Data Link Control

Raj Jain
Professor of CIS
The Ohio State University
Columbus, OH 43210
Jain@ACM.Org

http://www.cis.ohio-state.edu/~jain/cis677-98/

- Flow Control
- Effect of propagation delay, speed, frame size
- Error Detection
- Error Control
- HDLC

Flow Control

- Flow Control = Sender does not flood the receiver, but maximizes throughput
- Sender throttled until receiver grants permission

Flow Control

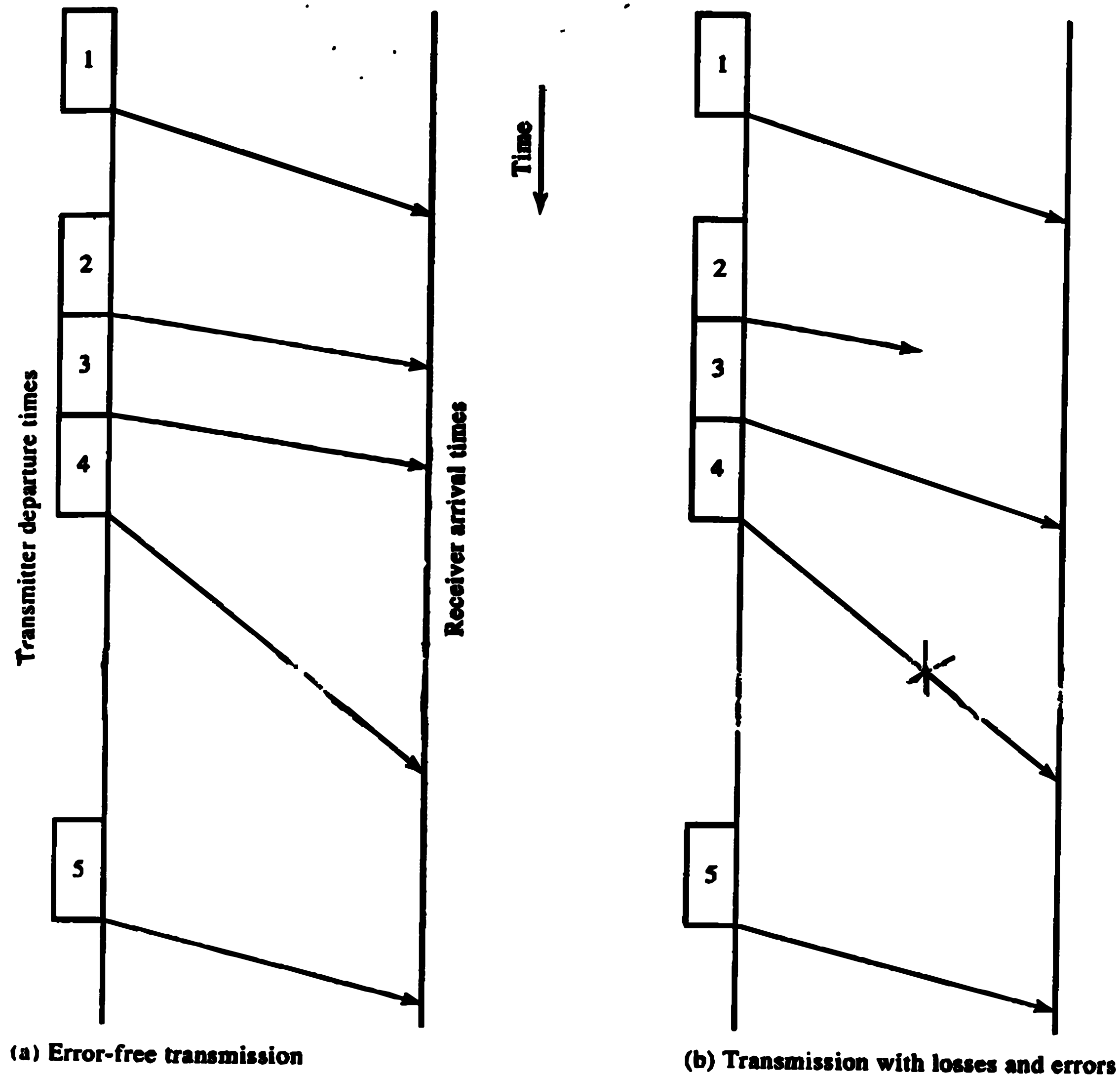
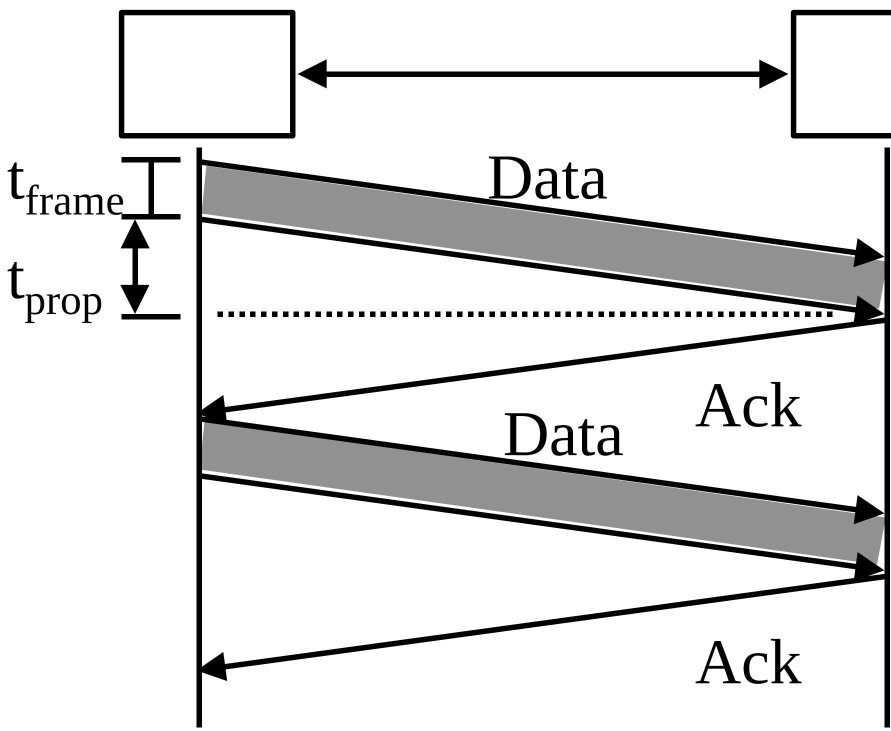


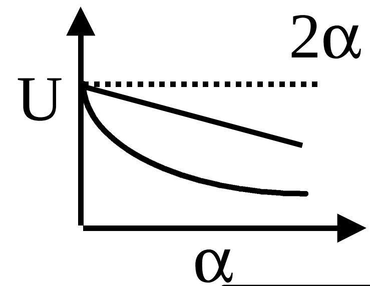
Fig 6.1 Stallings

Stop and Wait Flow Control



$$U = \frac{t_{frame}}{2t_{prop} + t_{frame}}$$

$$= \frac{1}{2\alpha + 1}$$



$$\alpha = \frac{t_{prop}}{t_{frame}} = \frac{\text{Distance/Speed of Signal}}{\text{Frame size /Bit rate}}$$

$$= \frac{\text{Distance} \times \text{Bit rate}}{\text{Frame size} \times \text{Speed of Signal}}$$

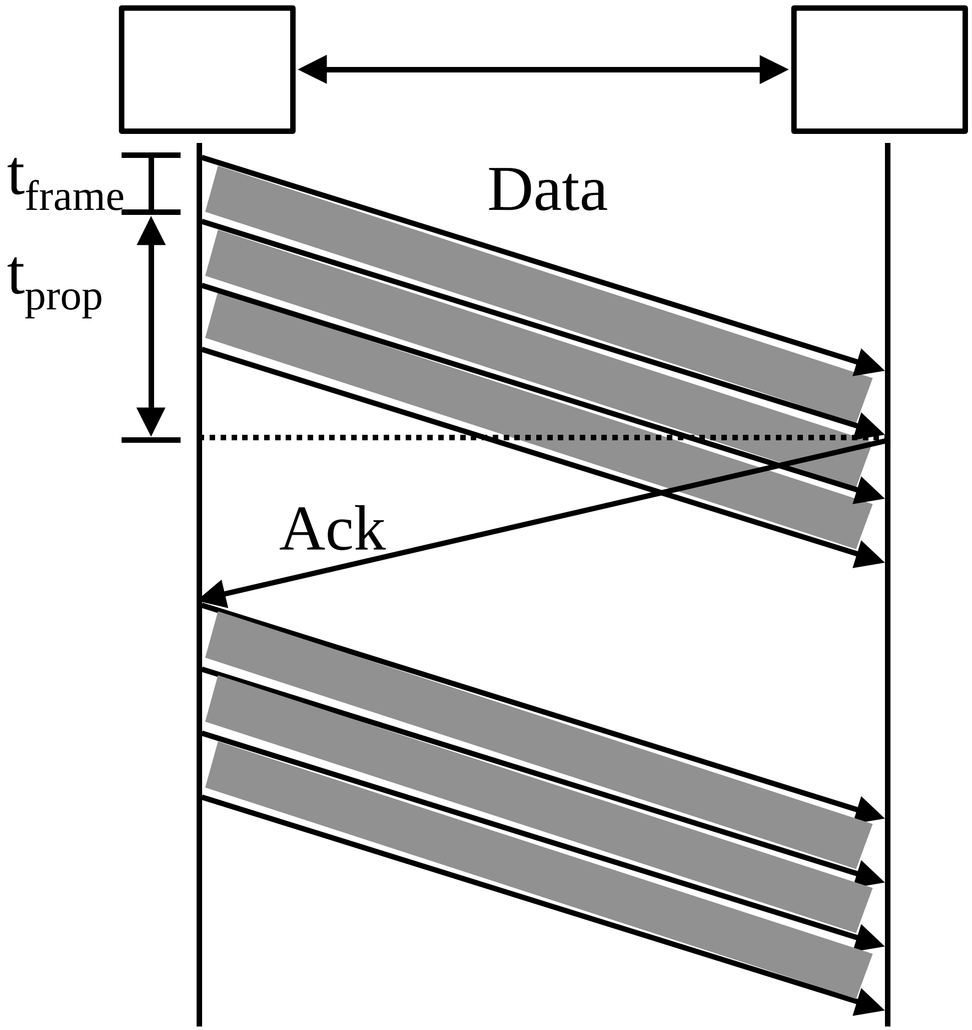
Light in vacuum
= 300 m/μs
Light in fiber
= 200 m/μs
Electricity
= 250 m/μs

Utilization: Examples

- Satellite Link: Propagation Delay t_{prop} = 270 ms
  Frame Size = 4000 bits = 500 bytes
  Data rate = 56 kbps $\Rightarrow$ t_{frame} = 4/56 = 71 ms
  $\alpha = t_{prop}/t_{frame} = 270/71 = 3.8$
  $U = 1/(2\alpha+1) = 0.12$

- Short Link: 1 km = 5 μs,
  Rate=10 Mbps,
  Frame=500 bytes $\Rightarrow t_{frame}$= 4k/10M= 400 μs
  $\alpha=t_{prop}/t_{frame}=5/400=0.012 \Rightarrow U=1/(2\alpha+1)=0.98$

Sliding Window Protocol



$$U = \frac{N t_{frame}}{2t_{prop} + t_{frame}} = \begin{cases} \frac{N}{2\alpha+1} \\ 1 \text{ if } N > 2\alpha+1 \end{cases}$$

Sliding Window Protocols

- Window = Set of sequence numbers to send/receive
- Sender window
  - Sender window increases when ack received
  - Packets in sender window must be buffered at source
  - Sender window may grow in some protocols

Sliding Window

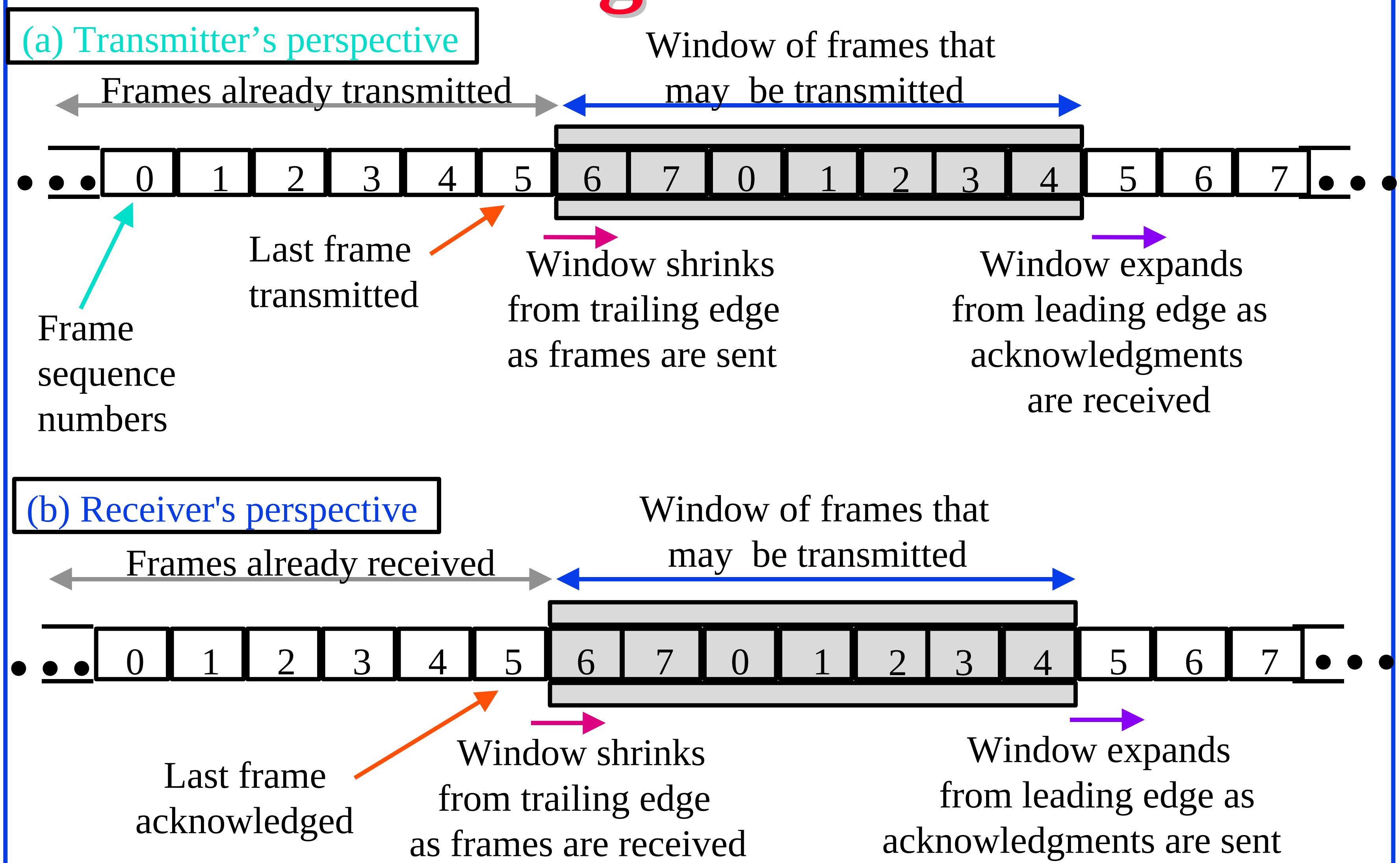


Fig 6.3 Stallings

Sliding Window: Example

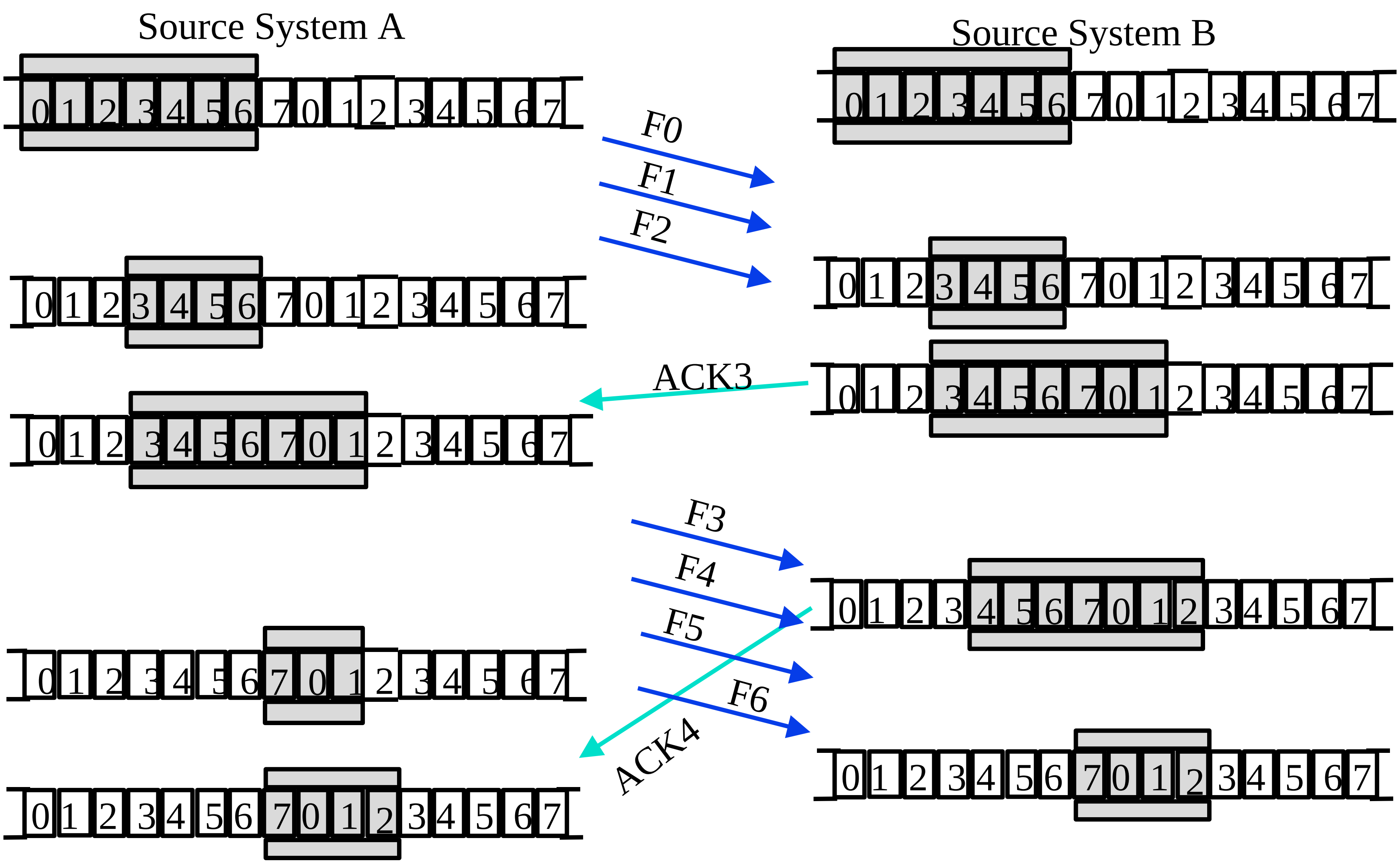


Fig 6.4 Stallings

Effect of Window Size

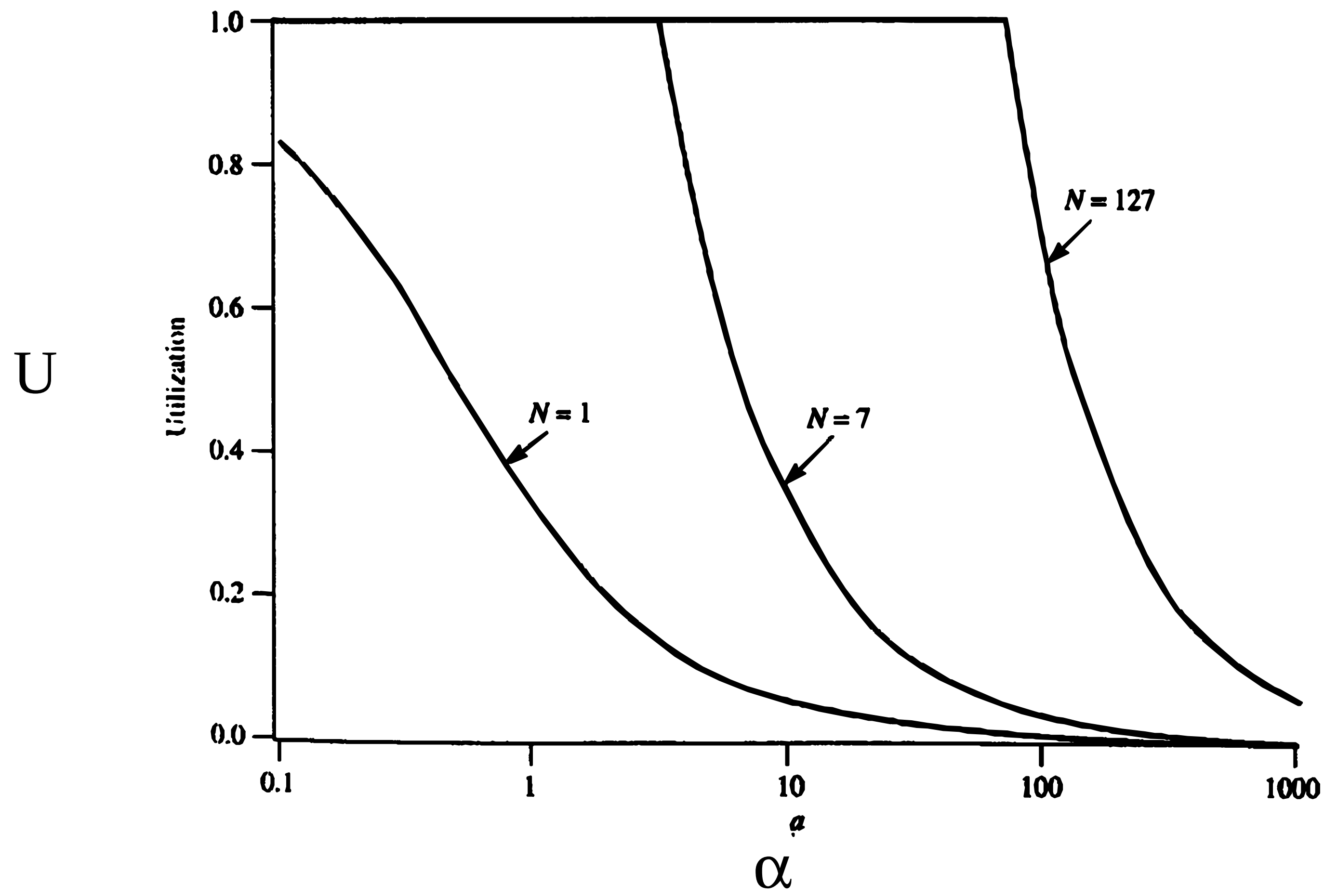
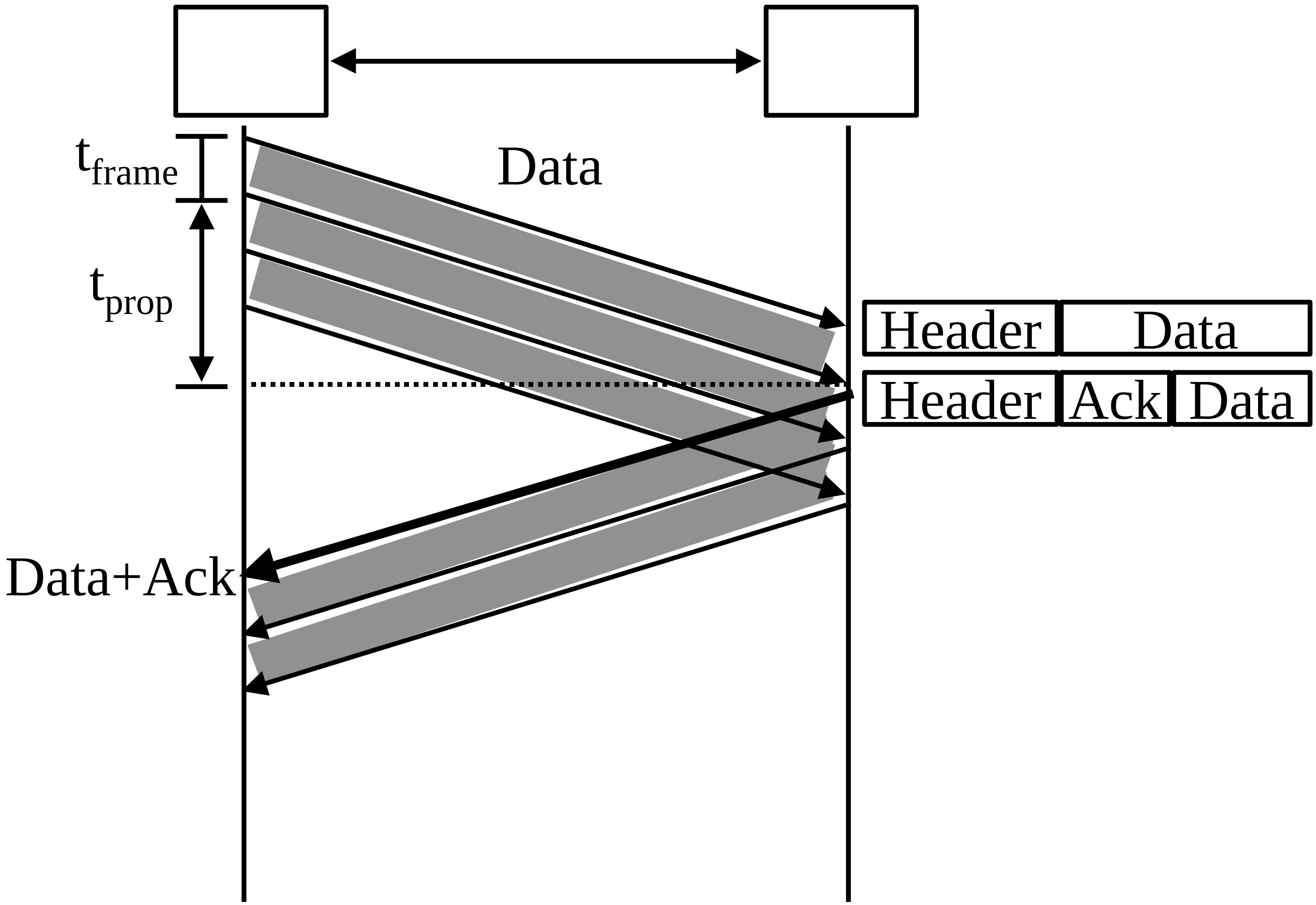


Fig 6.16 Stallings

Piggybacking



Error Detection

- Let Pb = Probability of bit error
  F = Frame size in bits
- P(No errors) = $(1-P_b)^F$
- P(one or more bits in error) = $1-(1-P_b)^F$
- Example: $P_b = 10^{-6}$, F=1000
  P(Frame error) = $1-(1-10^{-6})^{1000} = 10^{-3}$

Parity Checks

1 0 1 1 1 0 1 0 [ ]
1 2 3 4 5 6 7 8 9

- Odd Parity

1 0 1 1 1 0 1 0 0 | 0 0 1 1 1 0 1 0 0
1 2 3 4 5 6 7 8 9 | 1 2 3 4 5 6 7 8 9

1-bit error

0 0 0 1 0 0 1 0 0 | 0 0 0 1 1 0 1 0 0
1 2 3 4 5 6 7 8 9 | 1 2 3 4 5 6 7 8 9

3-bit error | 2-bit error

- Even Parity

1 0 1 1 1 0 1 1 0
1 2 3 4 5 6 7 8 9

Check Digit Method

- Make number divisible by 9

Example: 823 is to be sent

1. Left-shift: 8230
2. Divide by 9, find remainder: 4
3. Subtract remainder from 9: 9-4=5
4. Add the result of step 5 to step 1: 8235
5. Check that the result is divisible by 9.

Detects all single-digit errors: 7235, 8335, 8255, 8237

Detects several multiple-digit errors: 8765, 7346

Does not detect some errors: 7335, 8775, ...

Modulo 2 Arithmetic

```
  1111      11001          110
+1010     ×    11    11  | 1010
--------  --------        /  11
  0101     11001            ---------    010   2
          11001               x11      011   3
          ---------            11      ----  --
          101011              ---------  001   1  Mod 2
                                  x00    101   5  Binary
                                   00
                                -----
                                   x0
```

Cyclic Redundancy Check (CRC)

- **Binary Check Digit Method**
- Make number divisible by P=110101 (n+1=6 bits)

Example: M=1010001101 is to be sent

1. Left-shift M by n bits 2^nM= 101000110100000
2. Divide 2^nM by P, find remainder: R=01110
3. ~~Subtract remainder from P~~ ← Not required in Mod 2
4. Add the result of step 2 to step 1 :
   T=101000110101110
5. Check that the result T is divisible by P.

Modulo 2 Division

```
       Q=1101010110
P=110101)101000110100000=2^nM
         110101
          111011
          110101
           011101
           000000
            111010
            110101
             011111
             000000
              111110
              110101
```

```
010110
000000
 101100
 110101
  110010
  110101
   001110
   000000
    01110 = R
```

Checking At The Receiver

```
       1101010110
110101)101000110101110
       110101
        111011
        110101
         011101
         000000
          111010
          110101
           011111
           000000
            111110
            110101
```

```
010111
000000
 101111
 110101
  110101
  110101
   00000
```

Polynomial Representation

- Number the bits 0, 1, ..., from <u>right</u>

  $b_n b_{n-1} b_{n-2} \ldots b_3 b_2 b_1 b_0$

  $b_n x^n + b_{n-1} x^{n-1} + b_{n-2} x^{n-2} + \ldots + b_3 x^3 + b_2 x^2 + b_1 x + b_0$

- Example:

  543210

  ↓↓↓↓↓

  $110101 = x^5 + x^4 + x^2 + 1$

  $1101\ 1001\ 0011 = x^{11} + x^{10} + x^8 + x^7 + x^4 + x + 1$

  11 10 9 8 1 0

Cyclic Redundancy Check (CRC)

Polynomial Division Method

Make $T(x)$ divisible by $P(x) = x^5+x^4+x^2+1$ (Note: n=5)

Example: M=1010001101 is to be sent
$M(x) = x^9+x^7+x^3+x^2+1$

1. Multiply M(x) by x^n, $x^nM(x) = x^{14}+x^{12}+x^8+x^7+x^5+$ ....
2. Divide $x^nM(x)$ by $P(x)$, find remainder: $R(x)$=01110=x^3+x^2+x

CRC (Cont)

3. Add the remainder $R(x)$ to $x^nM(x)$:
   $T(x) = x^{14}+x^{12}+x^8+x^7+x^5+x^3+x^2+x$
4. Check that the result $T(x)$ is divisible by $P(x)$.

Transmit the bit pattern corresponding to $T(x)$:
   101000110101110

Popular CRC Polynomials

- CRC-12: $x^{12}+x^{11}+x^{3}+x^{2}+x+1$
- CRC-16: $x^{16}+x^{15}+x^{2}+1$
- CRC-CCITT: $x^{16}+x^{12}+x^{5}+1$
- CRC-32: Ethernet, FDDI, ...

  $$x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^{8}+x^{7}+x^{5}+x^{4}+x^{2}+x+1$$

Even number of terms in the polynomial

⇒ Polynomial is divisible by 1+x

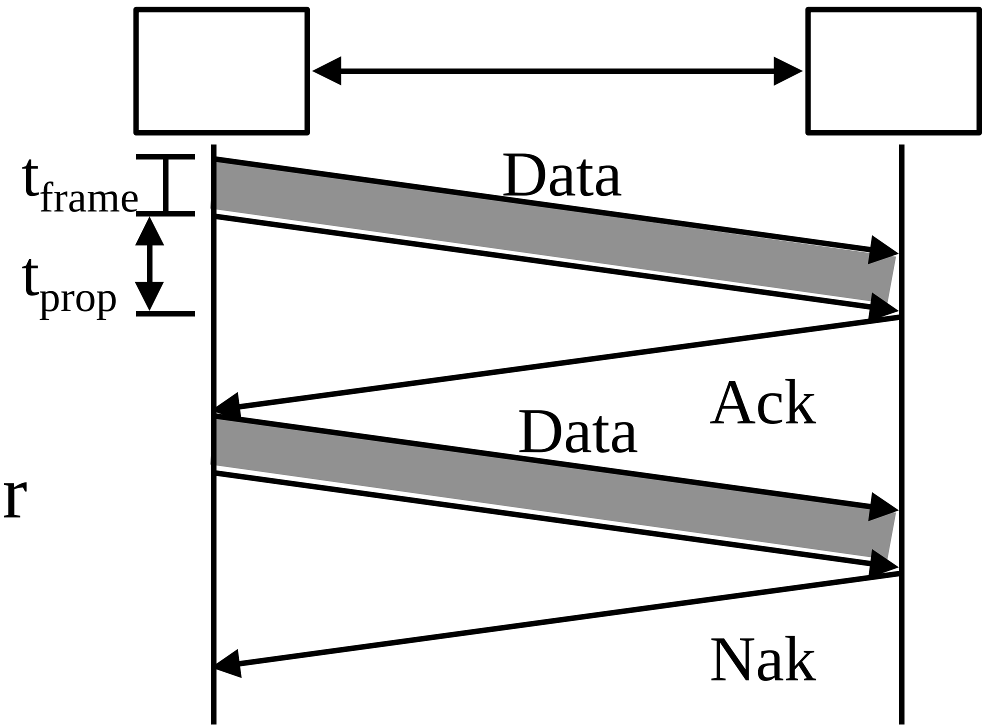
⇒ Will detect all odd number of bit errors

Error Control

- Error Control = Deliver frames without error, in the proper order to network layer
- Error control Mechanisms:
  - Ack/Nak: Provide sender some feedback about other end
  - Time-out: for the case when entire packet or ack is lost
  - Sequence numbers: to distinguish retransmissions from originals

Error Control

- Automatic Repeat Request (ARQ)
  - Error detection
  - Acknowledgment
  - Retransmission after timeout
  - Negative Acknowledgment



Stop-and-Wait ARQ

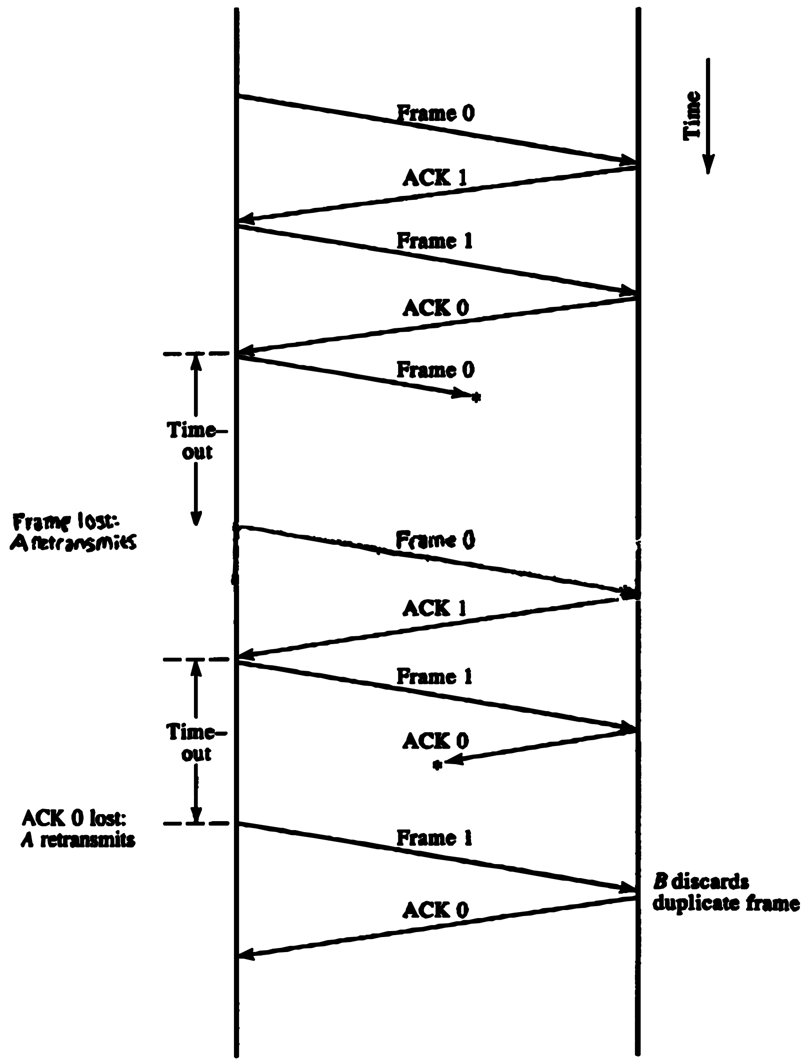
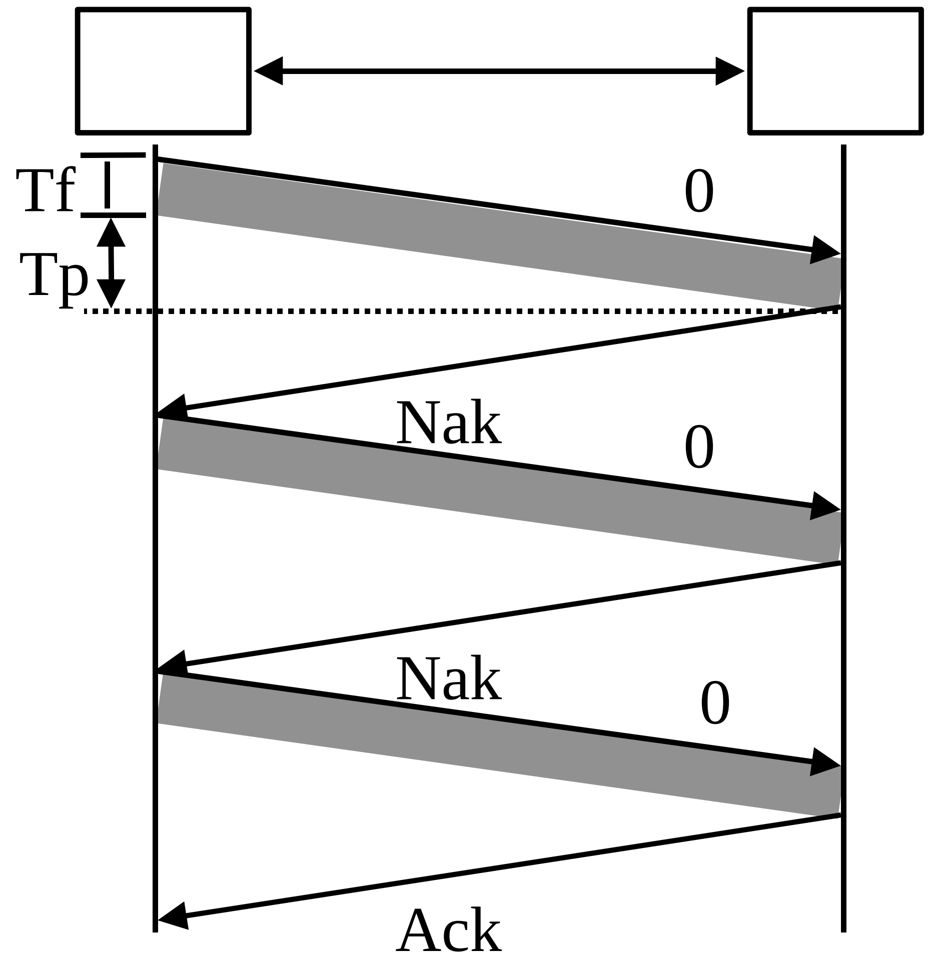


Fig 6.8 Stallings

Performance: Stop-and-Wait

- P=Probility of Frame Error
- $\alpha = T_p/T_f$
- $U=T_f/[N_r(T_f+2T_p)]$
  $= 1/[N_r(1+2\alpha)]$
- $N_r=\Sigma\ i\ P^{i-1}(1-P)$
  $=1/(1-P)$
- $U=(1-P)/(1+2\alpha)$



Go Back n: Example

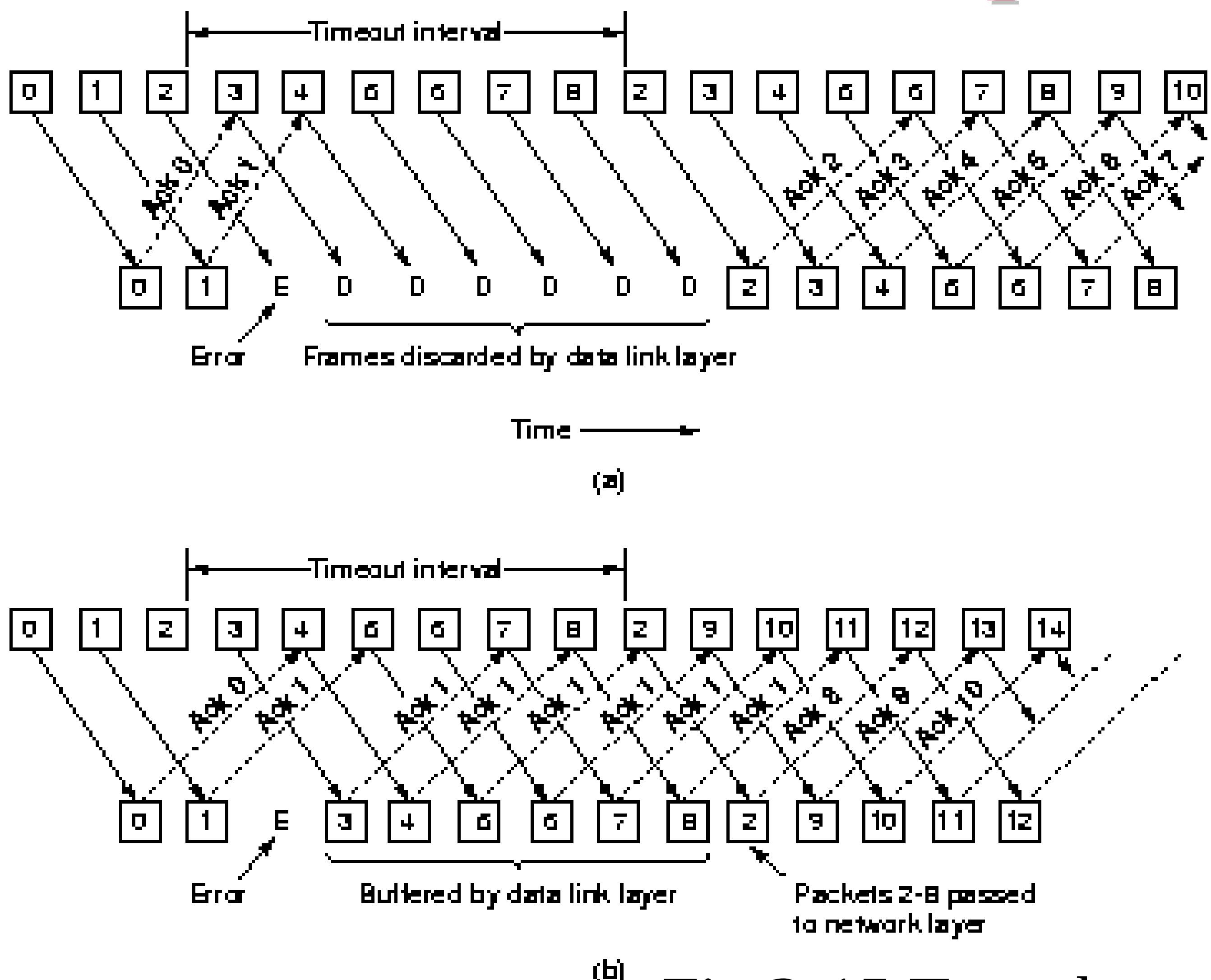
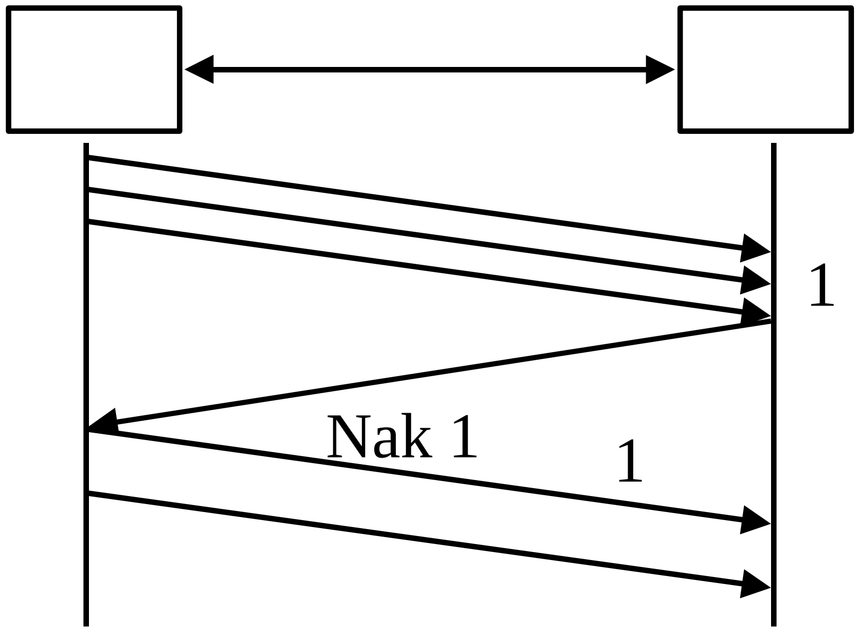


Fig 3-15 Tanenbaum

Go-back-N

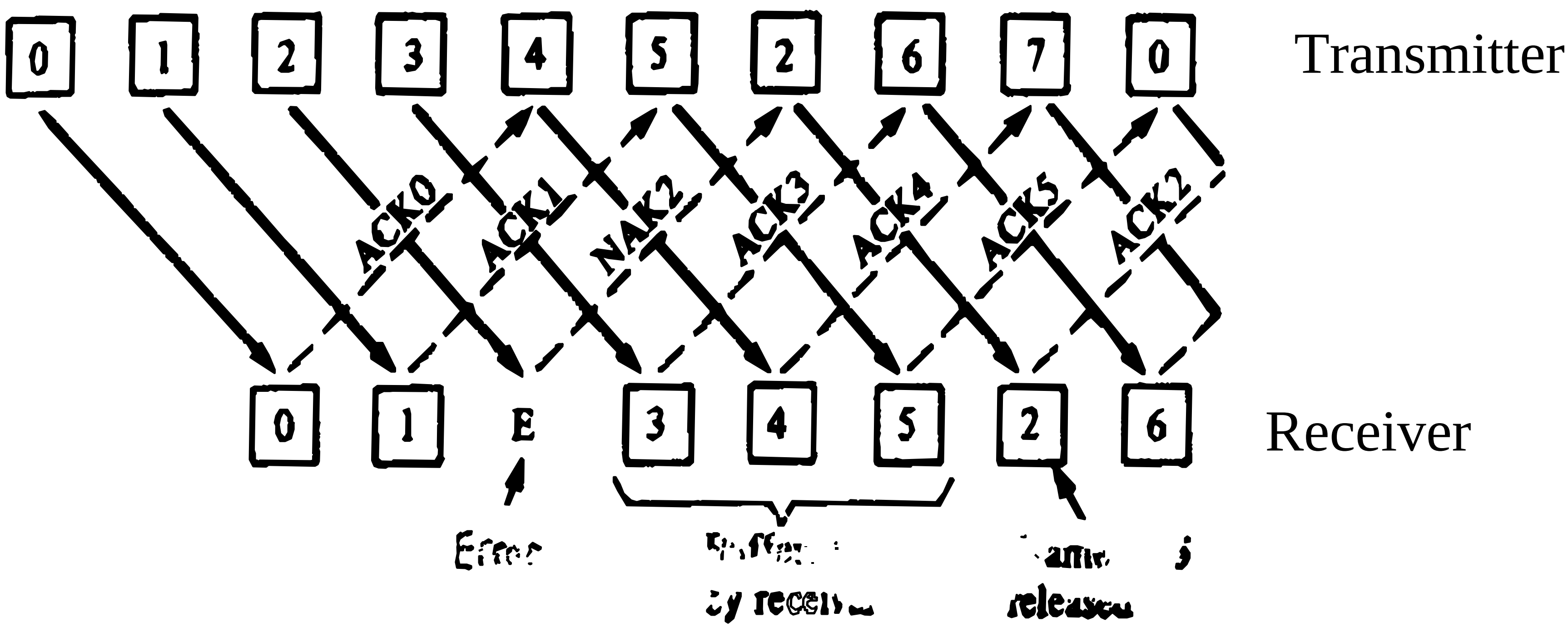
- Damaged Frame
  - Frame received with error
  - Frame lost
  - Last frame lost
- Damaged Ack
  - One ack lost, next one makes it
  - All acks lost
- Damaged Nak

Performance: Go-back-N

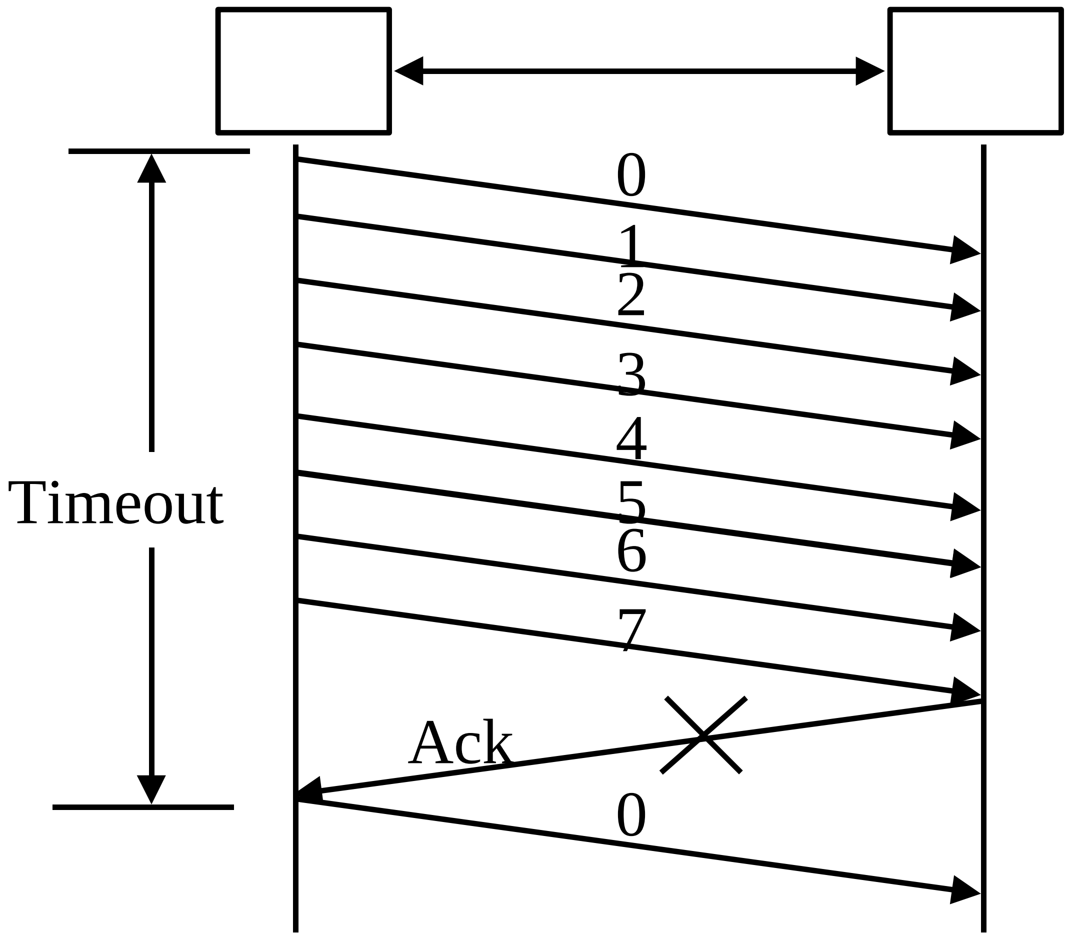


- Frames Retransmitted = $2\alpha+1$ if $N>2\alpha+1$
  N otherwise
- $U = (1-P)/(1+2\alpha P)$ if $N>2\alpha+1$
  $N(1-P)/[(2\alpha+1)(1-P+NP)]$ otherwise

Selective-Reject ARQ



Selective Reject: Window Size



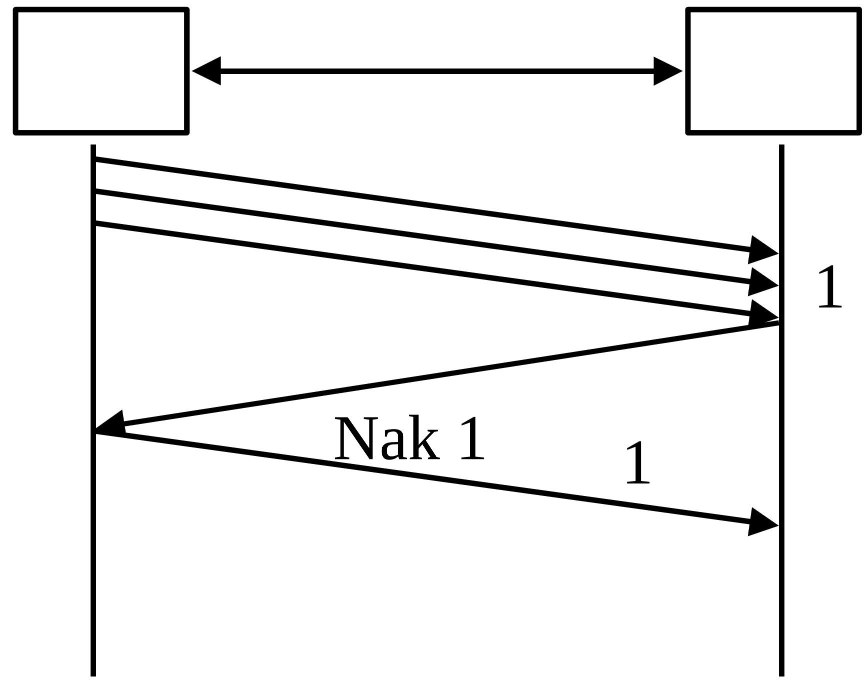
Sequence number space ≥ 2 window size

Performance: Selective Reject

- Error Free:
  U=1 if $N>2\alpha+1$
  $N/(2\alpha+1)$ otherwise

- With Errors:
  $N_r=\Sigma\, i\, P^{i-1}(1-P)$
  $=1/(1-P)$

- U=(1-P) if $N>(1+2\alpha)$
  $N(1-P)/(1+2\alpha)$ otherwise



Performance Comparison

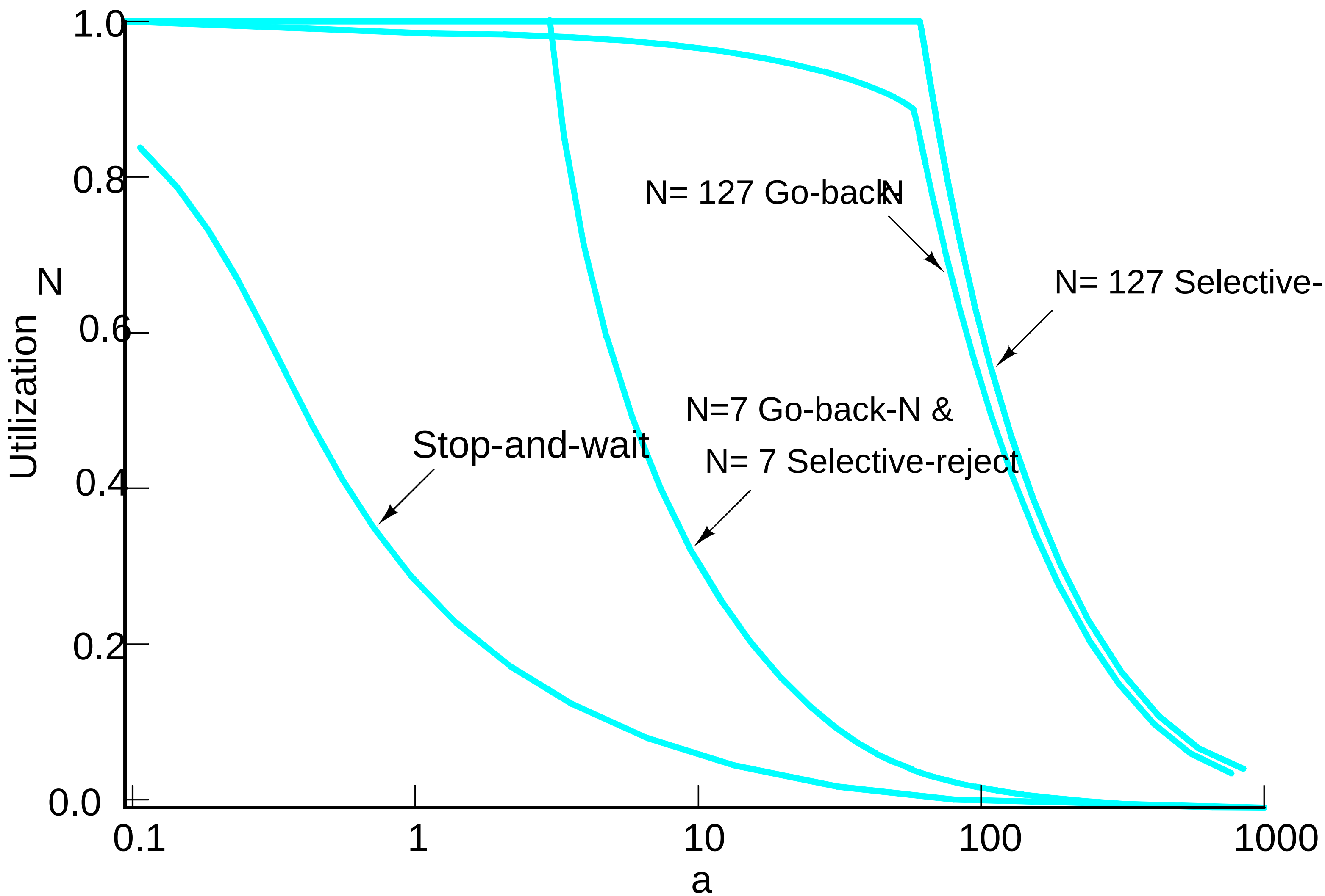
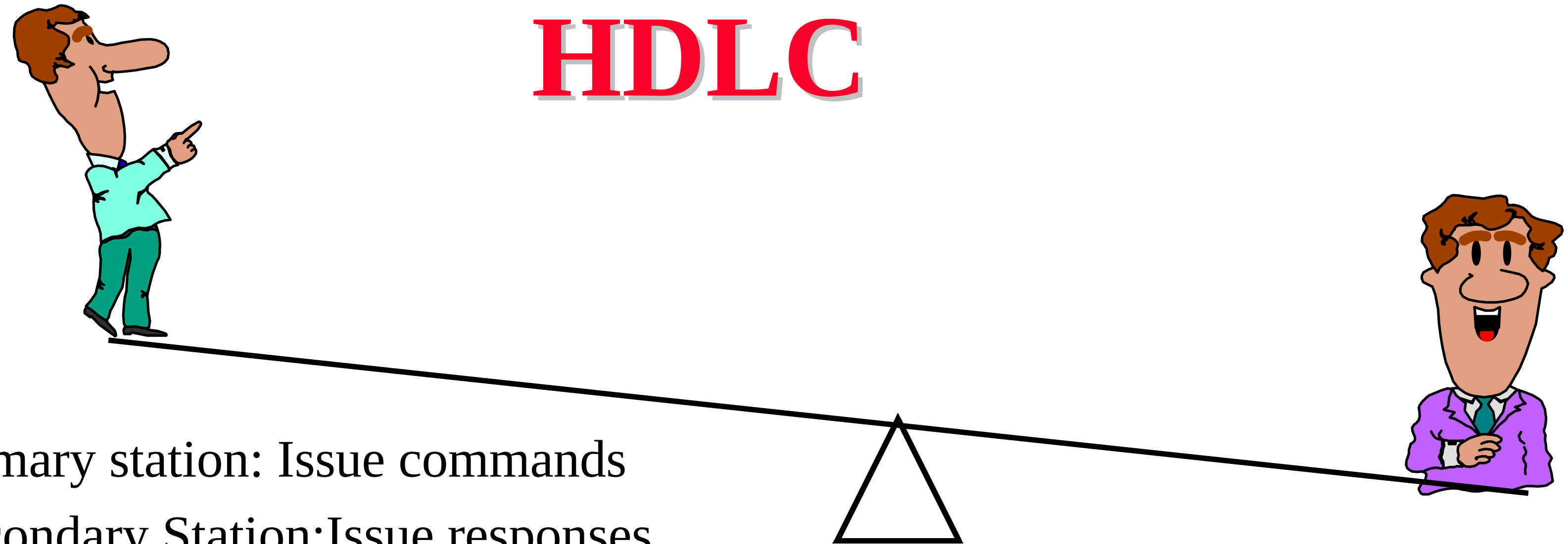


Fig 6.17 Stallings

HDLC Family

- Synchronous Data Link Control (SDLC): IBM
- High-Level Data Link Control (HDLC): ISO
- Link Access Procedure-Balanced (LAPB): X.25
- Link Access Procedure for the D channel (LAPD): ISDN
- Link Access Procedure for modems (LAPM): V.42
- Link Access Procedure for half-duplex links (LAPX): Teletex
- Point-to-Point Protocol (PPP): Internet
- Logical Link Control (LLC): IEEE
- Advanced Data Communications Control Procedures (ADCCP): ANSI
- V.120 and Frame relay also use HDLC

HDLC



- Primary station: Issue commands
- Secondary Station:Issue responses
- Combined Station: Both primary and secondary
- Unbalanced Configuration: One or more secondary
- Balanced Configuration: Two combined station
- Normal Response Mode (NRM): Response from secondary
- Asynchronous Balanced Mode (ABM): Combined Station
- Asynchronous Response Mode (ARM): Secondary may respond before command

HDLC Frame Structure

Frame Format

FLAG	ADDRESS	INFORMATION	FCS	FLAG
← 8 → bits	← 8 → Extendable	← 8 or 16 → Extendable	← 16 or 32 →	← 8 →

Control Field Format

	1	2	3	4	5	6	7	8
I: Information	0	N(S)			P/F	N(R)		
S: Supervisory	1	0	S		P/F	N(R)		
U: Unnumbered	1	1	M		P/F	M		

N(S)= Send sequence number N(R)= Recieve sequence number

S= Supervisory function bits M= Unnumbered bits P/F= Poll/final bit

Extended Address Field

1	2 3 4 5 6 7 8	9	10 11 12 13 14 15 16	......		8n
0		0		......	1	

Extended Control Field

	1	2 3 4 5 6 7 8	9	10 11 12 13 14 15 16
Information	0	N(S)	P/F	N(R)

	1	2	3 4	5	6	7	8	9	10 11 12 13 14 15 16
Supervisory	1	0	S	0	0	0	0	P/F	N(R)

Fig 6.10 Stallings

Bit Stuffing

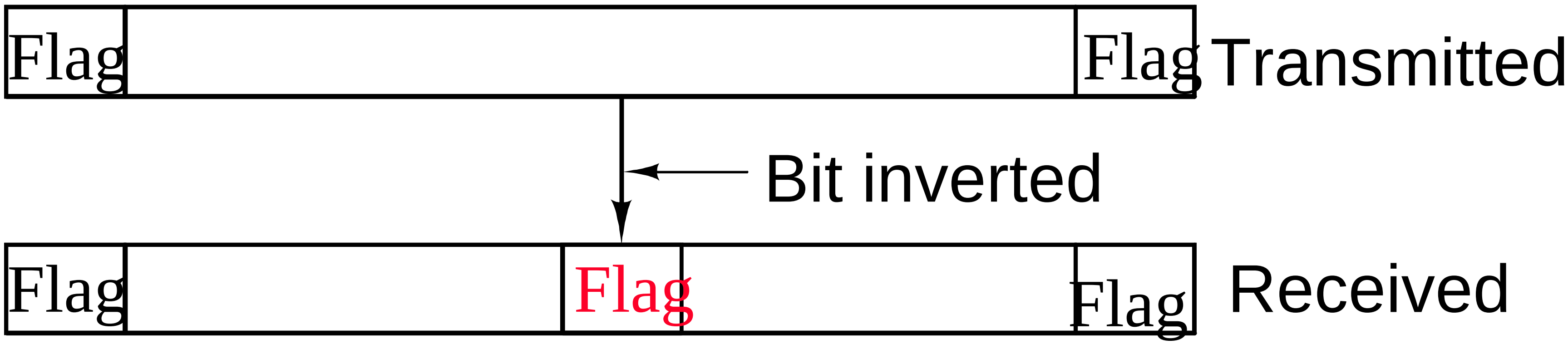
Original Pattern

111111111111011111101111110

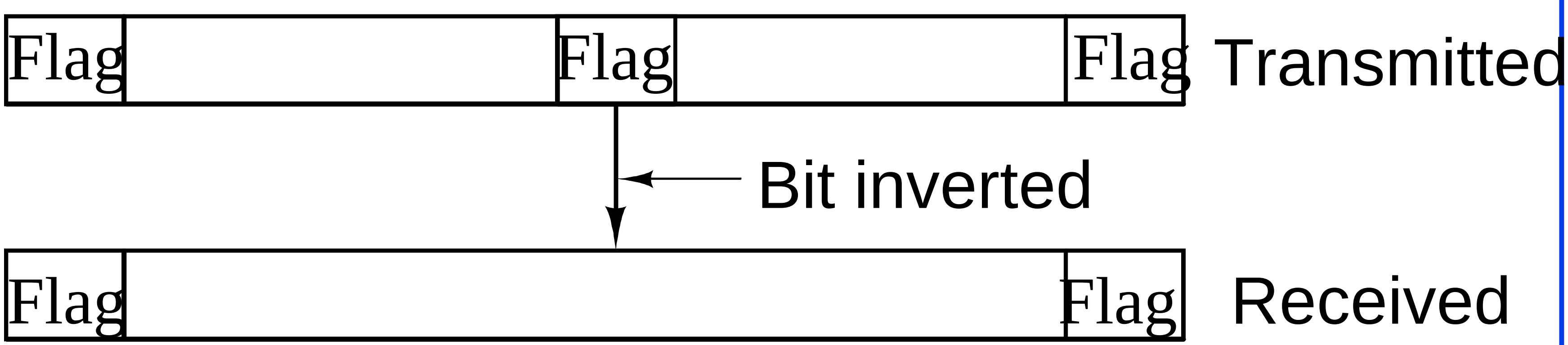
After bit-stuffing

1111101111101101111101011111010

Bit Stuffing (Cont)



(b) An inverted bit splits a frame in two



(c) An inverted bit merges two frames

HDLC Frames

- Information Frames: User data
  - Piggybacked Acks: Next frame expected
  - Poll/Final = Command/Response
- Supervisory Frames: Flow and error control
  - Go back N and Selective Reject
  - Final $\Rightarrow$ No more data to send
- Unnumbered Frames: Control
  - Mode setting commands and responses
  - Information transfer commands and responses
  - Recovery commands and responses
  - Miscellaneous commands and responses

HDLC Commands and Responses

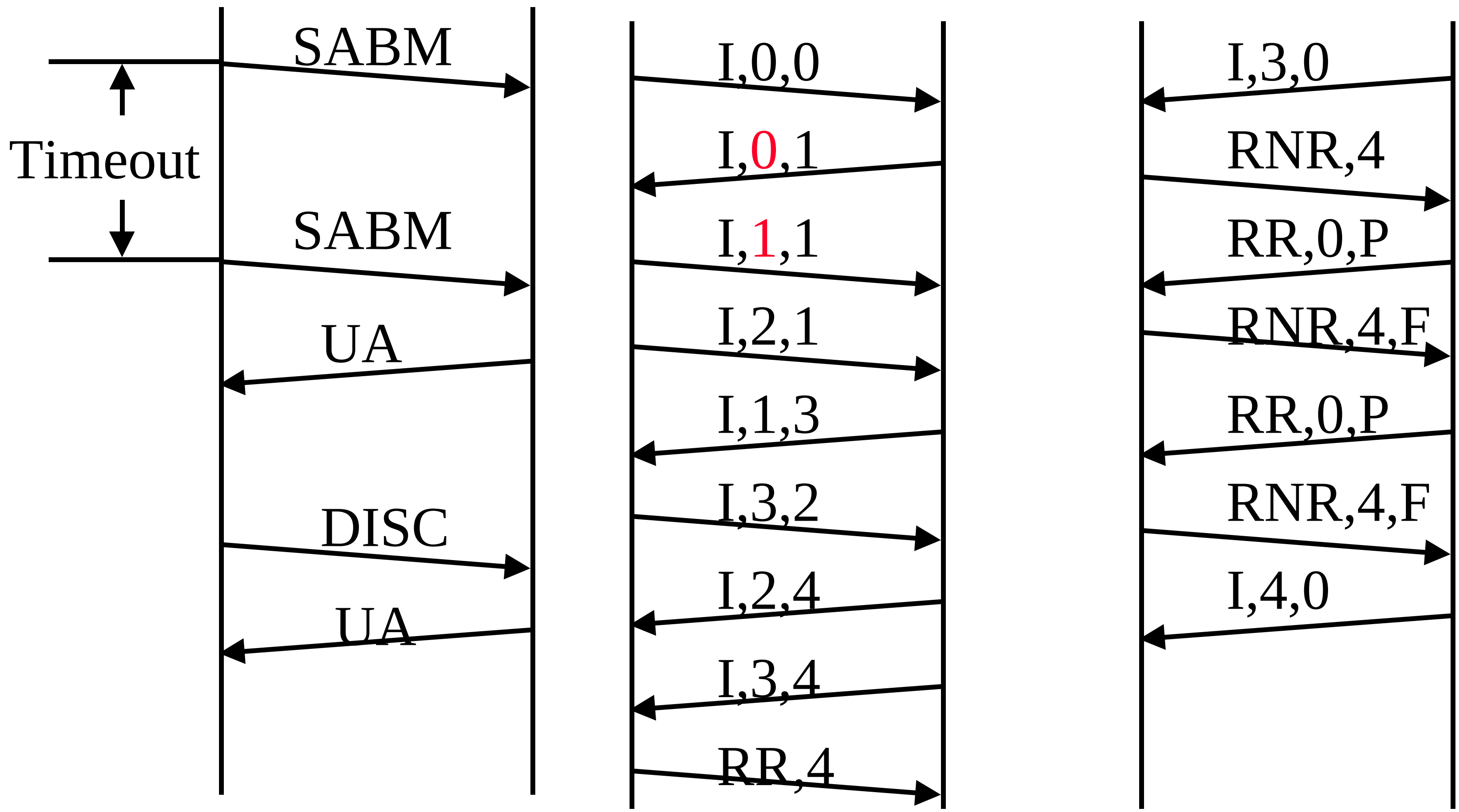
Name	Function	Description
Information (I)	C/R	Exchange user data
Supervisory (S)		
Recieve Ready (RR)	C/R	Positive Acknowledgement; ready to receive I-frame
Recieve Not Ready (RNR)	C/R	Positive acknowledgement; not ready to receive
Reject (REJ)	C/R	Negative acknowledgement; go back N
Selective Reject (SREJ)	C/R	Negative acknowledgement; selective reject
Unnumbered (U)		
Set Normal Response / Extended Mode (SNRM / SNRME)	C	Set mode;extended=two-octet control field
Set Asynchronous Response / Extended Mode (SARM / SARME)	C	Set mode;extended=two-octet control field
Set Asynchronous Balanced / Extended Mode (SABM / SABME)	C	Set mode;extended=two-octet control field
Set Initialization Mode (SIM)	C	Initialize link control functons in addressed station

Table 6.1 Stallings

HDLC Commands and Responses (cont)

Name	Function	Description
Disconnect (DISC)	C	Ter minate logical link connection
Unnumbered Acknowledgement (UA)	R	Acknowledges acceptance of one of the above set-mode commands
Disconnect Mode (DM)	R	Secondary is logically disconnected
Request Disconnect (RD)	R	Request for DISC command
Request Initialization Mode (RIM)	R	Initialization needed; request for SIM command
Unnumbered Information (UI)	C/R	Used to exchange control information
Unnumbered Poll (UP)	C	Used to solicit control information
Reset (RSET)	C	Used for recovery; resets N(R), N(S)
Exchange Identification (XID)	C/R	Used to request/report identity and status
Test (TEST)	C/R	Exchange identical information fields for testing
Frame Reject (FRMR)	R	Reports receipt of unacceptable frame

Examples of HDLC Operation



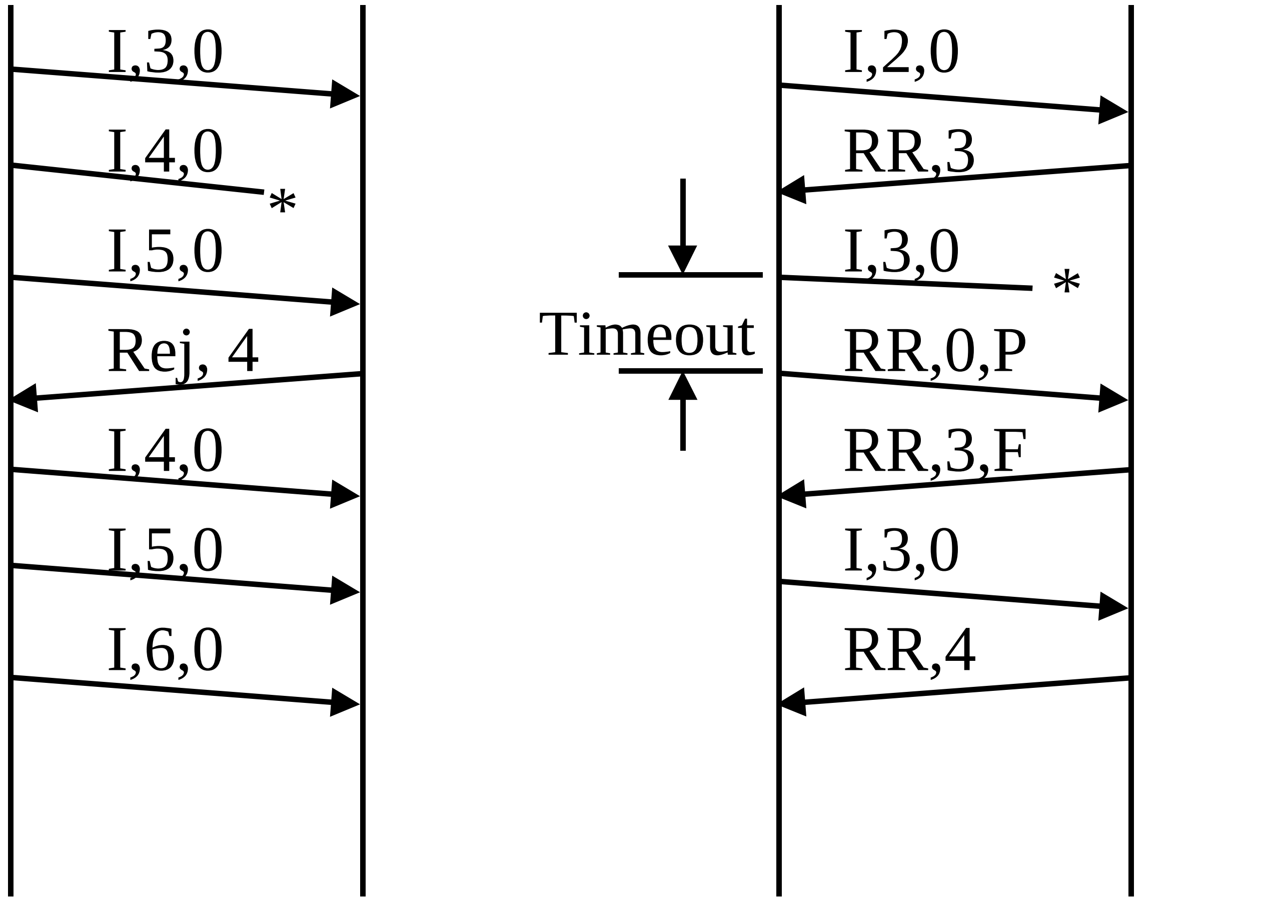
(a) Line setup and disconnect

(b) Two-way data exchange

(c) Busy condition

Fig 6.12 Stallings

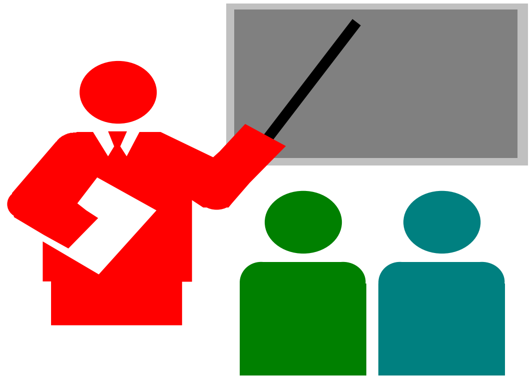
Examples of Operation (Cont)



(d) Reject Recovery (e) Timeout Recovery

Fig 6.12 Stallings

Summary



- Flow Control: Stop and Wait, Sliding window
- Effect of propagation delay, speed, frame size
- Error Detection: Parity, CRC
- Error Control: Stop and wait ARQ, Go-back-N, Selective Reject
- HDLC: Bit stuffing, Flag, I-Frame, RR, RNR

Homework

- Read chapter 6 of Stalllings.
- Homework: 6.7, **6.14, 6.18, 6.20**
  Due: Next class