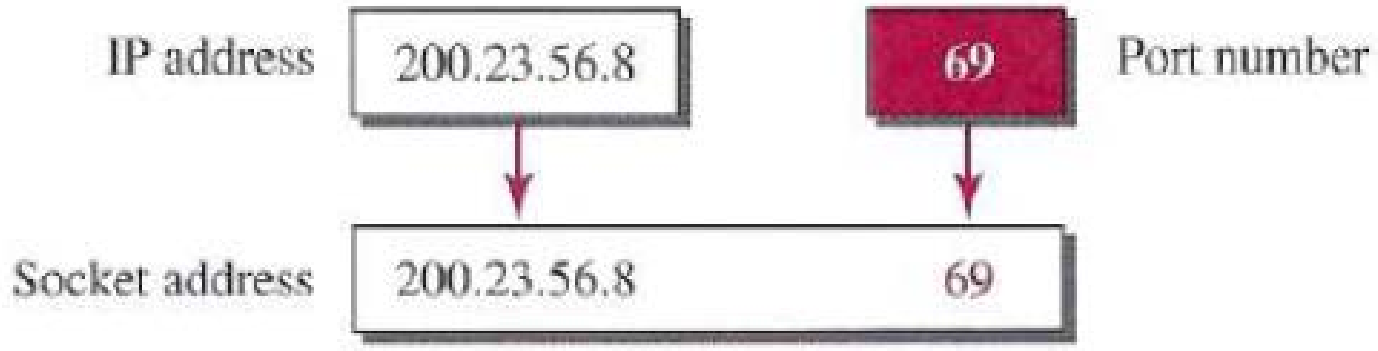
- Internet Corporation for Assigned Names and Numbers (ICANN)



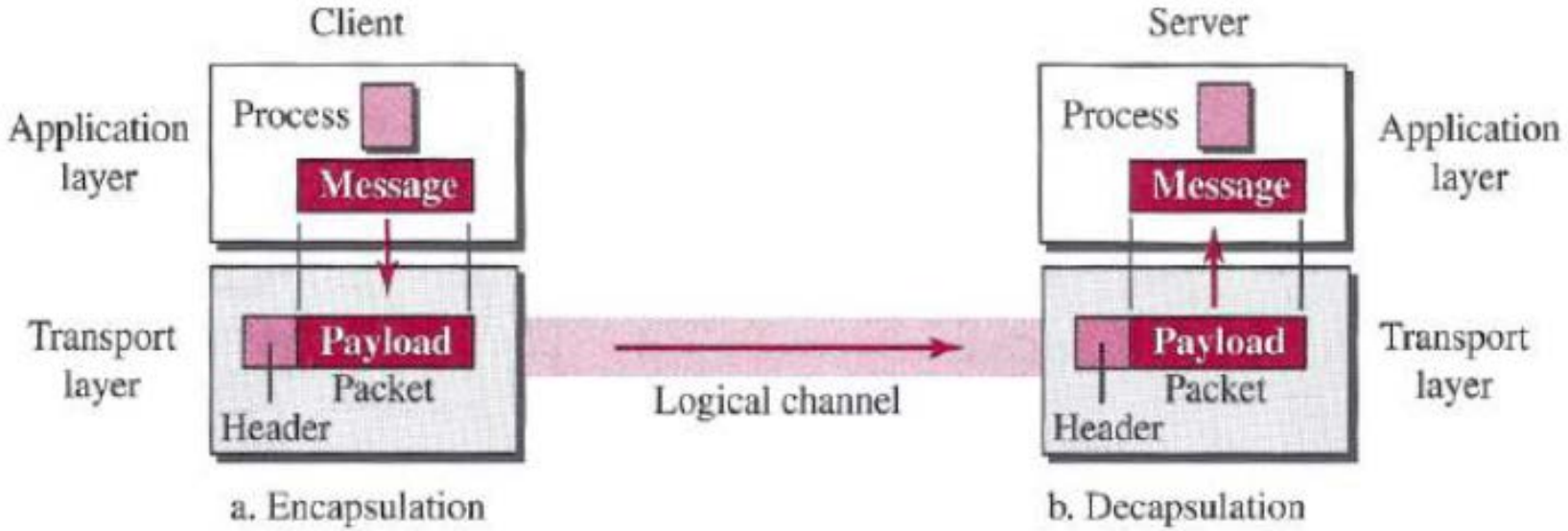
- Example:
  - Well-known ports are stored in /etc/services
  - FTP : 20, 21
  - SSH, SCP : 22
  - Echo : 7
  - DNS : 53
  - HTTP : 80
  - SNMP : 161,162
  - BGP : 179

Socket Address

- To use the services of the transport layer in the Internet, we need a pair of socket addresses:
  - the client socket address
  - the server socket address



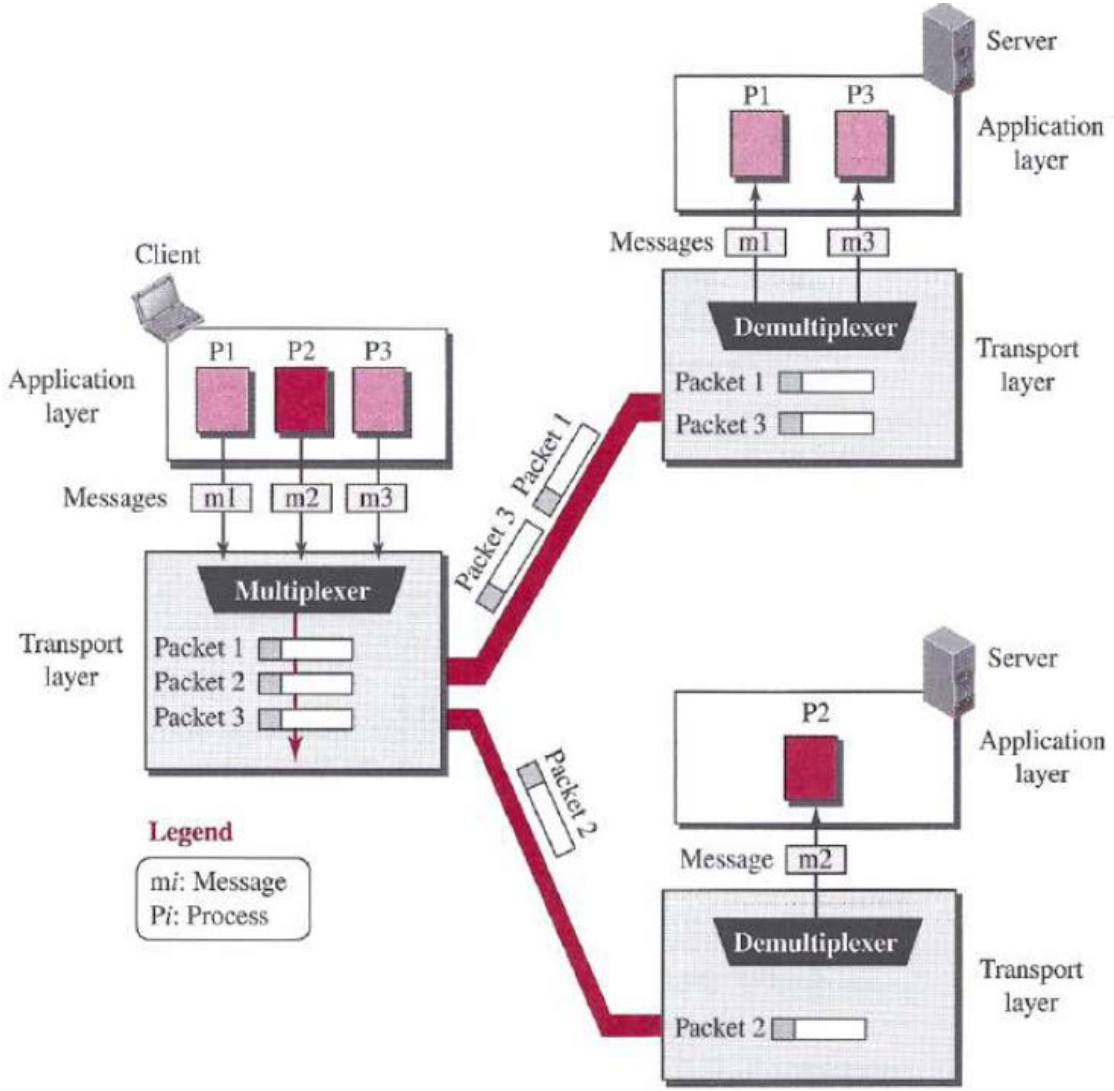
Encapsulation and Decapsulation



Multiplexing and Demultiplexing

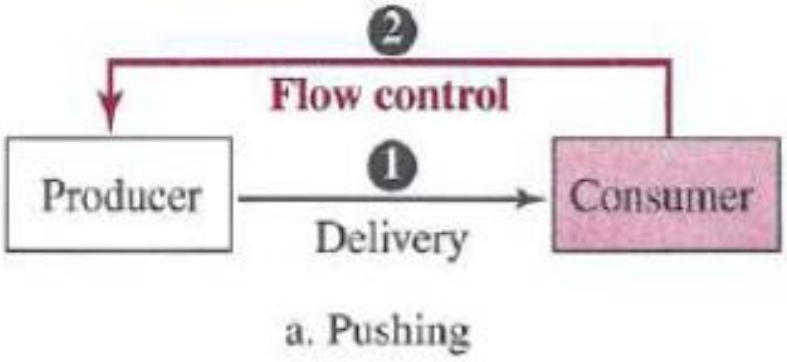
- Suppose you are sitting in front of a computer, and you are browsing Web pages while running one FTP session and two Telnet sessions.
- So, 4 processes
  - HTTP
  - FTP
  - 2 Telnet

- How does a segment forwarded to intended process?
  - using Socket Address
  - Method is called multiplexing and demultiplexing

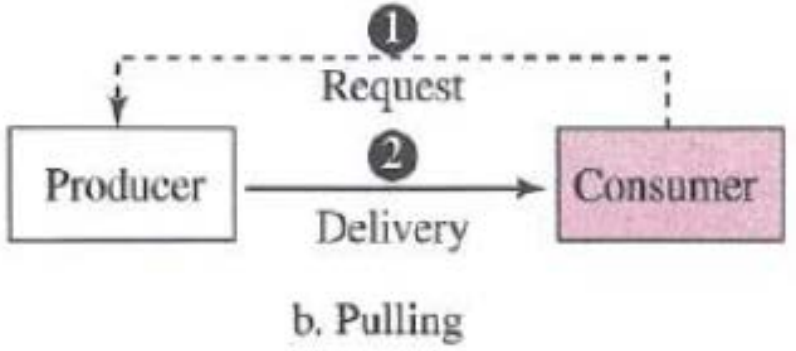


Flow Control

- Delivery of an item follows two approaches:
  - Pushing
  - Pulling
- Flow control need when a consumer is overwhelmed by the receiving items

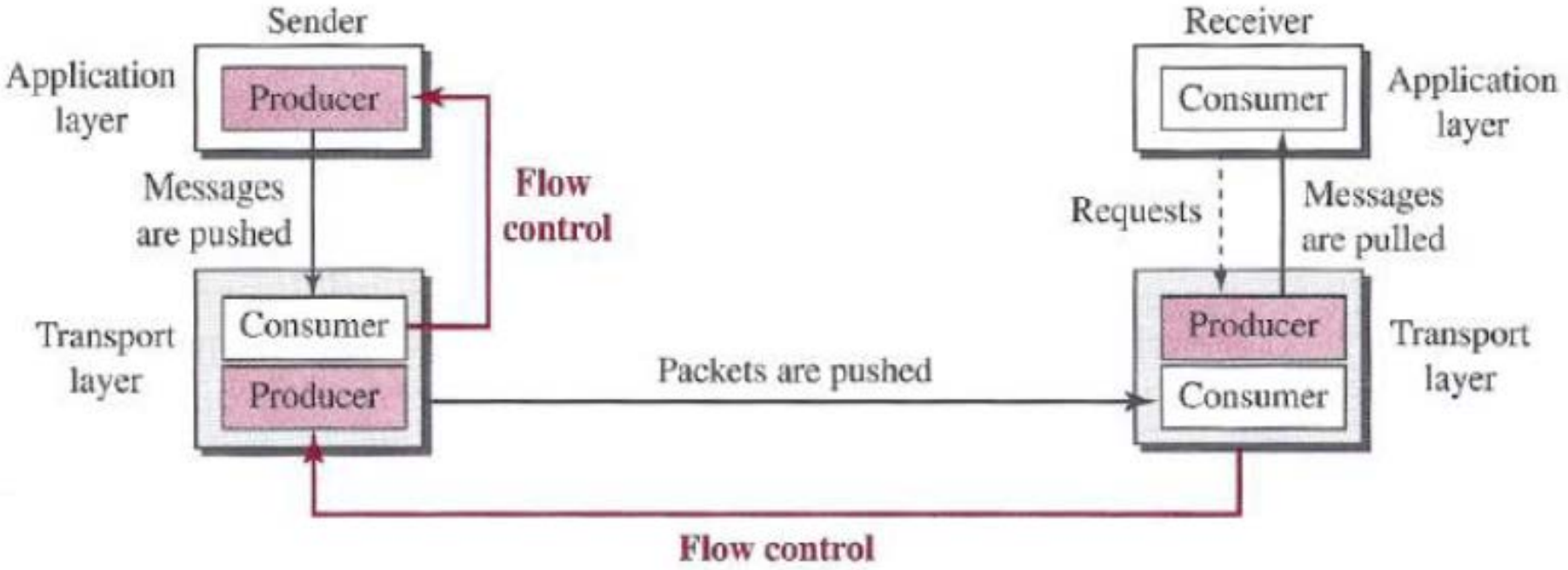


a. Pushing



b. Pulling

Flow Control in Transport Layer

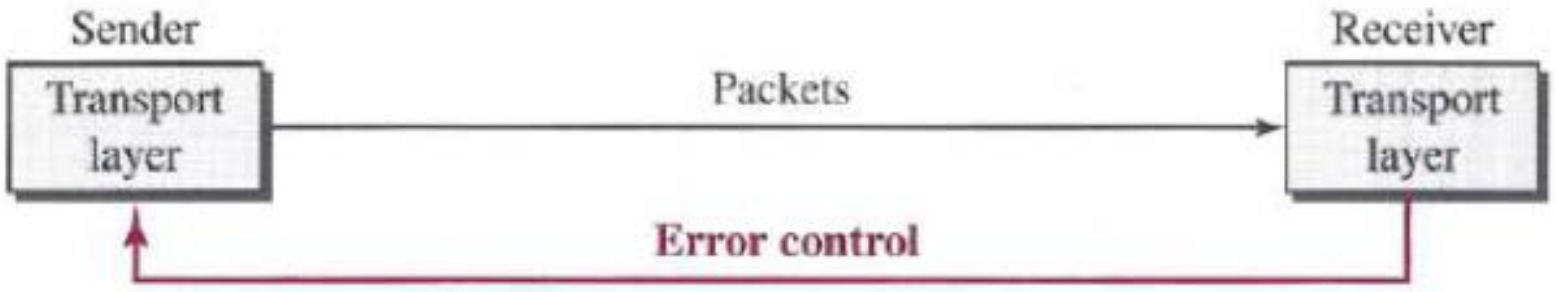


- Commonly used solution:
  - Using two buffers

Error Control

Error control at the transport layer is responsible for:

- Detecting and discarding corrupted packets
- Keeping track of lost and discarded packets and resending them.
- Recognizing duplicate packets and discarding them
- Buffering out-of-order packets until the missing packets arrive

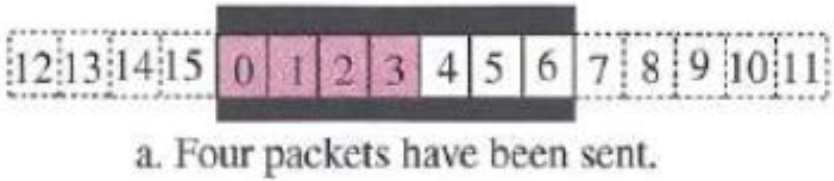


Requirements for Error Control

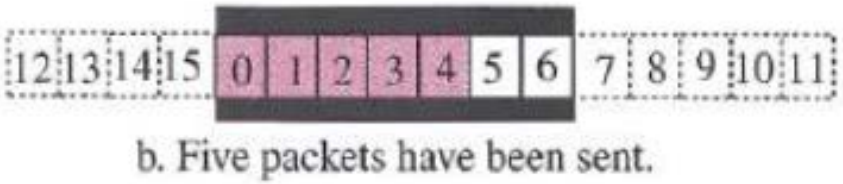
- Error detection mechanism
- Sequence Number
  - For identifying the position of segments
  - For deciding duplicate segments
- Acknowledgement

Sliding Window

- Protocol needs to maintain few information
  - How many segments have been sent
  - ACK has not been received for which segments
  - How many segments can be sent before receiving ACK



a. Four packets have been sent.



b. Five packets have been sent.

c. Seven packets have been sent; window is full.

d. Packet 0 has been acknowledged; window slides.

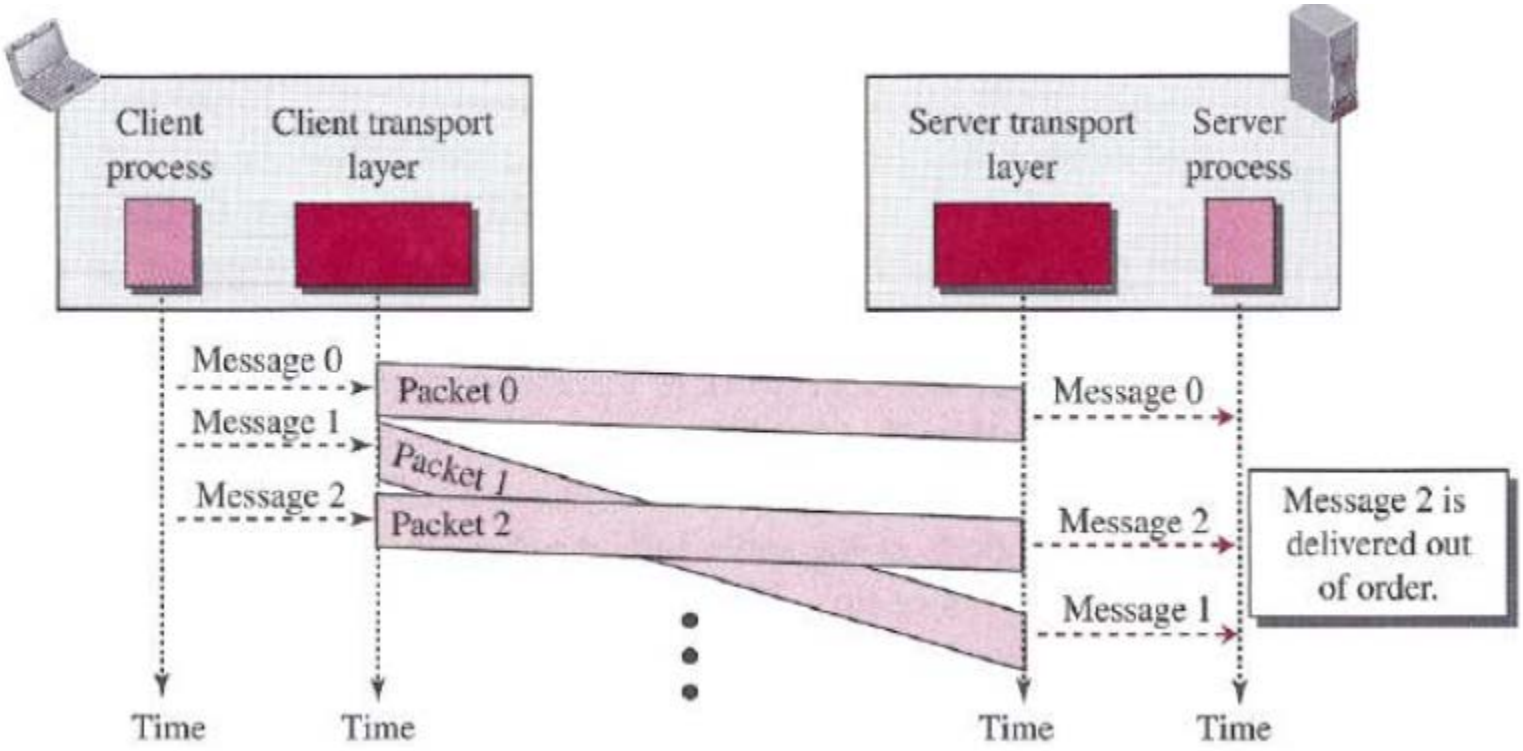
Congestion Control

- Congestion occurs if the load of a network greater than its present capacity

- TCP allows to have congestion control mechanism

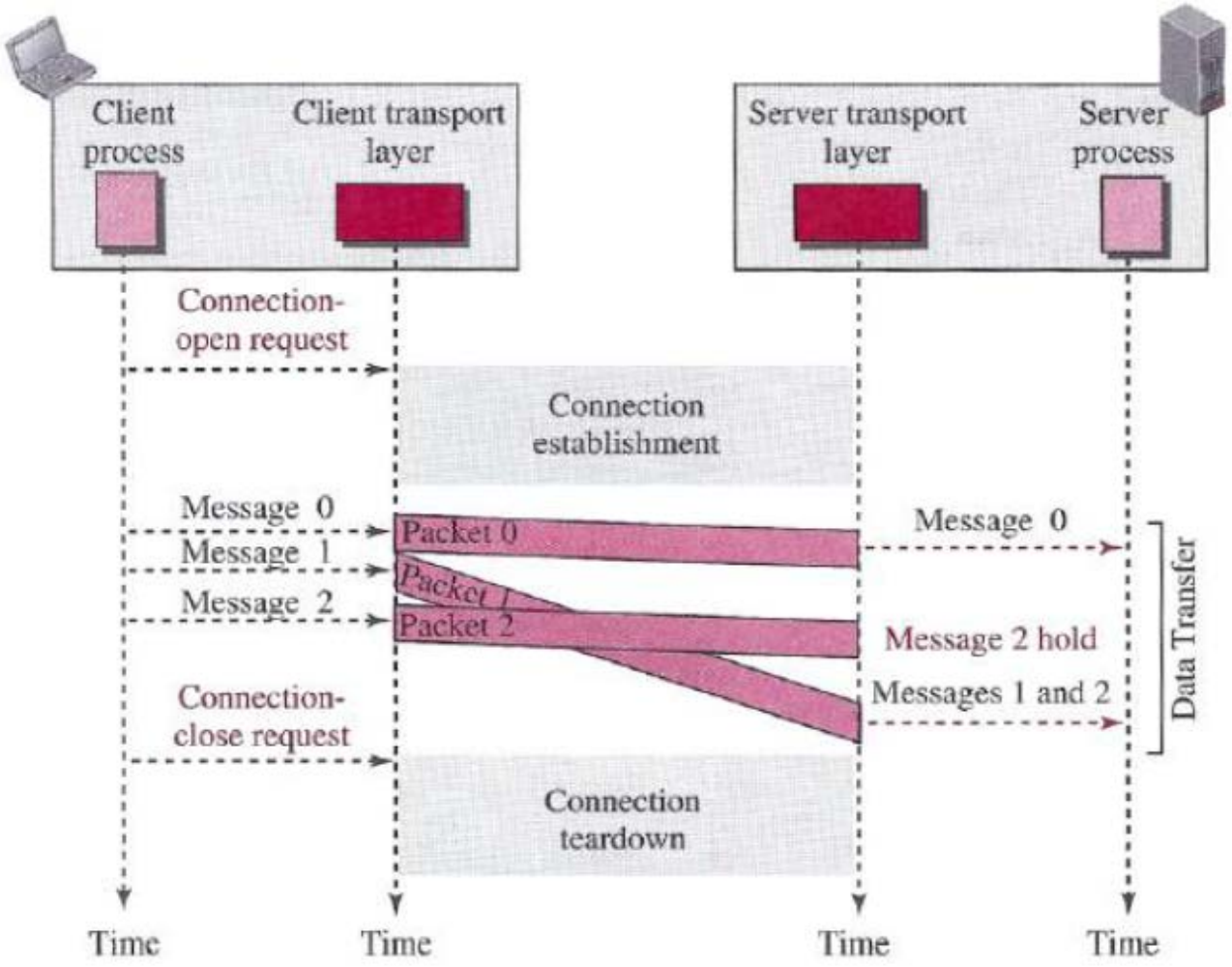
Connectionless and Connection-oriented

- Transport layer provides two types of services
  - Connectionless: independency between segments; no connection between sender & receiver; no segment numbering; no error control; no flow control; no congestion control



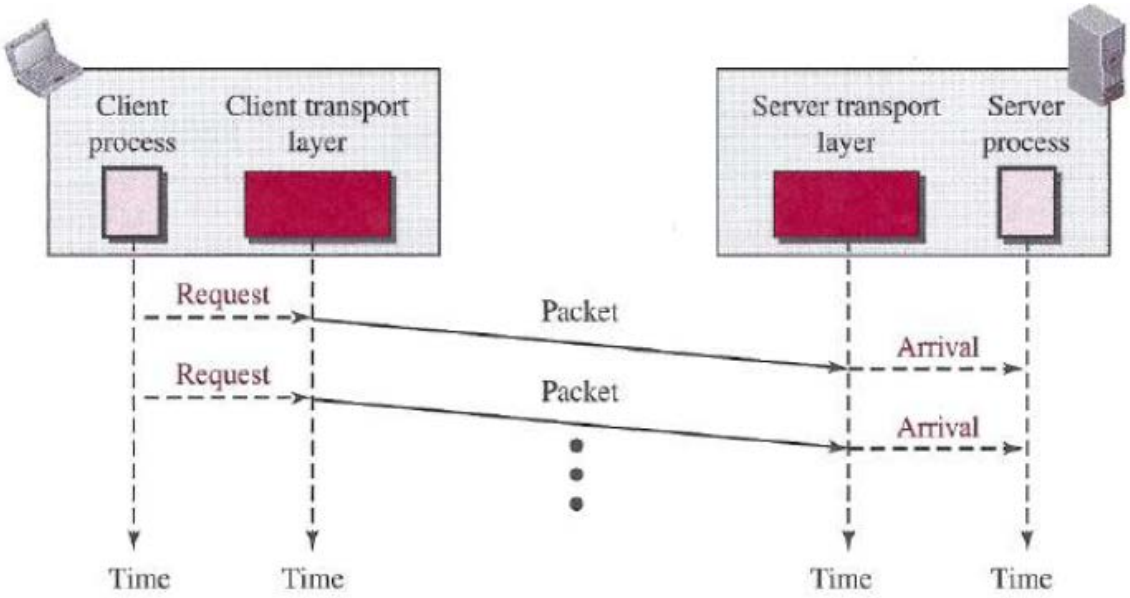
Cont…

– Connection-oriented: segments have relation; sender & receiver creates a connection to share segments of a message



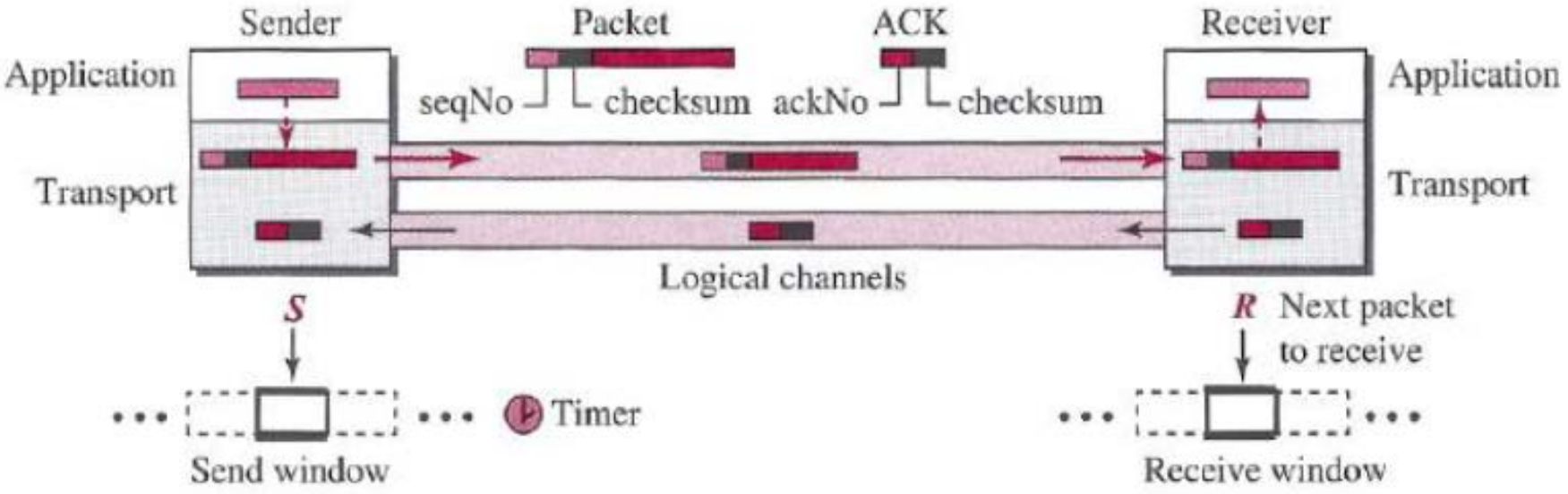
ARQ Protocols

- Simple Protocol (it is not ARQ protocol)
  - Connectionless
  - No error control ; No flow control

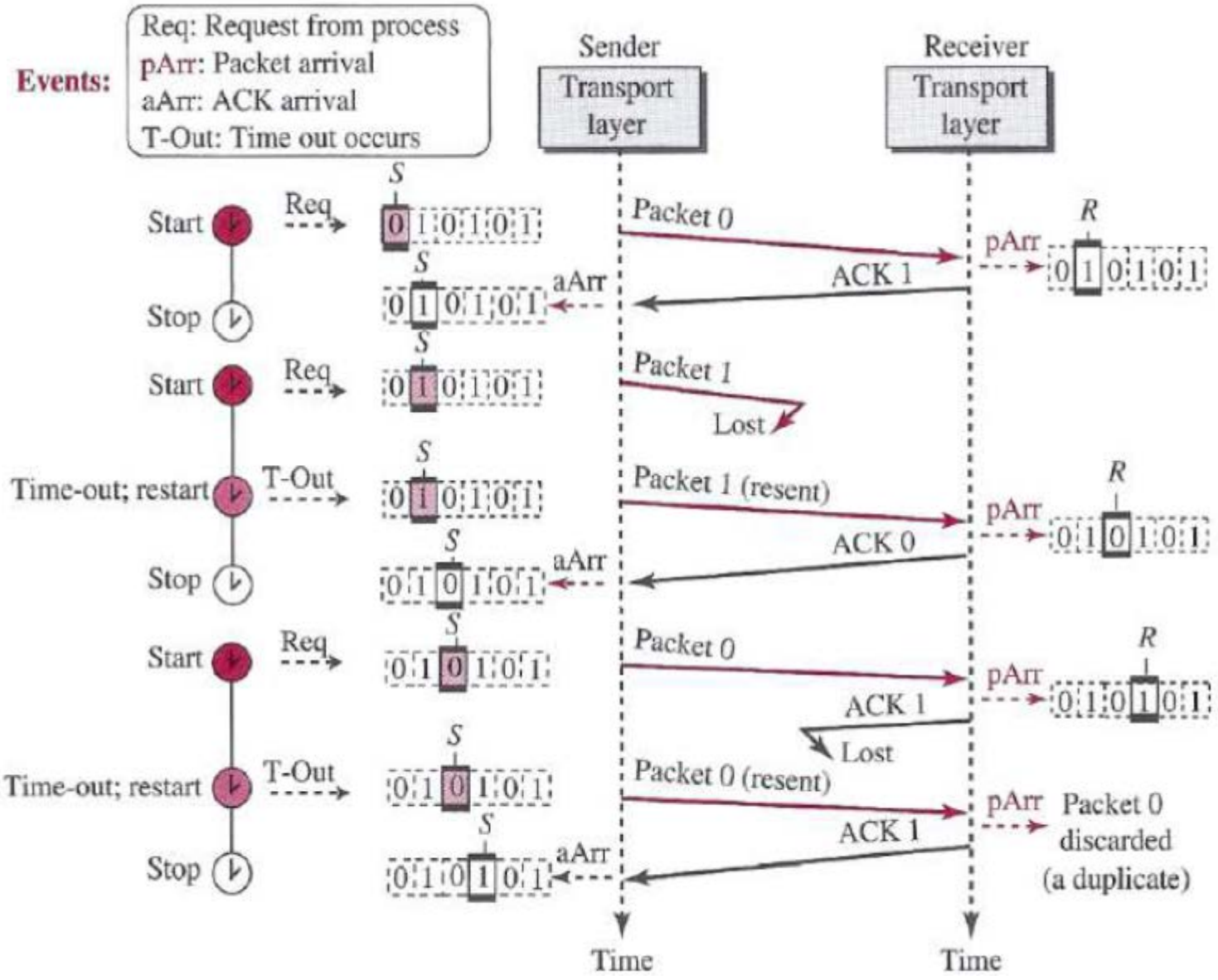


Stop-and-Wait Protocol

- Connection oriented
- Uses error control
- Uses flow control
- Sender & receiver use sliding window of size 1



Sequence Numbering



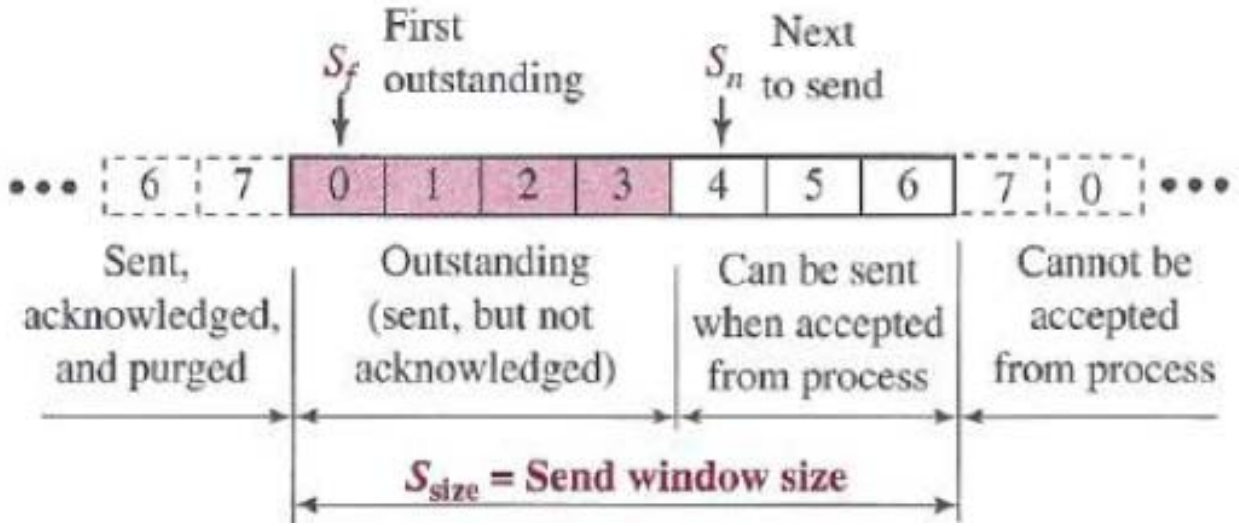
Efficiency of Stop-and-Wait

- Efficiency is very less if the channel has large bandwidth and round trip delay is long

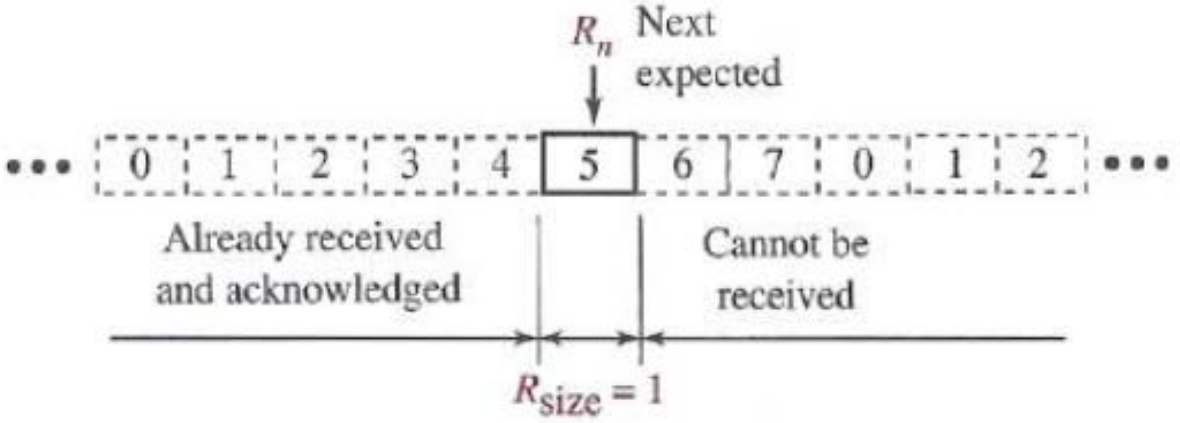
- Example:
  - Assume that, in a Stop-and- Wait system, the bandwidth of the line is 1 Mbps, and 1 bit takes 20 milliseconds to make a round trip.
  - What is the bandwidth-delay product?
  - If the system data packets are 1,000 bits in length, what is the utilization percentage of the link?
- Answer:
  - The bandwidth-delay product is $(1 \times 10^{6}) \times (20 \times 10^{-3}) = 20{,}000$ bits.
  - The system can send 20,000 bits during the time it takes for the data to go from the sender to the receiver and the acknowledgment to come back. However, the system sends only 1,000 bits. We can say that the link utilization is only 1,000/20,000, or 5 %.

Go-Back-N

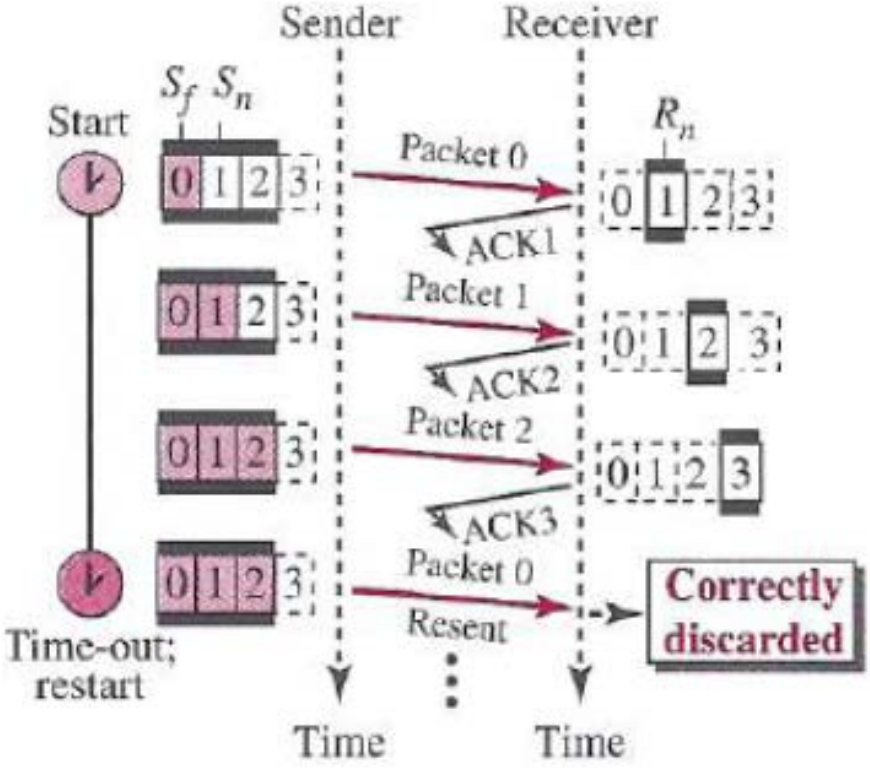
Send Window



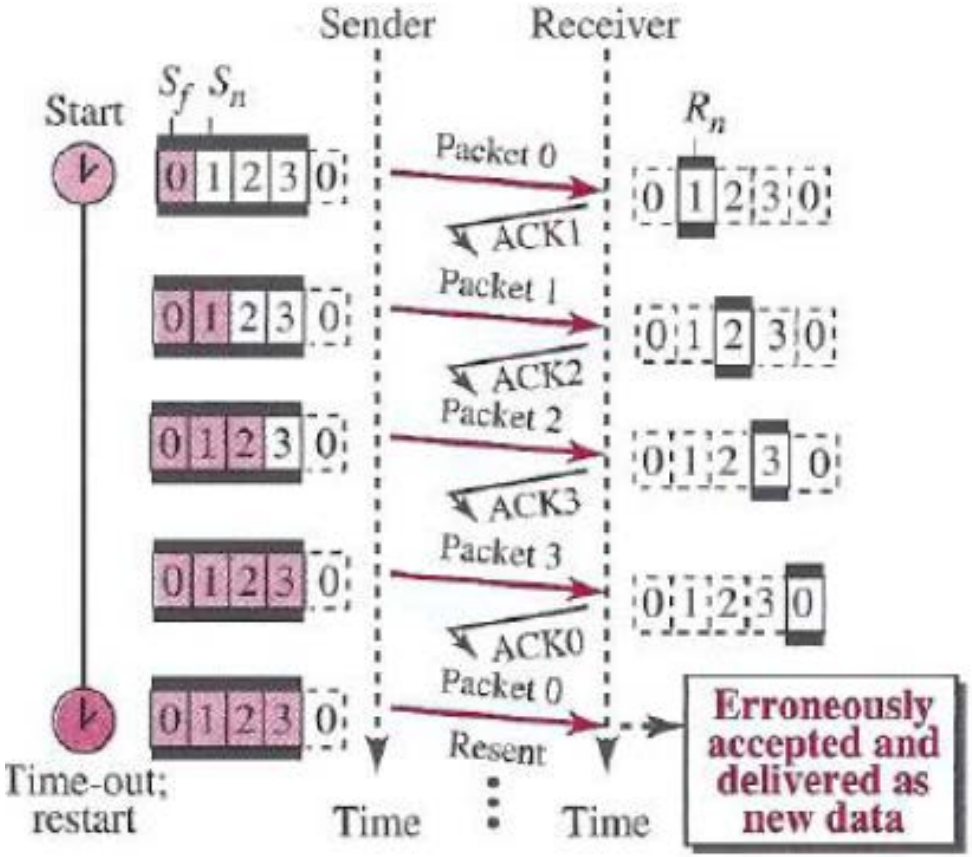
Receive Window



Send Window Size in Go-Back-N

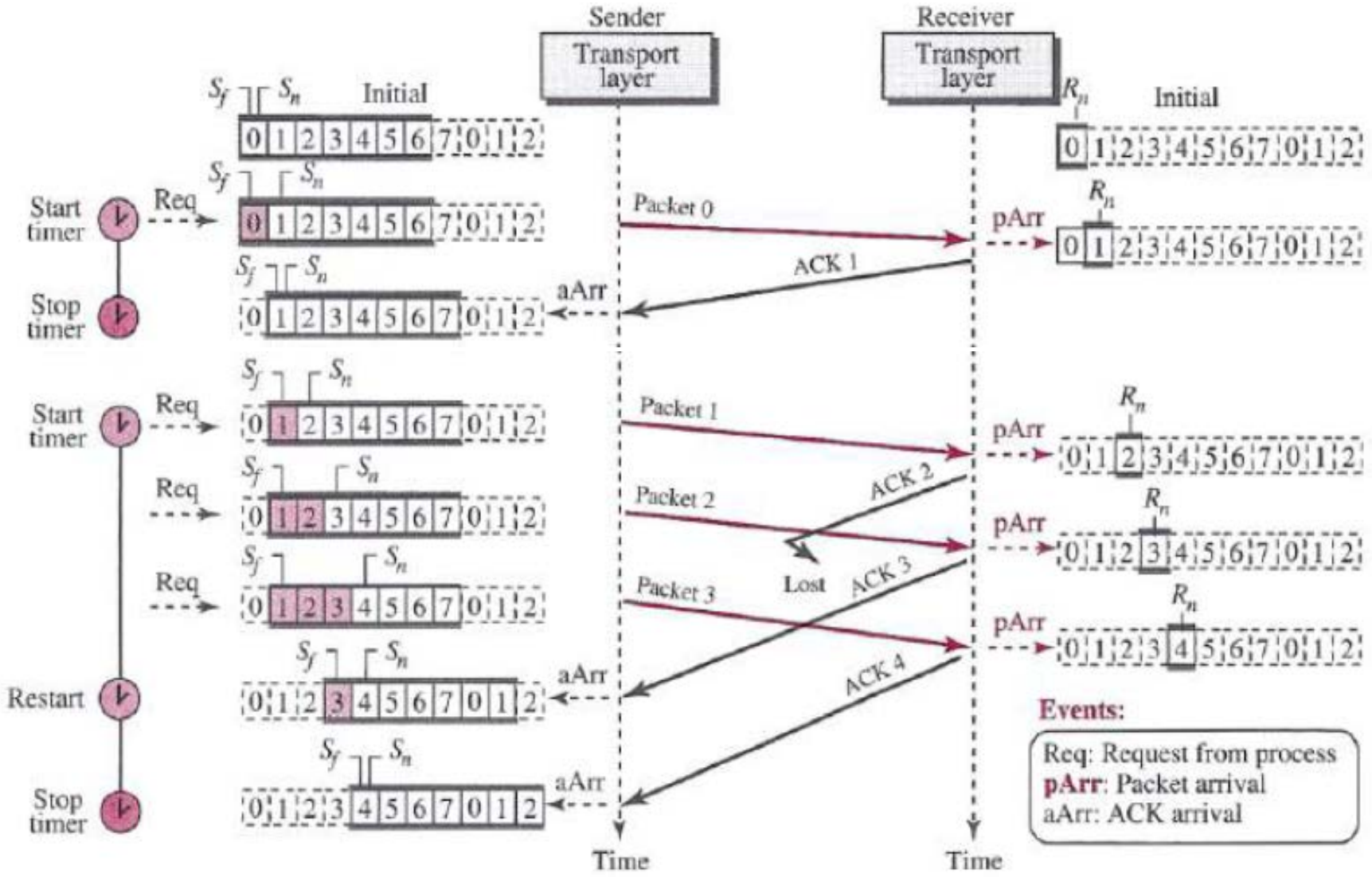


a. Send window of size $< 2^m$



b. Send window of size $= 2^m$

Flow Diagram in GBN



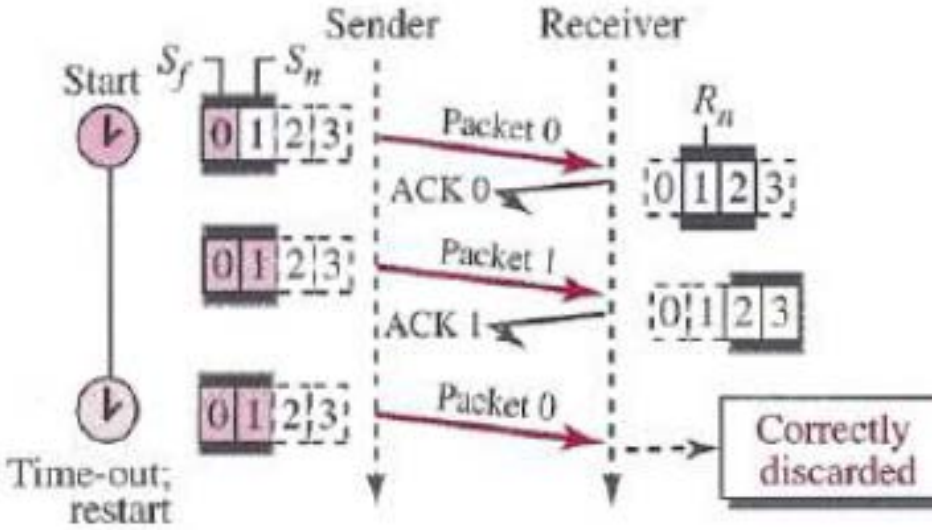
Selective-Repeat

- Disadvantages of Stop-and-Wait & Go-Back-N
  - The receiver keeps track of only one variable
  - So, many retransmission for few packet loss
  - Which increases congestion
  - Which in turn creates more loss of packet
  - And so on cyclically results in "total collapse"

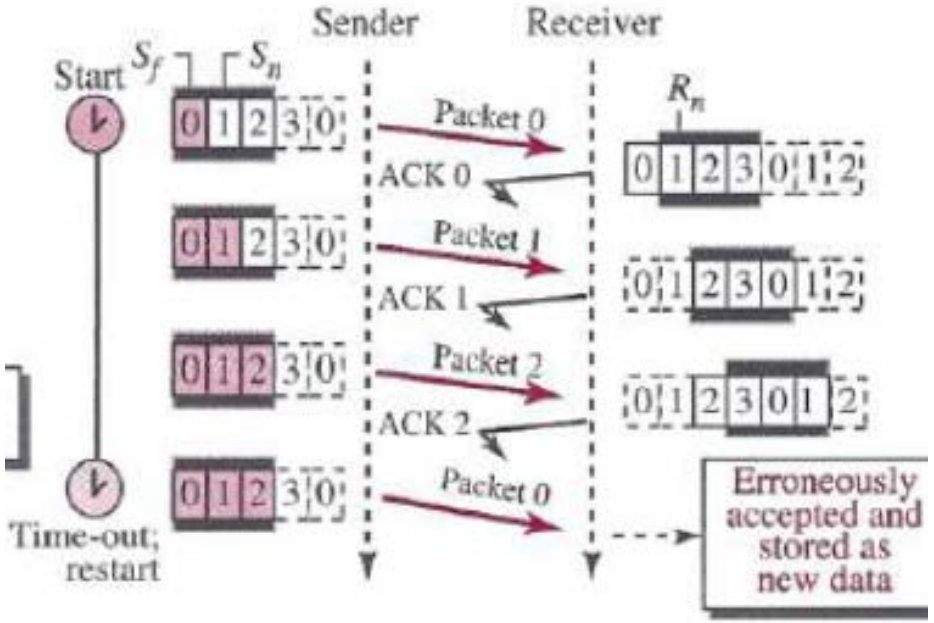
ACK in SR

- Assume a sender sends 6 packets: packets 0, 1,2,3,4, and 5.
- The sender receives an ACK with ackNo= 3.
- What is the interpretation if the system is using GBN or SR?
- Solution:
  - If the system is using GBN, it means that packets 0, 1, and 2 have been received uncorrupted and the receiver is expecting packet 3.
  - If the system is using SR, it means that packet 3 has been received uncorrupted; the ACK does not say anything about other packets.

Window Size in SR



a. Send and receive windows of size = $2^m - 1$



b. Send and receive windows of size > $2^m - 1$

Piggybacking

- Improve efficiency of bidirectional protocol

- When a packet is carrying data from A to B, it can also carry acknowledgment feedback about arrived packets from B

Thanks!