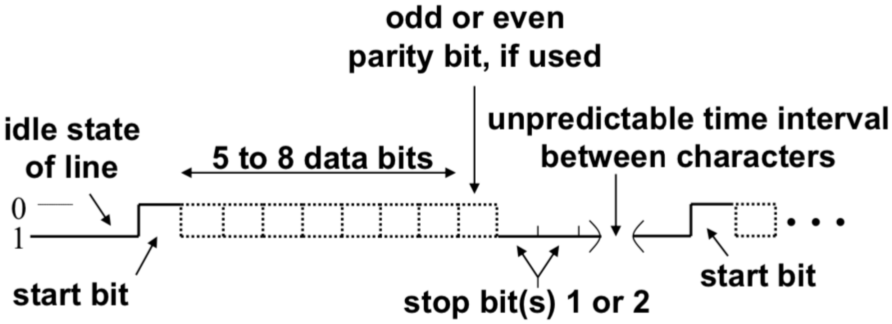
C2 Data Link Layer

1. DLL Flow Control (4 main functions)

1.1 Framing (not examinable)

- Encapsulate each network layer datagram within a link layer frame before transmission over the link
    - Byte (Character) Oriented
        - Information framed into fixed 8-bit (1 byte) basic unit with some used for signalling (protocol control)
        - Frame format: [`start bit`, `data bits`, `parity check bit`, `stop bit`]

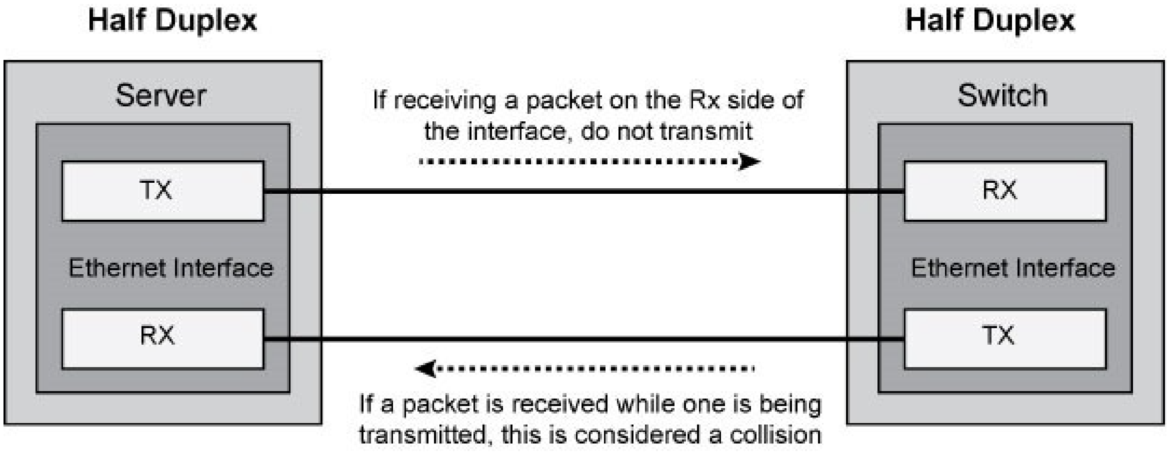


    - Bit Oriented (HDLC)
        - Flag used to frame bits sent
        - Header / Trailer (tail) used to describe content of a frame (which may be used for control)

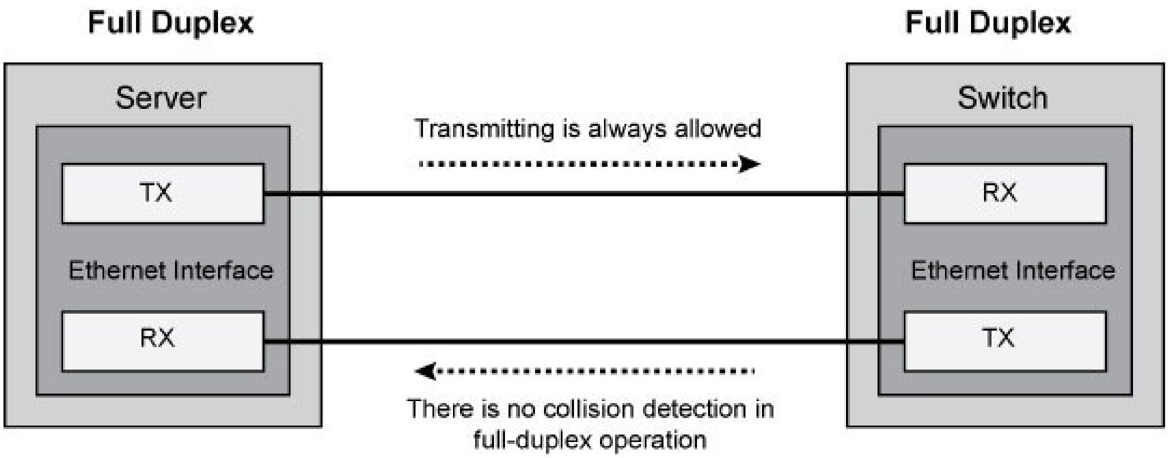
1.2 Link Access

- MAC protocol specifying rules by which a frame is transmitted into the link
    - Objective is to determine who gets to transmit at when on a link
    - Topology (physical arrangement of stations)
        - Point-to-Point: pairs of hosts are directly connected
        - Broadcast: all stations share a single channel (only one packet needs to be sent out)
        - **hosts / stations / machines are used interchangeably**
    - Duplexity
        - Half-Duplex: only one party transmit at a time
        - Full-Duplex: simultaneous transmission and reception allowed

1. Half-Duplex
The following figure shows a half-duplex link. In a half-duplex environment, the receiving (Rx) line is monitored. If a frame is present on the Rx link, no frames are sent until the Rx line is clear. If a frame is received on the Rx line while a frame is being sent on the transmitting (Tx) line, a collision occurs. Collisions cause the collision error counter to be incremented – and the sending frame to be retransmitted – after a random back-off delay.



2. Full-Duplex
The following figure shows a full-duplex link. In full-duplex operation, the Rx line is not monitored, and the Tx line is always considered available. Collisions do not occur in full-duplex mode because the Rx and Tx lines are completely independent.

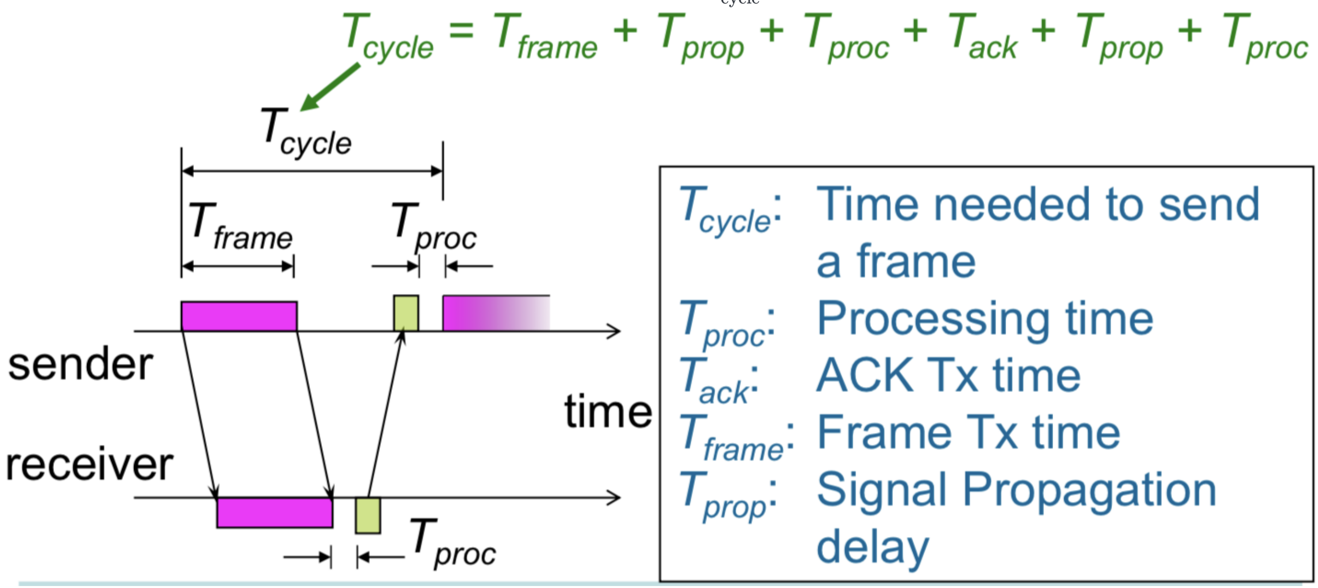


1.3 Flow Control

- Control of data flow to ensure sender does not overwhelm the receiver with data (i.e. buffers at receiver do not get overflowed)
  - No frame error (assumption for this chapter)

Throughput (Link Utilisation)

Link Utilisation $= \frac{\text{Time that link carries useful information}}{\text{total time}} = \frac{T_{\text{frame}}}{T_{\text{cycle}}}$



Stop-and-Wait

1. `A` packs binary information into frame and sends to `B`
2. `A` waits for `ACK` (acknowledgement) from `B`
   - `ACK1` will be sent after receiving frame 0, to indicate receiver is ready for frame 1
3. `B` sends `ACK` **after** receiving frame
4. `A` receives `B`'s `ACK` and repeats 1

- **Throughput**
  - Assume that input is saturated, no error, and ignore T_{ack}, T_{proc}
  - T_{prop} = time taken for one bit to travel from sender to receiver (Distance/Speed)
  - T_{frame} = time taken for signal to transmit given frame length (Length/Rate)
  - $T_{cycle} = T_{frame} + 2T_{prop}$ (propagation delay for Frame and `ACK`)
  - $\text{Then } U = \frac{T_{frame}}{T_{frame}+2T_{prop}} = \frac{1}{1+2a}$
    - U **is ratio**
  - $\text{Where } a = \frac{T_{prop}}{T_{frame}}$ (normalised propagation delay)
- **Disadvantages**
  - Lost frame / ACK $\implies$ longer waiting time expected
    - use TIMEOUT control in sender to fix
  - If normalised propagation delay is long, sender must wait long time before next transmission can be performed
    - Link Utilisation is low
    - use Buffers at sender / receiver (sliding window)

Sliding Window

- Allows multiple frames to be in transit
- Both Sender and Receiver have buffer of size N
  - Sender can send up to N frames without needing `ACK` from Receiver
  - Each `ACK` is numbered to correspond to an expected frame
- Sequence number is bounded by field size (k bits)
  - frames are numbered modulo 2^k
  - sequence number in range $[0, 2^k - 1]$
- Sender
  - Maintains a window which contains the frame numbers that can be transmitted
  - Window shrinks from left side (lower bound / trailing edge) as frames are sent
  - Window expands from right side (upper bound / leading edge) as `ACK`s are received
- Receiver
  - Maintains a window which determines the available buffer (?) size
  - Window shrinks from left side as frames are received
  - Window expands from right side as `ACK`s are sent
- **Window Size and Sequence Number size**
  - For $N = 3$ with $k = 1$ bit sequence number, when sender sends 0,1,0 to receiver, receiver does not know whether the second 0 is a new frame or retransmitted frame of the first 0. Receiver sends `ACK` for received frame 1, sender does not know whether receiver wants a retransmission of first 0 or second 0
    - Need to follow the rule $N \leq 2^k$
- **Features (not examined)**
  - Receiver can acknowledge received frames without allowing further transmission by sending `RNR` (Receive Not Ready) instead of `ACK`, and send `ACK` to resume transmission
  - `ACK` can be piggybacked on data frames in the reverse direction (**dk what it means, waiting for lecture**)
- **Performance**
  - depends upon (error free operation)
    - Parameter a (normalised propagation delay)
    - Window Size N
  - Assume T_{ack} and T_{proc} are negligible
  - Frame Transmission time = 1

- 2 cases:
    1. $N \geq 2a + 1$ - can transmit continuously without exhausting window
        - Link Utilisation = 1.0
    2. $N < 2a + 1$ - window exhausted at $t = N$, station cannot send additional frames until $t = 2a + 1$
        - Link Utilisation = $\frac{N}{1+2a}$

1.4 Reliable Delivery

- moving each network-layer datagram across the link without error

2. Error Control

- Useful for detecting and correcting errors that occur in frame transmission
- **Frame Error**
    - Lost Frame: receiver does not receive a frame / header corrupted (leads to unrecognisable frame)
    - Damaged Frame: receiver receives a frame, but some of the bits are in error

2.1 Error Detection

Parity Check

- A single bit is appended to the original message (7 + 1 bits) to attempt to satisfy a pre-determined parity scheme agreed by both R and S
    - Even parity: total number of 1s is even
    - Odd parity: total number of 1s is odd
    - **Both parities include the parity bit**, can only detect odd numbers of error

Cyclic Redundancy Check (not examinable)

- Multiple parity bits are appended to the original message
    1. Sender prepares to send message (M bits), appends k bits (CRC to the message)
    2. Receiver receives packet from Sender (M+k bits), passes it through CRC function
    3. CRC function will give a remainder, 0 indicates no error

2.2 Error Correction Techniques

Forward Error Correction (FEC) (not examinable)

- Sends more redundant bits in message

Automatic Repeat Request (ARQ)

- Retransmission after timeout

- Sender expects ACK from Receiver, if ACK not received within a predetermined time duration, Sender retransmits the frame

- Retransmission when requested
  - Receiver replies a negative ACK to inform Sender about an error, Sender then retransmits the corrupted frames accordingly

- **Stop-and-Wait ARQ**
  - Sender transmits a single frame and waits for ACK
  - Receiver sends an ACK if frame received correctly, otherwise, discard and do nothing or send a negative ACK (NAK)
  - If Sender:
    - receives ACK: transmit next frame
    - receives NAK: retransmit same frame
    - nothing received: retransmit same frame
    - receives wrong ACK (due to damage etc): transmitter timeouts and retransmit same frame, receiver ends up with duplicate frames, need to discard one
  - Performance

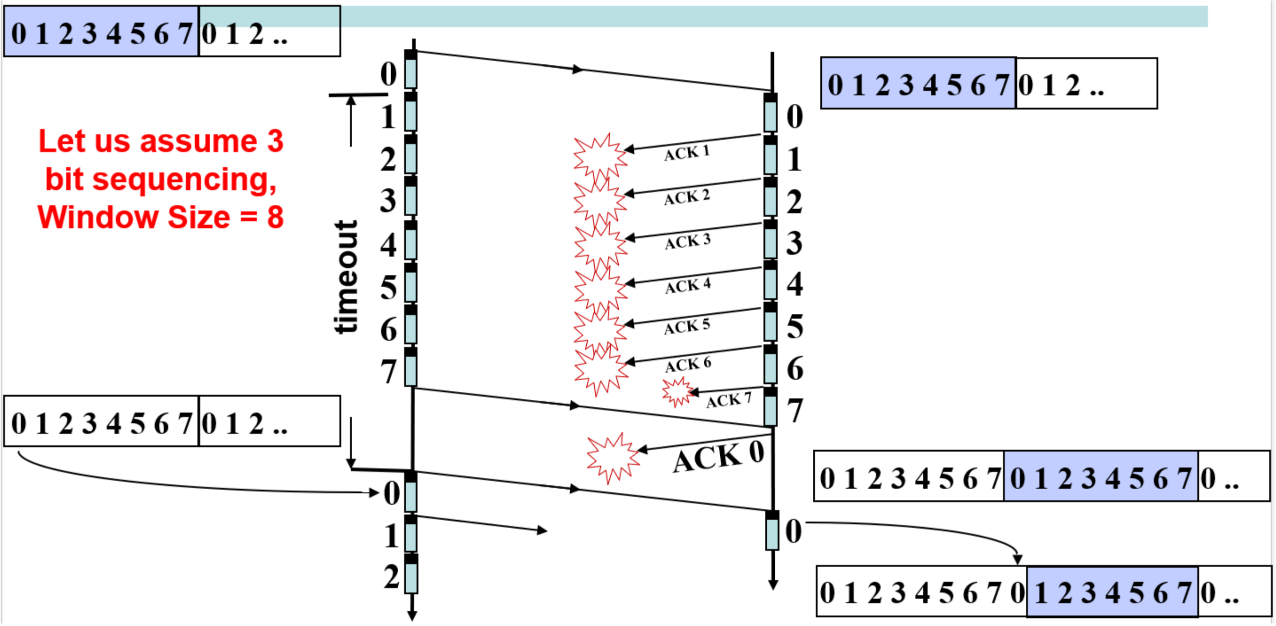
$$\text{Throughput (U) (Link Utilization)} = \frac{\text{The time that the link carries useful information}}{\text{The total time}} = \frac{T_{frame}}{T_{cycle}}$$

$$\begin{aligned} U_{SaW}^{ARQ} &= \frac{1}{1+2a}\Pr\{\text{no error}\} + 0 \cdot \Pr\{\text{frame error}\} \\ &= \frac{1}{1+2a}(1-P) + 0 \cdot P \\ &= \frac{1-P}{1+2a} \end{aligned}$$

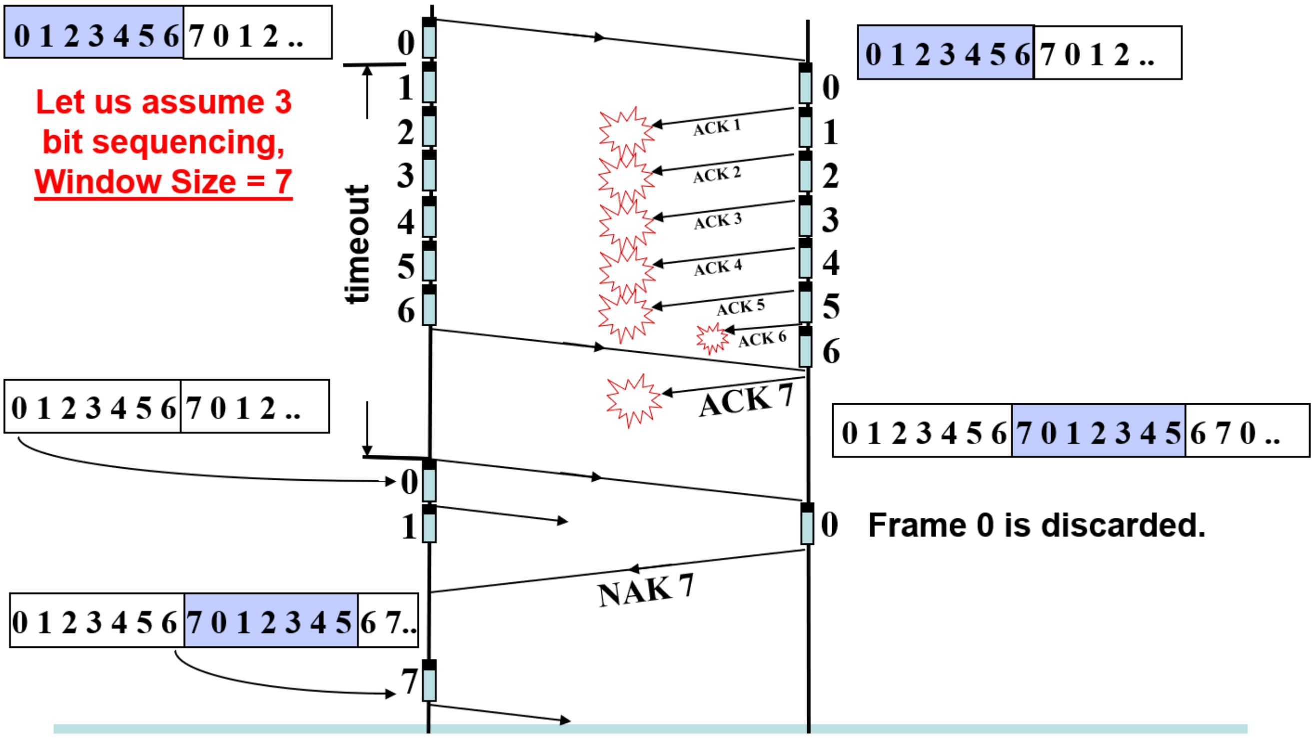
P: Frame loss probability
a: normalized prop. delay

- **Sliding Window - Go-back-N ARQ**
  - Sender transmit frames sequentially based on sliding window
  - Receiver expects F_1 from Sender, but receives a non-F_1 (erroneous) frame and sends NAK_1 to Sender (all received frames will be discarded until F_1 arrives)

- Sender receives NAK_1 and go back to transmitting F_1



  - As S sends 8 frames over to R, S's window size becomes 0 (Left side)
  - As R sends ACKs, R's window size increases (Right side)
  - However, ACK0 gets lost in transmission, resulting in timeout, causing S to send frame 0 again
  - R does not know if ACK_0 has been received by S, and proceeds as if nothing happened
  - **Reason why maximum window size allowed is** $2^k - 1$



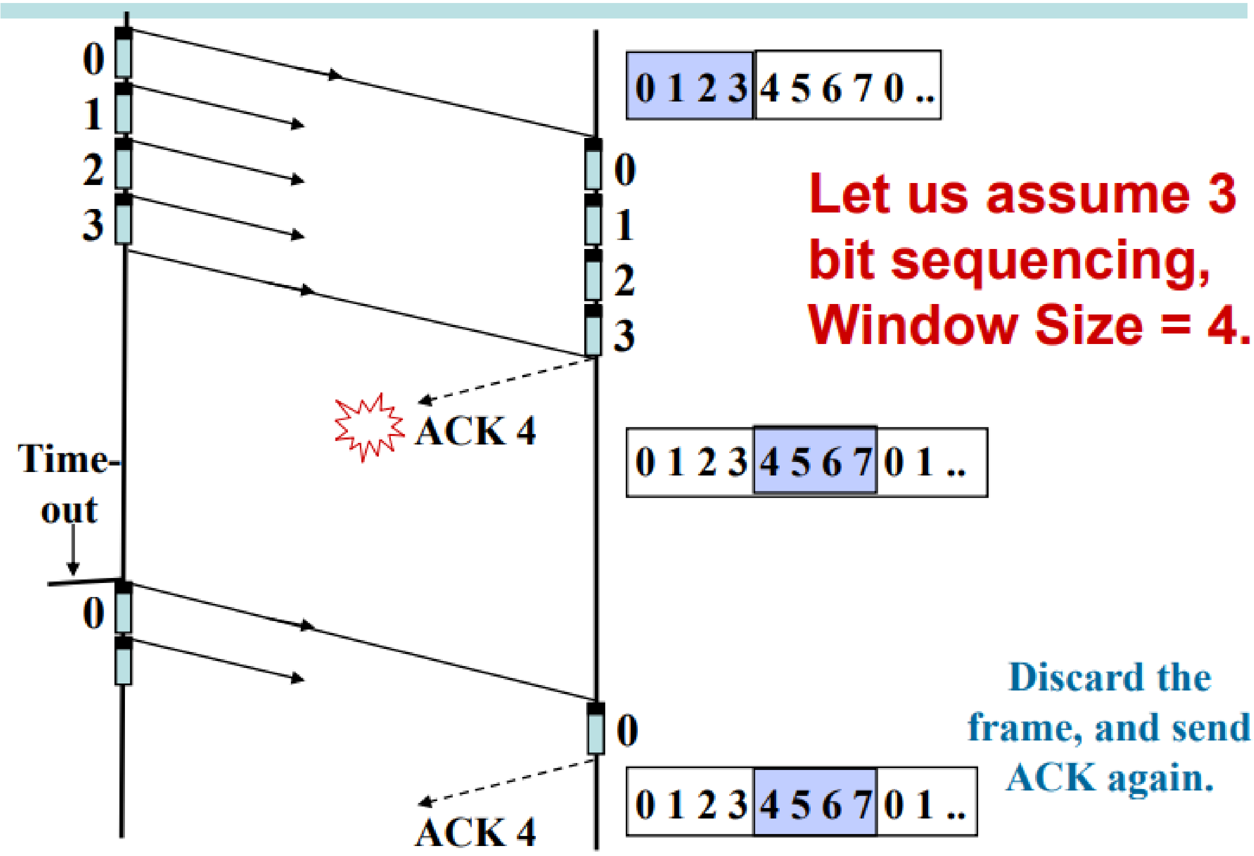
- S sends 7 frames, shrinking S's window size to 0
- R receives 7 frames, sends ACK requesting for frame 7, which gets lost in transmission.
- R doesn't receive ACK 7, resents frames 0 to 6 again
- S receives frame 0, sends NAK_7 to R to signal wrong frame received
- R corrects window and sends frame 7

- Performance (with assumptions)

- T_{ack} and T_{proc} are negligible
- frames are never completely lost
- ACKs and NAKs are never in error
- each frame is acknowledged immediately
- S always has frames to send

$$U_{GBN}^{ARQ} = \begin{cases} \dfrac{1-P}{1+2aP} & N \geq 2a+1 \\ \dfrac{N(1-P)}{(1-P+NP)(1+2a)} & N < 2a+1 \end{cases}$$

The equations in this slide are not examinable.

- **Sliding Window - Selective Reject ARQ**
  - Only rejected frames and timed out frames are retransmitted
  - R informs S of rejected frame n by sending NAK_n
  - Frames received by R after rejected frame gets buffered until R receives valid copy of rejected frame, after which, frames are put in proper order and passed to higher layer
  - Minimises retransmission but R requires more complex buffer management
  - 

    - R sends 4 frames (0,1,2,3) to S, S sends ACK_4 which gets lost in transmission (buffer S now 4,5,6,7)
    - R assumes time out and resends same 4 frames
    - S receives unexpected frame (0), resends ACK_4

- max window size for this error correction is 2^{k-1}

- Performance

  P: Frame loss probability
  a: normalized prop. Delay

  Since frame loss prob for each tx is independent, in 1+2a cycle, we expect N transmissions, each with prob P of failure due to errors.

  $$U = \frac{N\overline{F}}{1+2a}, N < 2a+1$$

  where $\Pr\{F=n\} = \begin{cases} P, n=0 \\ 1-P, n=1 \end{cases}$

  and $\overline{F} = 1-P$.

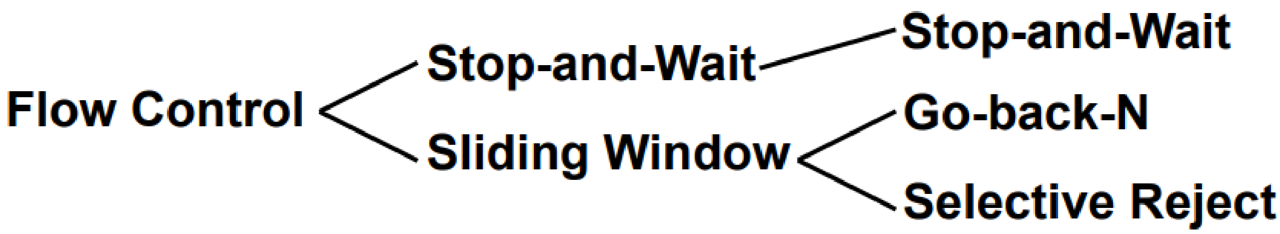
  Hence $U_{Selective\ reject} = \frac{N(1-P)}{1+2a}$

  Examinable:

  $$\boxed{U_{SR}^{ARQ} = \begin{cases} 1-P & N \geq 2a+1 \\ \frac{N(1-P)}{1+2a} & N < 2a+1 \end{cases}}$$

  Setting P=0 reduces the above to that of Sliding Window.

- Which ARQ variant to use depends on the flow control method used



3. Important formulas

Stop-and-Wait ARQ

$$U_{Stop-and-Wait} = \frac{1-P}{1+2a}$$

Go-back-N ARQ

$$U_{Go-back-N} = \begin{cases} \text{[illegible]} & N \geq 2a+1 \\ \dfrac{\text{[illegible]}}{(1-P+NP)(1+2a)} & N < 2a+1 \end{cases}$$

Sliding Window (no errors)

$$U_{Sliding\ \ Window} = \begin{cases} 1 & N \geq 2a+1 \\ \dfrac{N}{1+2a} & N < 2a+1 \end{cases}$$

Selective Reject ARQ

$$U_{Selective\ \ reject} = \begin{cases} 1-P & N \geq 2a+1 \\ \dfrac{N(1-P)}{1+2a} & N < 2a+1 \end{cases}$$

P: **frame error probability**

a: **normalized propagation delay**

N: **window size**

U: **Channel Utilization (between 0 and 1)**

(X is not examinable)