NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Week 4: Data Link Layer

EE3017/IM2003 Computer Communications

School of Electrical and Electronic Engineering

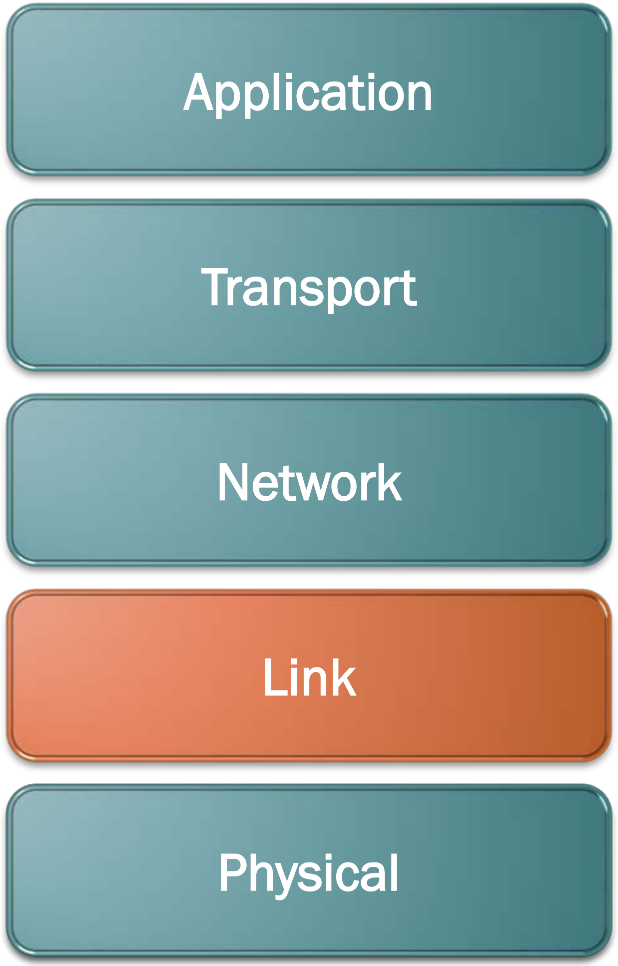
Dr. SHAO Xuguang, Michelle

Email: XGSHAO@ntu.edu.sg

Room: S1-B1a-10

Phone: 6513-7648

Topic Outline



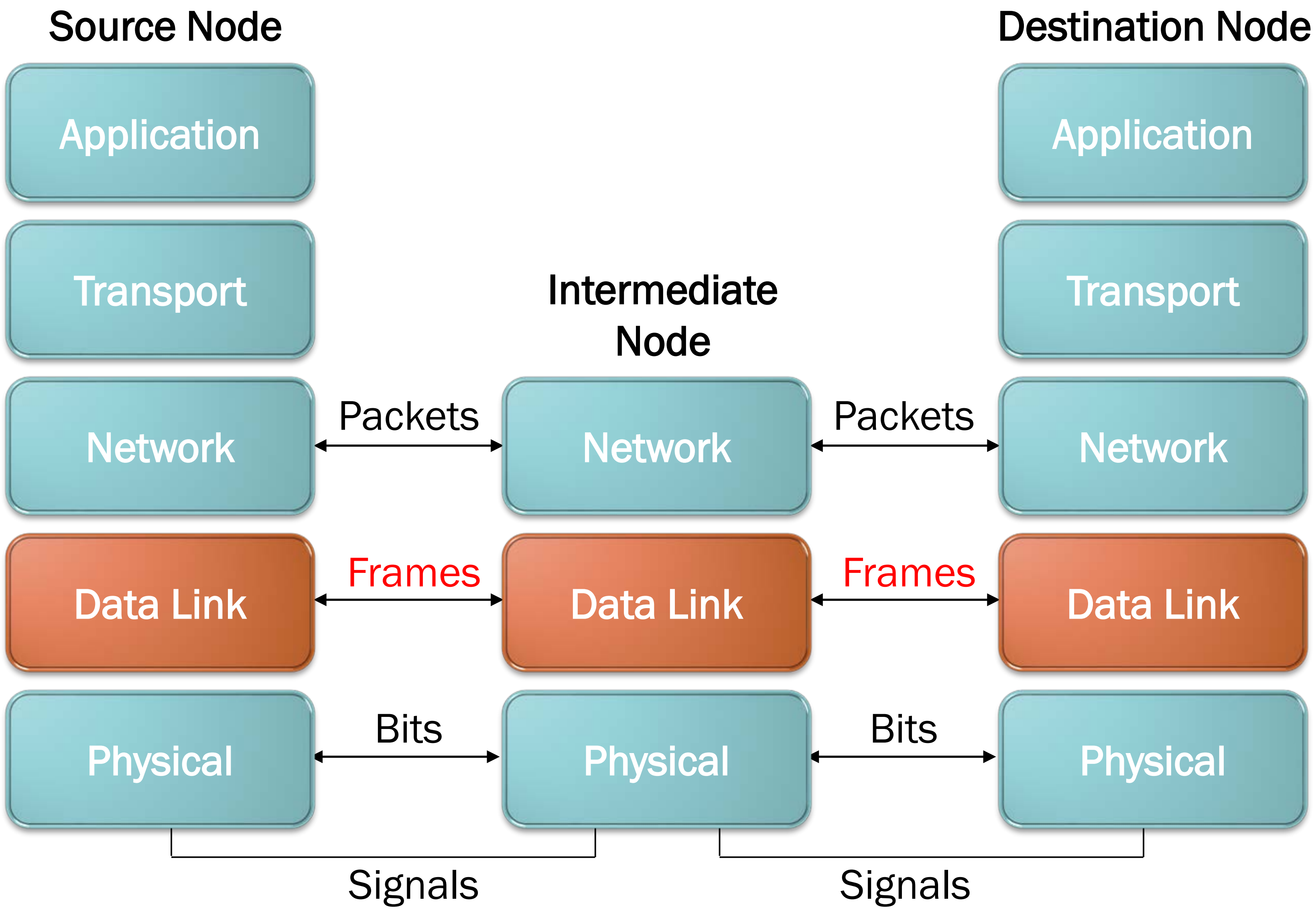
Learning Objectives

By the end of this topic, you should be able to:

- Describe the major services that the data link layer offers.
- Describe the framing and stuffing method in High-Level Data Link Control (HDLC) and Point-to-Point Protocol (PPP).
- Familiarise with the terminology used in Data Link Layer with the example of HDLC protocol.
- Describe the Stop-and-Wait and Sliding Window flow control techniques.
- Construct the comprehensive delay model by using various flow control techniques acquired based on different scenarios.

Data Link Layer Overview

Where are we?


Source Node
Destination Node
Intermediate Node
Application
Application
Transport
Transport
Network
Packets
Network
Packets
Network
Data Link
Frames
Data Link
Frames
Data Link
Physical
Bits
Physical
Bits
Physical
Signals
Signals

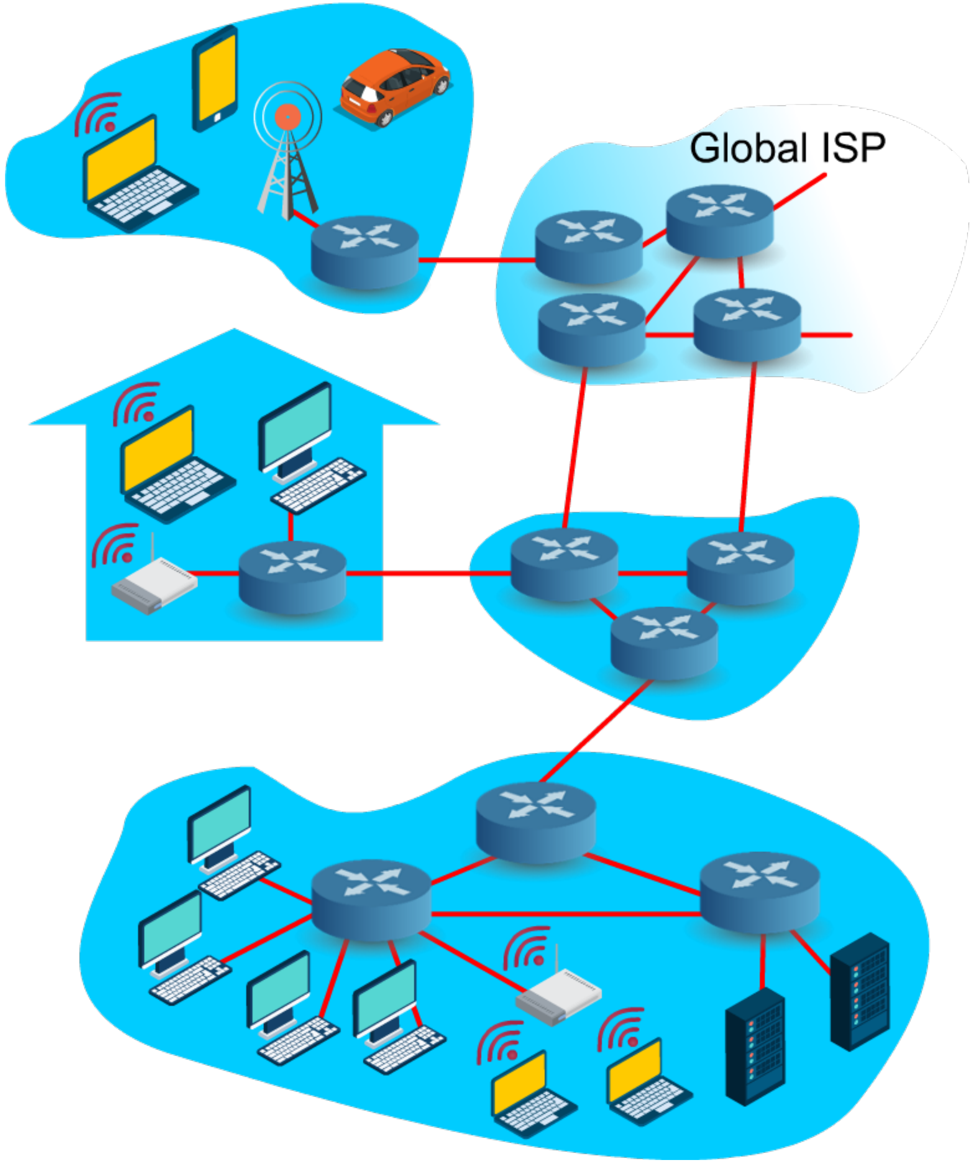
Terminology (in this topic):

- Communication channels that connect adjacent nodes along communication path are links
  - o Wired, Wireless, LANs, etc.

- Two types of link layer channels:
  - o Point-to-point (e.g. HDLC)
  - o Broadcast (e.g. LAN segment)

- Layer-2 PDU (Protocol Data Unit) is a frame, encapsulates datagram (packet)

Data-link layer has the responsibility of transferring datagram from one node to physically adjacent node over a link.



Data Link Layer Overview

- A layer of logic added above the Physical layer, which provides services to network layer.
- Implemented in “adaptor” (NIC).
  - e.g., Ethernet card, PCMCIA card.
- Basic service:
  - Moves a datagram from one node to an adjacent node over a single communication link.
- Datagram transferred by different link protocols over different links from end to end.
  - e.g., Ethernet on first link, frame relay on intermediate links, WiFi on last link.
- Different data link protocols provide different services.

Data Link Layer Services

- Framing
  - Encapsulate datagram into frame, adding overhead (header and/or trailer)

- Flow control
  - Pacing between adjacent sending and receiving nodes

- Error control
  - Receiver detects presence of errors
  - Receiver drops frame and/or signals sender for retransmission

- Link access (Prof. Cheng's part)
  - Medium Access Control (MAC) protocol specify the rule for channel access

Framing and Stuffing

Framing and Bit Stuffing in HDLC

HDLC frame

Flag	Address	Control	Information	FCS	Flag
01111110			Any number of bits		01111110

- Frame delineated by flag character (i.e. octet/byte)
- HDLC uses bit stuffing to prevent occurrence of flag pattern 01111110 **inside** the frame
- Transmitter inserts extra 0 after each consecutive five 1's inside the frame
- Receiver checks for five consecutive 1's
  - o if next bit = 0, the '0' is then removed
  - o if next two bits are 10, then flag is detected
  - o If next two bits are 11, error indication

Example: Bit Stuffing and De-stuffing

a) Data to be sent

0110111111111100

After stuffing and framing?

01111110 011011111011111000 *01111110*

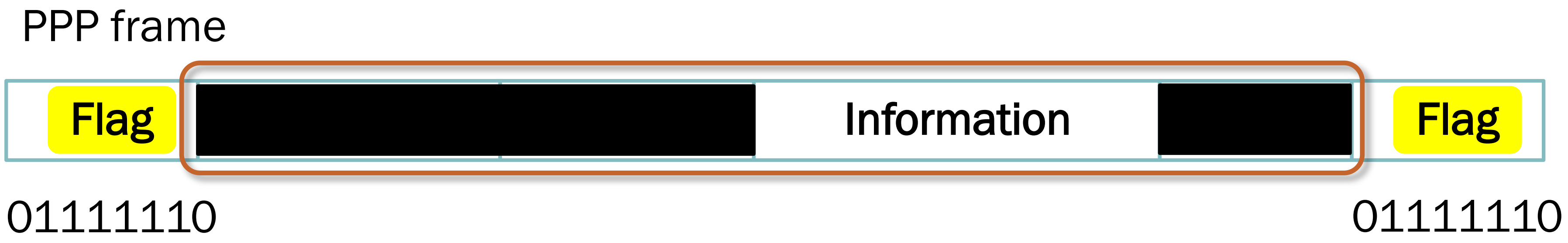
b) Data received

01111110000111011111011111011001111110

After destuffing and deframing?

000111011111-11111-110

Framing and Byte Stuffing in PPP

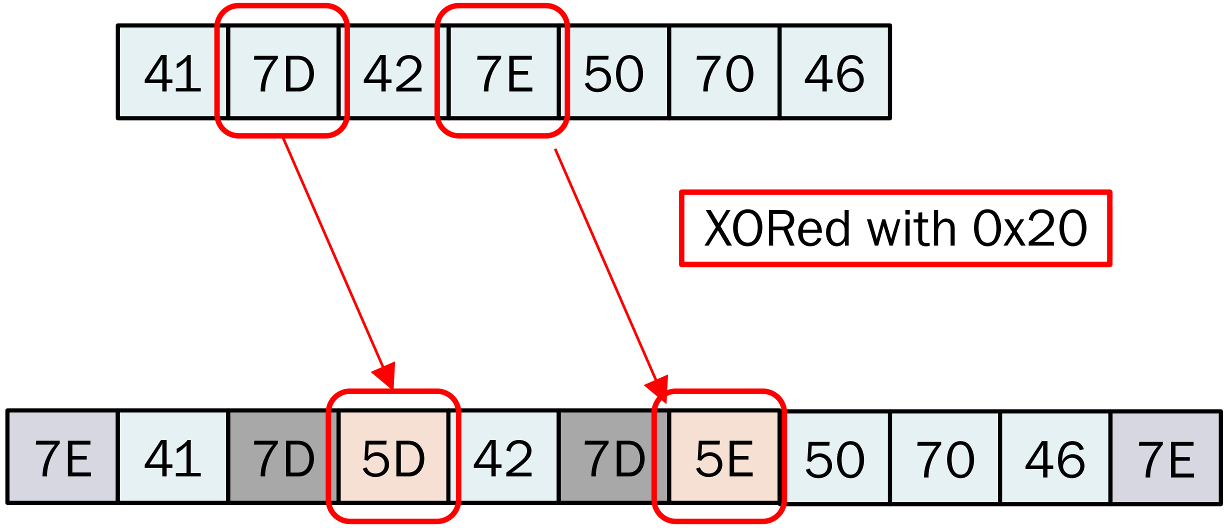


- Point-to-Point Protocol (PPP) is a character-oriented version of HDLC and uses similar frame structure as HDLC
- PPP uses the same flag, but uses byte stuffing
- Flag is 0x7E (01111110)
- Control escape 0x7D (01111101)
- Any occurrence of flag or control escape inside of frame is replaced with 0x7D followed by original octet XORed with 0x20 (00100000)

Example of Byte-Stuffing in PPP

(At Transmitter)

Data to be sent



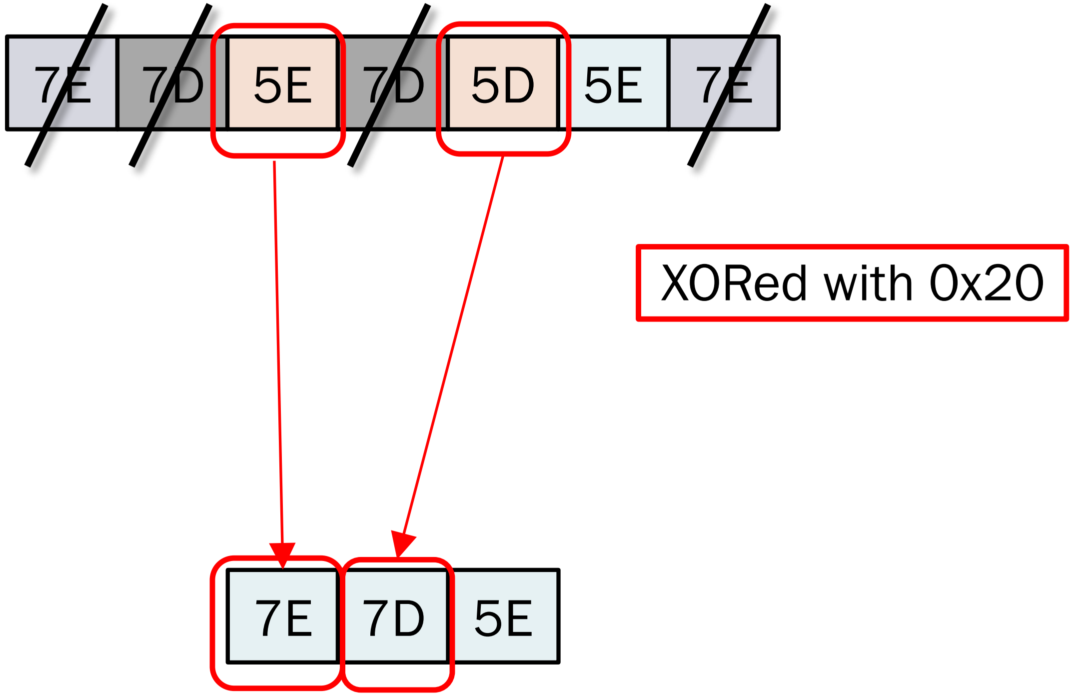
After stuffing and framing?

INPUT		OUTPUT
A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

Example of Byte-Stuffing in PPP

(At Receiver)

Bits receives:

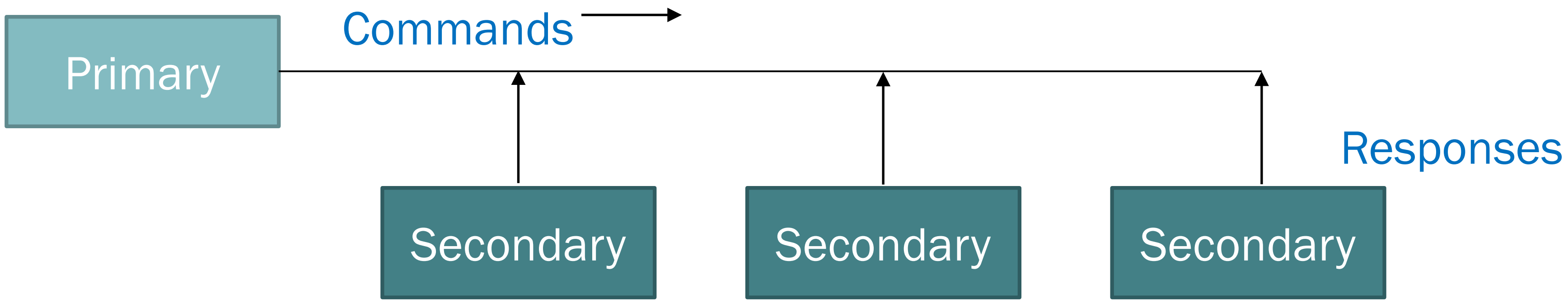


After byte destuffing and deframing

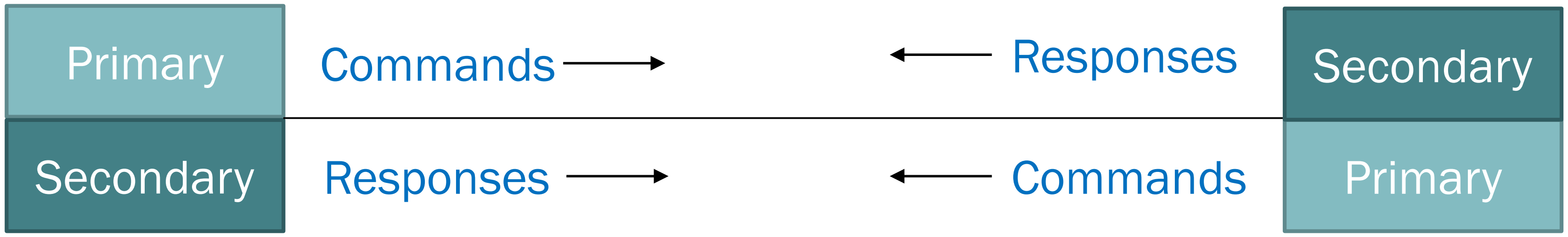
High-level Data Link Control (HDLC)

HDLC Data Transfer Modes

- Response Mode
  - Used in polling multidrop lines.

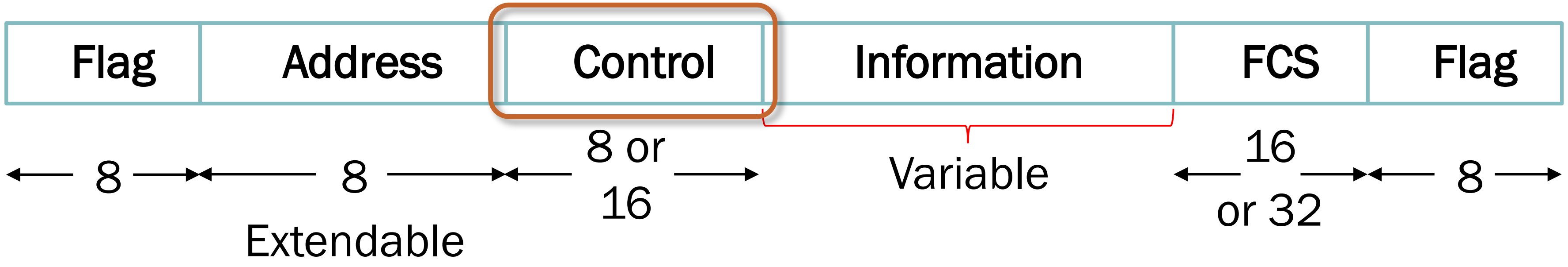


- Asynchronous Balanced Mode (ABM)
  - Used in full-duplex point-to-point links.



- Mode is selected during connection establishment

HDLC Frame Format



- Flag: delineate frame boundaries (01111110)
- Address: identify secondary station
- Control: purpose and functions of frame
- Information: contains user data
- Frame Check Sequence: 16- or 32-bit CRC

Control Field

- Control field defines three types of frames
- First one or two bits of control field identify frame type

Information Frame

1	2-4	5	6-8
0	N(S)	P/F	N(R)

Supervisory Frame

1	0	S	S	P/F	N(R)

Unnumbered Frame

~~1~~	~~1~~	~~M~~	~~M~~	~~P/F~~	~~M~~	~~M~~	~~M~~

Legend			
N(R):	Receive Sequence Number	M:	Unnumbered Function Bits
N(S):	Send Sequence Number	P/F:	Poll/final bit used in interaction between primary and secondary
S:	Supervisory Function Bits:		

Information Frames

1	2-4	5	6-8
0	N(S)	P/F	N(R)

- Each I-frame contains ‘send’ sequence number N(S)
- Positive ACK piggybacked
  - ‘receive’ sequence number = N(R) = Sequence number of *next* frame expected acknowledges all frames up to and including N(R)-1
- 3 or 7 bit sequence numbering
- Poll/Final Bit
  - NRM: Primary polls station by setting P = 1; Secondary sets F = 1 in *last* I-frame in response
  - Primaries and secondaries always interact via *paired* P/F bits

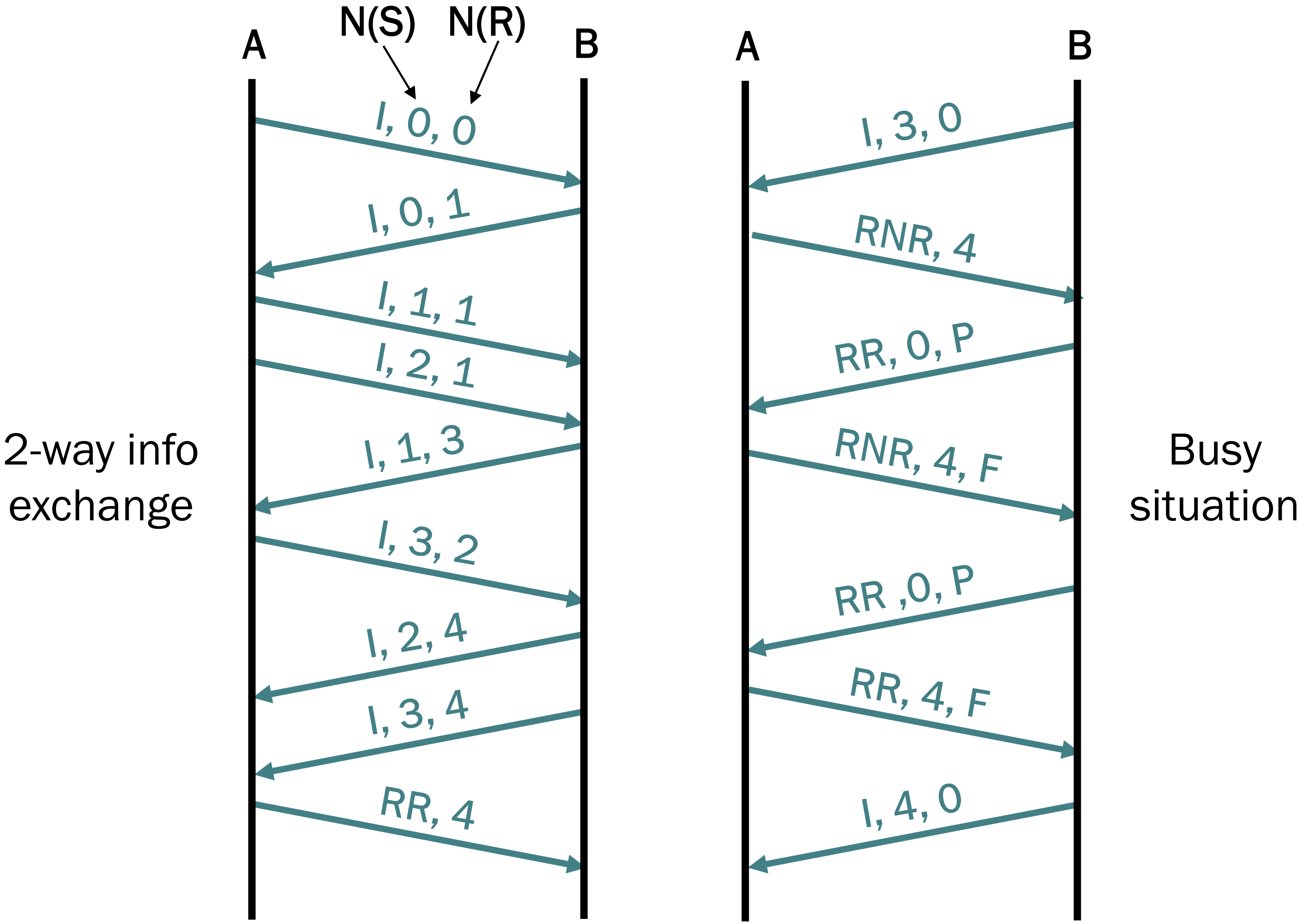
Supervisory Frames

1	0	S	S	P/F	N(R)

Used for error control and flow control:

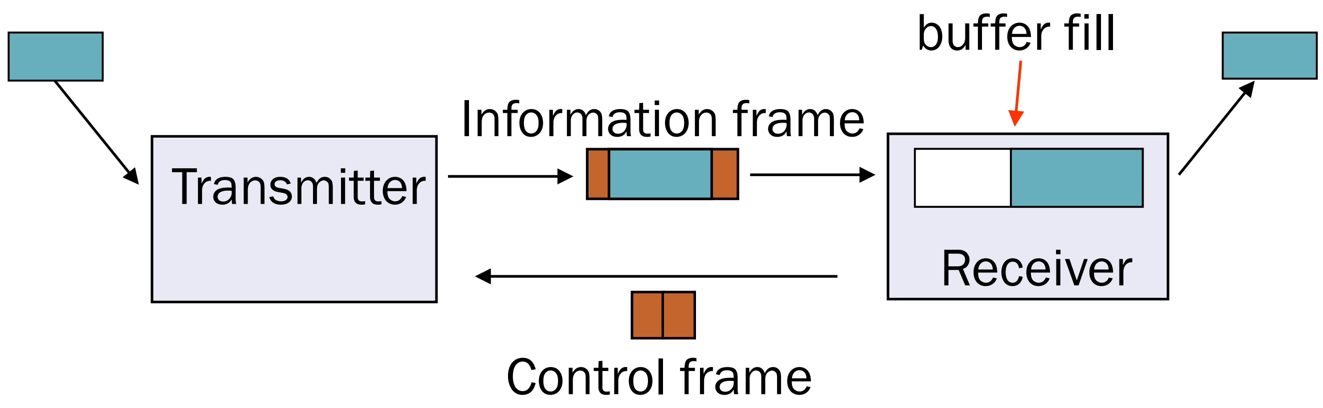
- *Receive Ready* (RR), SS = 00
  - ACKs frame up to N(R)-1 when piggybacked ACK not available
- *REJECT* (REJ), SS = 01
  - Negative ACK indicating N(R) is first frame not received correctly; Transmitter must resend N(R) and later frames
- *Receive Not Ready* (RNR), SS = 10
  - ACKs frame N(R)-1 and request that no more I-frames be sent
- *Selective REJECT* (SREJ), SS = 11
  - Negative ACK for N(R) requesting that N(R) be selectively retransmitted

Example: HDLC in ABM



Flow Control

Information Frames



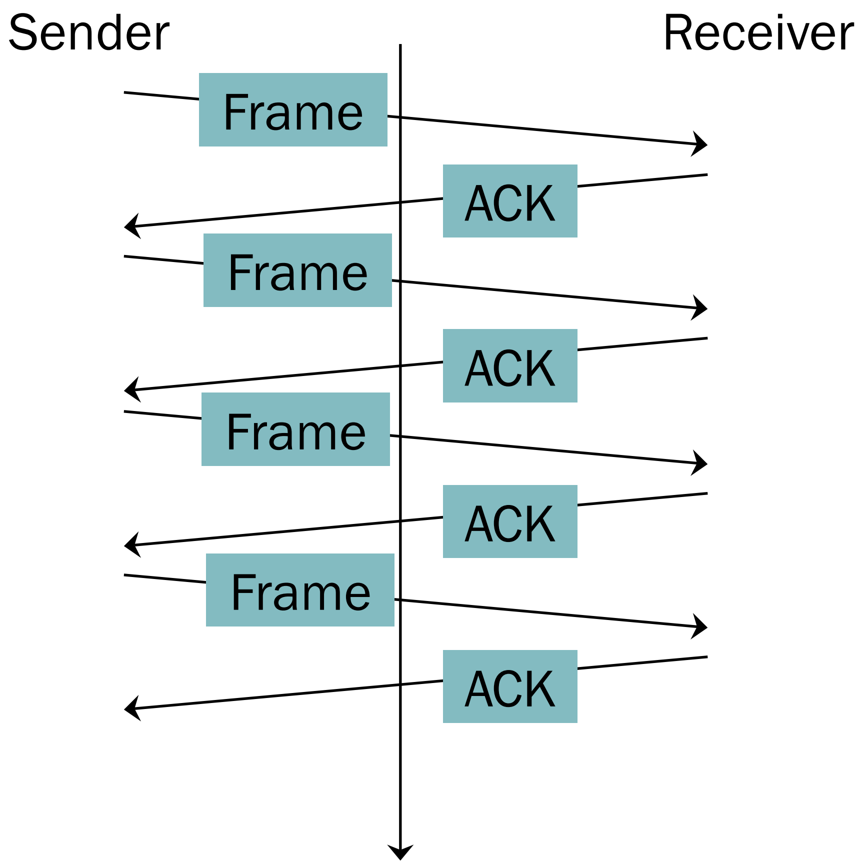
- Receiver has limited buffering to store arriving frames.
- If the sender sends data faster than the receiver can process, then buffer over-flow will occur.
- Flow control is a technique for preventing the sender from overwhelming the receiver with data.
- **Assumption:** all frames are successfully received without lost frames or erroneous frames (for the study of flow control part).

Stop and Wait (S&W) Scheme

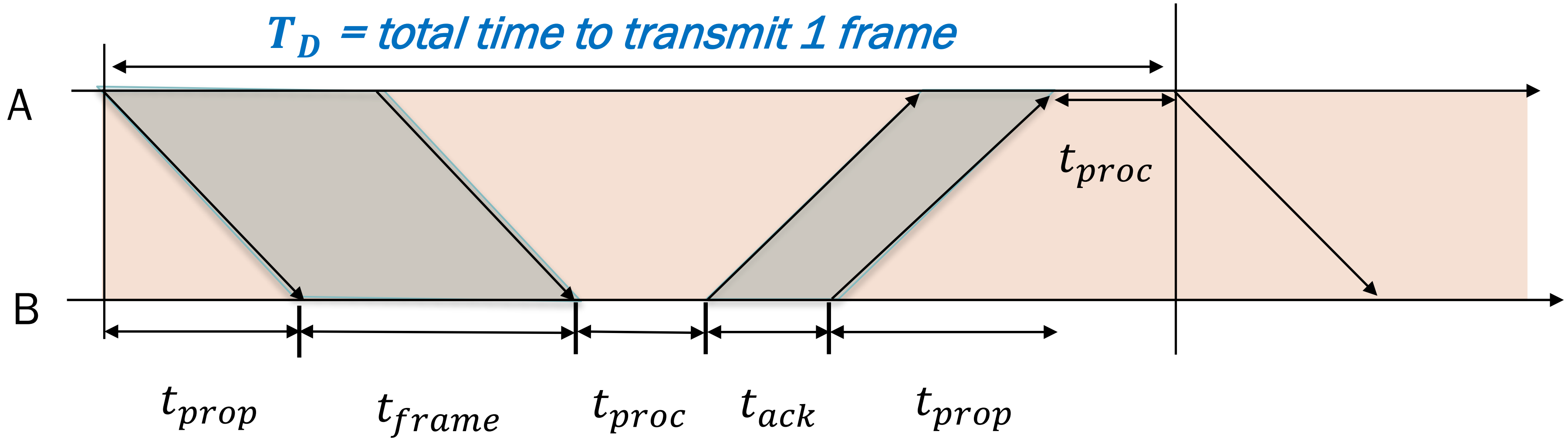
- Simplest flow control protocol

- By this protocol:
  - Source transmits a frame
  - When the destination receives the frame:
    - sends an ACK to sender
    - may withhold the ACK until it is ready to accept another frame
  - Source sends another frame only upon receipt of ACK

- Works well for a message sent in a few large frames
  - Inadequate if large block of data is split into small frames by source (why?)



Stop and Wait Model (comprehensive)



$$T_D = 2t_{prop} + 2t_{proc} + t_{frame} + t_{ack}$$

$$= 2t_{prop} + 2t_{proc} + \frac{L_{frame}}{R} + \frac{L_a}{R}$$

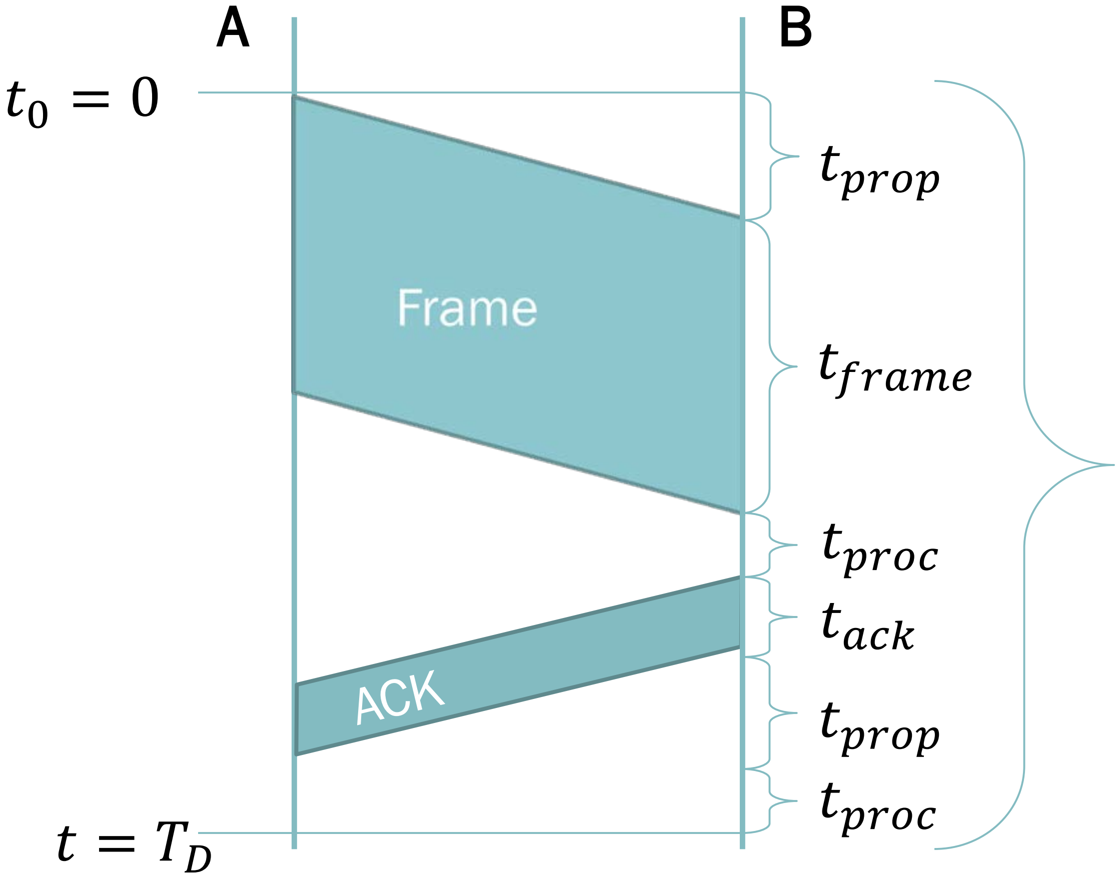
info frame length

ACK frame length

channel transmission rate

Performance of S&W

- Let t_{prop} be the propagation delay, t_{frame} be the frame transmission time, t_{ack} be the acknowledgement transmission time, t_{proc} be the nodal processing delay



$$T_D = 2t_{prop} + 2t_{proc} + t_{frame} + t_{ack}$$

$$= 2t_{prop} + 2t_{proc} + \frac{L_{frame}}{R} + \frac{L_a}{R}$$

Performance of S&W

- Of this time T_D only t_{frame} is actually used to transmit data. Therefore, the efficiency or utilisation U is:

U : fraction of time sender busy sending information data

$$U = \frac{t_{frame}}{T_D}$$

Effective data rate (throughput in bps)

$$R_{eff} = \frac{L_{frame}}{T_D}$$

If Header needs to be excluded, L_{frame} needs to be replaced with payload length.

$$U = \frac{R_{eff}}{R\,(\text{channel rate})}$$

Performance of S&W (Simplified Model)

- Of this time T_D only t_{frame} is actually used to transmit data.
  Therefore, the efficiency or utilisation U is:
  $$U = \frac{t_{frame}}{T_D}$$
- Ignoring t_{ack}, t_{proc} the total delay T_D is $2t_{prop} + t_{frame}$
- If we define $a = t_{prop}/t_{frame}$, then
  $$U = \frac{1}{1 + 2a}$$
- If d is the link length, V the velocity of propagation, R the data rate, and L the frame length, then
  $$a = \frac{d/V}{L/R} = \frac{Rd}{LV}$$

Effective data rate
$$R_{eff} = \frac{L_{frame}}{T_D}$$

$$U = \frac{R_{eff}}{R(\text{channel rate})}$$

Assumption: ACK transmission time, node processing time, queuing time is negligible. Header bits in data frame is ignored.

Example 1

A link has a data rate of 4 Mbps, and a distance of 1000 km. The propagation delay is 5 μs/km. How long does it take a frame of 1000 bytes <u>to get to the other end of the link</u>?

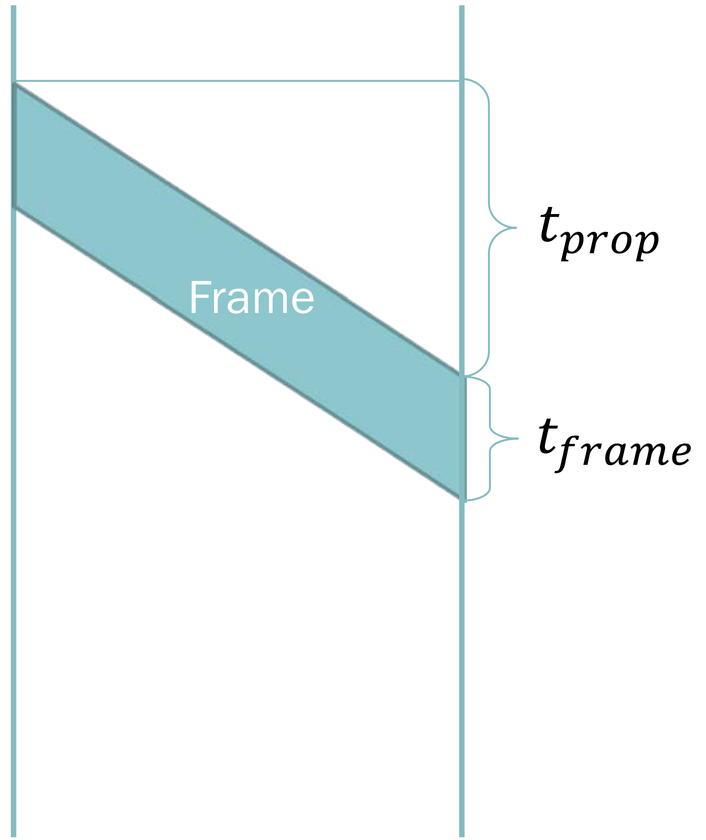
Transmission time = frame length/data rate

= $(1000 \times 8)/(4 \times 10^6)$ = 2 ms

Propagation time = 5 μs/km × 1000km = 5 ms

Total delay = Transmission time + Propagation time

= 7 ms



Example 2

Compare the value $a = t_{prop}/t_{frame}$ of Example 1 with another of lower data rate at 1 Mbps.

From Example 1:

R = 4 Mbps, d = 1000 Km, V = 2 x 10^8 m/s,
L = 1000 bytes = 8000 bits

$$a = \frac{R}{L} \times \frac{d}{V}$$

For R = 4 Mbps, a = 2.5

For R = 1 Mbps, a = 0.625

⇒For a given frame size, reducing the transmission rate can increase the transmission time. This reduces '$\boldsymbol{a}$'.

⇒ If S&W is used for flow control, this will lead to increased link utilisation.

Example 3

In Example 1, if S&W flow control is used, what is the link utilisation?

Recall that: t_{frame} = 2 ms, t_{prop} = 5 ms

Link utilisation: $U = t_{frame}/(2t_{prop} + t_{frame})$ = 2/(10 + 2) = 0.17

or alternatively, a = 2.5 $U = 1/(1 + 2a)$ = 0.17

What if the frame size is increased from 1000 bytes to 5000 bytes?

t_{frame} = (5000 x 8)/(4 x 10^8) = 10 ms

$U = t_{frame}/(2t_{prop} + t_{frame})$ = 10/(10 + 10) = 0.5

or alternatively, a = 0.5 $U = 1/(1 + 2a)$ = 0.5

⇒ Increasing frame size will increase the link utilisation.

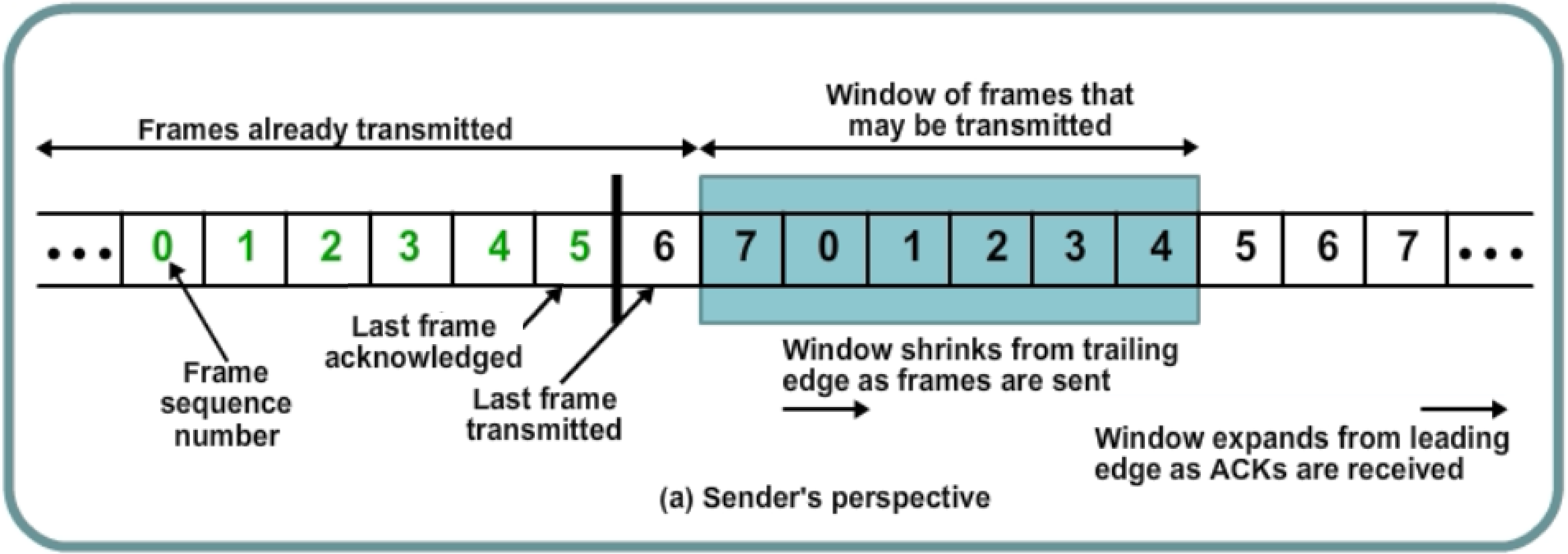
S&W 1st page

Sliding Window Scheme

- Allow multiple frames to be in transit
- Transmitter can send up to W frames without receiving ACK
- Each (data) frame carries a sequence number
- Sequence number bounded by size of sequence number field (k bits)
    - o Frames are numbered modulo 2^k
        - E.g. 3-bit sequence number implies frames and ACKs are numbered as 0,1,2,3,4,5,6,7
- ACK includes the sequence number of the next frame expected
- Cumulative Acknowledgment frame RR i indicates that up to i-1^{th} frame received and receiver is expecting the i^{th} frame.

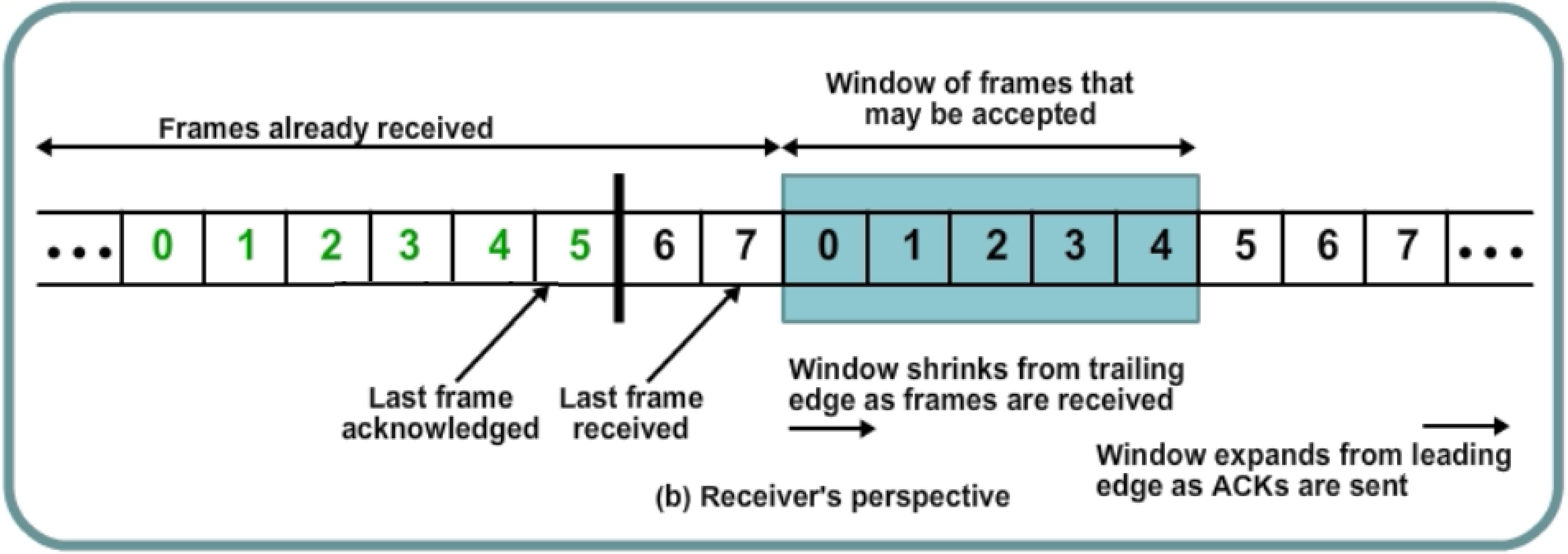
Sliding Window Diagram

Example: W = 7, up to 7 consecutive frames can be sent without getting ACK



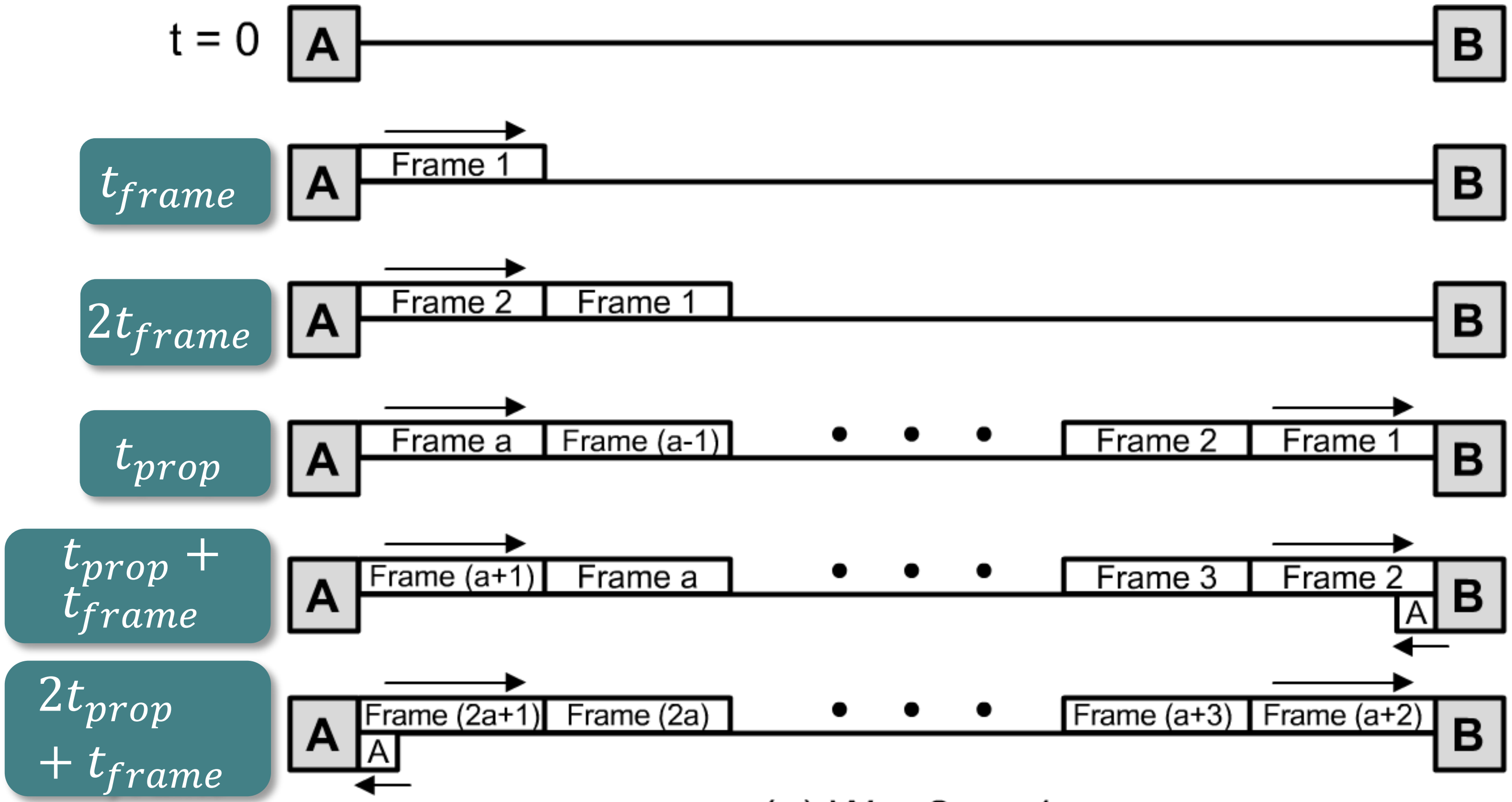
Sliding Window Diagram

Example: W = 7, up to 7 consecutive frames can be sent without getting ACK



Sliding Window (Simplified Model)

Case 1: $Wt_{frame} \geq 2t_{prop} + t_{frame}$



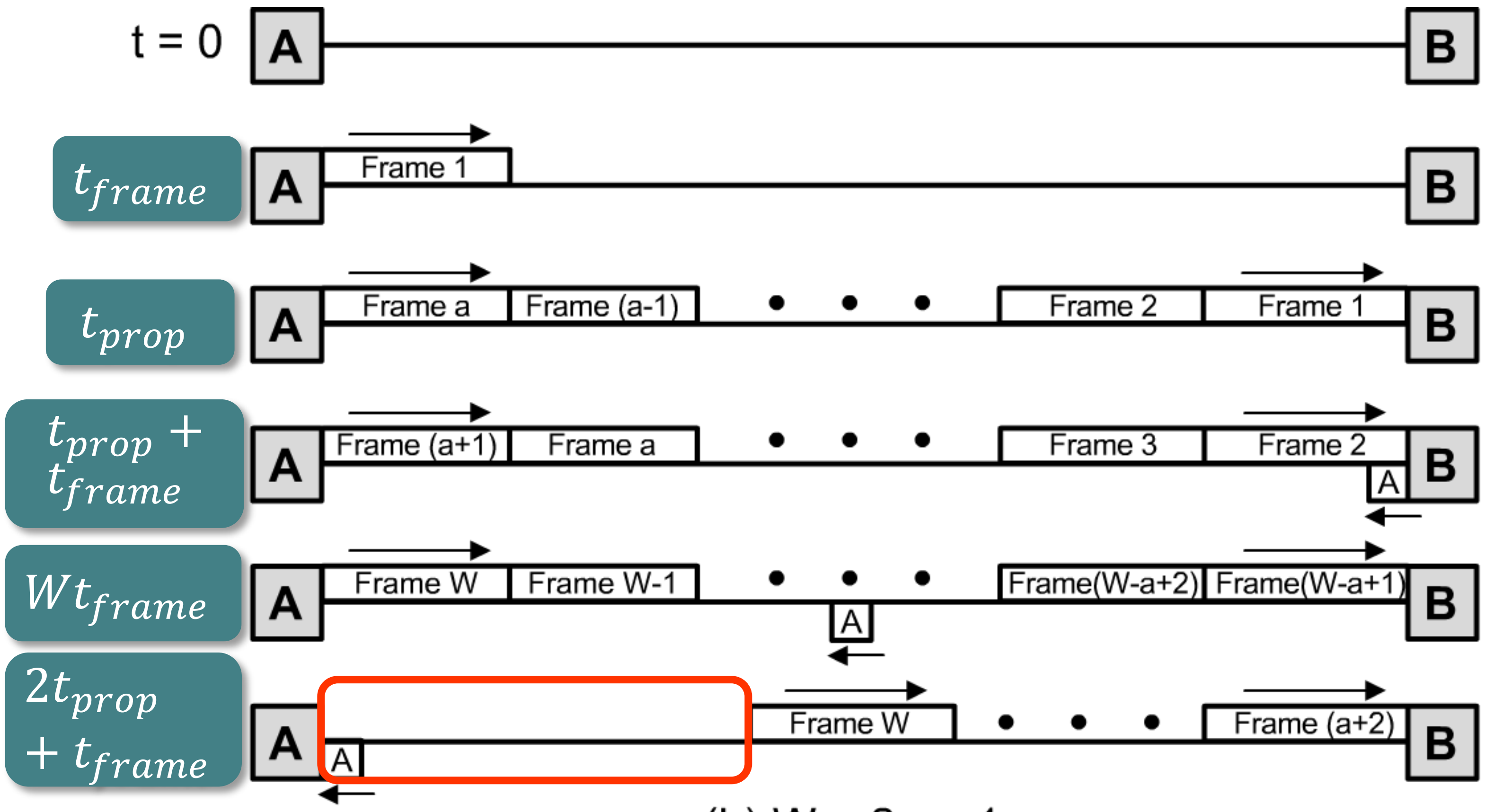
(a) W ≥ 2a + 1

Assumption:
Only consider information frame transmission time, and propagation time.
Ignore the headers in the information frame.

Assume: ACK is sent out for every frame received by the receiver.

Sliding Window (Simplified Model)

Case 2: $Wt_{frame} < 2t_{prop} + t_{frame}$



(b) W < 2a + 1

Assumption:
Only consider information frame transmission time, and propagation time. Ignore the headers in the information frame.

Assume: ACK is sent out for every frame received by the receiver.

Sliding Window Link Utilisation (Simplified Model)

- Assume only t_{frame} and t_{prop} are considered, and overhead in data frame is ignored
- link utilisation depends on the window size *W* and the value of *a*.

$$a = \frac{t_{prop}}{t_{frame}} = \frac{d/V}{L/R} = \frac{Rd}{LV}$$

d: distance of link
V: signal speed
L: frame Length in bits
R: data rate

$$U = \begin{cases} 1 & W \geq 2a+1 \\ \dfrac{W}{2a+1} & W < 2a+1 \end{cases}$$

frame 1 ACK reaches source before window exhausted

window exhausted at *t*=*W* out of *2a + 1* period

- Sliding Window link efficiency is at most W times that of Stop-and-Wait $(= 1/(2a+1))$

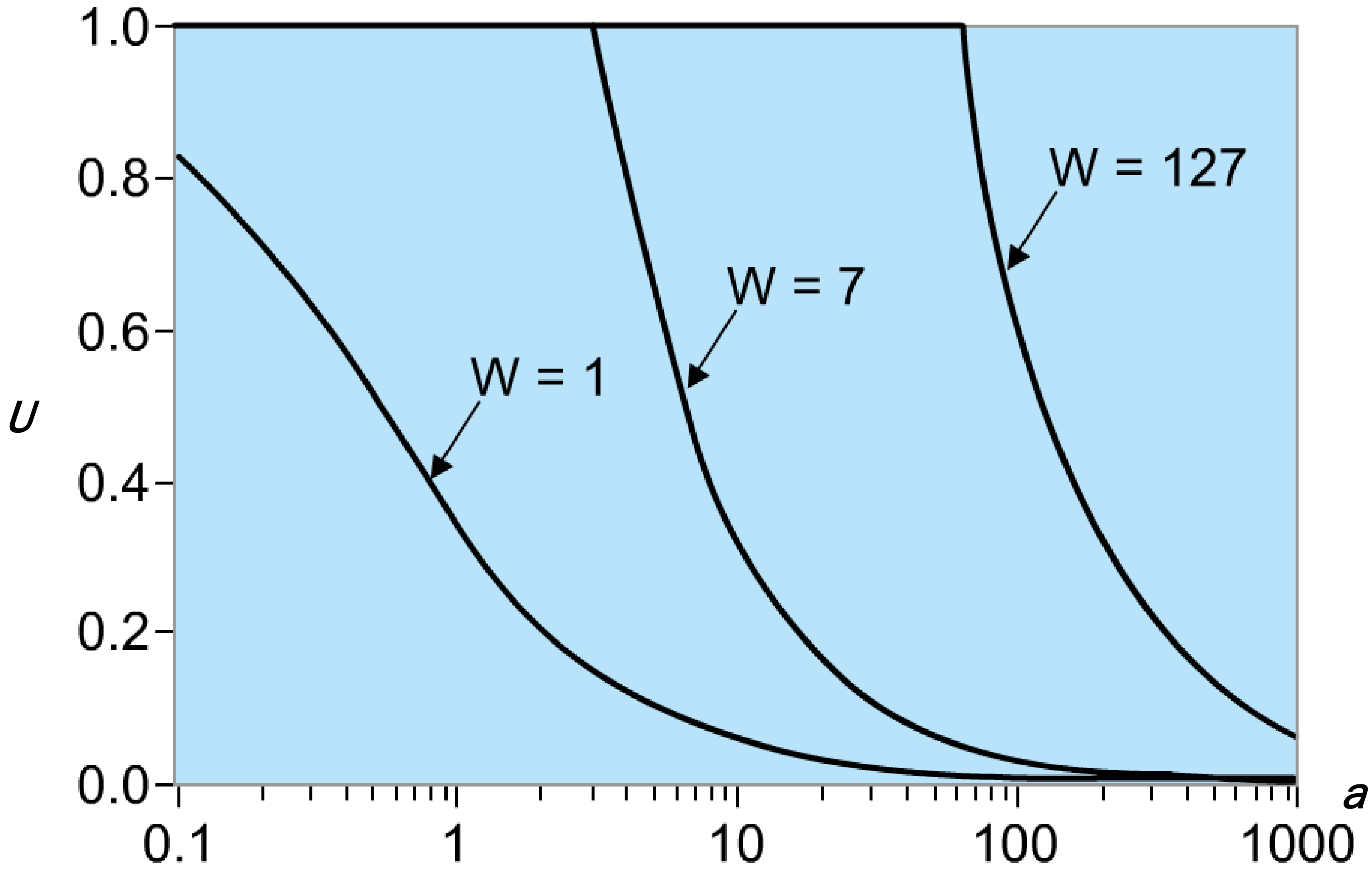
Sliding Window Link Utilisation (In General)

- Link utilisation depends on the window size *W* and total delay upon receipts of the ACK of the first frame by the source (T_1).

- Ignore the overhead bits in the information frame, still assume ACK is sent for every frame received:
  - If $\boldsymbol{W} \times \boldsymbol{t_{frame}}$ **>=** $\mathbf{T_1}$ $\Rightarrow$ frame 1 ACK reaches source before window exhausted
    - U = 1
  - If $\boldsymbol{W} \times \boldsymbol{t_{frame}}$ **<** $\mathbf{T_1}$ $\Rightarrow$ window exhausted at $t = Wt_{frame}$ out of T_1 period
    - U = $W \times t_{frame} / T_1$

- Sliding Window link efficiency is at most *W* times that of Stop-and-Wait.

Performance



Sliding Window Utilisation as a function of *a*

Example 4

Consider an error-free 64 kbps satellite channel used to send 512-byte data frames in one direction, with a very short ACK in the other direction. What is the maximum utilisation for window sizes of 1, 7, 15, and 128? Round trip propagation delay is 540 ms.

- t_{frame} = (512 x 8)/64000 = 64 ms
- $2t_{prop}$ = 540 ms
- $T_1 = t_{frame} + 2t_{prop}$ = 604 ms
- 604ms/64ms = 9.43 => Need a window size of at least 10 frames to keep the channel busy.
- W = 1, U = t_{frame}/T_1 = 10.6%
- W = 7, U = 7 x t_{frame}/T_1 = 7 x10.6% = 74.2%
- W = 15 or 128, U = 100%

Example 4

Consider an error-free 64 kbps satellite channel used to send 512-byte data frames in one direction, with a very short ACK in the other direction. What is the maximum utilisation for window sizes of 1, 7, 15, and 128? Round trip propagation delay is 540 ms.

- A 512 byte frame occupies the channel for (512 x 8)/64000 = 64 ms
- Round trip propagation delay is 540 ms, 2a = 540/64 = 8.43
- Need a window size of at least 2a + 1 = 10 frames to keep the channel busy.
- W = 1, U = 1/(2a + 1) = 10.6%
- W = 7, U = 7 x 10.6% = 74.2%
- W = 15 or 128, U = 100%

Summary

Summary

Key points discussed in this topic:

- The major services that the data link layer offers include framing, flow control, error control and link access.

- HDLC uses bit stuffing to prevent occurrence of flag pattern 01111110 inside the frame.

- PPP is a character-oriented version of HDLC and uses similar frame structure as HDLC. It uses the same flag, but uses byte stuffing.

- Flow control is a technique for preventing the sender from overwhelming the receiver with data. The two flow control techniques discussed are:
  - o Stop-and-Wait
  - o Sliding Window