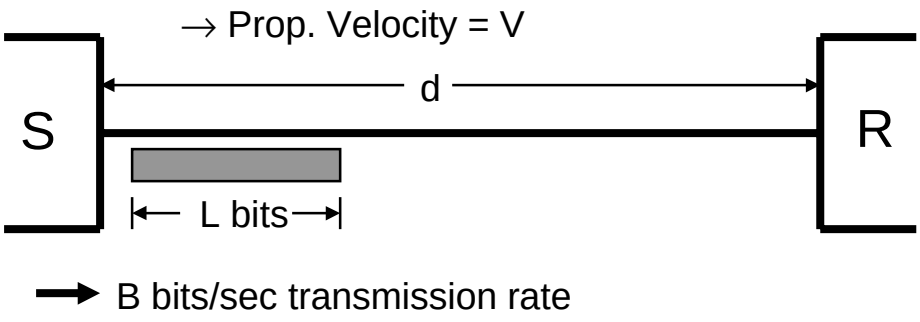
- Data Link Layer Functions
  - Frame Sync. - Create abstraction of “frame-at-a-time” channel for higher layer
  - Addressing - When many nodes share transmission link
  - Flow Control - Control rate of transmission to prevent over-run
  - Error Control - Correct transmission errors (by retransmission)
  - Sequence Control -
  - Link Management - Initiation, maintenance, & termination of connections

1. Flow Control

- Technique for controlling data rate so that sender does not over-run receiver
  - 1. Stop-and-wait
  - 2. Sliding-window

- Stop & Wait Flow Control
  - 1. Sender sends a frame
  - 2. Receiver receives frame & acknowledges it
  - 3. Sender waits to receive “ack” before sending next frame (If receiver is not ready to receive another frame it holds back the ack)
- Utilization (Efficiency) of Stop & Wait

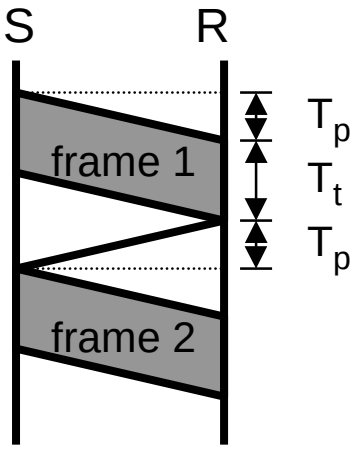


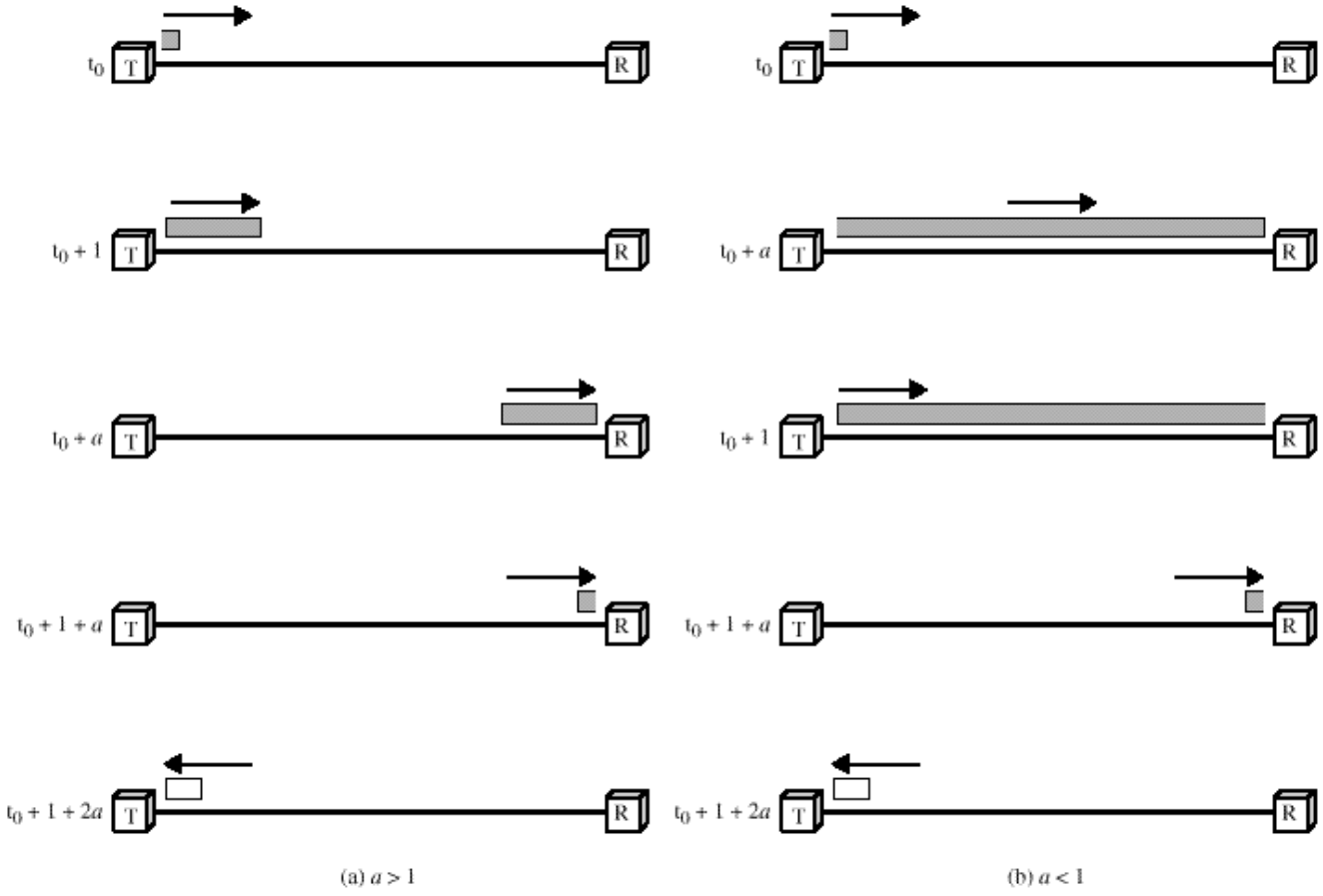
  - Propagation time: time taken for signal to travel from S to R. Thus first bit transmitted at t=0 arrives at R at t = T_p = d / V.
  - Transmission time: time taken to emit all bits of frame at sender = T_t = L / B.
  - $$a = \frac{\text{Propagation time}}{\text{Transmission time}} = \frac{T_p}{T_t} = \frac{d}{V} \cdot \frac{B}{L}$$

– Utilization (Efficiency) = U
  = Tt / (Time bet. Succ. Trans.)

$$= \frac{T_t}{T_p + T_t + T_p}$$

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





The effect of *a* on utilization

- With stop & wait scheme, for high channel utilization, we need a low a (since U= 1/(1+2a))

  $a = \frac{d \bullet B}{V \bullet L}$ Low a $\Rightarrow$ L as large as possible

- In practice it is desirable to limit frame length L
  - error probability increases with L
  - starvation/high avg. delay with multipoint lines
  - buffer size limitations
- So a more efficient scheme is called for, especially with WAN/satellite communication
- Examples
  - 1). WAN using ATM
    - L = 53 bytes = 424 bits, B= 155.52 Mbps, d=1000km

      $\Rightarrow T_t = 424 / (155.52 \times 10^6) = 2.7 \times 10^{-6}$ seconds

      assuming optical fiber, $T_p = 10^6$ m / (3×10^8 m/sec)

      $= 0.33 \times 10^{-2}$ seconds $\Rightarrow a = (0.33 \times 10^{-2})/(2.7 \times 10^{-6}) = 1200 \Rightarrow U = 1/2401 = 0.0004$
  - 2). LAN
    - d = 0.1 ~ 10 km, B = 10 Mbps, V = 2×10^8 m/sec
    - L = 1000 bits $\Rightarrow$ a = 0.005 ~ 0.5 $\Rightarrow$ U = 0.5 ~ 0.99
  - 3). Digital trans. vis modem over voice-grade line
    - B = 28.8 kbps, L = 1000 bits, d = 1000 m

      $\Rightarrow$ a = (28.8 kbps * 1000 m) / (2×10^8 m/sec x 1000 bits)

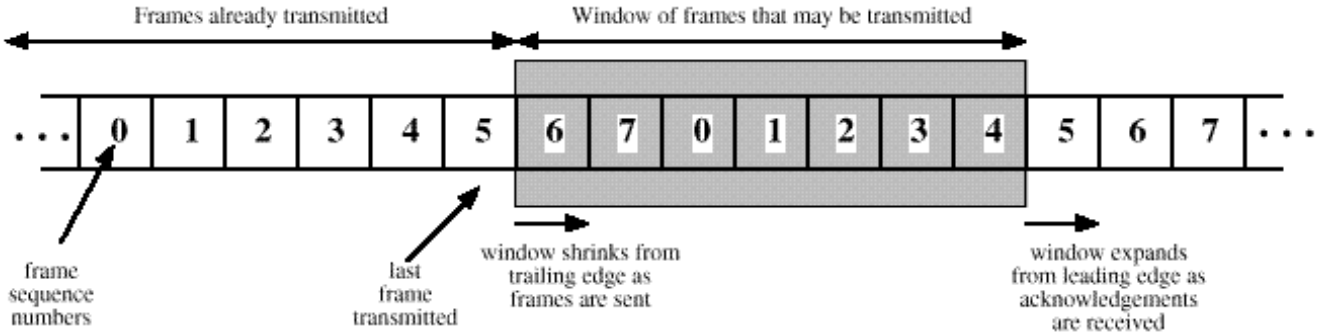
      $= 1.44 \times 10^{-4} \Rightarrow$ U = 1.0,

      if d = 5000 km,

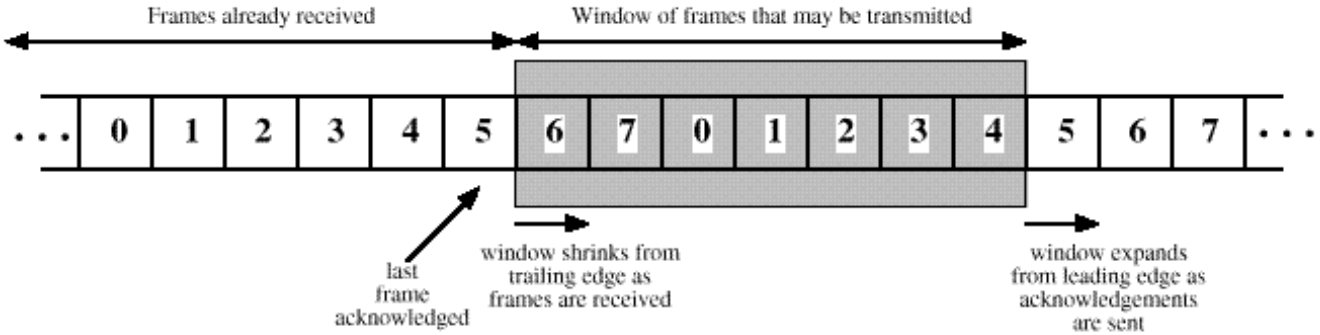
      a = (28.8kbps x 5000km) / (2×10^8 x 1000bits) = 0.72

      $\Rightarrow$ U = 0.4

- Sliding-Window Flow Control
  - Pipeline transmission of successive frames
  - Transmit up to “N” frames if necessary without receiving acks.
  - Wait for acks when “N” unacked frames in transit
  - For duplex transmission each station nees a sending window & receiving window.



(a) Transmitter's Perspective



(b) Receiver's Perspective

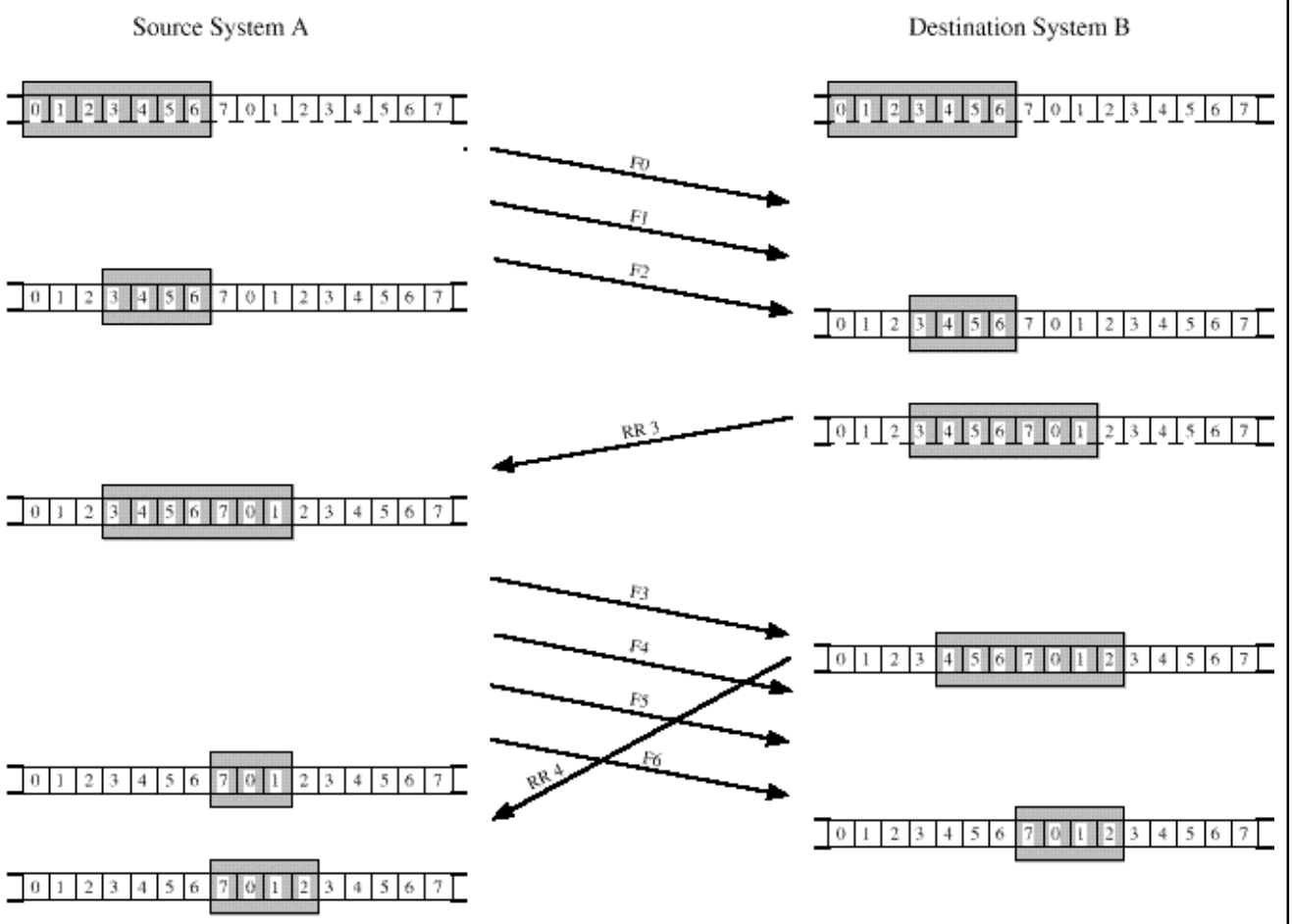
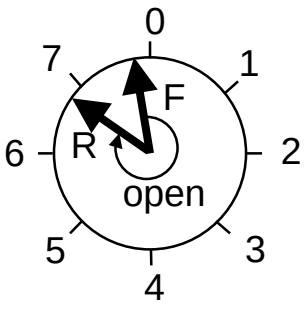
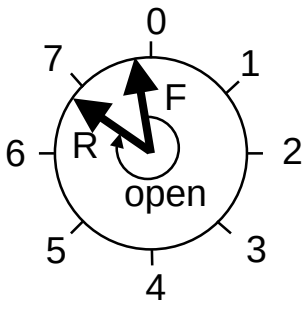


Figure 6.4 Example of a sliding-window protocol

Sliding-Window Protocol



Sender



Receiver

- Sender/Receiver window has two regions - open & closed, delimited by two edges F and R
- Open region includes all numbers bet. F and R clockwise, including F, but excluding R. If F=R, F is not included in open region
- Closed region includes all numbers bet. R and F clockwise, including R, but excluding F. If R=F, it is included in closed region.

Sender

Open Window: All frame #s currently permissible to use. Next frame to be used is #F

Closed Window: Frame #s currently disallowed from use

Send Frame:

if (F≠R) then {use frame #F;
F←(F+1)mod(N+1)}
else {cannot send now; need to wait till ack received}

Receive Ack #X:

if ((X-1)mod(N+1) in closed region)
then R←(X-1)mod(N+1)
else {error}

Receiver

Open Window: Frame #s expected to receive in future. Next expected frame is #F

Closed Window: #s in closed window excluding #R are frame #s so far received but not yet acked.

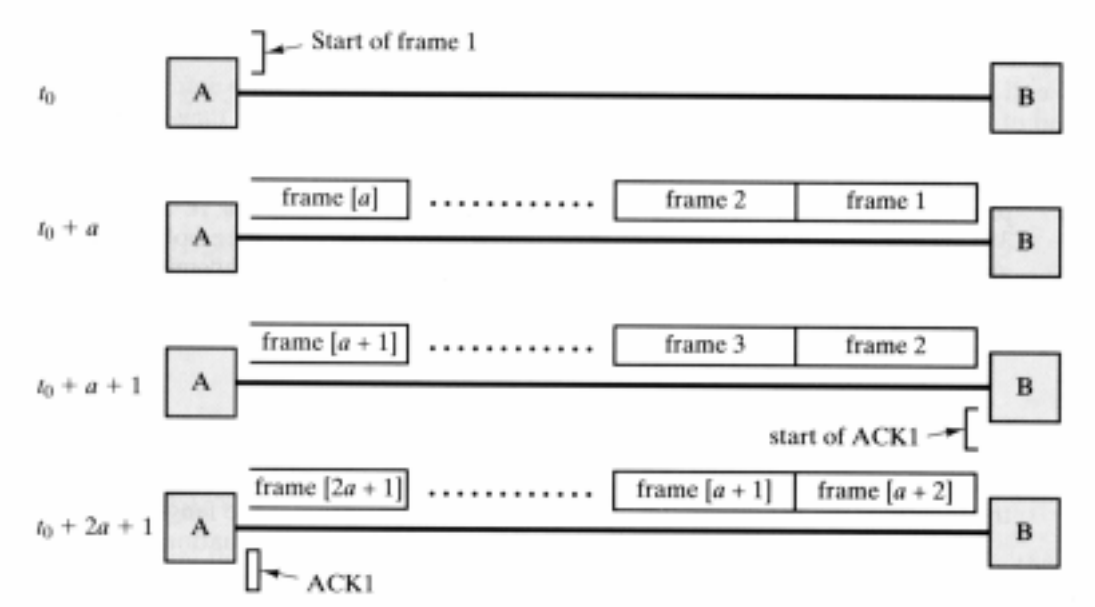
Receive Frame #X:

if (F=X) then F←(F+1)mod(N+1)
else {error}

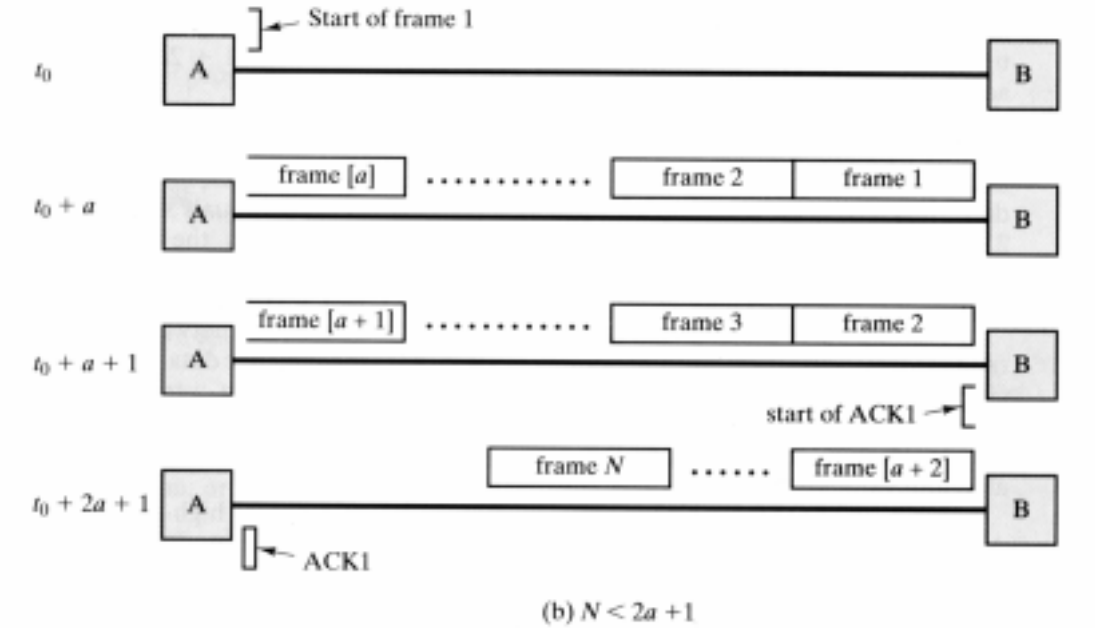
Send Ack for frame X:

Send Ack #(X+1)mod(N+1);
R ← X;

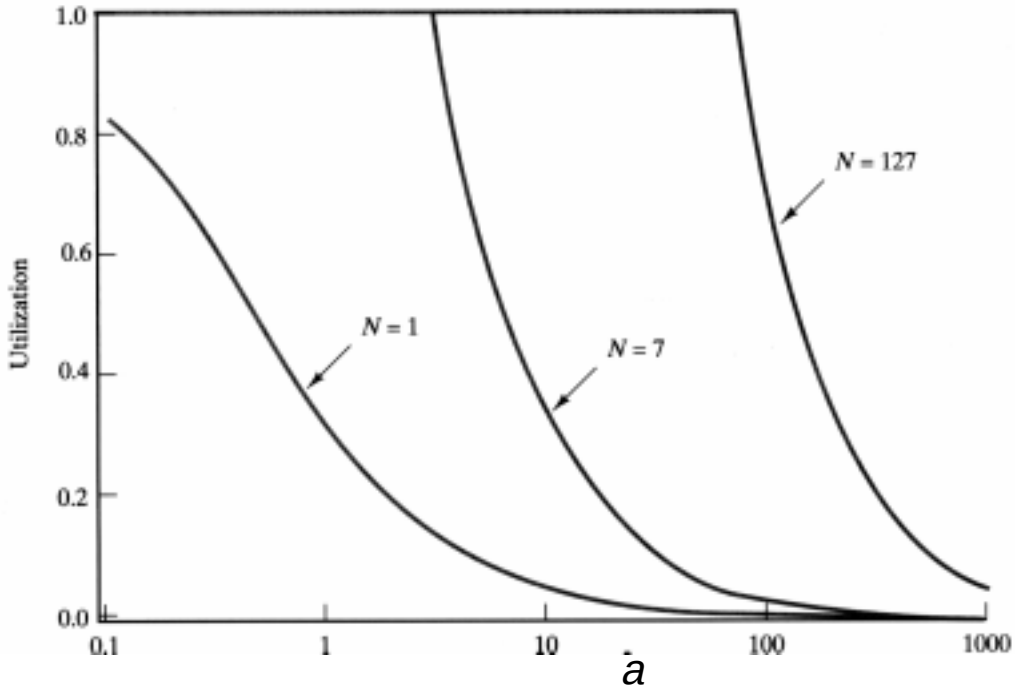
- Utilization: U is a function of a and N
  - Case 1: N > 1 + 2a : U = 1
    - Ack for frame 1 reaches S before transmission of N^{th} frame $\Rightarrow$ continuous transmission possible
  - Case 2: N < 1 + 2a : U = N / (1 + 2a)
    - Wasted time between N and 1 + 2a



(a) $N > 2a + 1$



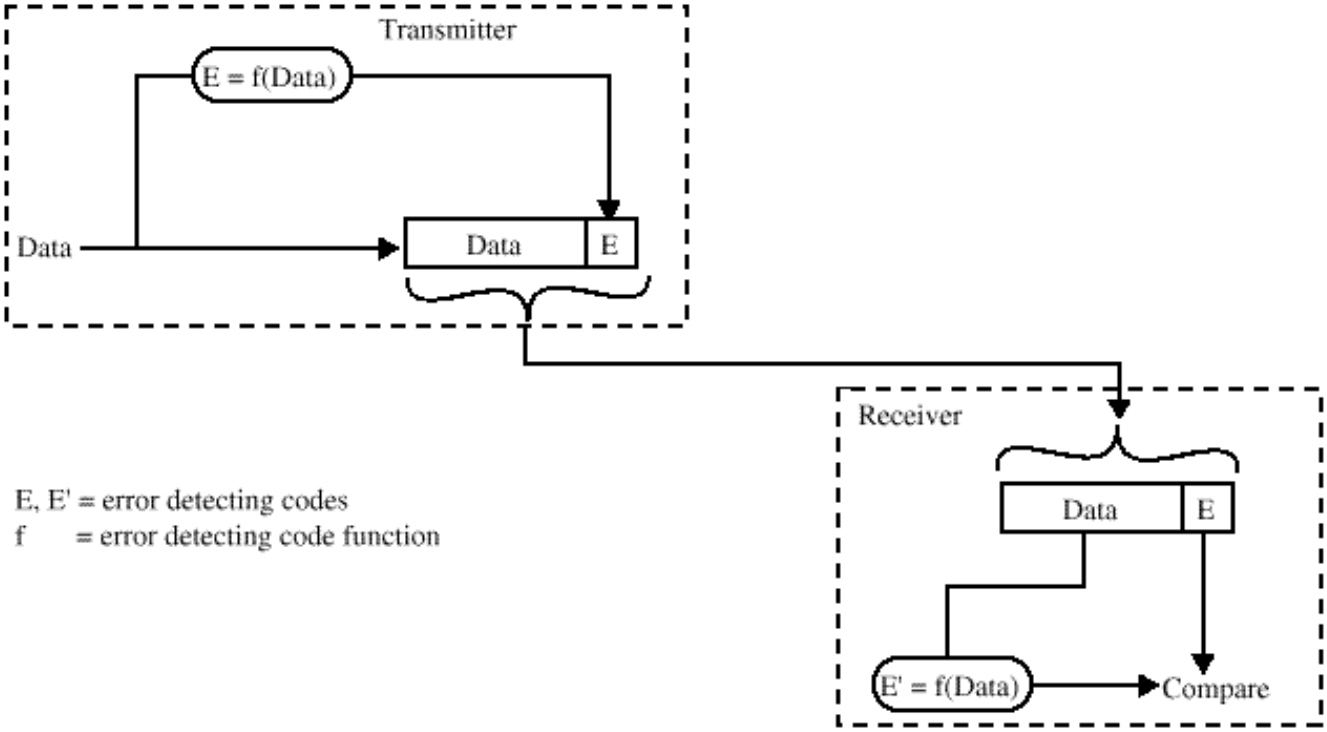
(b) $N < 2a + 1$



Line utilization as a function of window size

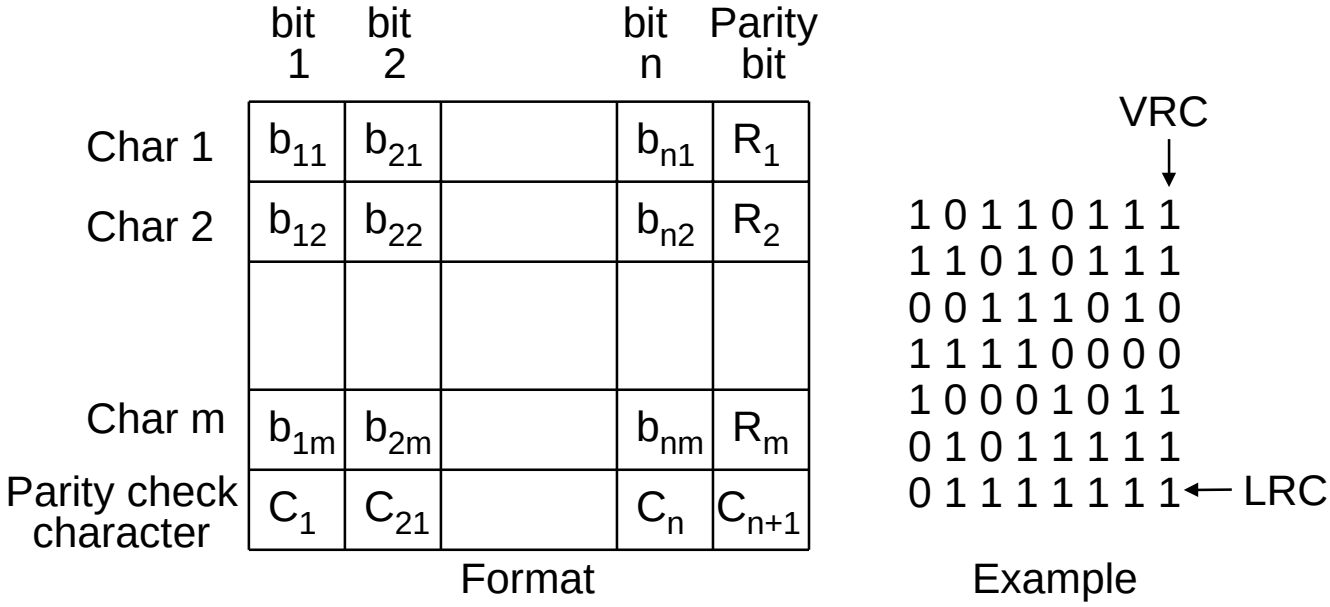
2. Error Detection

- Basic Principle
  - Transmitter: For a given bit stream M, additional bits (called error-detecting code) are calculated as a function of M and appended to the end of M
  - Receiver: For each incoming frame, perform the same calculation and compares the two results. A detected error occurs iff there is a mismatch



- Two common techniques
  - Parity checks
  - Cyclic redundancy checks (CRC)
- Parity Check
  - One extra “parity” bit is added to each word
    - Odd parity: bit added so as to make # of 1’s odd
    - Even parity: makes total # of 1’s even
  - Single parity is very effective with white noise, but not very robust with noise bursts (which may extend over whole word duration.)

- Additional block check characters (longitudinal check characters) are used: parity bit for first bits of all chars, 2nd bits of all chars and so on
- This decreases residual error rates by 2 to 4 orders of magnitude.



Vertical and longitudinal redundancy checks

– Cyclic Redundancy Checks

- Powerful error detection method, easily implemented
- Message (M) to be transmitted is appended with extra frame checksum bits (F), so that bit pattern transmitted (T) is perfectly divisible by a special “generator” pattern (P) - (divisor)
- At destination, divide received message by the same P. If remainder is nonzero $\Rightarrow$ error
- Let
  - T = (k+n)-bit frame to be transmitted, n<k
  - M = k-bit message, the first k bits of T
  - F = n-bit FCS, the last n bits of T
  - P = n+1 bits, generator pattern (predetermined divisor)
- Use modulo-2 arithmetic
  - no carries/borrows; add $\equiv$ subtract $\equiv$ xor
- Method
  - Extend M with n ‘0’s to the right ($\equiv 2^n M$)
  - Divide extended message by P to get R ($2^n M / P = Q + R/P$)
  - Add R to extended message to form T ($T = 2^n M + R$)
  - Transmit T
  - At receiver, divide T by P. Nonzero rem. $\Rightarrow$ error

$$\frac{T}{P} = \frac{2^n M + R}{P} = Q + \frac{R}{P} + \frac{R}{P} = Q + \frac{R+R}{P} = Q$$

- Example:
  - M = 110011, P = 11001, R = 4 bits
  - 1. Append 4 zeros to M, we get 1100110000
  - 2.

```
                 1 0 0 0 0 1
            ---------------------
11001 ) 1 1 0 0 1 1 0 0 0 0
      ⊕ 1 1 0 0 1
        -----------
        0 0 0 0 1
      ⊕ 0 0 0 0 0
        -----------
          0 0 0 1 0
        ⊕ 0 0 0 0 0
          -----------
            0 0 1 0 0
          ⊕ 0 0 0 0 0
            -----------
              0 1 0 0 0
            ⊕ 0 0 0 0 0
              -----------
                1 0 0 0 0
              ⊕ 1 1 0 0 1
                -----------
                  1 0 0 1
```

  - 3. ∴ T = 110011 1001

- Exercise: Compute the frame to be transmitted for message 1101011011 using P = 10011
  - Answer: 1101011011 1110

- Can view CRC generation in terms of polynomial arithmetic
- Any bit pattern ≡ polynomial in dummy varaible X
  - e.g., M = 110011 ≡ $1\cdot X^5+1\cdot X^4+0\cdot X^3+0\cdot X^2+1\cdot X+1\cdot X^0$

    $\therefore$ $M(X) = X^5+X^4+X+1$
- CRC generation in terms of polynomial
  - Append n '0's ≡ $X^nM(X)$
  - Modulo 2 division → $\frac{X^nM(X)}{P(X)} = Q(X) + \frac{R(X)}{P(X)}$
  - Transmit $X^nM(X)+R(X) = T(X)$
  - At receiver: $\frac{X^nM(X)+R(X)}{P(X)} = \frac{X^nM(X)}{P(X)} + \frac{R(X)}{P(X)}$

    $= Q(X) + \boxed{\frac{R(X)+R(X)}{P(X)}}$

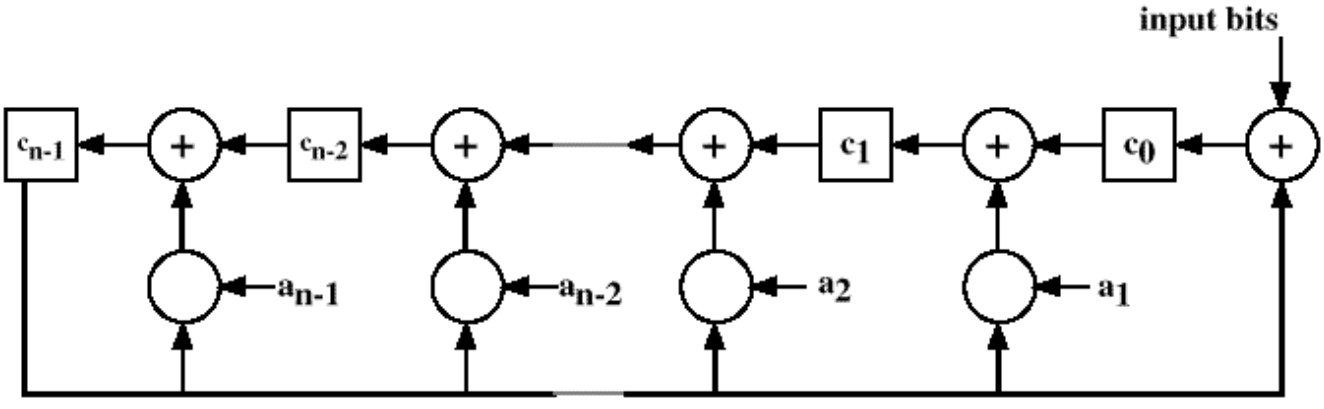
    → If no error 0 always
- Commonly used polynomials, P(X)
  - CRC-16 = $X^{16}+X^{15}+X^2+1$
  - CRC-CCITT = $X^{16}+X^{12}+X^5+1$
  - CRC-32 = $X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^8+X^7+X^{12}+X^4+X^2+X+1$

- Can detect
  - 1. All single-bit errors
  - 2. All double-bit errors, as long as P(X) has a factor with at least three terms
  - 3. Any odd number of errors, as long as P(X) contains a factor (X+1)
  - 4. Any burst error for which the length of the burst is less than the length of the FCS
  - 5. Most larger burst errors
- Why?
  - Error can also be rep. by polynomial, E(X).
    - T'(X) = T(X) + E(X)
    - Error is undetectable iff E(X) is divisible by P(X)
  - P has at least two terms, X^n, 1
  - ① Single-bit error: $E(X) = X^i$
    - $P(X) = X^n + \ldots + 1$ ∴ P(X) cannot divide E(X)
  - ② Double-bit error: $X^i+X^j = X^i(1+X^k)$, k=j-I>0
    - P(X) does not divide X^i
    - P(X) can be chosen which does not divide $1+X^k$ up to the maximum value of k (i.e., up to the practical frame length). (e.g., $X^{15}+ X^{14}+1$ will not divide $1+X^k$ for any k below 32768)

- ③ No polynomial with an odd # of terms is divisible by (X+1)
  - Assume E(X) has an odd # of terms and is divisible by (X+1). Then E(X) = (X+1)Q(X). E(1) = (1+1)Q(1) = 0. However, E(1) cannot be zero since it has an odd # od 1's
- ④ A burst error of length r < n can be represented by $X^i(X^{r-1}+ \cdots +1)$.
  - P(X) does not divide X^i
  - P(X) which is a polynomial of degree n cannot divide $X^{r-1}+ \cdots +1$ since r-1<n.

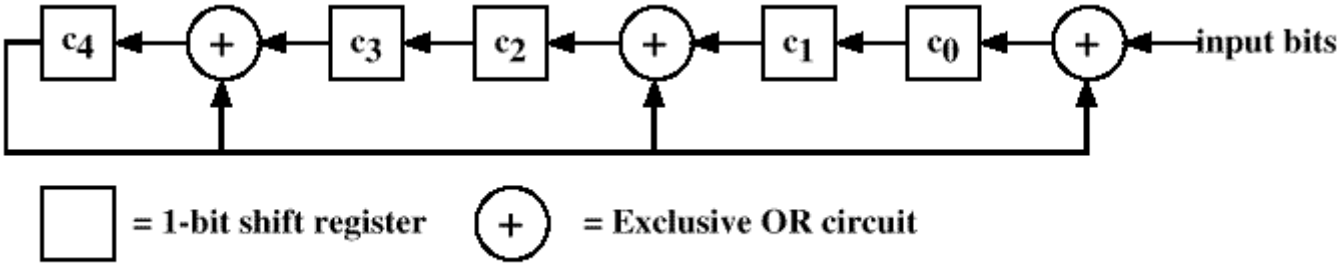
Implementation

- Implemented by a circuit consisting of exclusive-or gates and a shift register
  - The shift register contains n bits (length of FCS)
  - There are up to n exclusive-or gates
  - The presence or absence of a gate corresponds to the presence or absence of a term in P(X)



General CRC architecture to implement divisor $1 + a_1X + a_2X^2 + \ldots + a_{n-1}X^{n-1} + X^n$

Example



= 1-bit shift register + = Exclusive OR circuit

(a) Shift-register implementation

	c_4	c_3	c_2	c_1	c_0	c_4	c_3	c_4	c_1	c_4	input	input
Initial	0	0	0	0	0	0		0			1	1
Step 1	0	0	0	0	1	0		0			0	0
Step 2	0	0	0	1	0	0		1			1	1
Step 3	0	0	1	0	1	0		0			0	0
Step 4	0	1	0	1	0	1		1			0	0
Step 5	1	0	1	0	0	1		1			1	0
Step 6	1	1	1	0	1	0		1			0	1
Step 7	0	1	1	1	0	1		1			1	1
Step 8	1	1	1	0	1	0		1			1	0
Step 9	0	1	1	1	1	1		1			1	1
Step 10	1	1	1	1	1	0		0			1	0
Step 11	0	1	0	1	1	1		1			0	0
Step 12	1	0	1	1	0	1		0			1	0
Step 13	1	1	0	0	1	0		1			1	0
Step 14	0	0	1	1	1	0		1			0	0
Step 15	0	1	1	1	0	1		1			0	—

Message to be sent

Five zeros added

(b) Example with input of 1010001101

Circuits with shift registers for dividing by the polynomial $X^5+X^4+X^2+1$

3. Error Control

- Error control techniques
  - Forward error control:
    - Error recovery by correction at the receiver
      (Forward Error Correction (FEC))
  - Backward error control:
    - Error recovery by retransmissions
      (Automatic Repeat Request (ARQ))
- ARQ
  - Based on
    - Error detection
    - Positive ack
    - Retransmission after timeout
    - Negative ack. And retransmission
  - ARQ
    - Stop-and-wait ARQ
    - Continuous ARQ
      - Go-back-N ARQ
      - Selective-reject ARQ

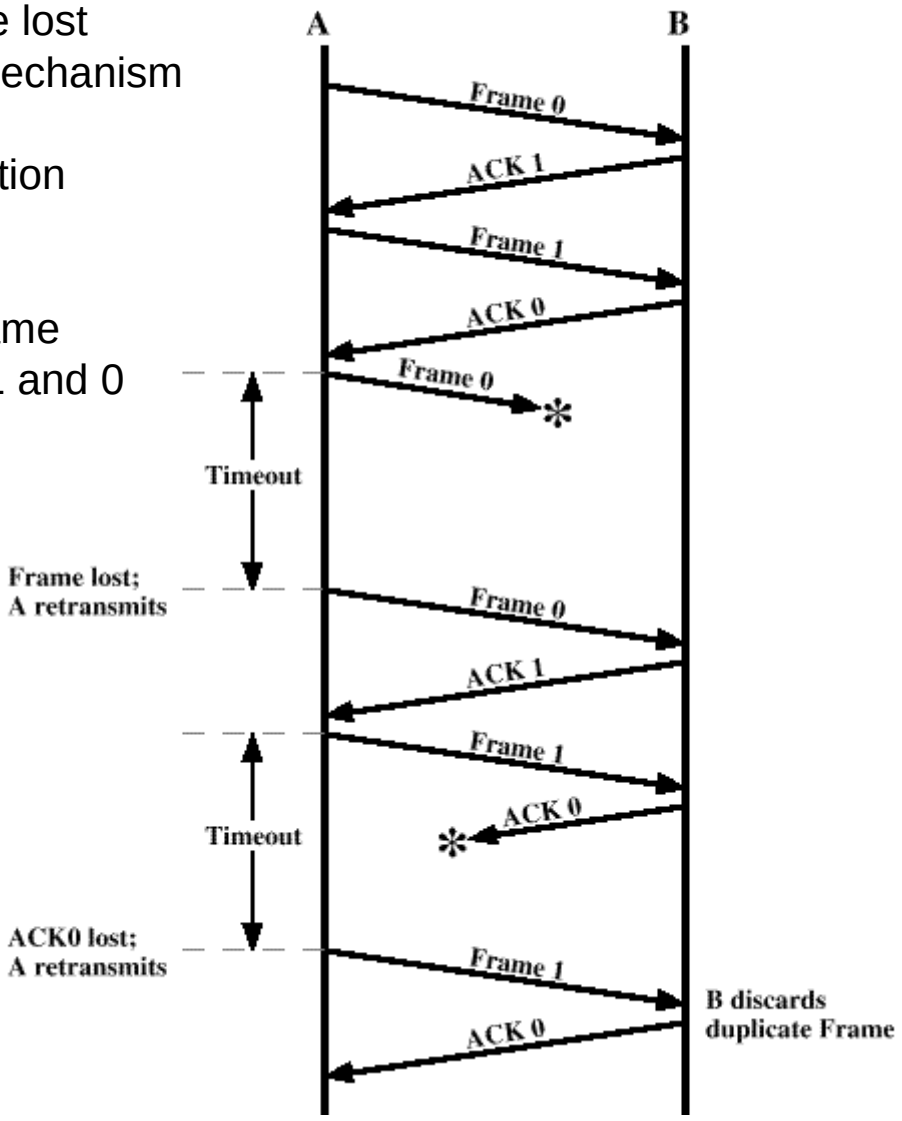
- Stop & Wait ARQ
  - Simple and minimum buffer requirement, but inefficient
    - Sender transmits message frame
    - Receiver checks received frame for errors; sends ACK/NAK
    - Sender waits for ACK/NAK.

      NAK ⇒ retrans; ACK ⇒ next frame

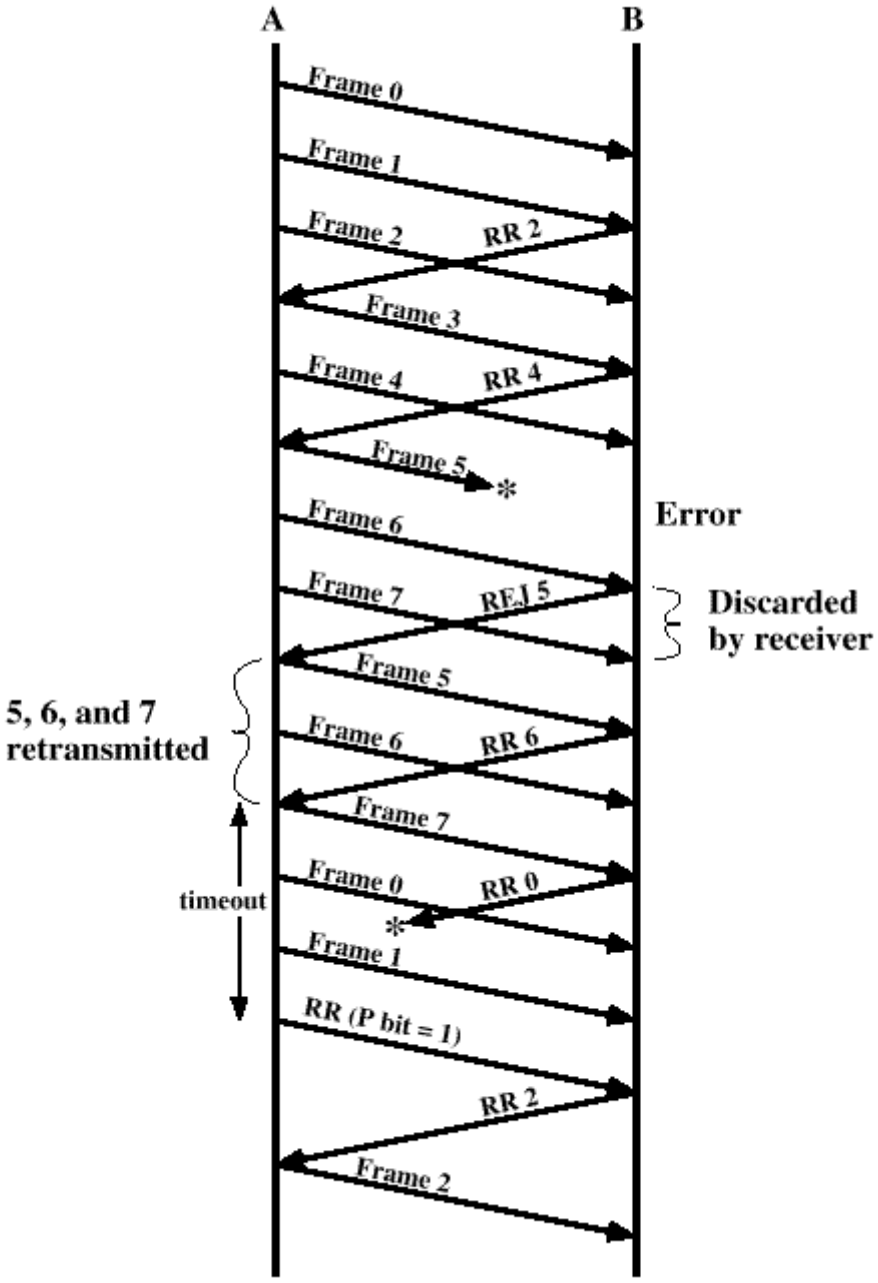
Frame/ACK could be lost
⇒ Uses a timeout mechanism

Possibility of duplication
⇒ Number frames

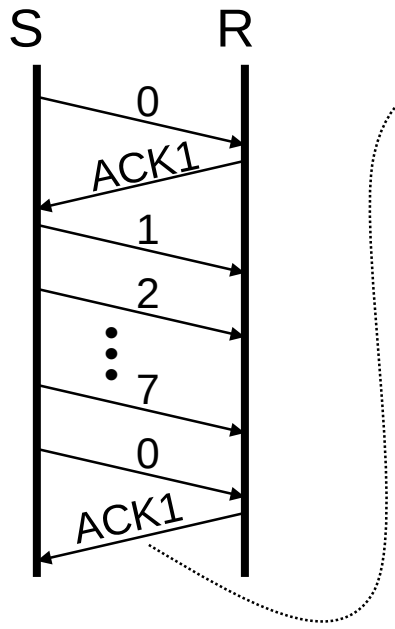
Only need a 1-bit frame number alternating 1 and 0



- Go-back-N ARQ
  - If the receiver detects an error on a frame, it sends a NAK for that frame. The receiver will discard all future frames until the frame in error is correctly received. Thus the sender, when it receives a NAK or timeout, must retransmit the frame in error plus all succeeding frames. (Sender must maintain a copy of each unacknowledged frame.)

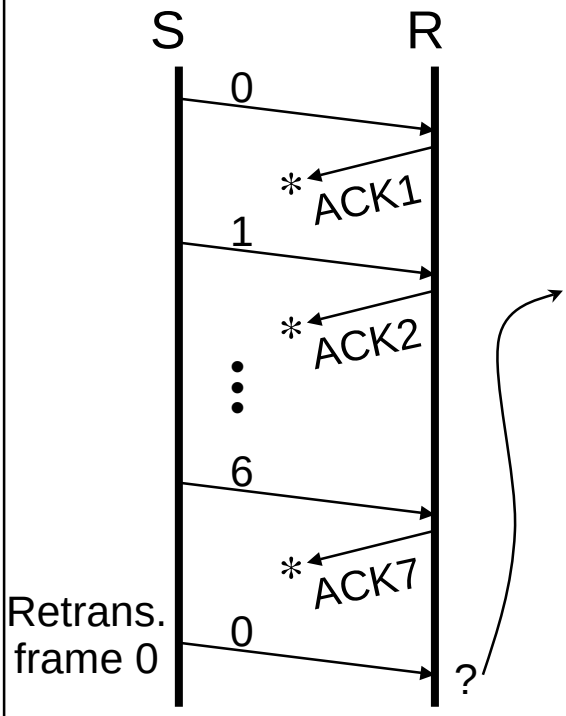


- Selective-reject ARQ
  - The only frames retransmitted are those that receive a NAK or which timeout
  - Can save retransmissions, but requires more buffer space and complicated logic
- Maximum window size (with n-bit sequence number)
  - Go-back-N : $2^n - 1$
  - Selective-reject : 2^{n-1}
- Why $2^n - 1$ instead of 2^n in Go-back-N ?



Question: Did all eight frames arrive successfully, or did all eight frames get lost (or errors) ? In both cases the receiver would send ACK1. The sender has no way of telling

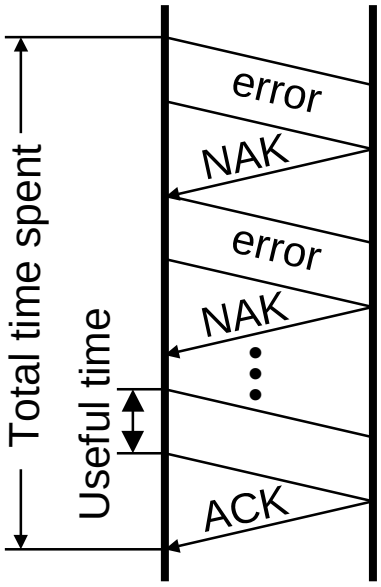
- Why $2^n/2$ instead of $2^n - 1$ in selective-reject?



At this point receiver has already advanced its window to accept frames 7, 0, 1, 2, 3, 4, 5. Thus it assumes that frame 7 has been lost and that this is a new frame 0, so it accepts.

$\therefore$ There should be no overlap between the sender and receiver windows.
$\Rightarrow$ Max window size $\leq 2^n/2$

- Link Utilization of Stop & Wait ARQ



$$U = \frac{\text{Useful time}}{\text{Total time}} = \frac{T_t}{N_r (T_t + 2T_p)}$$

Where N_r = expected # of attempts needed for successful transmission
Let P = prob. of error in frame transmission.

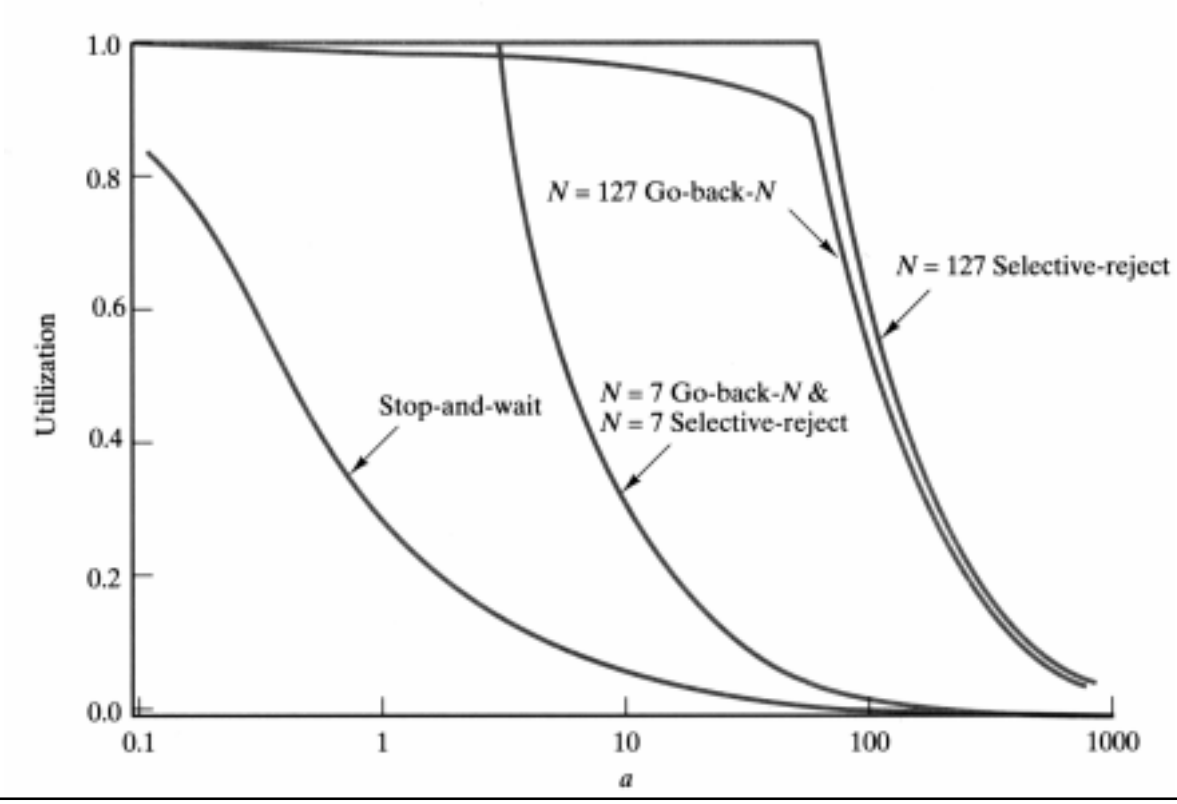
Prob(k attempts are needed)
$= P^{k-1}(1-P)$

$$\text{Then, } N_r = \sum_{k=1}^{\infty} k \bullet \text{Prob(k attempts are needed)}$$

$$= \sum_{k=1}^{\infty} k \bullet P^{k-1}(1-P) = \frac{(1-P)}{P} \sum_{k=1}^{\infty} k \bullet P^k = \frac{1-P}{P} \bullet \frac{P}{(1-P)^2} = \frac{1}{1-P}$$

$$\therefore \; U = \frac{1}{N_r(1+2a)} = \frac{1-P}{1+2a}$$

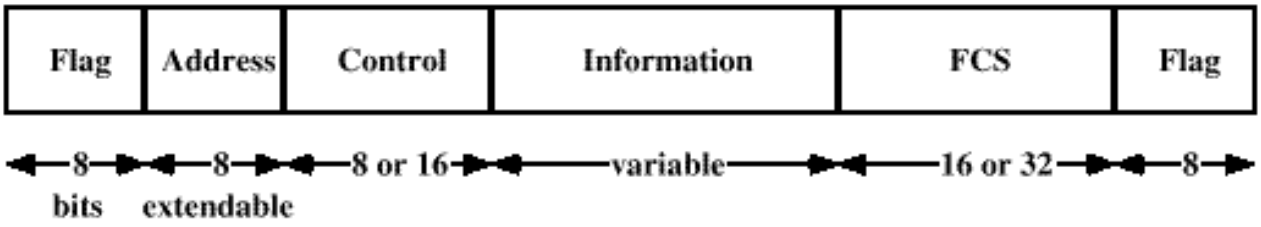
- Link Utilization of Sliding-Window Protocols
  - Selective-reject
    - 1 - P for N > (1+2a),
    - N(1-P)/(1+2a) for N < (1+2a)
  - Go-back-N
    - (1-P)/(1+2aP) for N > (1+2a)
    - N(1-P)/{(1+2a)(1-P+NP)} for N < (1+2a)



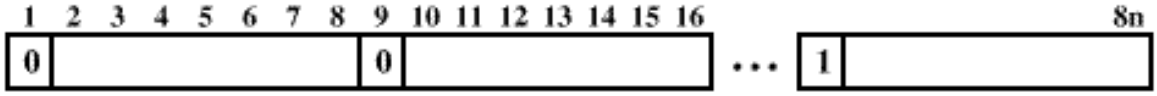
3. High-Level Data Link Control (HDLC)

- A synchronous data link protocol
- Widely used, and basis for many other data link control protocols
- Connections can be multipoint or point-to-point
- Can be used in half-duplex or full-duplex
- Three types of stations
  - Primary station - (Mainframe) Controls the operation of the link, issues commands, and receive responses
  - Secondary station - (Terminal) Usually only communicates (response) to a primary station
  - Combined station - Can be both a primary and a secondary
- Two link configurations
  - Unbalanced configuration - One primary and one or more secondary stations
  - Balanced configurations - Two combined stations
- Three data transfer modes
  - Normal response mode (NRM) - Unbalanced config. Primary station always dictates who sends and receives
  - Asynchronous balanced mode (ABM) - Two combined stations
  - Asynchronous response mode - Unbalanced config. Secondary station can send at any given time, but only one secondary can be active at a time

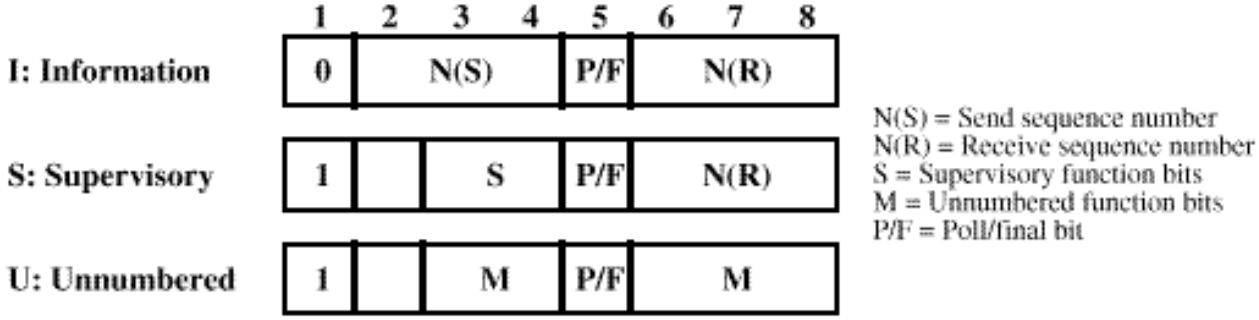
- HDLC Frame Structure



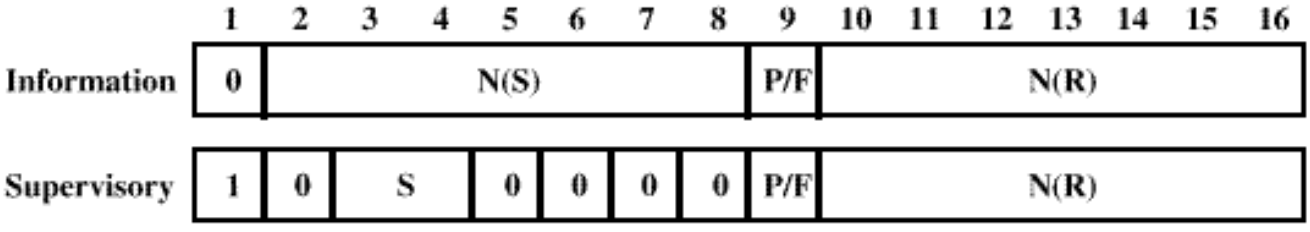
(a) Frame format



(b) Extended Address Field



(c) 8-bit control field format



(d) 16-bit control field format

– Flag fields
  - 8 bits (01111110)
  - “Bit stuffing” is used for data transparency
  - Bit stuffing: whenever five 1’s are transmitted, extra zero is inserted

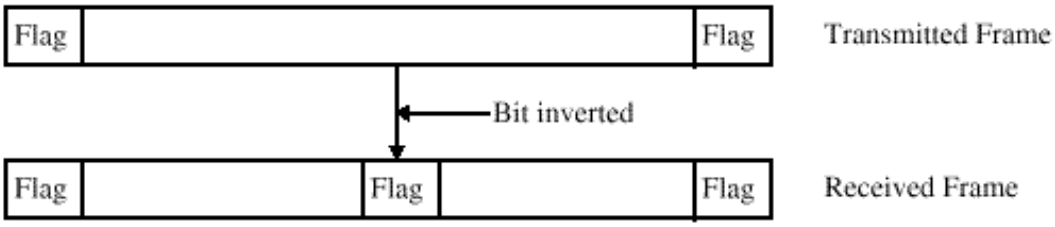
Original Pattern:

```
111111111111011111101111110
```

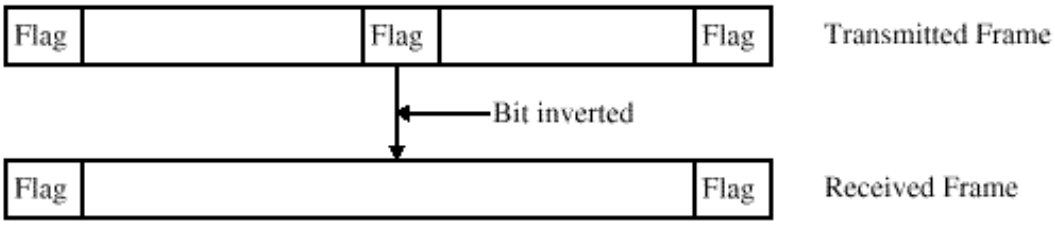
After bit-stuffing

```
111110111111011011111010111111010
```

(a) Example



(b) An inverted bit splits a frame in two



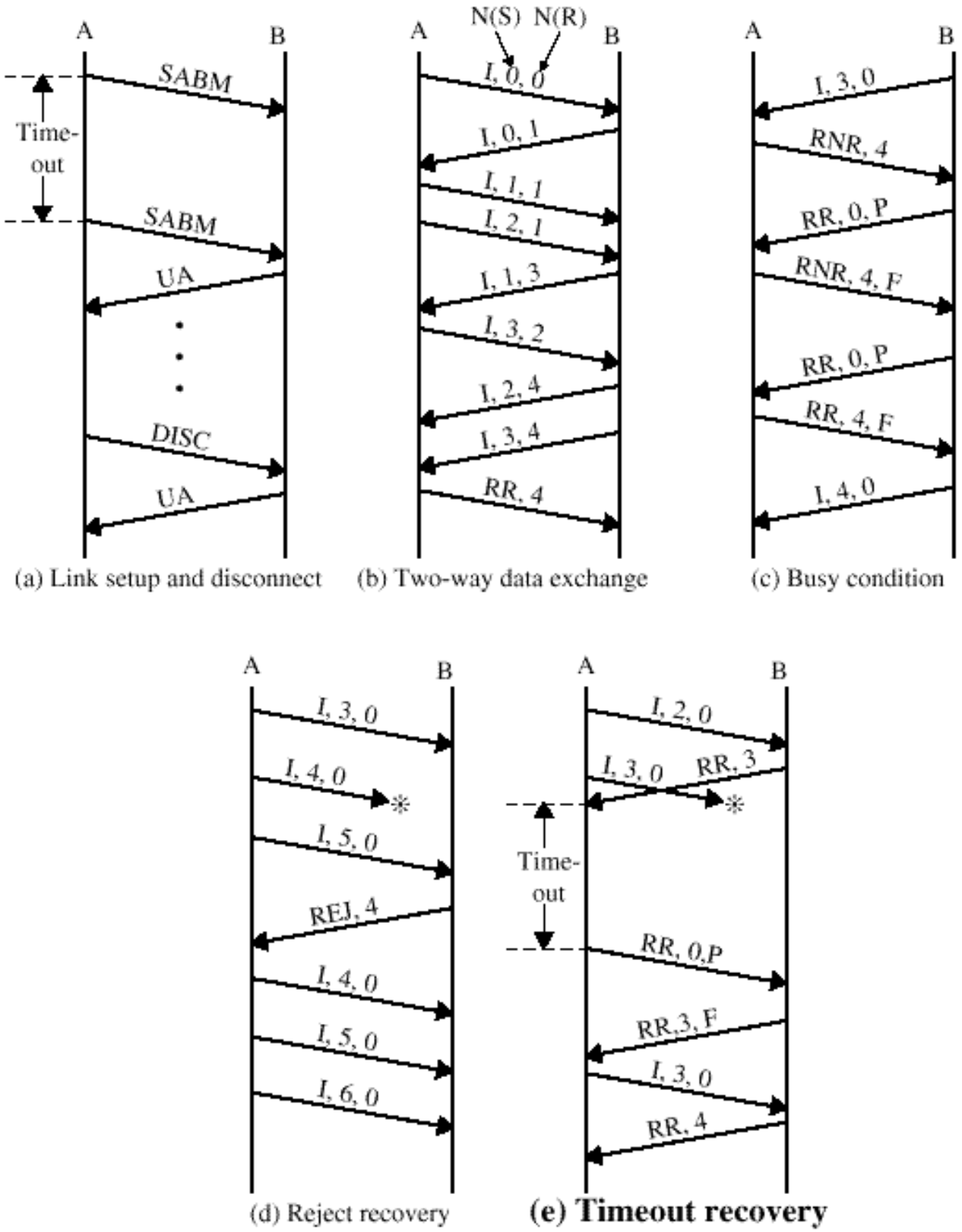
(c) An inverted merges two frames

- Flag field
  - 8 bits (01111110)
  - "Bit stuffing" is used for data transparency
  - Bit stuffing: whenever five 1's are transmitted, extra zero is inserted
- Address field
  - 8 bits. If needed longer address, the LSB can be set to zero and the address field is then assumed to be 8 bits longer.
  - All 1's indicates this is a broadcast frame
- Control field
  - 8 bits. If the 1st bit is 0, then "information frame", Otherwise, 10 indicates "supervisory frame" and 11 indicates "unnumbered frame"
  - Information frame
    - two 3 bits sequence numbers, P/F bit
    - sequence number can be extended to 7
    - used to send data
  - Supervisory frame: bits 3 and 4 determine types
    - 00: "Receive Ready". Used to ACK
    - 01: "Reject". Essentially a NACK in Go-back-N
    - 10: "Receive Not Ready". Indicates busy condition
    - 11: "Selective Reject". Request for a single frame retransmission
    - Bit 5 is P/F. Bits 6,7,8 are sequence number
  - Unnumbered frame:
    - More control functions. Has a variety of purposes, but most often used for establishing the link setup and disconnect.
    - Sets up the data transfer mode, sequence number size. Also used to reset the link and other miscellaneous stuff.

HDLC Commands and responses

Name	Command/ Response	Description
Information (I)	C/R	Exchange user data
Supervisory (S)		
Receive ready (RR)	C/R	Positive acknowledgment; ready to receive I-frame
Receive not ready (RNR)	C/R	Positive acknowledgment; not ready to receive
Reject (REJ)	C/R	Negative acknowledgment; go back N
Selective reject (SREJ)	C/R	Negative acknowledgment; selective reject
Unnumbered (U)		
Set normal response/extended mode (SNRM/SNRME)	C	Set mode; extended = 7-bit sequence numbers
Set asynchronous response/extended mode (SARM/SARME)	C	Set mode; extended = 7-bit sequence numbers
Set asynchronous balanced/extended mode (SABM, SABME)	C	Set mode; extended = 7-bit sequence numbers
Set initialization mode (SIM)	C	Initialize link control functions in addressed station
Disconnect (DISC)	C	Terminate logical link connection
Unnumbered Acknowledgment (UA)	R	Acknowledge acceptance of one of the set-mode commands
Disconnected mode (DM)	C	Terminate logical link connection
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 status
Test (TEST)	C/R	Exchange identical information fields for testing
Frame reject (FRMR)	R	Reports receipt of unacceptable frame

- Examples of Operation



(a) Link setup and disconnect (b) Two-way data exchange (c) Busy condition

(d) Reject recovery **(e) Timeout recovery**