NANYANG TECHNOLOGICAL UNIVERSITY

CE3005/SC2008/CZ3006: Computer Networks

Module 1-3: Part 1

Data Link Layer – Intro and Flow control

School of Computer Science and Engineering

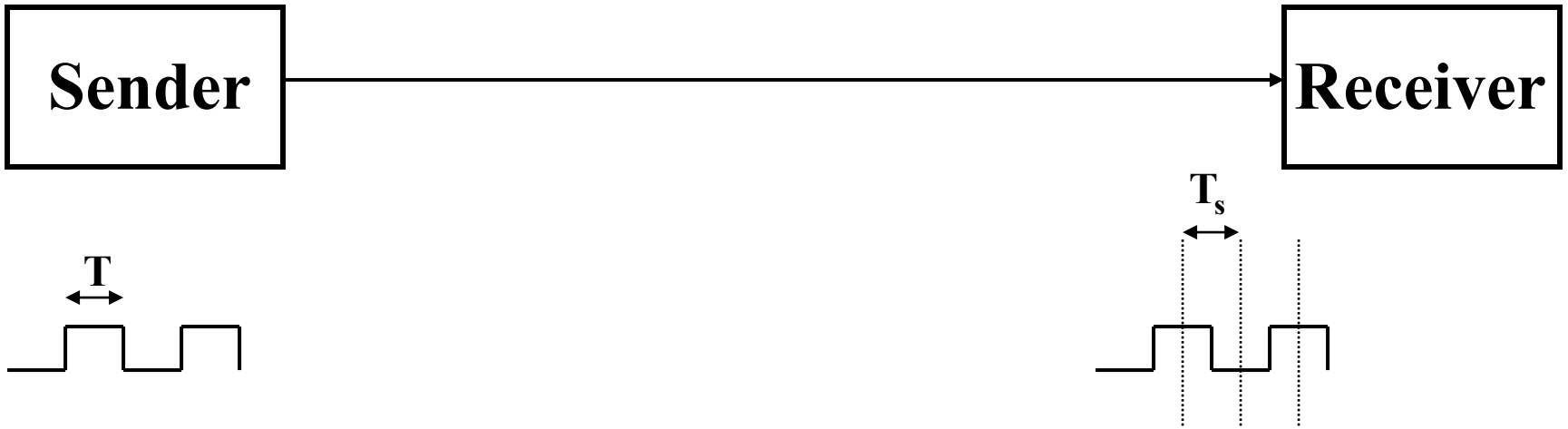
Data Link Control Protocol

Data and Computer Communication (William Stalling)
Chapter 6: Data Link Control

Functions of Data Link Control Protocol:

- Synchronizing sender and receiver (Synchronization)
- Managing Transmission on Link (Link Configuration; Who transmits when)
- Controlling Flow of Data/Frames (Flow Control)
- Detection and Recovery from Errors (Error Control)

Data Link Control Protocol: Synchronization



Sampling at the center of each bit time

Loss of Synchronization:

In practice, T_s is different than T. As a consequence, the timing of the receiver may slowly drift relative to the received signal, which will result in loss of synchronization.

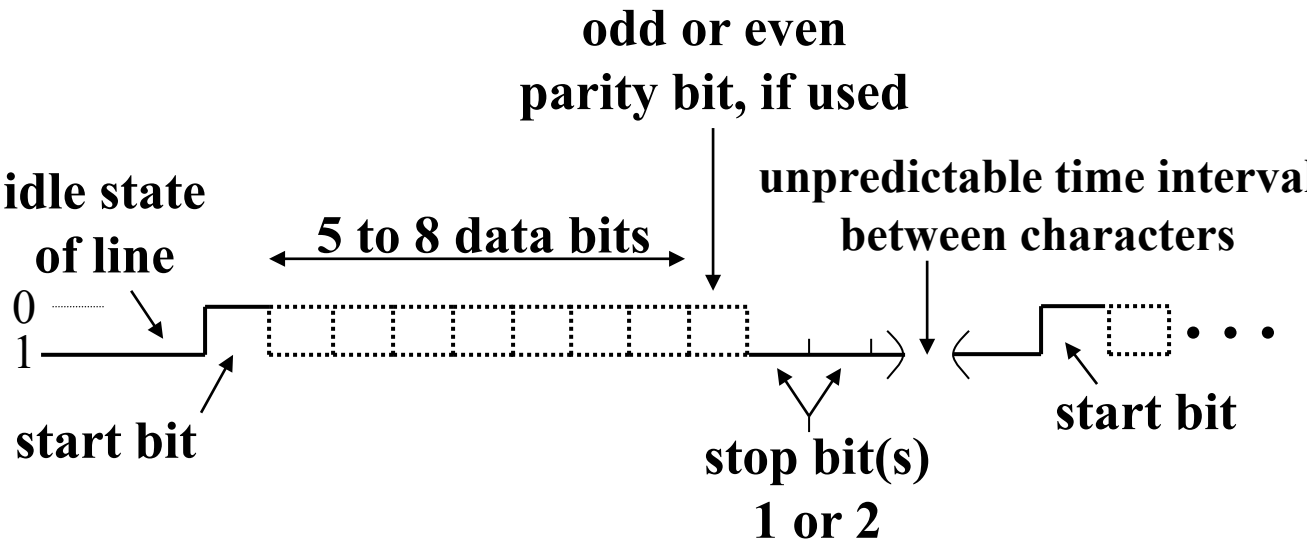
Synchronization

There are two approaches for achieving synchronization:

- **Asynchronous Transmission - Data is transmitted one character at a time.**
- **Synchronous Transmission - Data is transmitted in blocks.**

Asynchronous Transmission

- Data is transmitted one character at a time - 5 to 8 bits
- Timing or synchronization needs to be maintained within each character
- The receiver has the opportunity to resynchronize at the beginning of each new character



Asynchronous Transmission

Behavior:

- In a steady stream, interval between characters is uniform (length of stop element).
- In idle state, receiver looks for transition 1 to 0, and then samples next seven intervals (character length).
- Then looks for next 1 to 0 for next character.

Advantages and Disadvantages of Asynchronous Transmission:

- It is simple (timing requirements are modest) and inexpensive.
- It has high overheads. Overheads = Control bits / Total bits

Synchronous Transmission

Behavior:

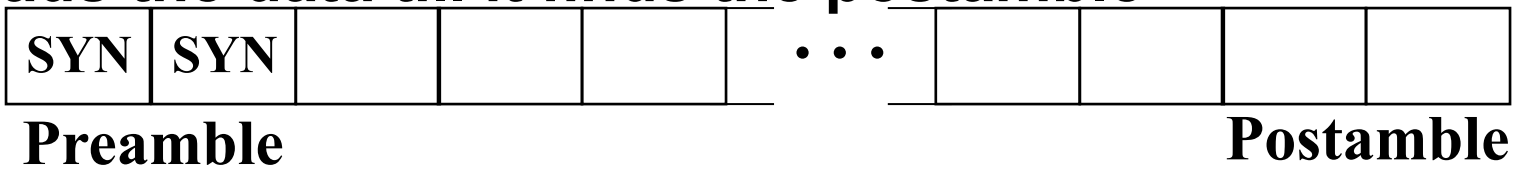
- In this mode, data is transmitted in blocks.
- Each block begins with a preamble (and sometimes ends with a postamble).
- There are two types of synchronous transmission:
  - Character oriented, and
  - Bit oriented.

Synchronous Transmission: Character Oriented

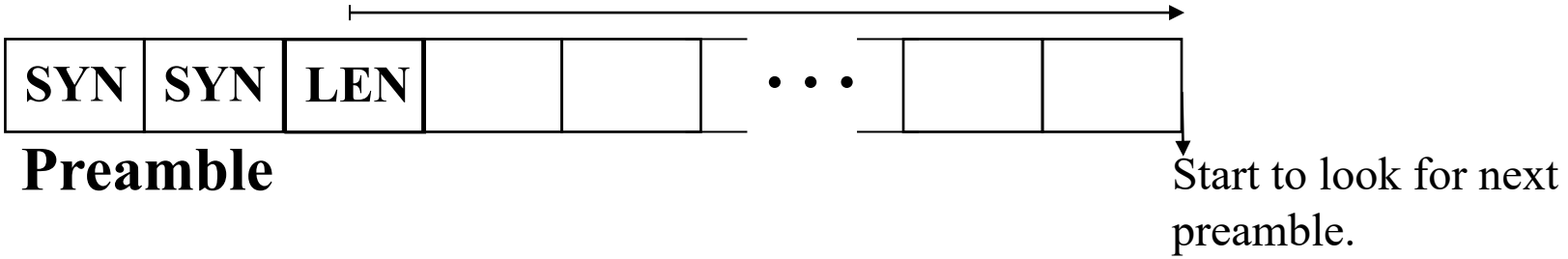
The frame consists of a sequence of bytes.

There are two possibilities:

1. **Frame consists of preamble and postamble - The receiver having detected the beginning of the frame, reads the data till it finds the postamble**



2. **Frame consists of preamble and length of frame - The receiver having received the preamble, looks for extra information regarding the length of the frame**



Note: SYN is a unique bit pattern that signals the receiver the beginning of a block.

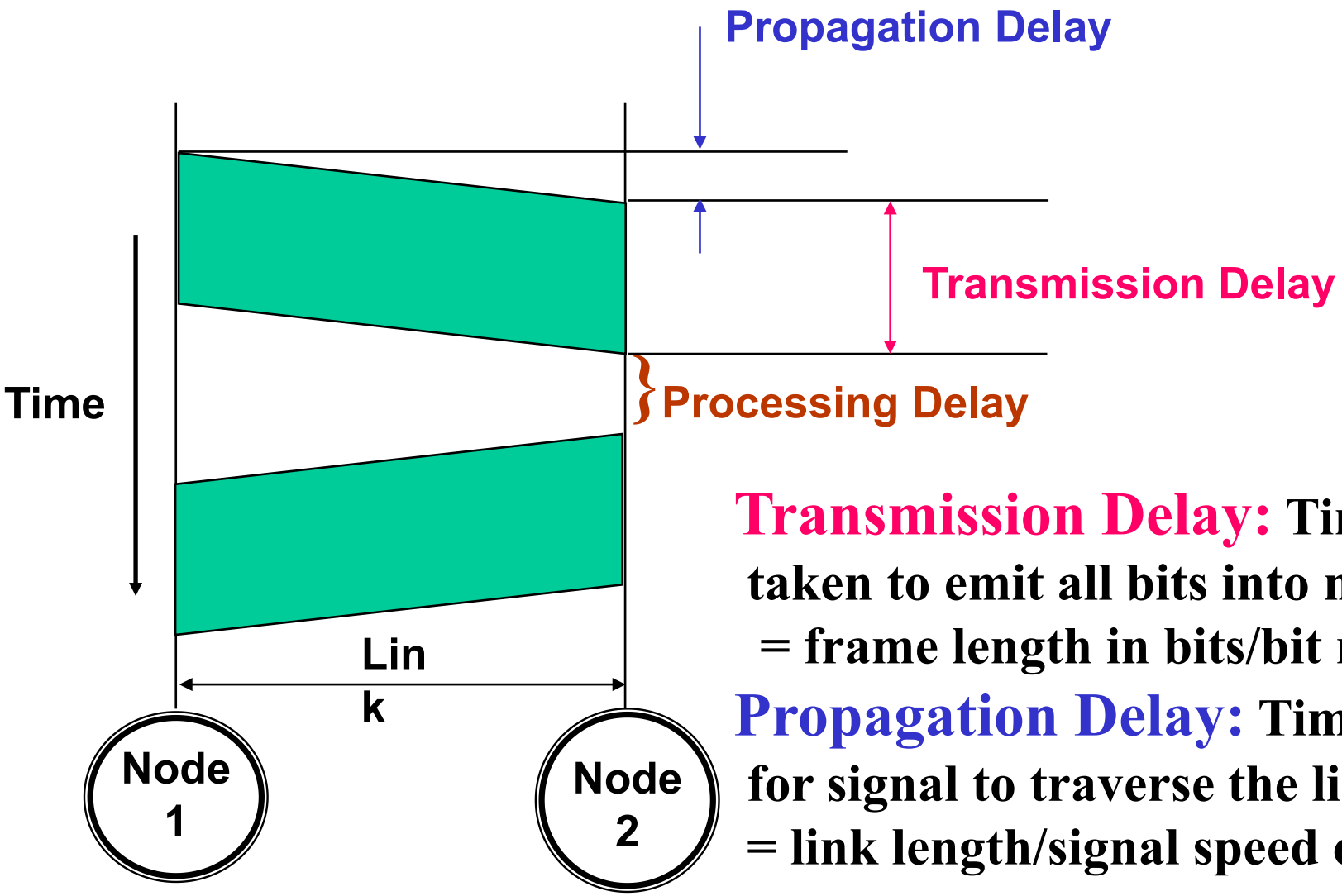
Synchronous Transmission: Bit Oriented

Behavior:

- In this mode, the frame is treated as a sequence of bits. Neither data nor control information (header & trailer) is interpreted in units of x-bit characters.
- A special bit pattern (called flag) indicates the beginning of a frame. A similar bit pattern indicates end of the frame.
- Control information and data cannot contain bit pattern used for flags.

Example: High-level Data Link Control (HDLC) protocol uses 01111110 bit pattern for flags.

Different Type of Delays on a Link



Transmission Delay: Time taken to emit all bits into medium, = frame length in bits/bit rate

Propagation Delay: Time taken for signal to traverse the link, = link length/signal speed on link

Link Configuration

Link configuration determines who gets to transmit when on a link.

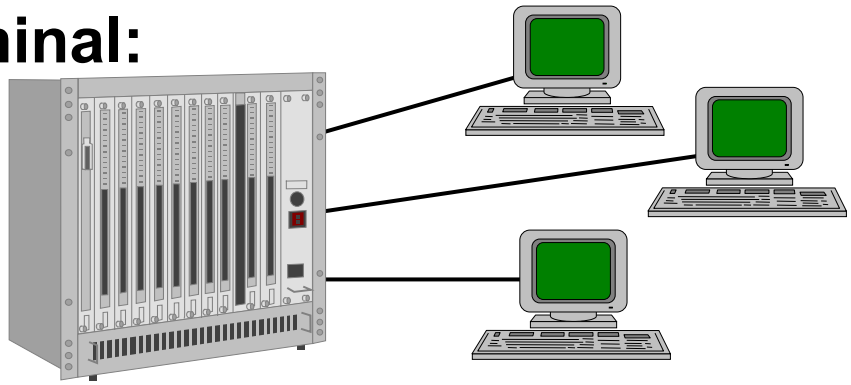
There are two characteristics that influence link configurations:

- **Topology -** refers to physical arrangement of stations on a link. Two topologies are of interest:
  - Point-to-point
  - Multi-point (not much in use now)
- **Duplexity -**
  - **Half Duplex:** Only one party may transmit at a time. It requires only one data path.
  - **Full Duplex:** Allows simultaneous transmission and reception between two parties. It requires two data paths or two different carrier frequencies (in analog transmission).

Topology: Point-to-Point

Examples:

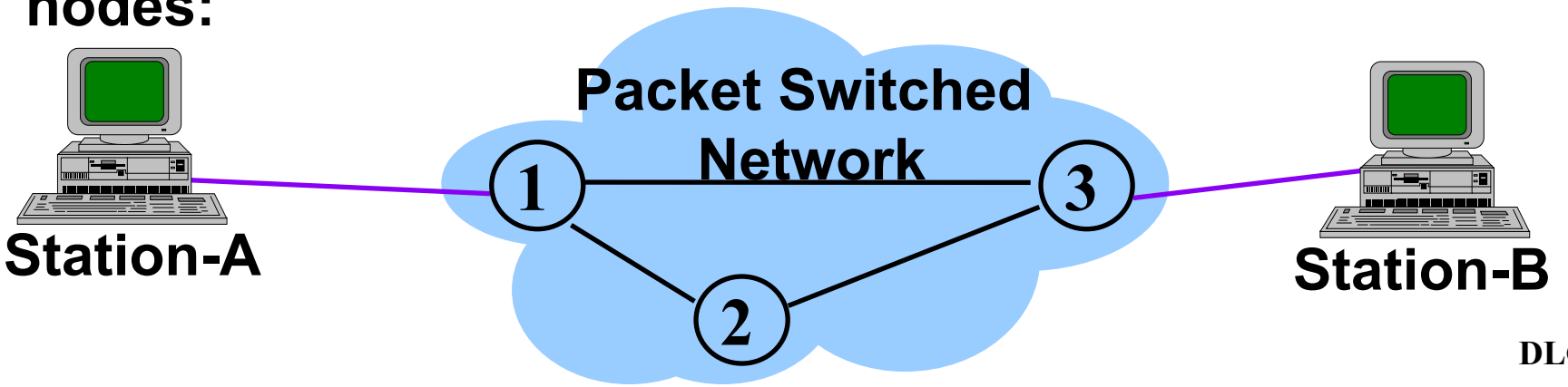
1. A separate line from a computer to each terminal. In this case, the computer must have an I/O port for each terminal:



Terminals (secondary)

Computer (primary)

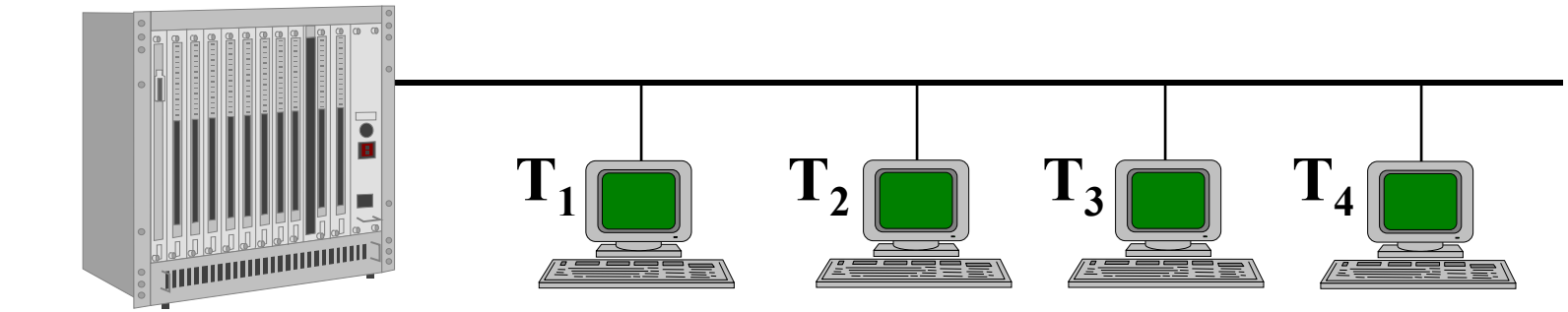
2. A link connecting a remote station with a switching node, or a link providing connection between two switching nodes:



Topology: Multi-point

Example:

Only a single transmission line is needed, and the computer requires only one I/O port for connecting terminals:

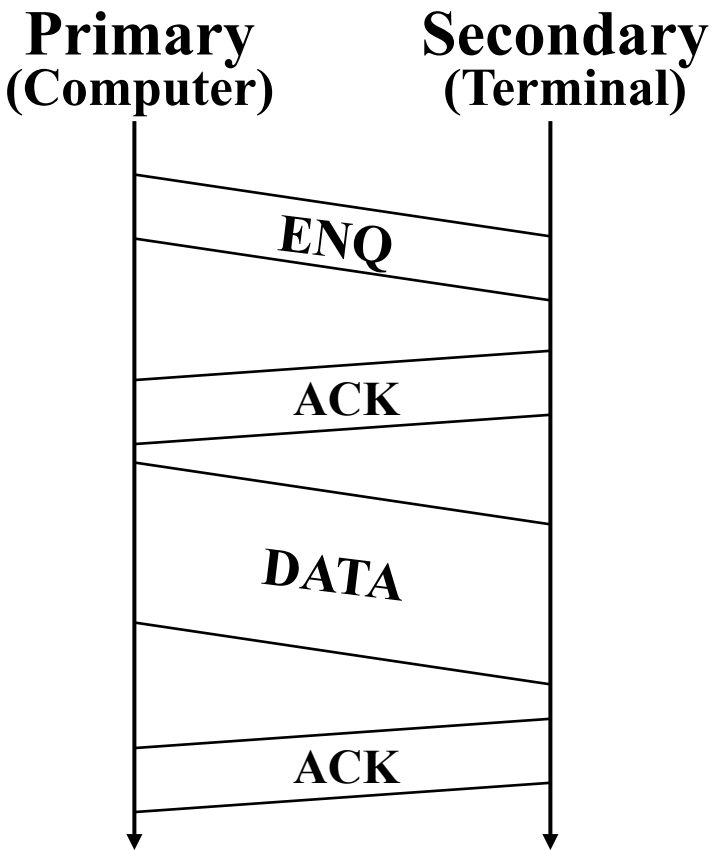


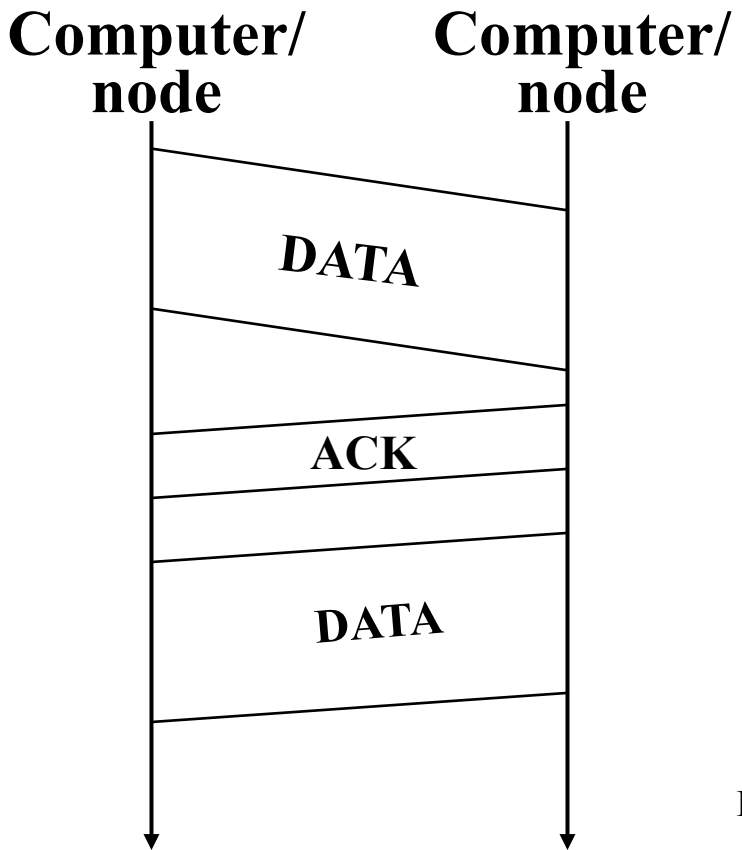
Notes:

- **Multi-point line is useful when the terminals are transmitting only for a fraction of the time.**
- **More complex versions of the multi-point topology are found in Local Area Networks.**

Data Link Configurations: Point-to-Point Links

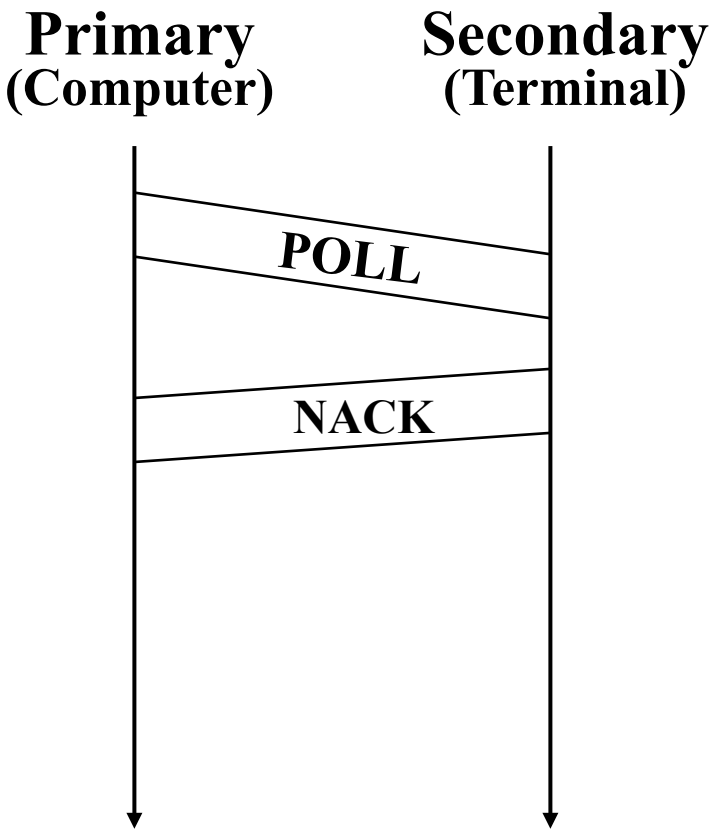
- In this case, the line discipline is simple since the link is dedicated to the communication between two parties.
- It can be used for connecting a terminal to a computer or between two computers (host/switching node).



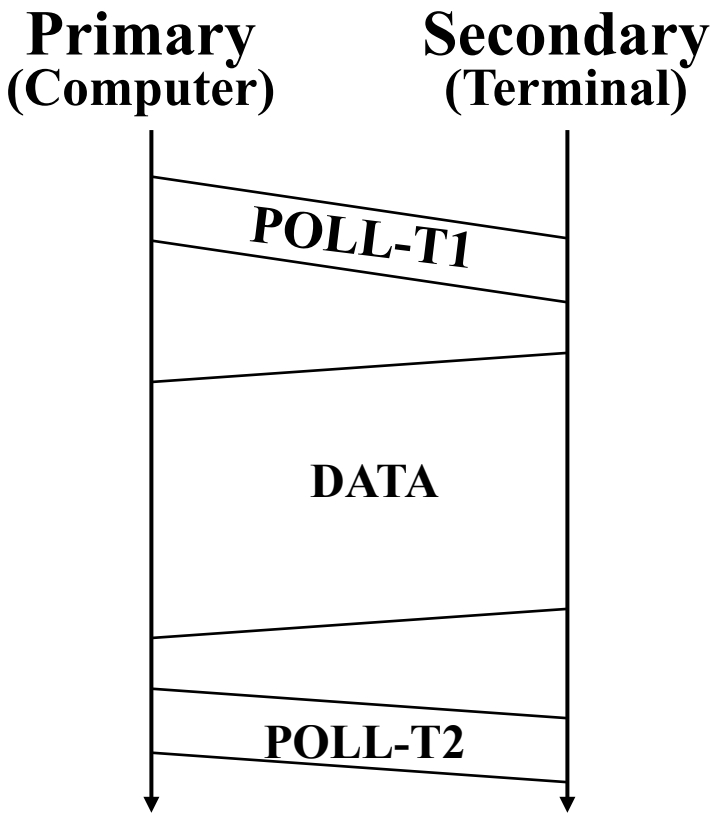


Data Link Configurations: Multi-Point Links

Primary station polls the secondary stations (terminals) one by one.



a) Polled terminal has nothing to send

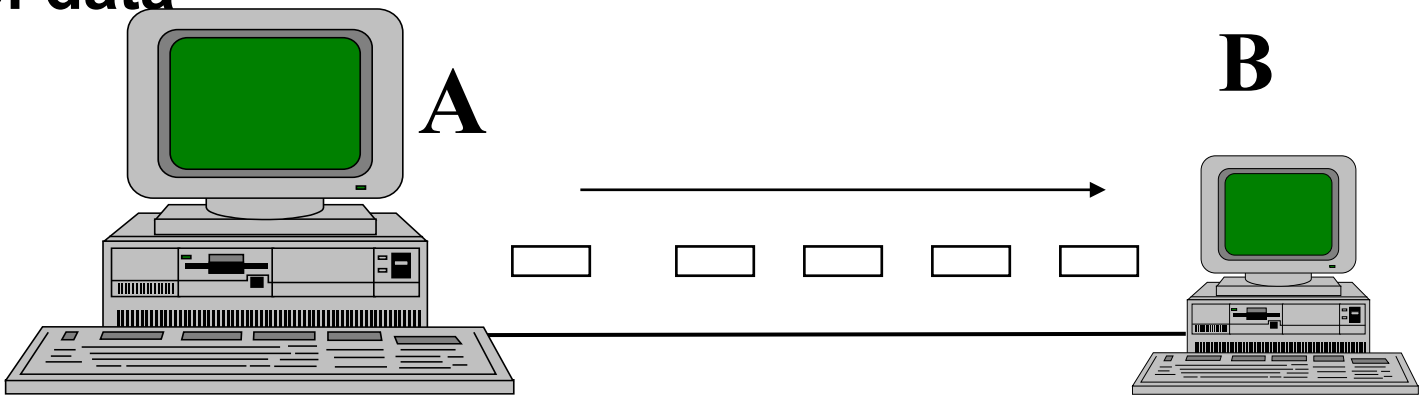


b) Polled terminal has data to send

Data Link Layer: Flow Control

Definition:

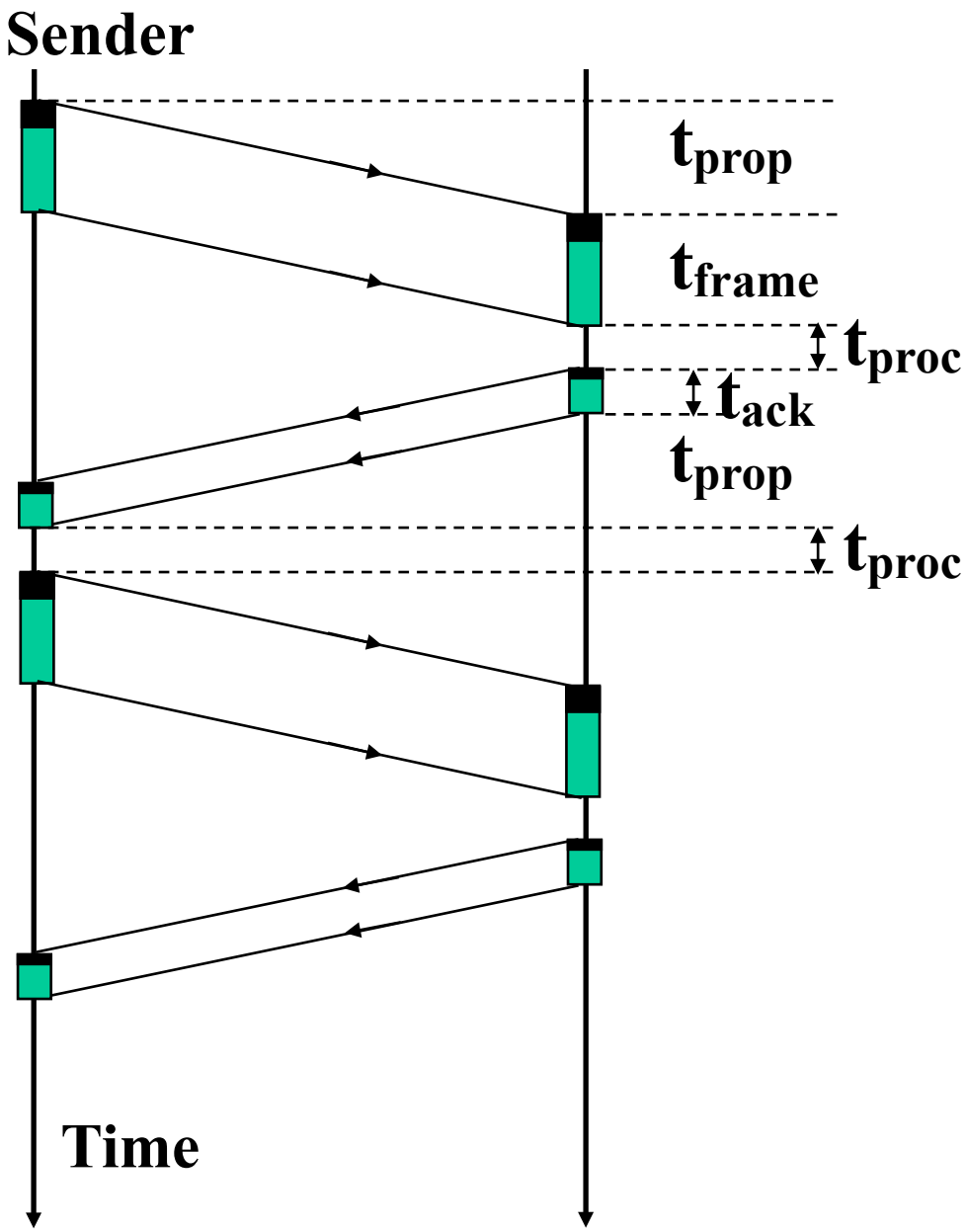
Flow control is a technique for assuring that a transmitting station does not overwhelm a receiving station with data, i.e., buffers at the receiver do not overflow resulting in loss of data



Two flow control mechanisms are of interest:

- **Stop-and-Wait (also referred as "send and wait")**
- **Sliding Window**

Stop-and-Wait Flow Control



Flow Control Procedure:

- Source transmits frame.
- Destination receives frame and replies with ACK (acknowledgement).
- Source waits for ACK before sending next frame.
- Destination can stop flow by not sending ACK.

Performance of Stop-and-Wait Flow Control

Performance Considerations: Inefficient line utilization for high data rate and for very long distance between the sender and the receiver.

- Time required to send one frame and receive an ack, $T_F = T_{frame} + T_{ack} + 2 \times T_{prop} + 2 \times T_{proc}$, assuming the transmission on the link is error-free, and processing time at the receiver and sender are equal.
- Link Utilization U = Link Efficiency = T_{frame} / T_F
- Link Utilization or Link Efficiency in % = $T_{frame} \times 100 / T_F$ %
- If we ignore T_{ack} and T_{proc}, $U = T_{frame} / (T_{frame} + 2 \times T_{prop})$

$$= 1 / (1 + 2\ T_{prop} / T_{frame})$$

Performance of Stop-and-Wait Flow Control:

Definition of Parameter '*a*' - Normalized Propagation Time

(Refer to next slide)

- Ignoring T_{ack} and T_{proc}, $U = 1 / (1 + 2\ T_{prop} / T_{frame})$
- Let us define parameter $a = T_{prop} / T_{frame}$

  = Propagation Time/Transmission Time

  = Normalized Propagation Time
- $U = 1/(1 + 2a)$

Performance of Stop-and-Wait Flow Control:

'*a*' - Normalized Propagation Time

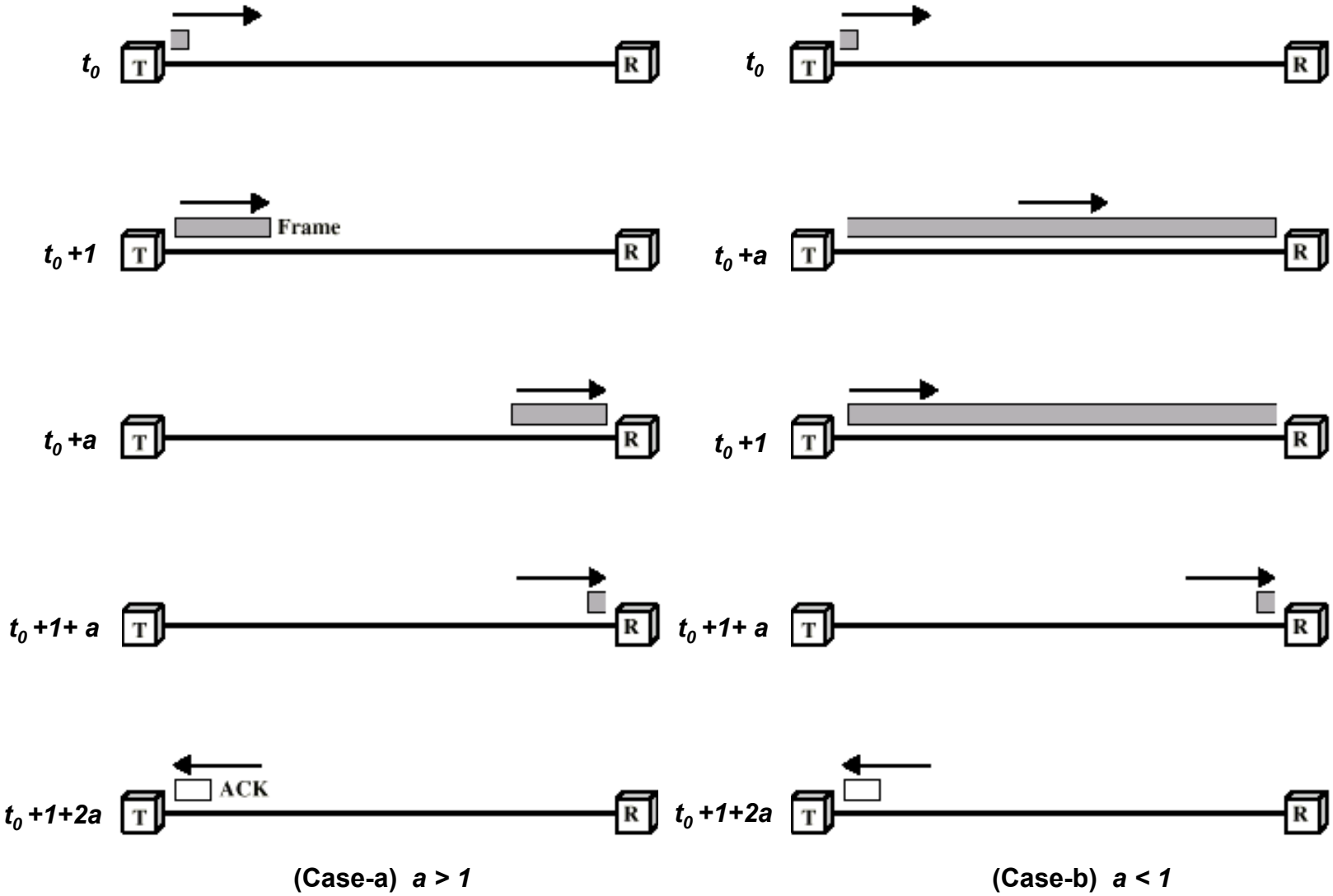
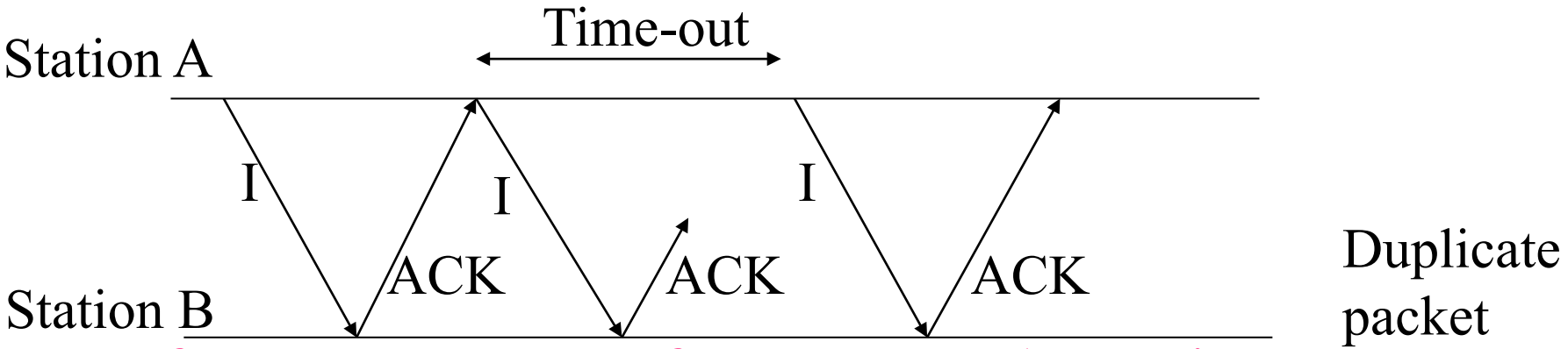


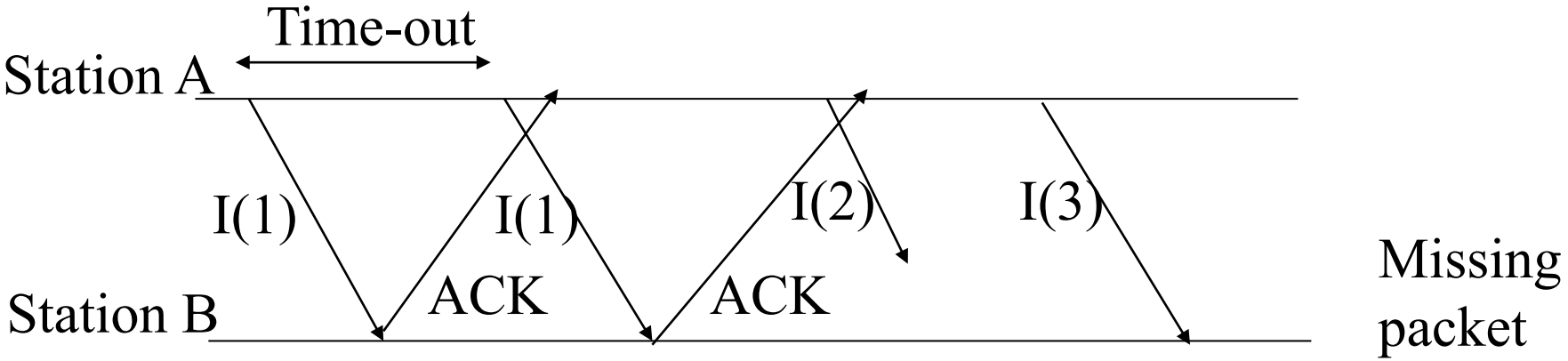
Figure 7.2 Stop-and-Wait Link Utilization (transmission time = 1; propagation time = *a*)

Stop-and-Wait Flow Control

Need for sender information frame number:

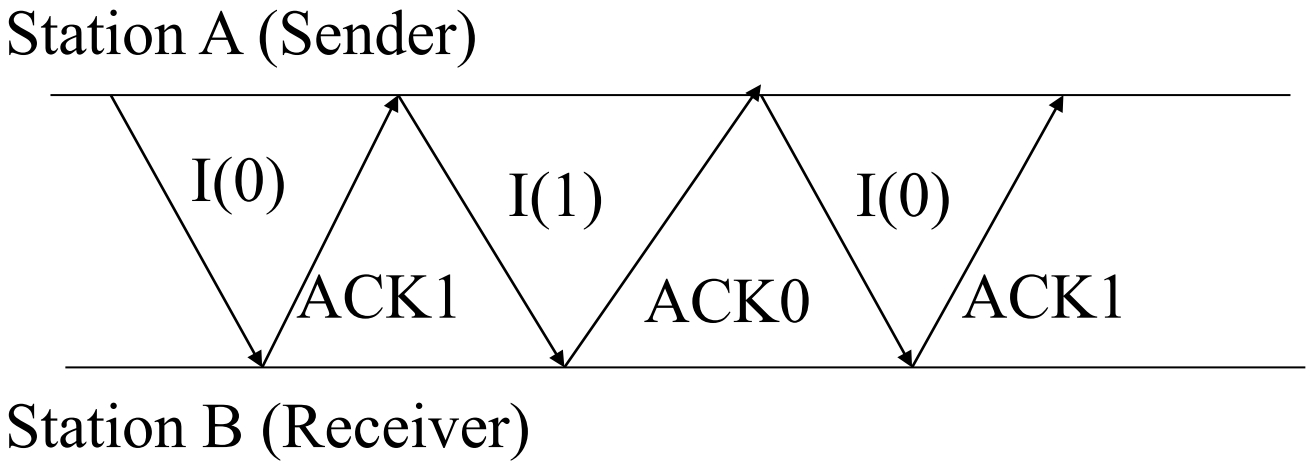


Need for ackowledgement frame number(when timeout is shorter than roundtrip time).



Stop-and-Wait Flow Control

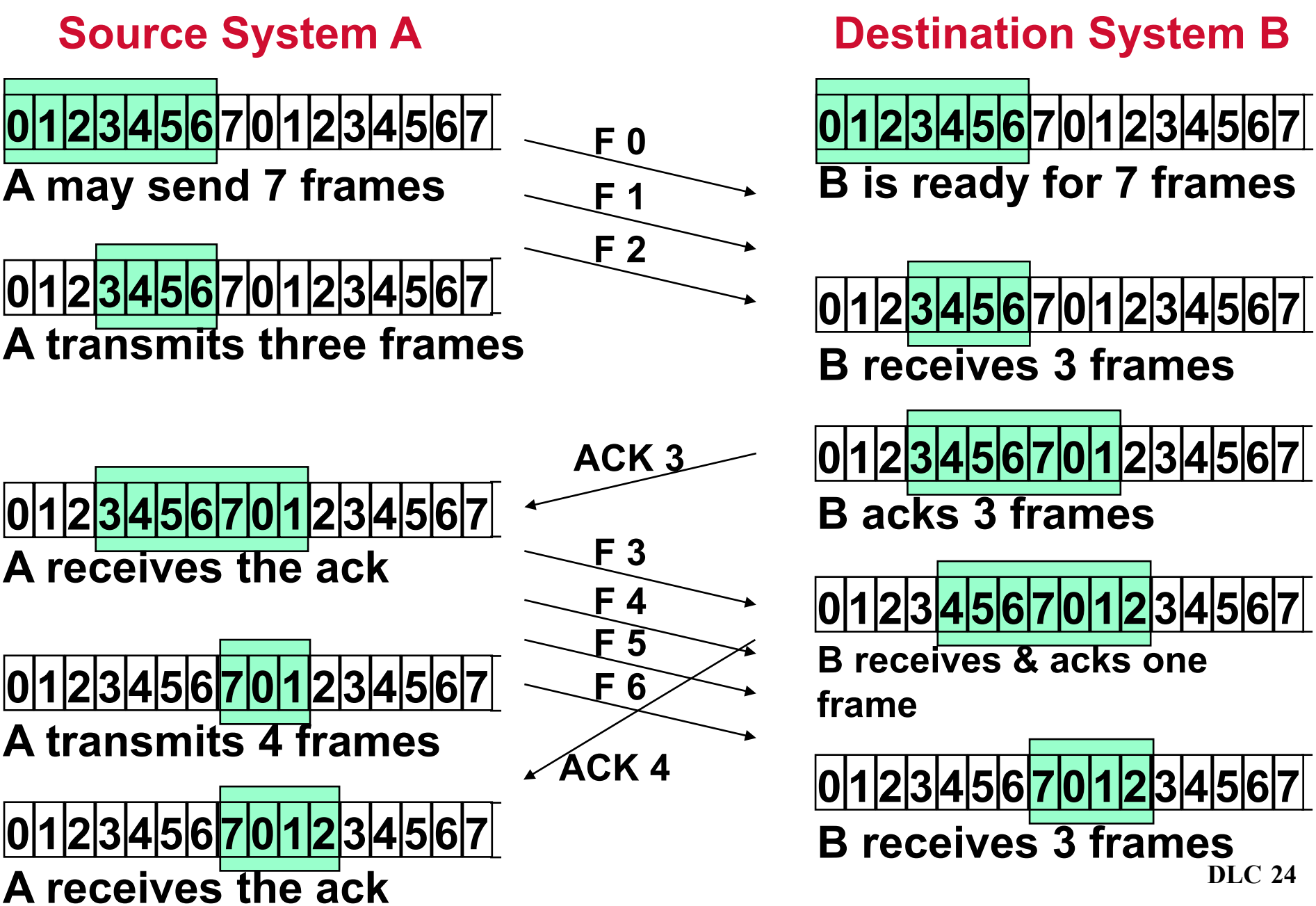
Packet with frame sequence number



Sliding Window Flow Control

- It is widely used method of flow control in traditional networks.
- Allows multiple frames to be in transit.
- Sender and Receiver have buffer *W* frame long.
- Sender can send up to *W* frames without receiving ACK
- Each frame is numbered.
- ACK includes number of next frame expected.
- Sequence number bounded by size of field (*k*)
  - Frames are numbered modulo 2^k

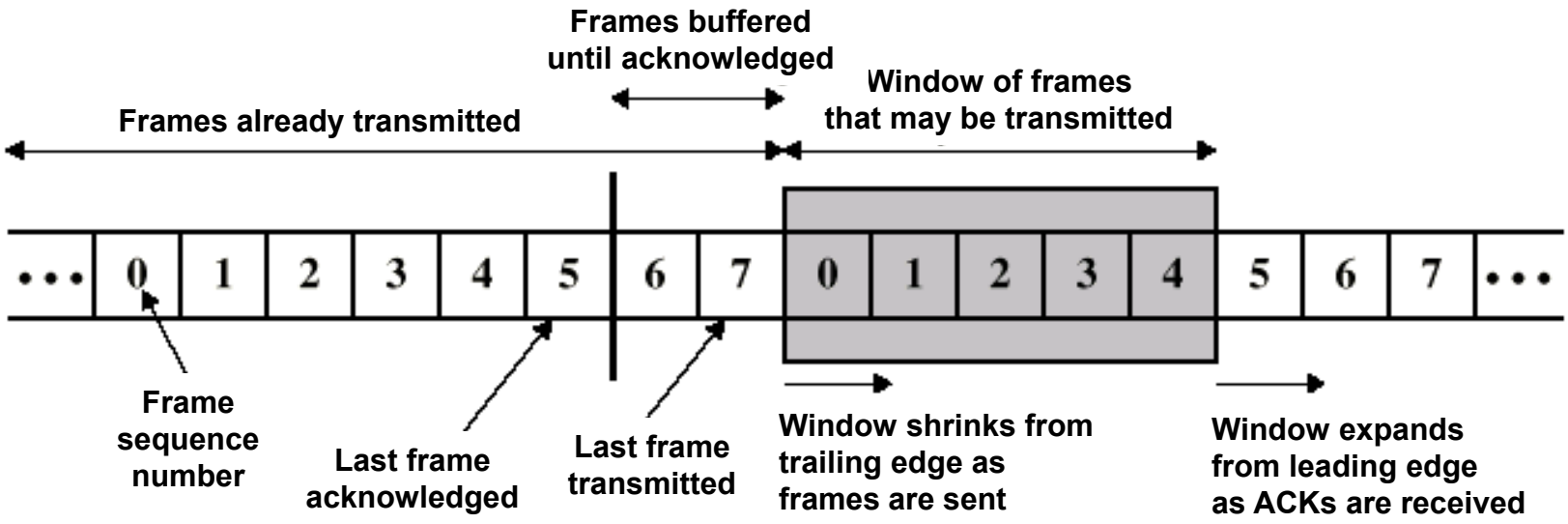
Sliding Window Flow Control: Example



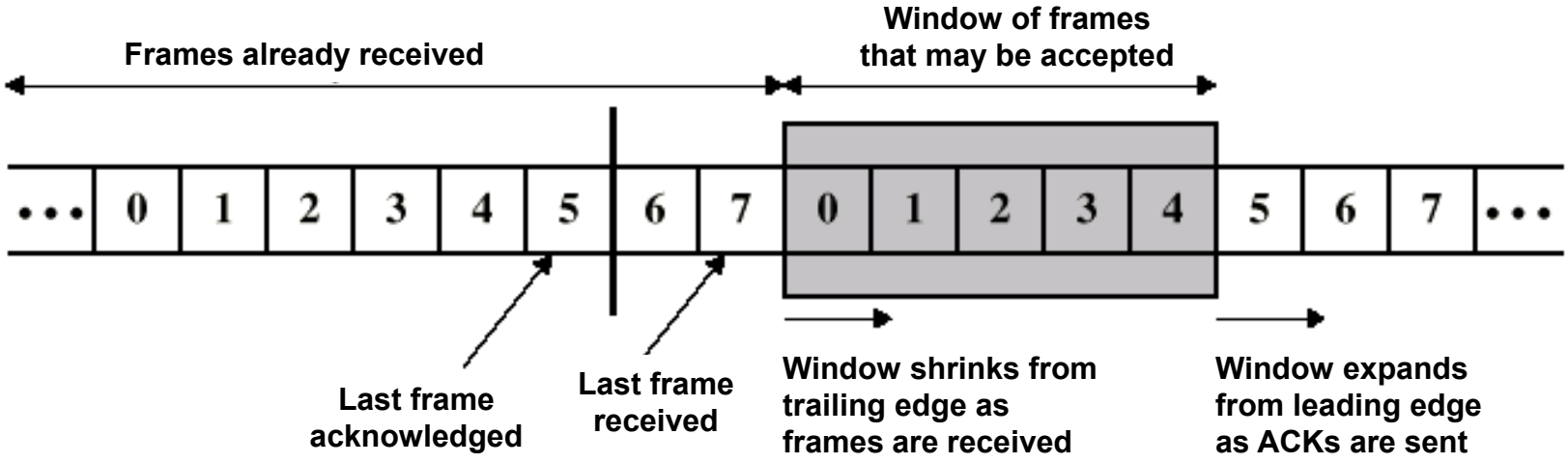
Sliding Window Flow Control: How does it work?

- Sender maintains a window, containing frame numbers that can be transmitted.
- Sender window shrinks from trailing edge (left side) as frames are sent.
- At the receiver window shrinks from trailing edge as frames are received.
- Window at the receiver expands from the leading edge (right side) as ACKs are sent.
- Frames are buffered at the sender until acknowledged.
- Window at the sender expands from the leading edge as ACKs are received.

Sliding Window Flow Control: How does it work?



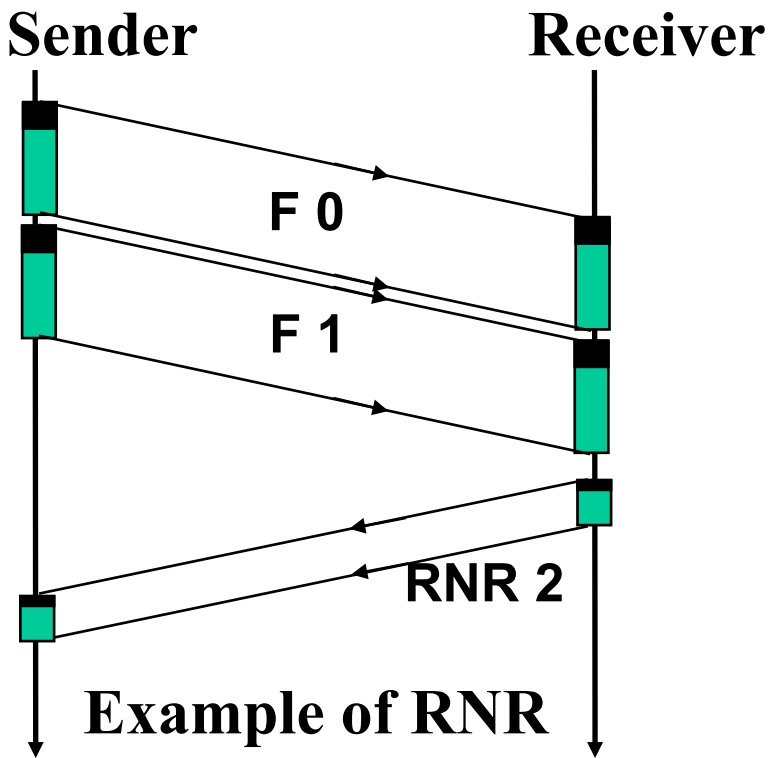
(a) Sender's perspective



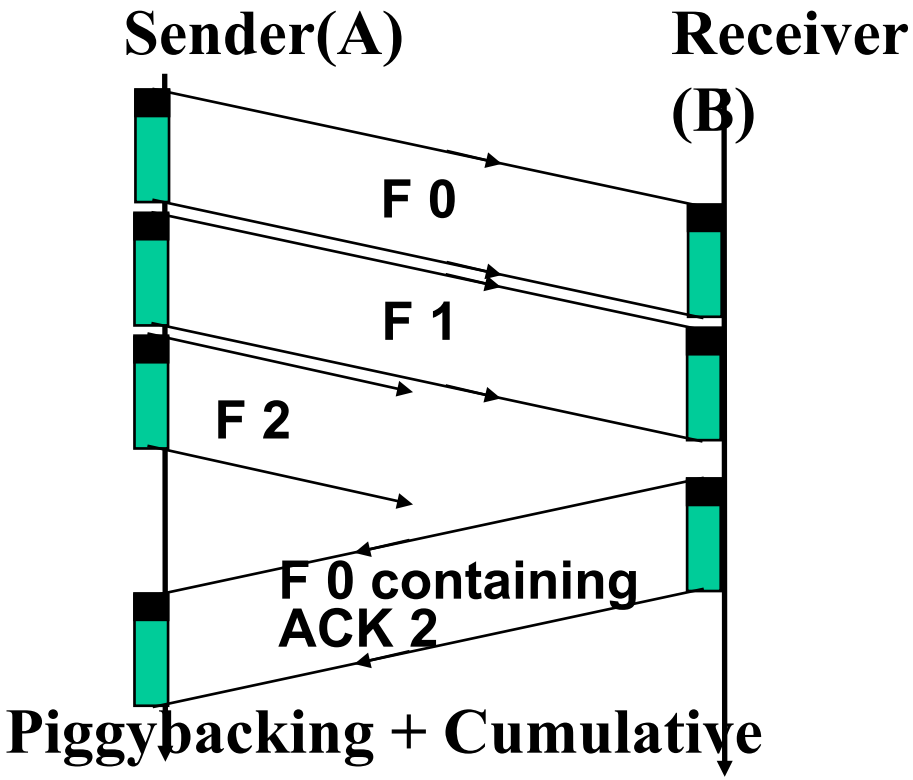
(a) Receiver's perspective

Sliding Window Flow Control: Other Features

- Receiver can acknowledge frames without permitting further transmission (by sending 'Receiver Not Ready' frame). Must send a normal acknowledgement to resume.
- ACK can be piggybacked on the data frames in the reverse direction.



Example of RNR

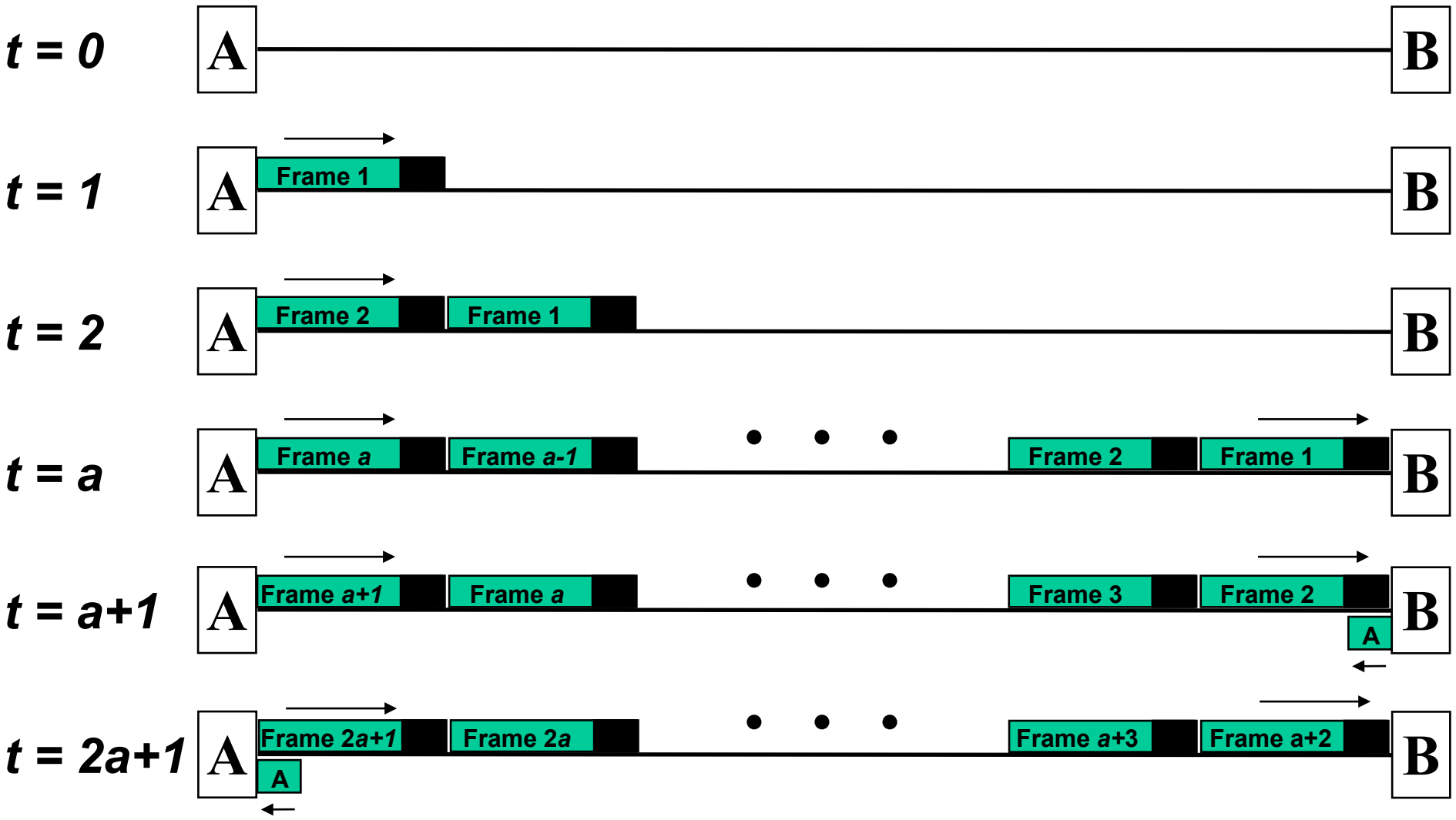


Piggybacking + Cumulative Acknowledgement

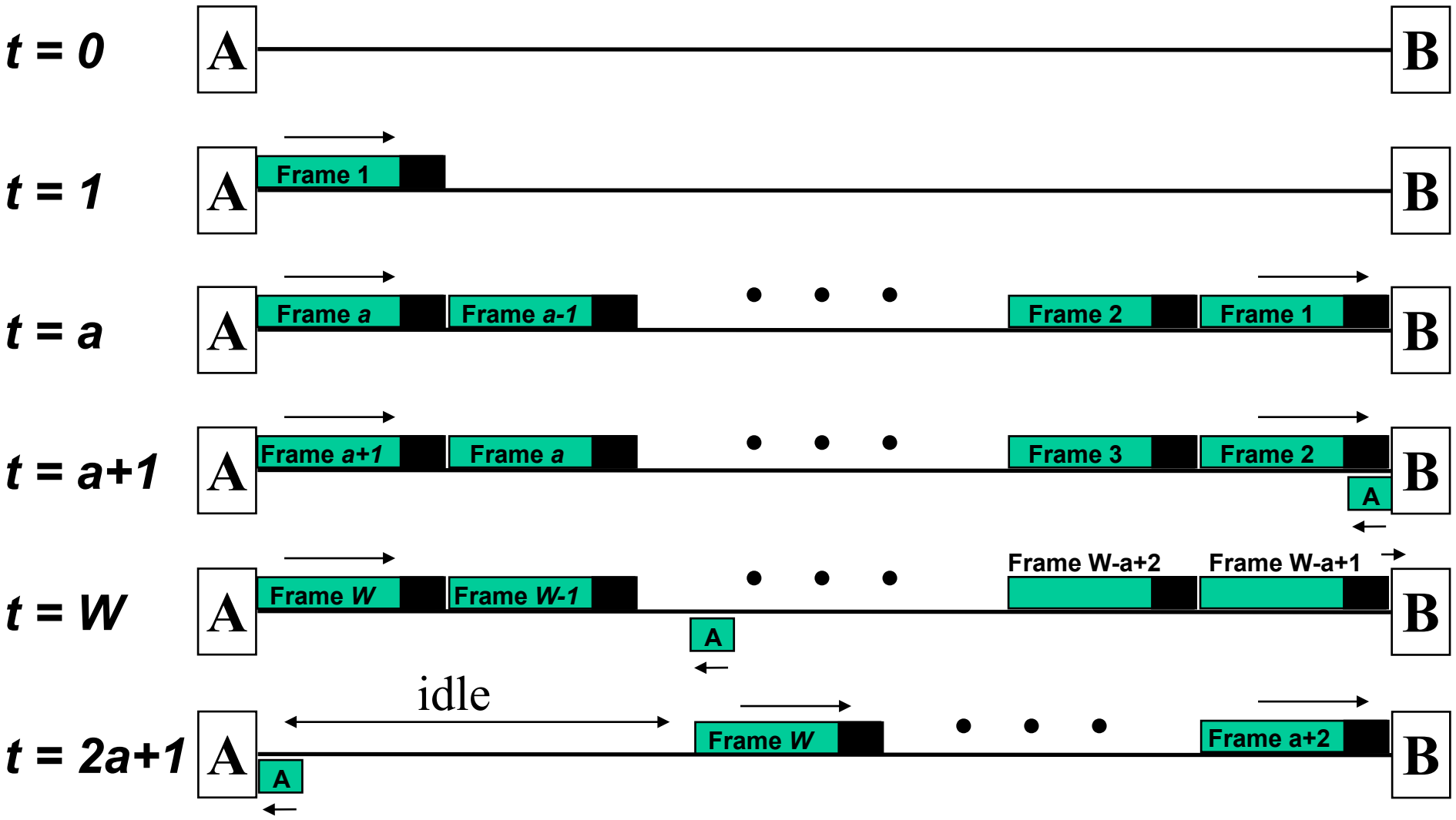
Sliding window parameters

- W, Window size
- k, number of bits allocated for frame sequencing
- Frame number from 0 till 2^k-1
- $W < 2^k$

Performance of Sliding Window Flow Control: $W \geq 2a + 1$



Performance of Sliding Window Flow Control: *W < 2a +1*



Sliding Window Flow Control: Performance

Performance depends upon (under error-free operation):

- **Parameter *a*, and**
- **Window size, *W*.**

Assumption: T_{ack} and T_{proc} are negligible.

Frame transmission time = 1 (normalized with respect to itself).

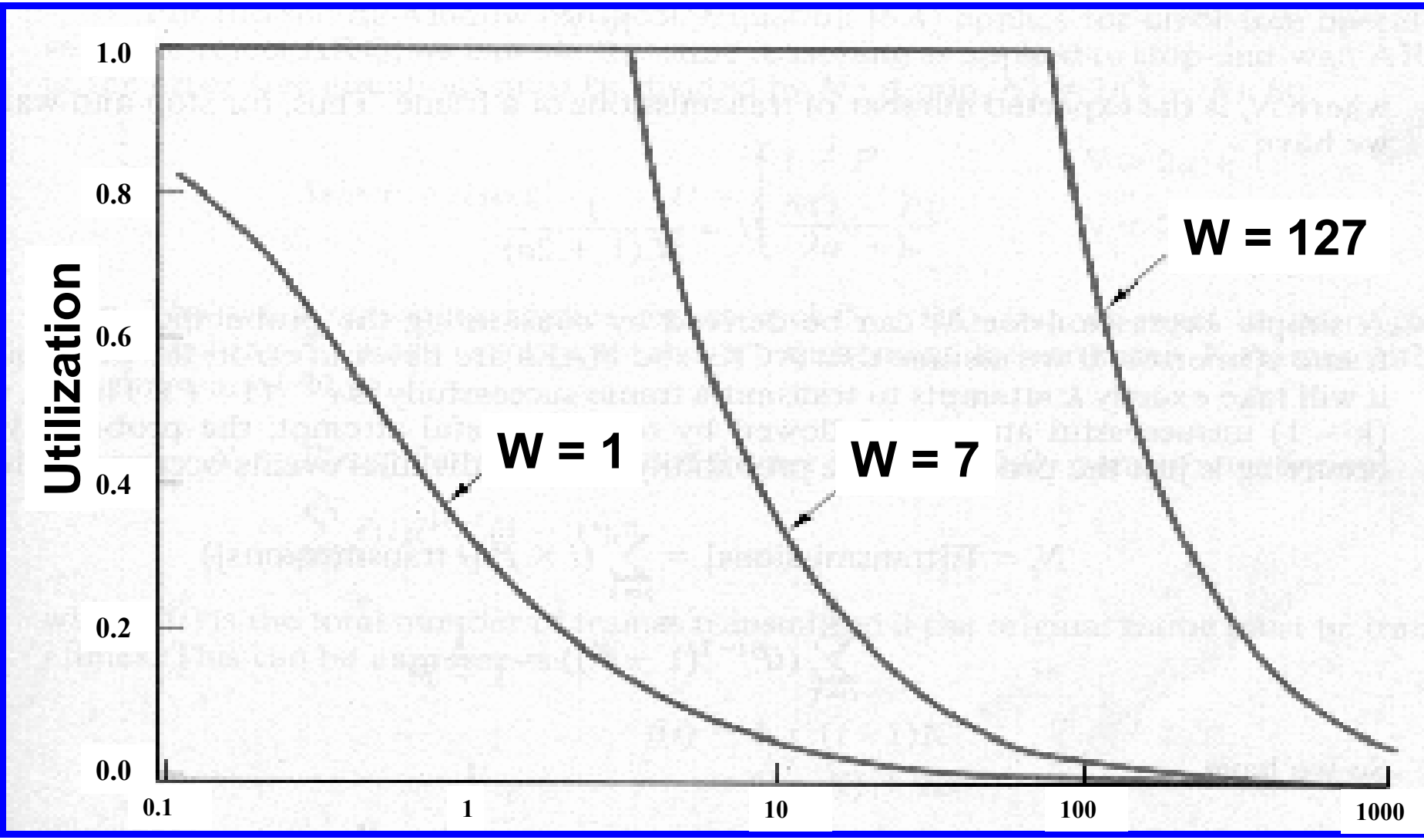
Propagation delay = *a*

We need to consider two cases:

- ***W* ≥ *2a* +*1*** : **Station can transmit continuously without exhausting its window.** ➡ ***U* = *1.0***
- ***W* < *2a* +*1*** : **Station's window is exhausted at *t* = *W*, and the station cannot send additional frames until *t* = *2a* +*1*.**

  ➡ ***U* = *W*/(*1* + *2a*)**

Performance of Sliding Window Flow Control



Performance(Flow control)

	W< 1+2a	W> 1+2a	Comments
Start-stop	1/ (1 + 2a)	1/ (1 + 2a)	W= 1
Sliding Window	W/(1 + 2a)	1	