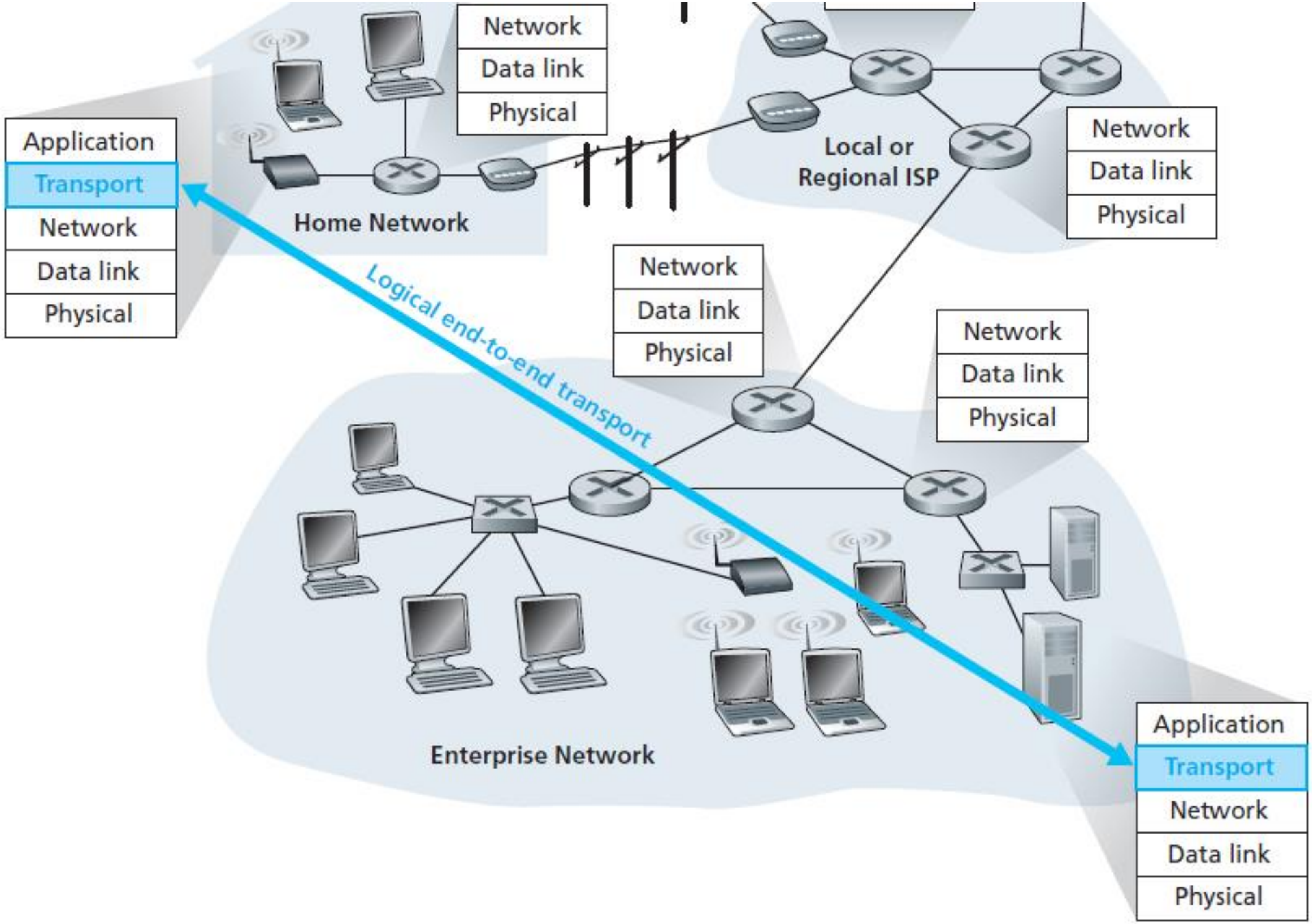
CS321: Computer Networks

Introduction to Transport Layer

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

Introduction



Cont…

- 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

Network v/s Transport Layer

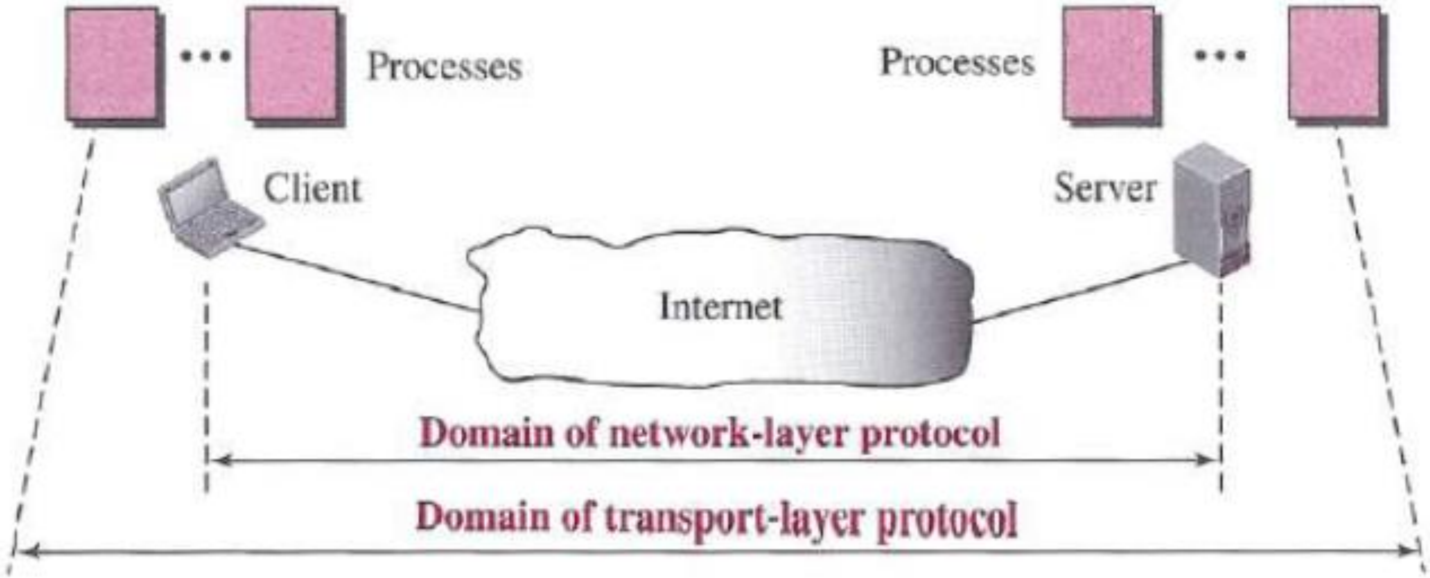
- Transport layer protocols are implemented in the end systems
- Network layer protocols are implemented in network routers

- IP provides communication between hosts
- TCP or UDP provide communication between processes

- Transport layer packets: segments
- Network layer packet: datagram

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

Process-to-Process Communication



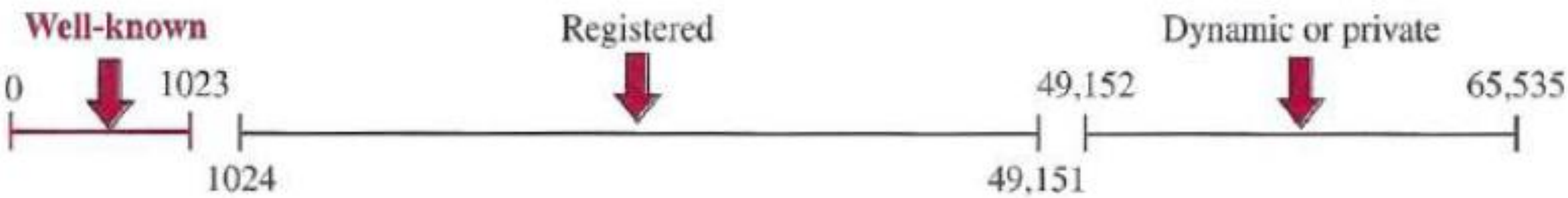
- One method for this: client-server approach

- 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

Addressing

- Internet Corporation for Assigned Names and Numbers (ICANN)



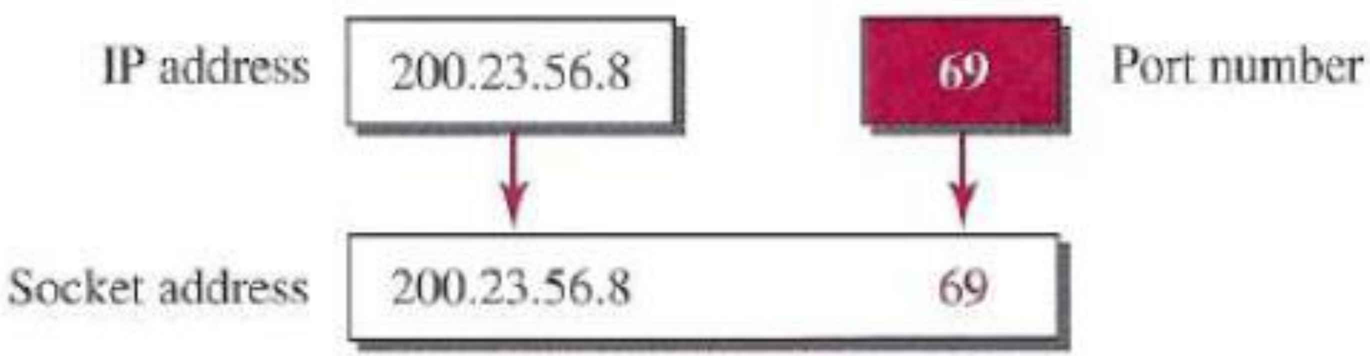
- Example:
  - ports are stored in /etc/services
  - FTP : 20, 21
  - SSH, SCP : 22
  - Echo : 7
  - DNS : 53
  - HTTP : 80
  - SNMP : 161,162
  - BGP : 179

Private port numbers are available for use by any application to use in communicating with any other application, using the Internet's TCP or UDP.

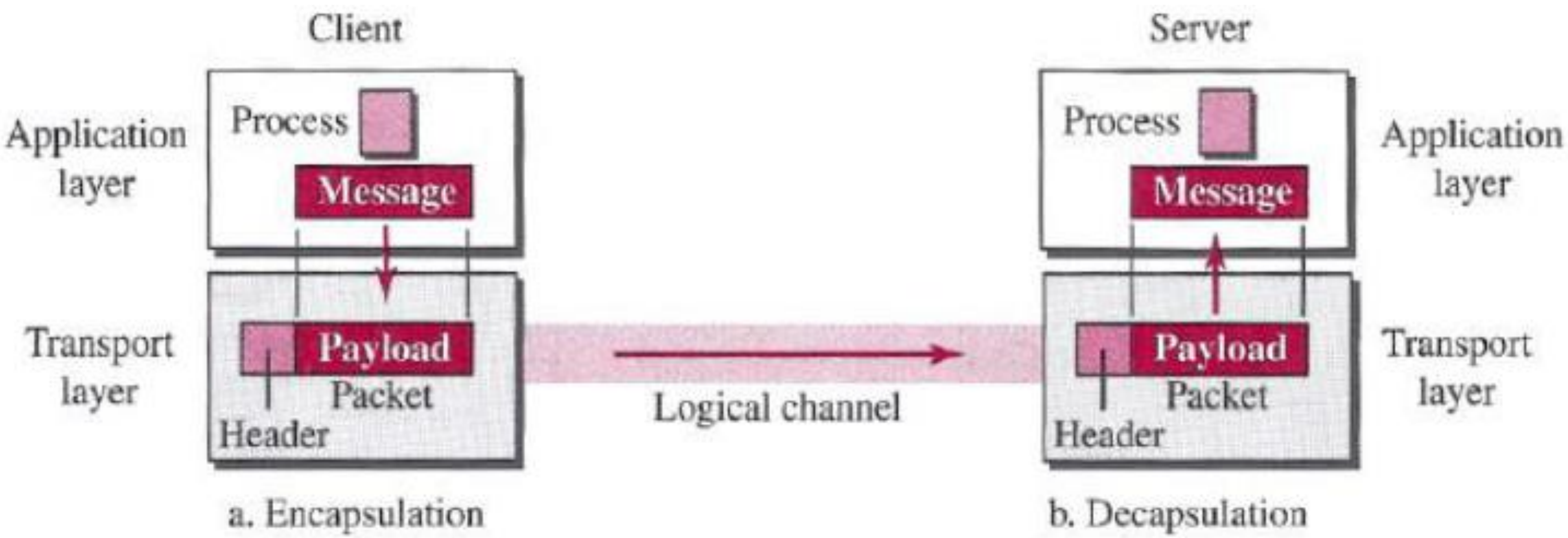
Companies and other users should register Registered Port Number with the ICANN for use by their applications.

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

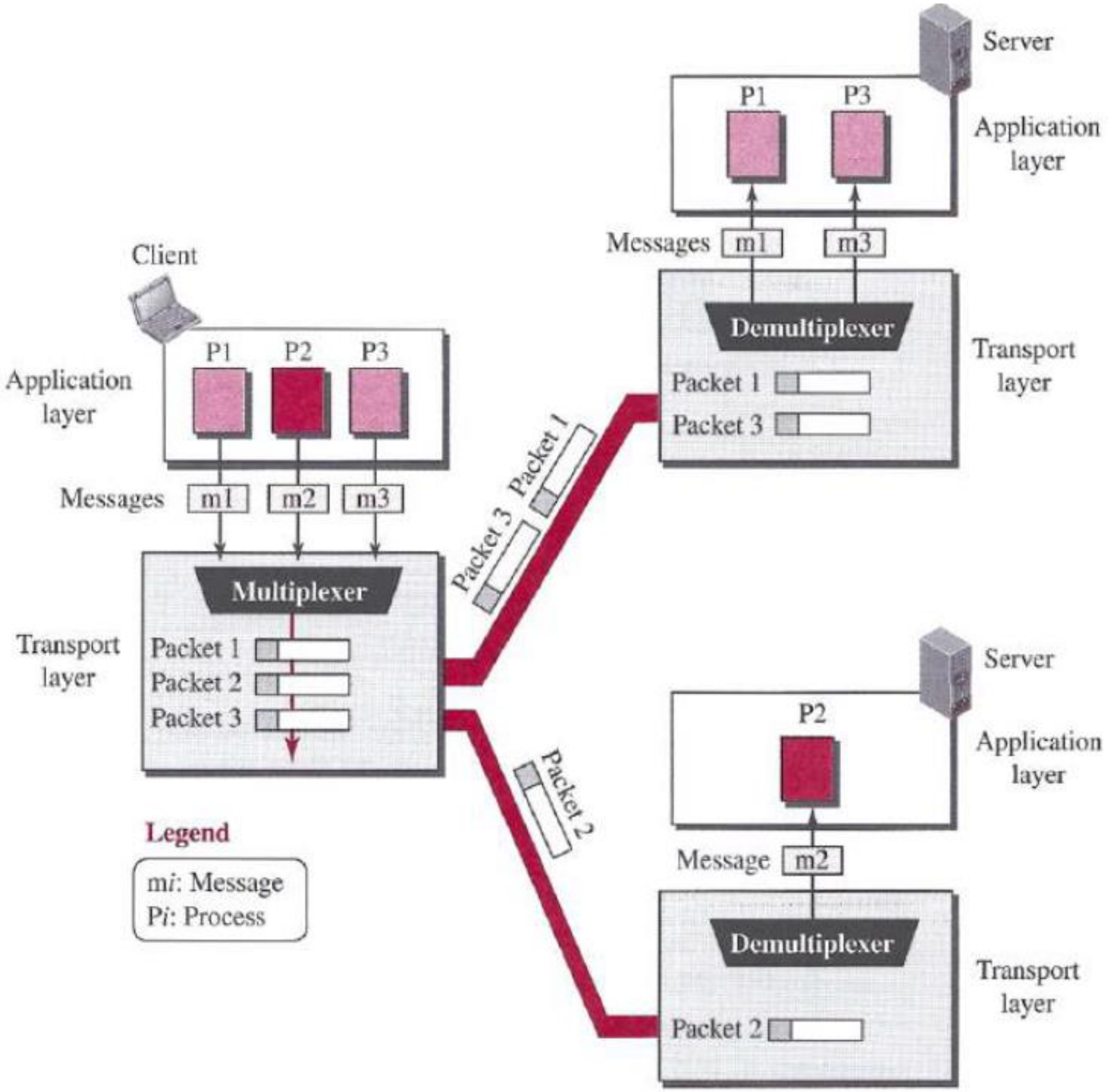


Encapsulation and Decapsulation



Multiplexing and Demultiplexing

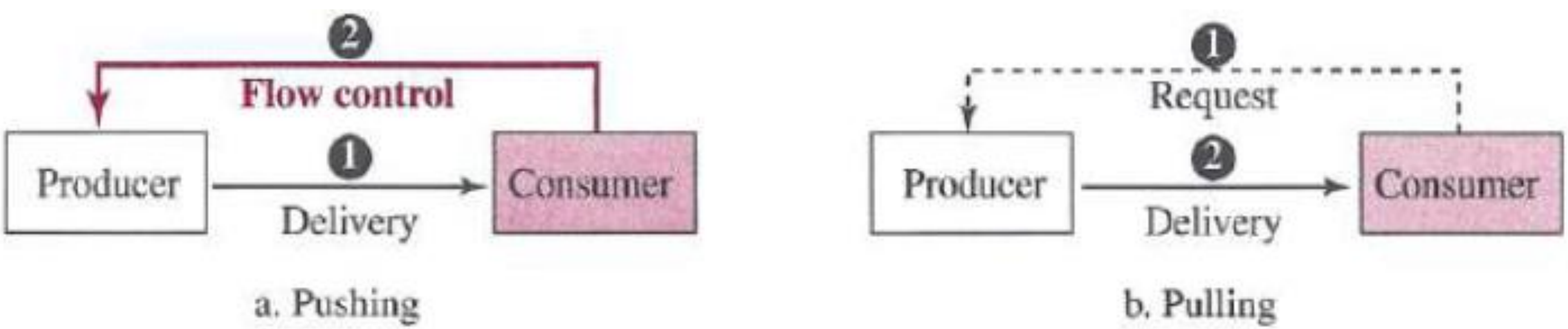
- Suppose you are sitting in front of a computer, and you are browsing Web pages while running one FTP session and two Telnet sessions.
- So, 4 processes
  - HTTP
  - FTP
  - 2 Telnet
- How does a segment forwarded to intended process?
  - using Socket Address
  - Method is called multiplexing and demultiplexing



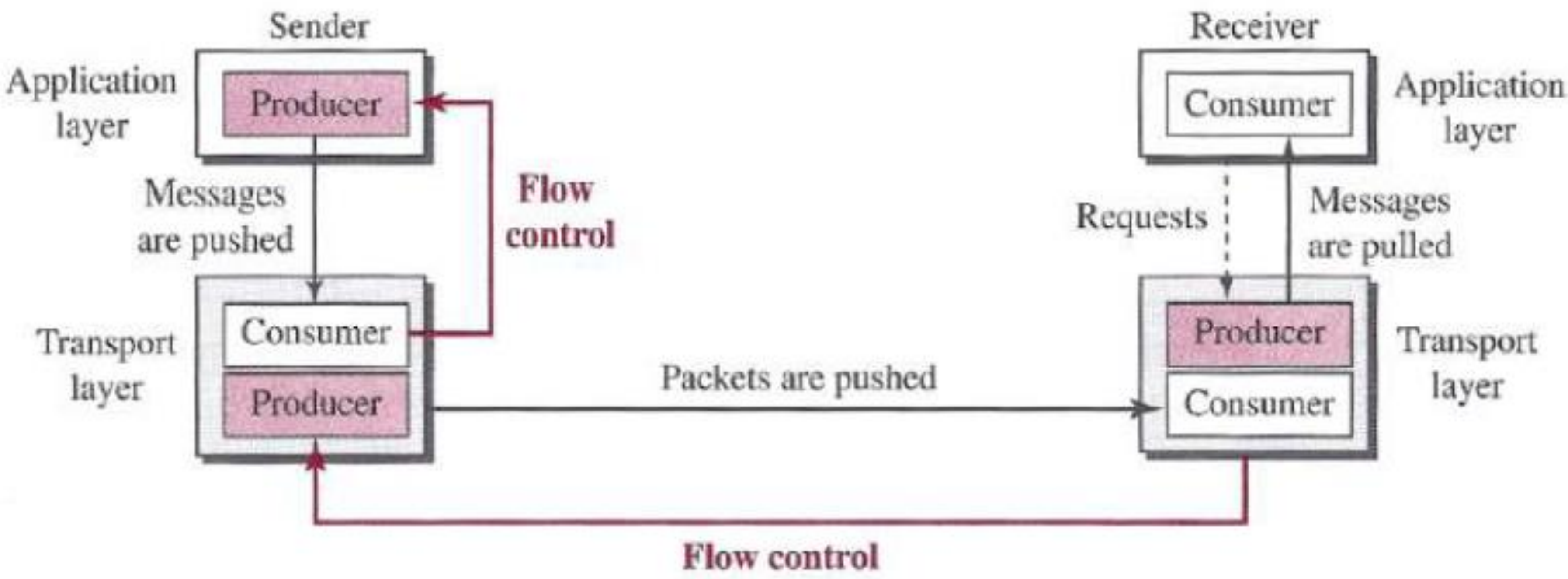
Flow Control

- Delivery of an item follows two approaches:
  - Pushing
  - Pulling

- Flow control need when a consumer is overwhelmed by the receiving items



Flow Control in Transport Layer

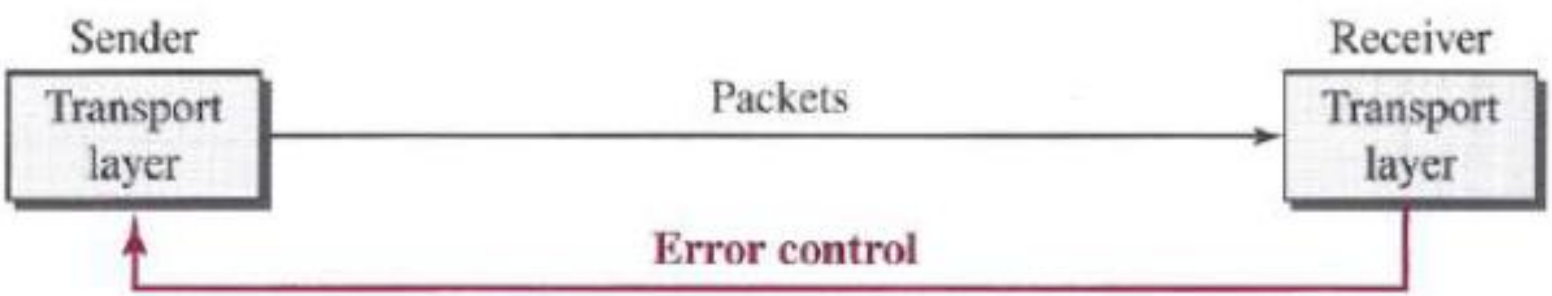


- Commonly used solution:
  - Using two buffers

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

- Three basic requirements:
  - Error detection mechanism
  - Sequence Number
    - For identifying the position of segments
    - For deciding duplicate segments
  - Acknowledgement (ACK)

- 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

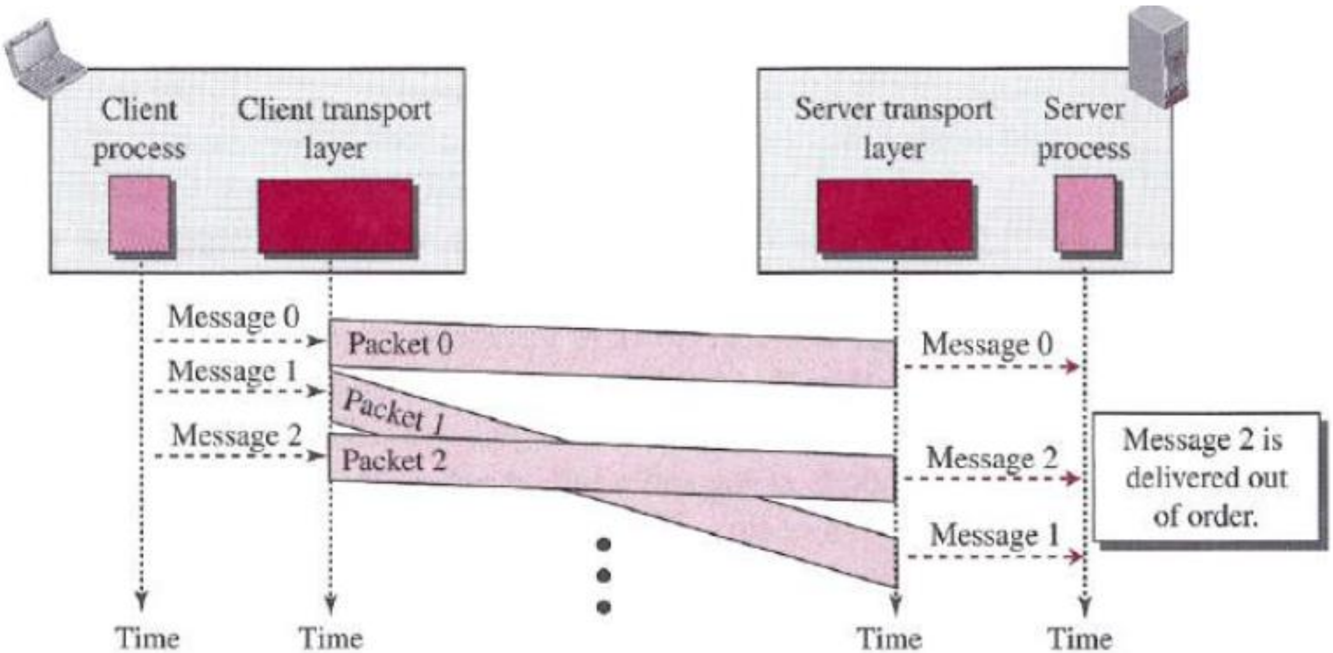
- TCP detects congestion through packet loss and changes in round trip time or throughput
  - Slow Start algorithm
  - Tri-S
  - DUAL
  - TCP Vegas

- Congestion control method when gateway provides an indication of congestion
  - Random Early Detection (RED)
  - Explicit Congestion Notification (ECN)

Source: https://www.cse.wustl.edu/~jain/cis788-95/ftp/tcpip_cong/index.html

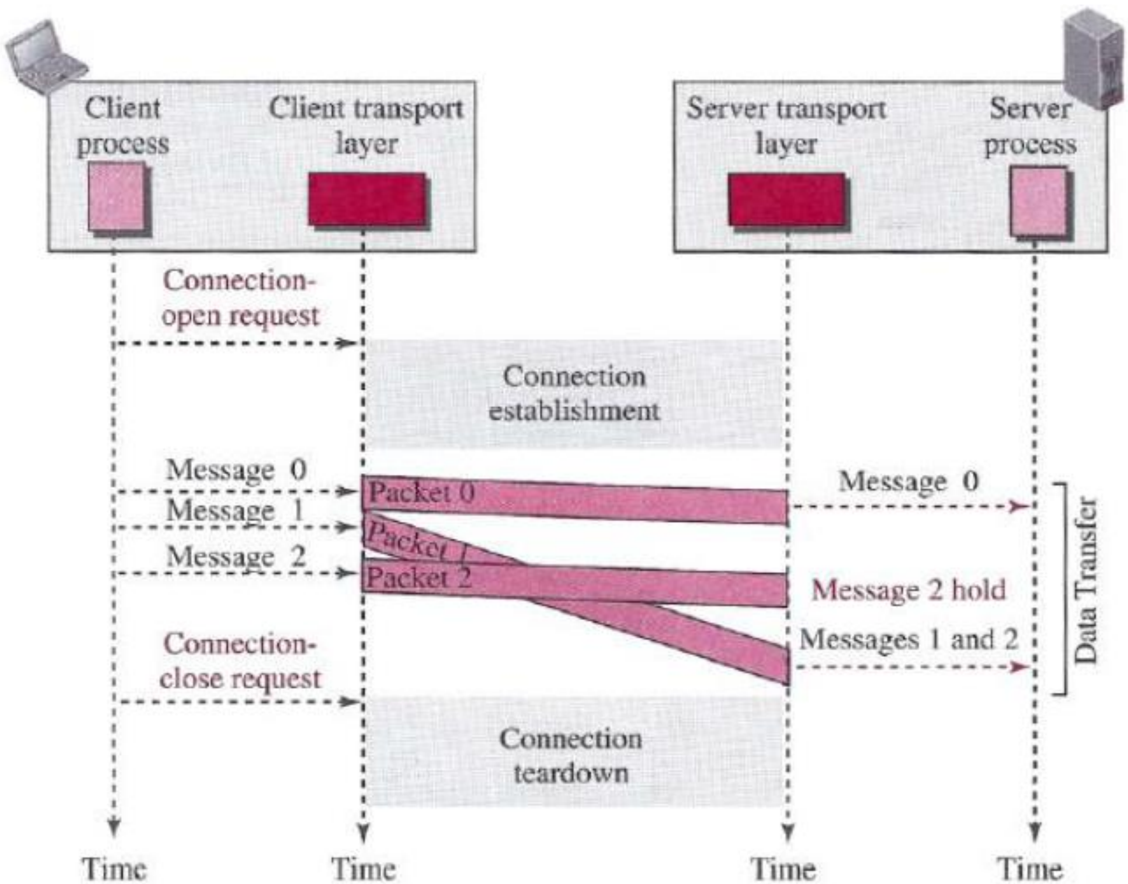
Connectionless and Connection-oriented

- 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



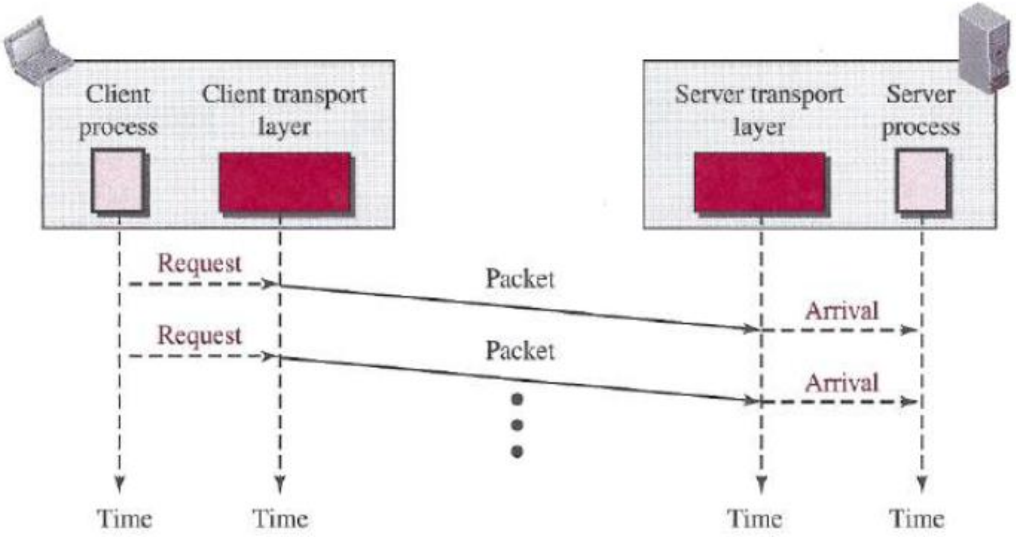
Cont…

– Connection-oriented:
  - segments have relation;
  - sender & receiver creates a connection to share segments of a message
  - Exist: segment numbering; error control; flow control; congestion control



Transport Layer Protocol Strategies

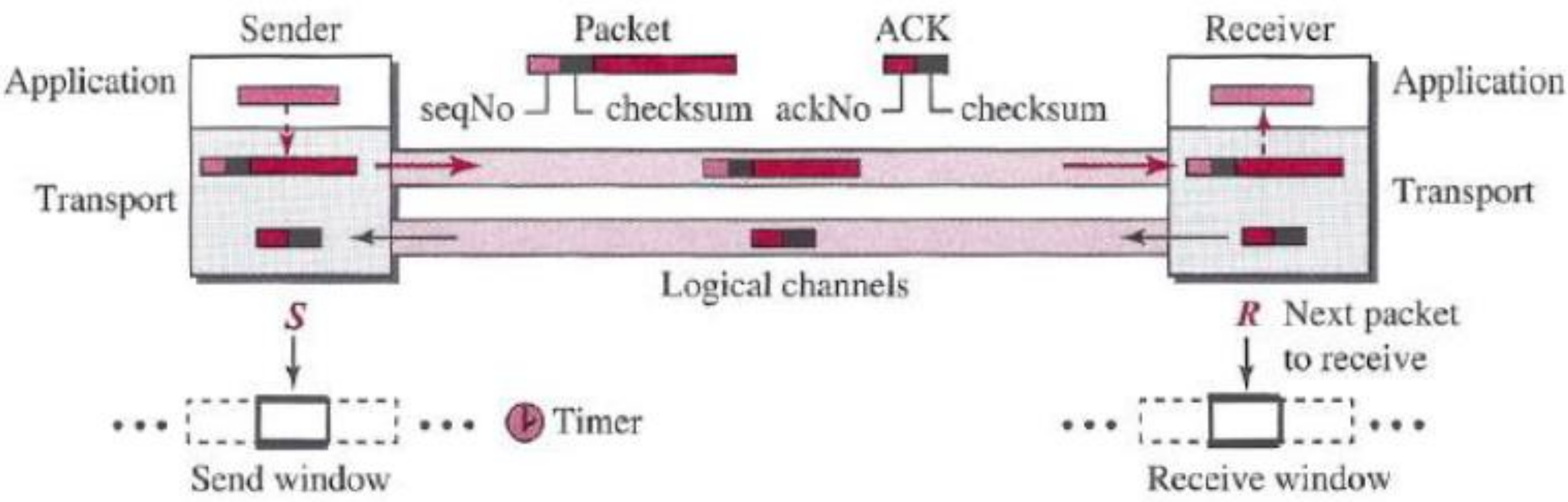
- Simple (it is not ARQ protocol)
  - Connectionless
  - No error control ; No flow control



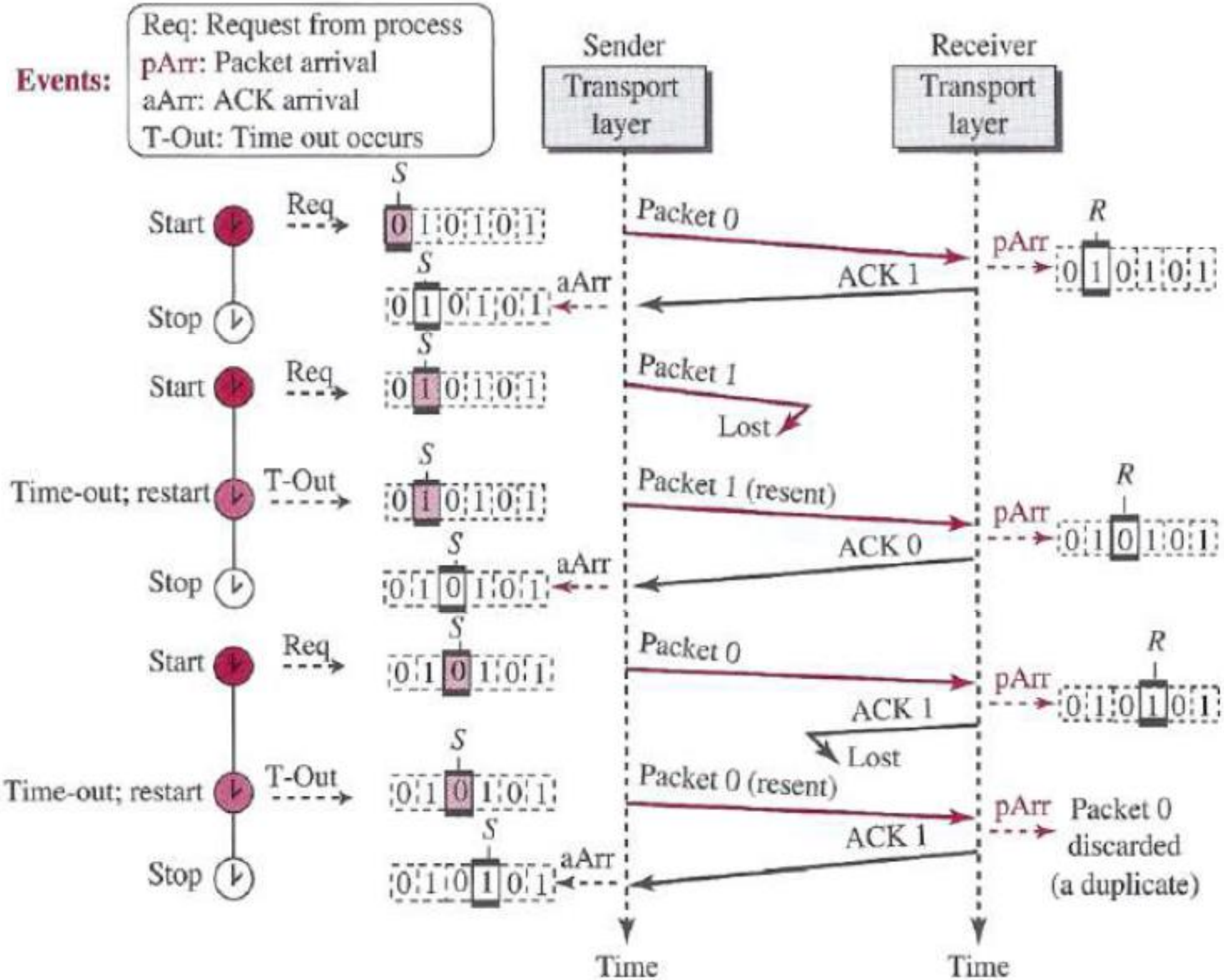
- Automatic Repeat Request (ARQ)
  - for error control in data transmission using acknowledgement and timeout
  - achieve reliable data transmission over an unreliable service
  - Three protocols: Stop-and-Wait, Go-back-N, Selective Repeat

Stop-and-Wait Protocol

- Connection oriented
- Uses error control
- Uses flow control
- Sender & receiver use sliding window of size 1
- Also known as *alternating-bit protocol*



Sequence Numbering



Efficiency of Stop-and-Wait

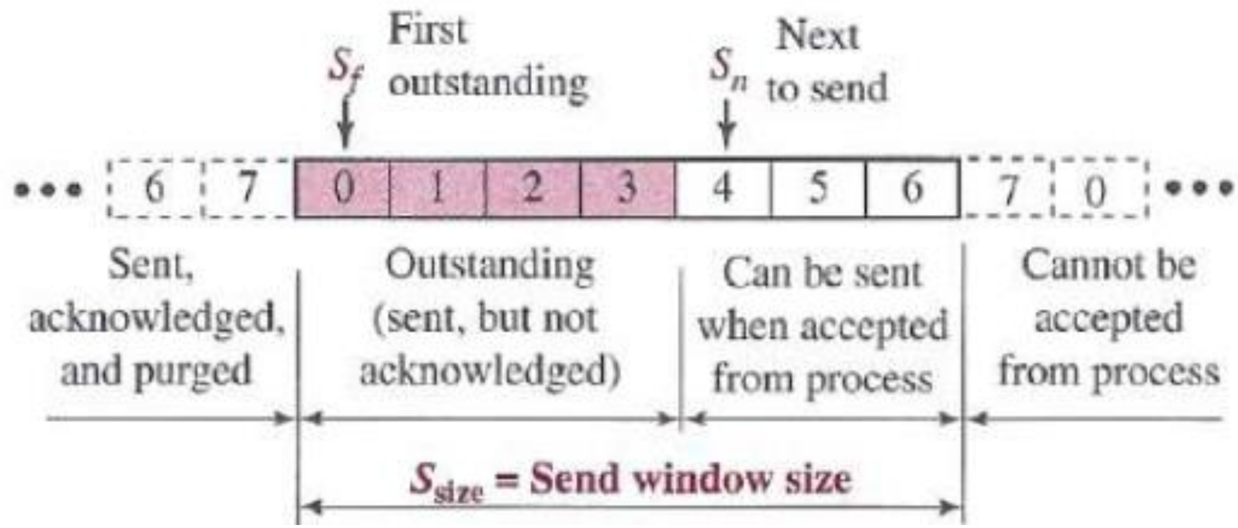
- Efficiency is very less if the channel has large bandwidth and round trip delay is long

- 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?

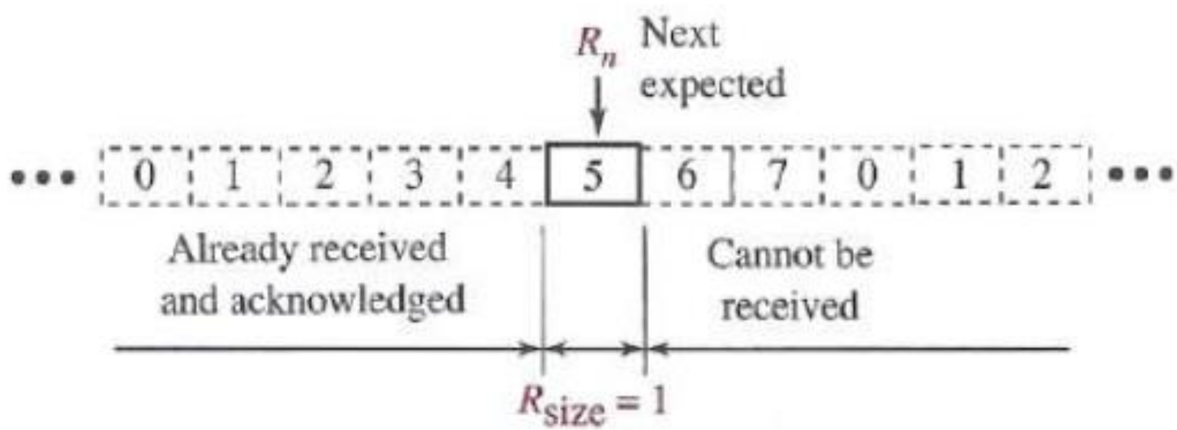
- Answer:
  - The bandwidth-delay product is $(1 \times 10^6) \times (20 \times 10^{-3}) = 20{,}000$ bits.
  - The system can send 20,000 bits during the time it takes for the data to go from the sender to the receiver and the acknowledgment to come back. However, the system sends only 1,000 bits. We can say that the link utilization is only 1,000/20,000, or 5 %.

Go-Back-N

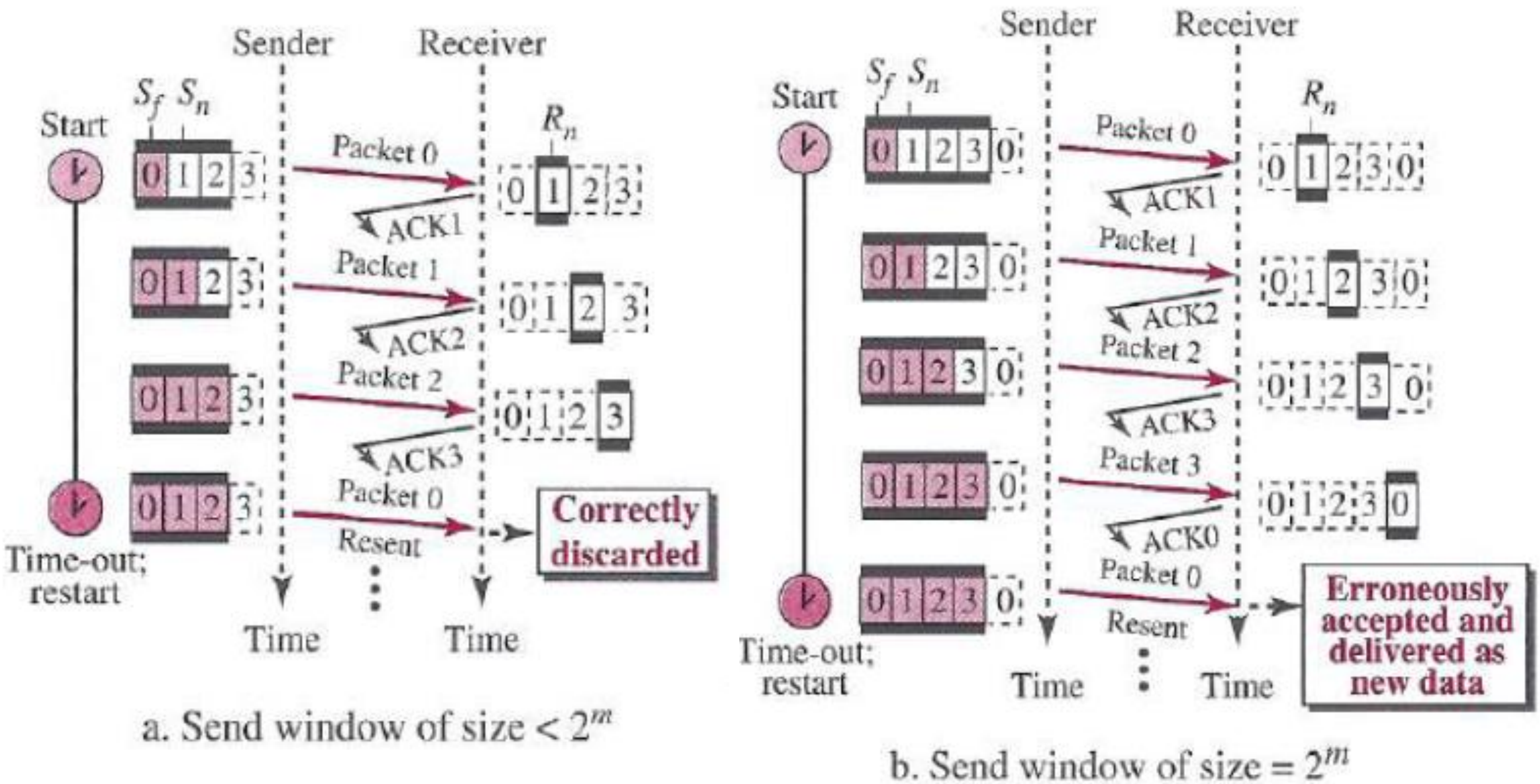
Send Window



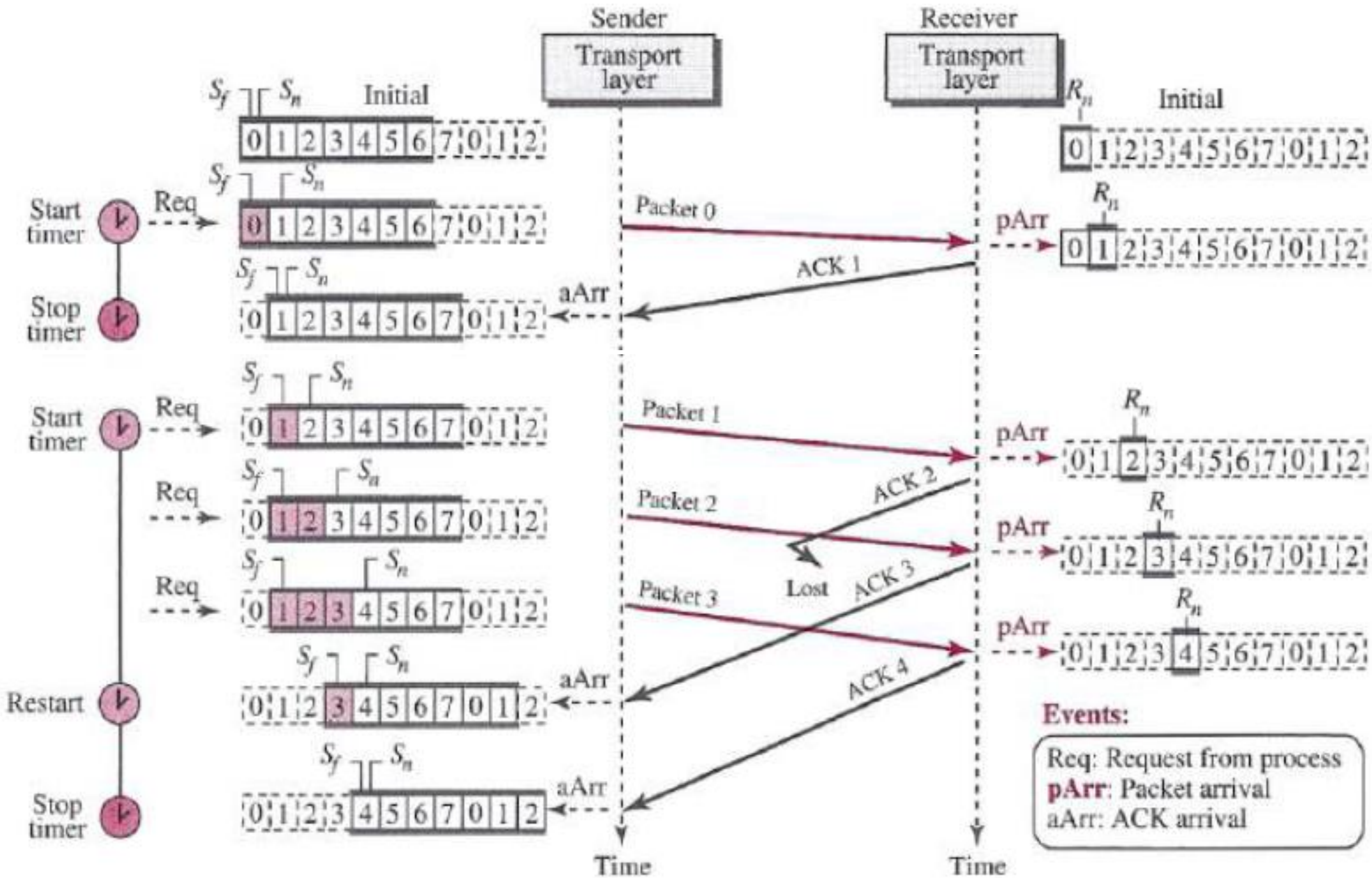
Receive Window



Send Window Size in GBN



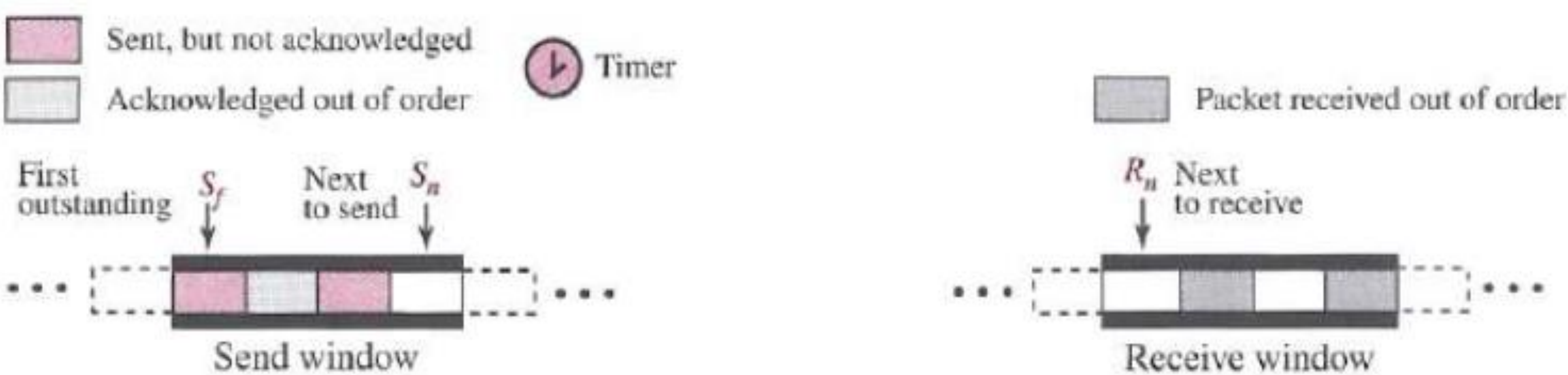
Flow Diagram in GBN



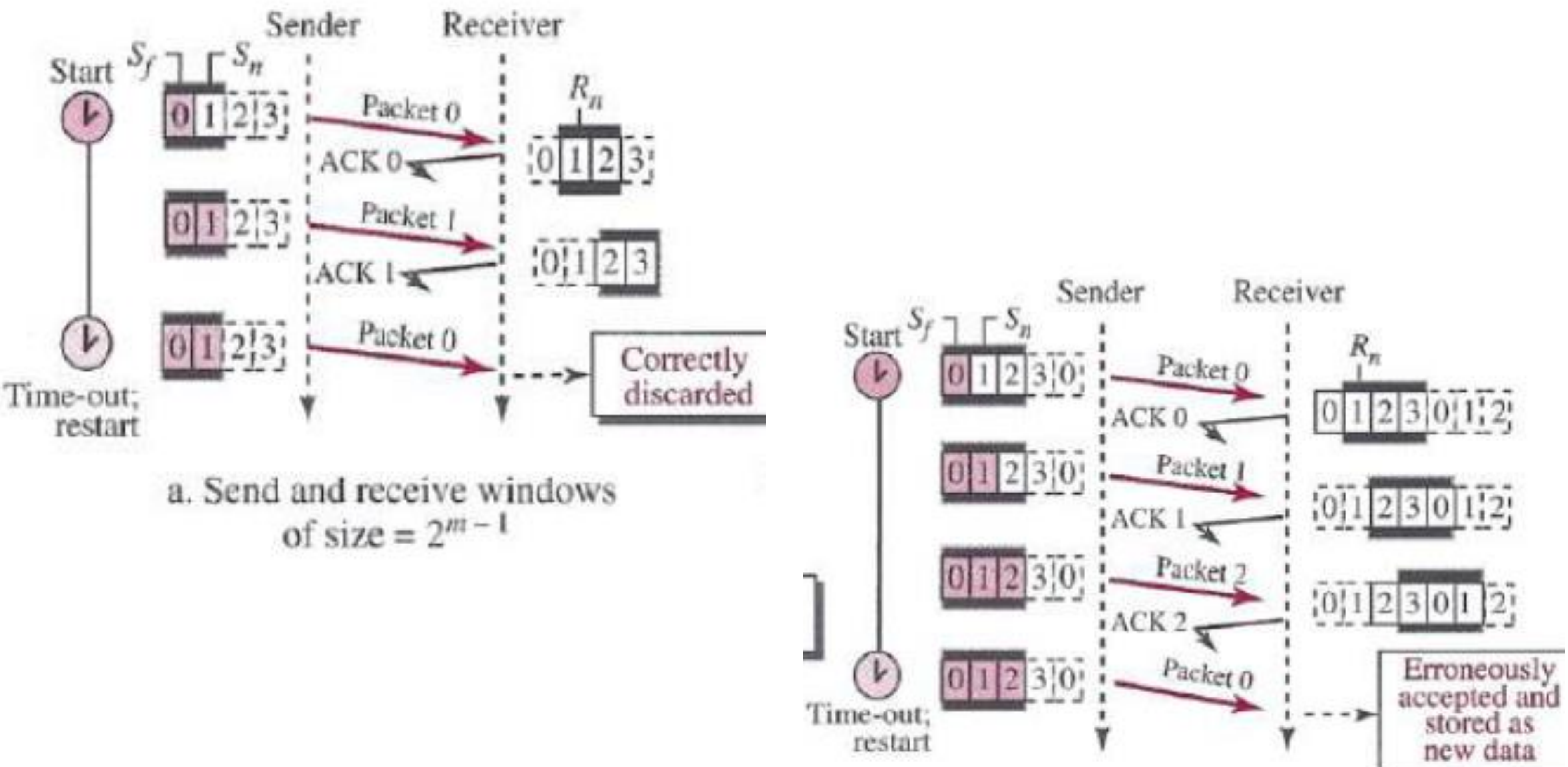
Selective-Repeat (SR)

- 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"

- Solution:
  - Selective Repeat using both side window



Window Size in SR



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

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

ACK in GBN v/s 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 of “ACK 3” if the system is using
  - GBN
  - 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.
  - It follows cumulative ACK

  - If the system is using SR, it means that packet 3 has been received uncorrupted; the ACK does not say anything about other packets.
  - It follows selective ACK

ACK in TCP

- ACK in TCP can be described as a hybrid of GBN and SR

- TCP is similar to GBN because both protocols have a limit on the number of unACK'd packets that the sender can send into the network.

- However, TCP is different from GBN because GBN requires the retransmission of every unACK'd packet when packets are lost, but TCP only retransmits the oldest unACK'd one.

- TCP is similar to SR because, when packets are lost due to congestion, the protocols do not require the sender to retransmit EVERY unACK'd packet sent by the sender. The sender just retransmits the oldest unACK'd packet.

- TCP is different from SR because SR requires individual acknowledgement of each packet that was sent by the receiver; but rather than selectively ACKing every packet, TCP sends an ACK for the next packet that it is expecting (like GBN) and buffers the ones that it has received so far, even if they're out of order (like SR).

Reliable Data Transfer

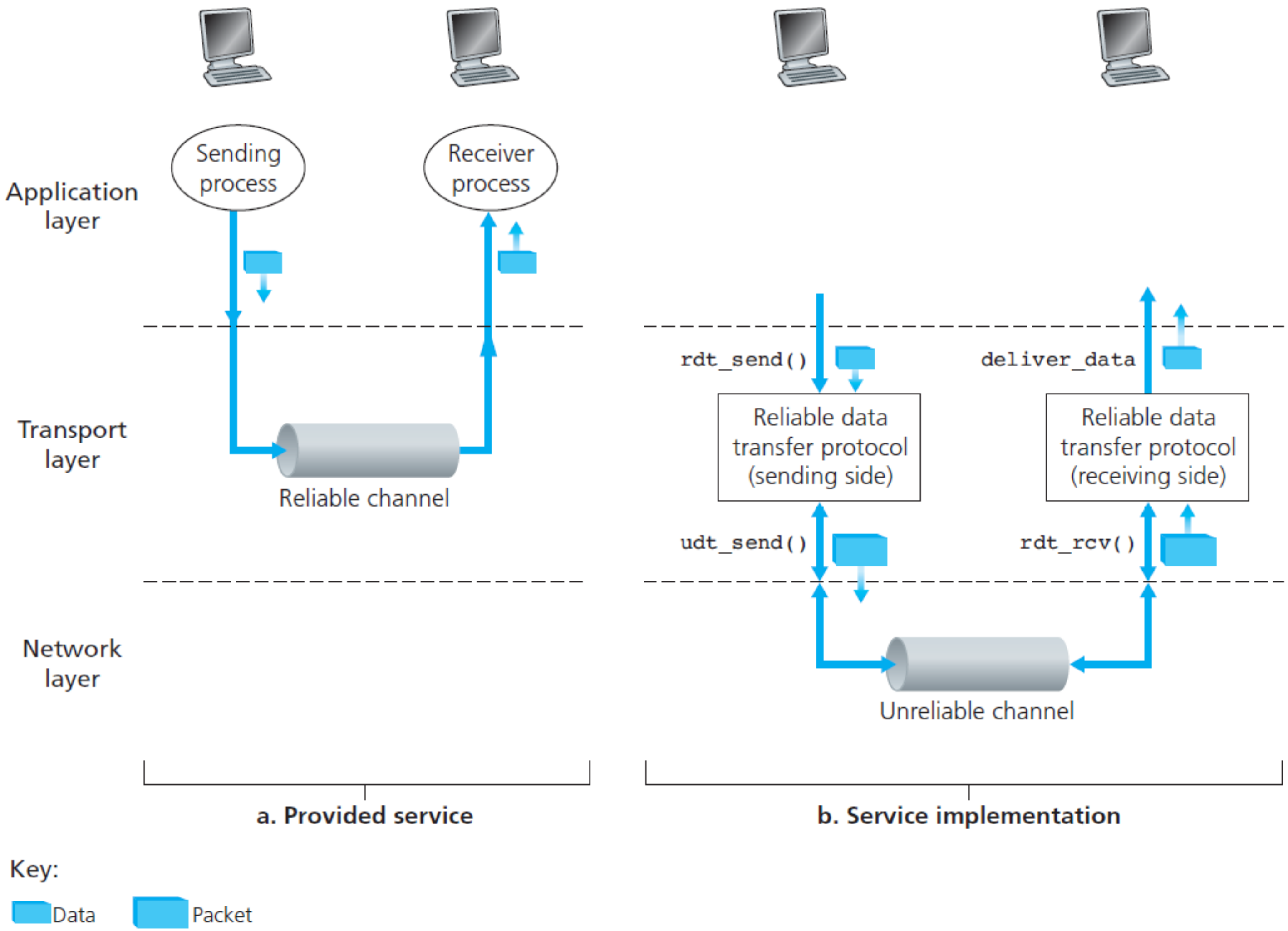


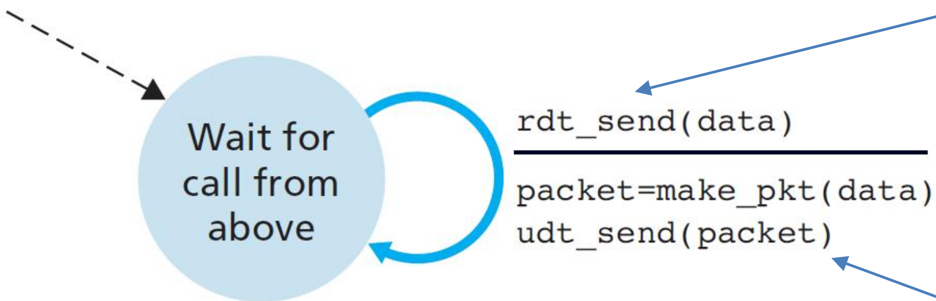
Figure 3.8 ♦ Reliable data transfer: Service model and service implementation

Finite State Machine for Reliable Data Transfer

- **Case1**: Over perfectly reliable channel

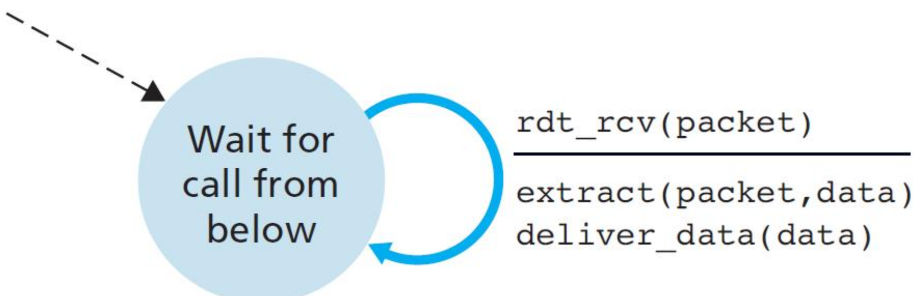
The initial state of the FSM

The event causing the transition



the actions taken when the event occurs

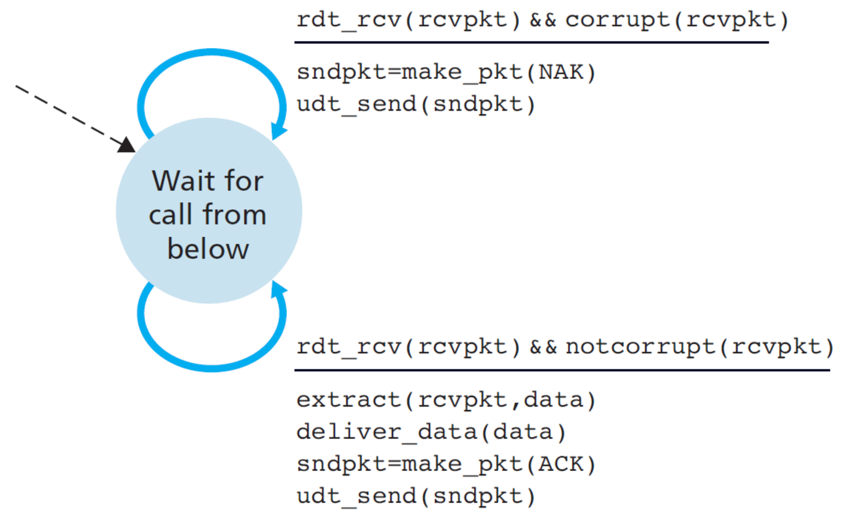
a. rdt1.0: sending side



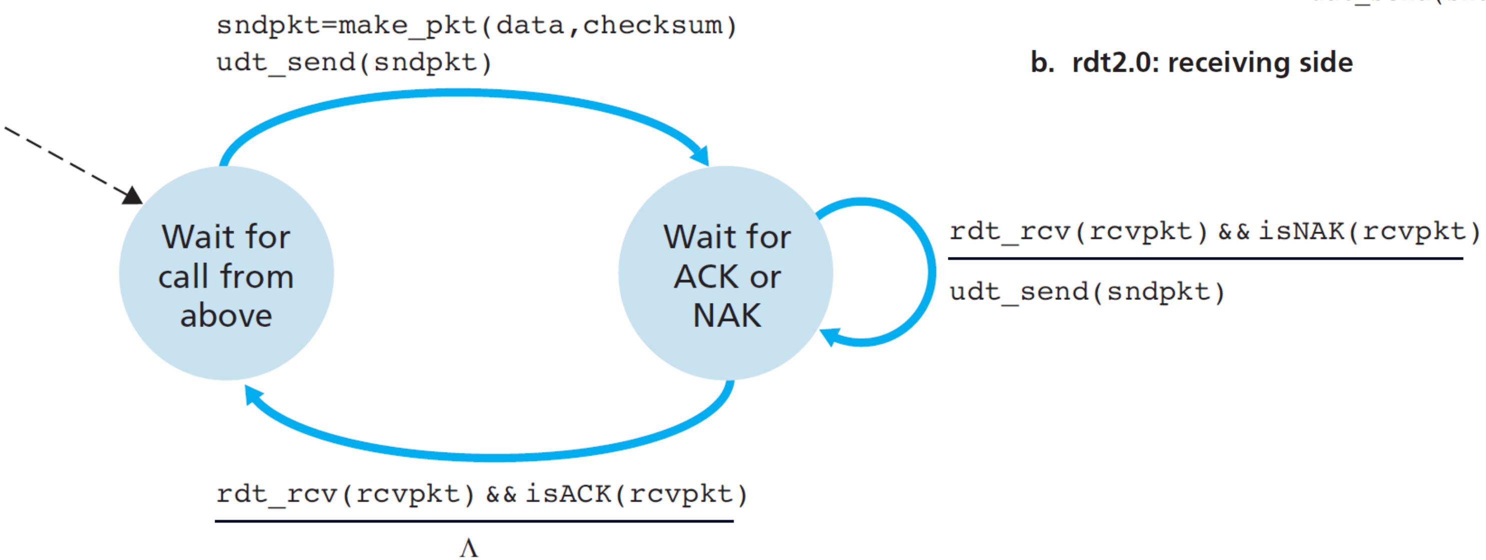
b. rdt1.0: receiving side

FSM

- **Case2**: Over channel with bit error
- Requirements:
  - Detection of Error
  - Inform the sender – use ACK/NAK
  - Retransmission



b. rdt2.0: receiving side

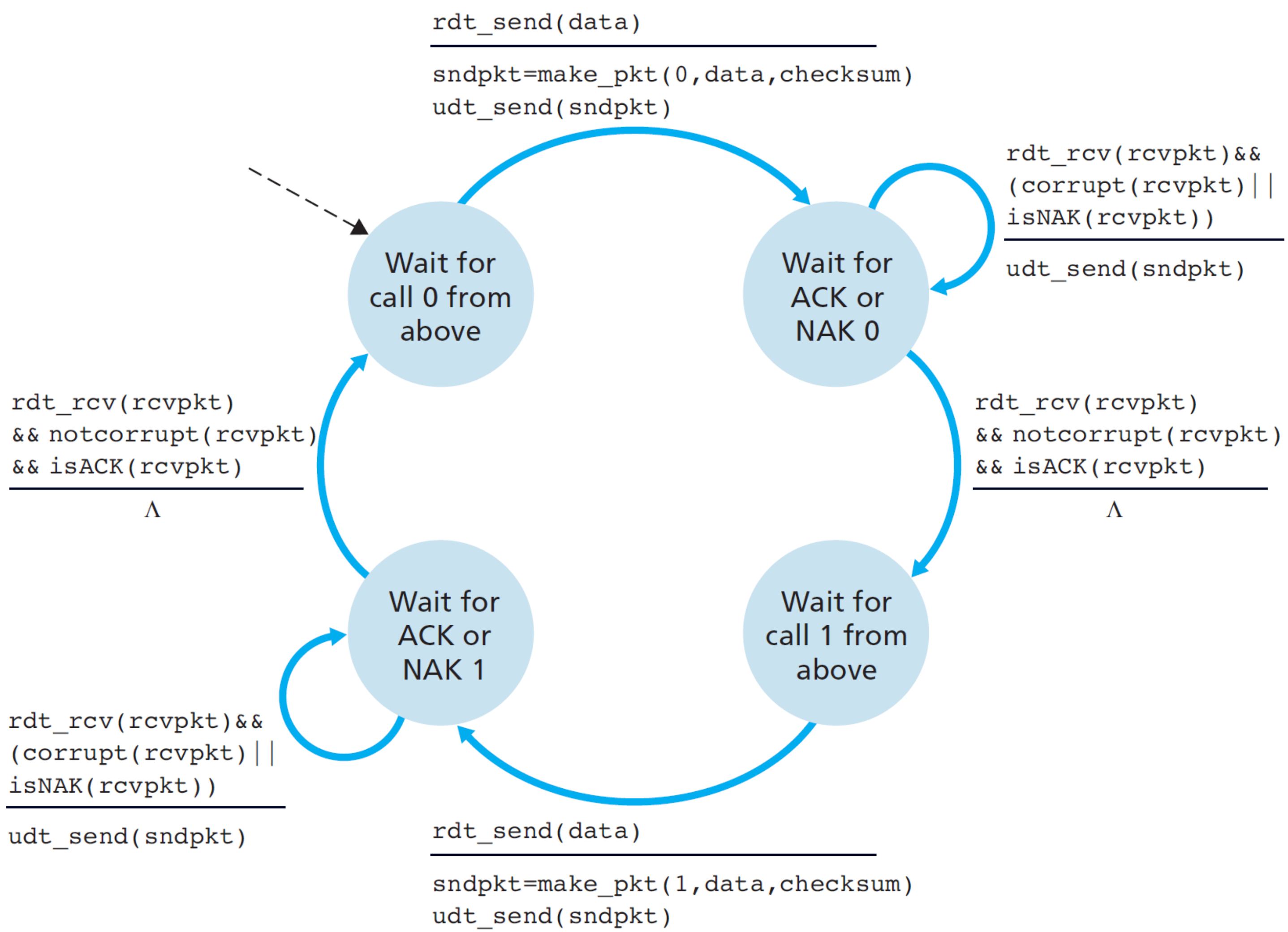


a. rdt2.0: sending side

FSM

- How to handle duplicate packet?

- Solution: Use packet sequence number



3.11 ♦ rdt2.1 sender

FSM

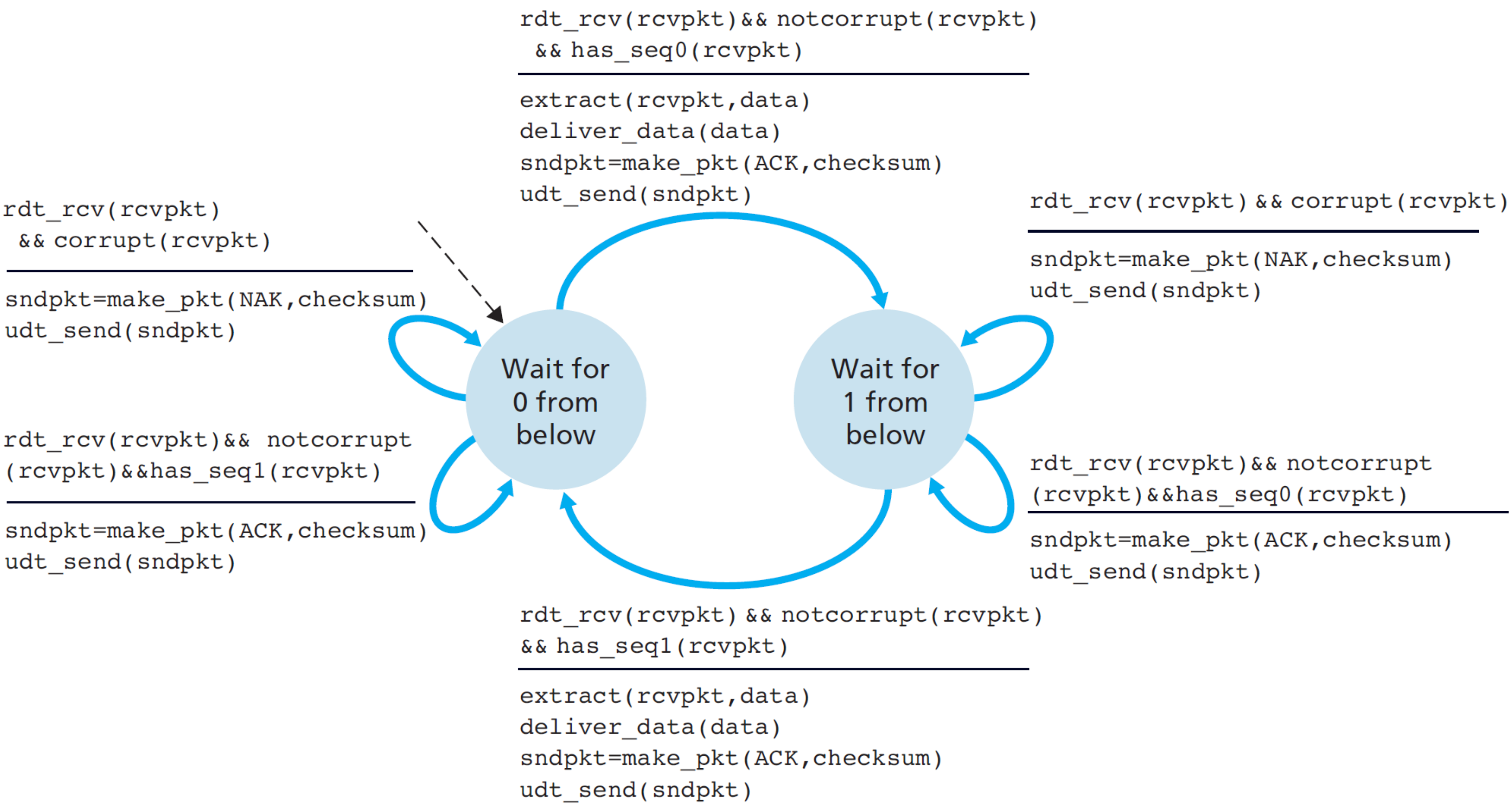
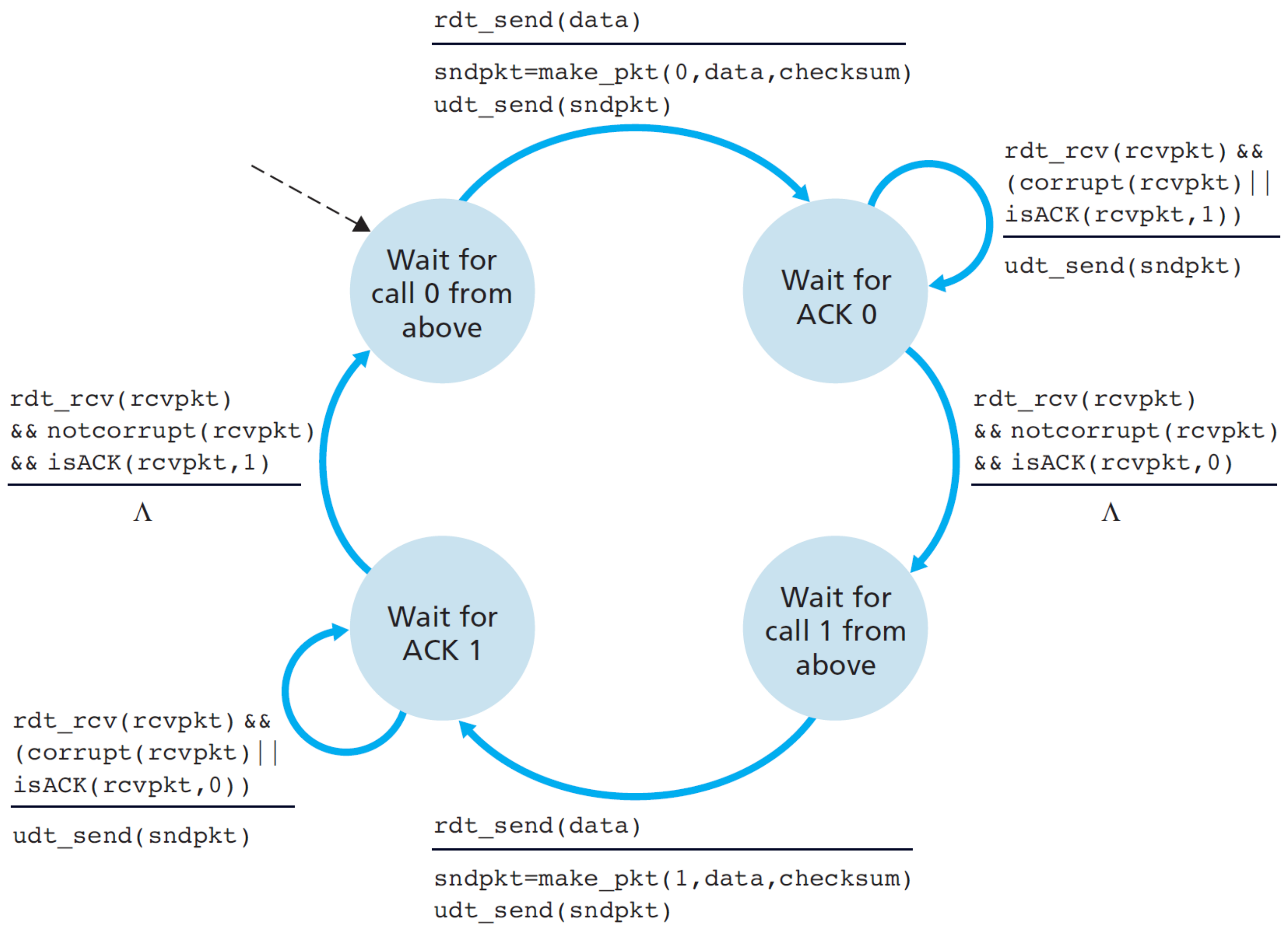


Figure 3.12 ♦ rdt2.1 receiver

FSM

- Does NAK required of out of order packets?
- Solution: No. Use packet sequence number with ACK



3.13 ♦ rdt2.2 sender

FSM

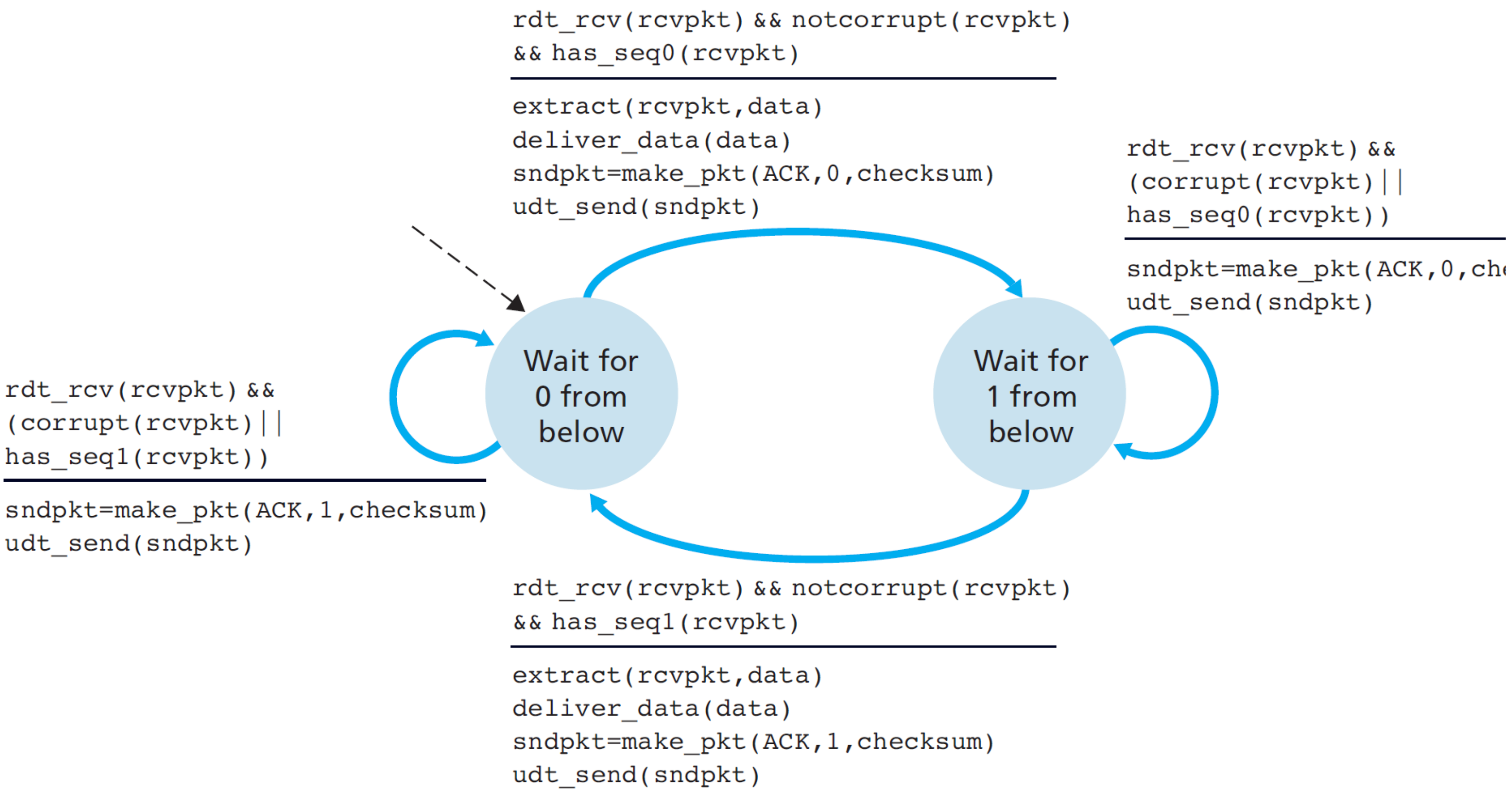
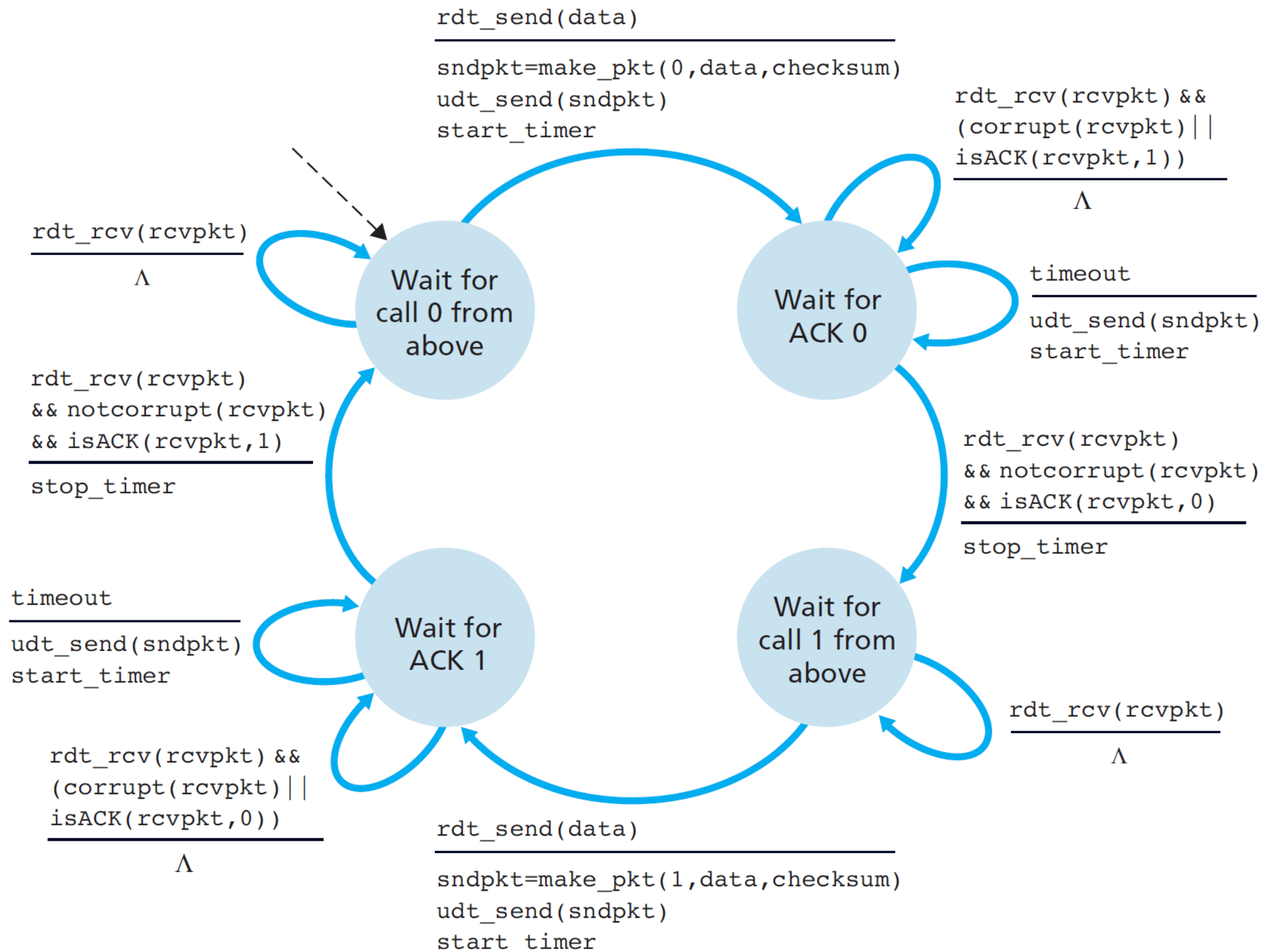


Figure 3.14 ♦ rdt2.2 receiver

FSM

- What will happen if ACK is lost?

- Solution: Use countdown timer



3.15 ♦ rdt3.0 sender

- Home task:
  - Final FSM of the receiver and sender for
    1) Stop-and-Wait
    2) Go-back-N
    3) Selective Repeat

Thanks!