Chapter 3 The Data Link Layer

3.1 Data Link Layer Issues

3.2 Error Detection and correction (Revise numericals from DCOM)

3.3 Elementary Data Link Protocols, Sliding Window protocols Example Data Link Protocols: HDLC: High-Level Data Link Control, The Data Link Layer in the Internet

---

Data Link Layer Design Issues (Functions of Data Link Layer)

- Addressing
- Link Management
- Framing
- Frame Synchronization
- Flow control
- Error control
- Access control

Framing Methods

- Character count Method
- Character oriented framing & Character Stuffing
- Byte oriented framing & Byte Stuffing
- Bit oriented framing & bit Stuffing

Frame Synchronization

- Asynchronous frame format
- Synchronous frame format.

Error control
- Error detection → checksum
- Error correction

- What is error control? Why it is required?
- Types of errors
  - Content errors
  - Flow integrity errors
  - Single bit error
  - Burst errors.
- Error control mechanism
  - ARQ
- Elements of ARQ
  - Timer
  - Information (I-frame)
  - Control frames (ACK & NAK)
    - +ve Ack
    - -ve Ack
- Types of ARQ
  - Sliding window ARQ
    - Go-Back-N ARQ
    - Selective repeat ARQ
  - Stop & wait ARQ.
- Transmission efficiencies of
  - stop & wait ARQ
  - Go back N ARQ
  - Selective repeat ARQ.
- Numericals on ARQ schemes.

Flow control
- Sliding window flow control

Piggybacking ?
- Advantages ?
- Drawbacks ?

Data link layer protocols
- HDLC (High Level Data Link control)
- SLIP (Serial Line IP)
- PPP (Point to point)

HDLC
- features
- Types of stations
  - Primary
  - Secondary
  - Combined
- Modes of operation
  - Asynchronous response Mode (ARM)
  - Asynchronous balanced Mode (ABM)
  - Normal response Mode (NRM)
- Types of frames
  - Information (I-frames)
  - control frames (~~ACK/NAK~~)
    - Supervisory frames (S-frames)
    - Unnumbered frames (U-frames)
- Transparency in HDLC
- Bit stuffing & destuffing in HDLC.

- Numericals of HDLC frame formats.
- Data Link Layer in the Internet.

SLIP

- Serial Line IP
- used for Internet connection
- over dial up connection using MODEM
- works with IP n/w's only
  (does not work with non IP n/w's)
- ~~Does~~ not support flow control & error control
- Does not provide authentication.
  (users does not ~~at~~ understand with whom they are communicating)
- It is not an approved Internet protocol.

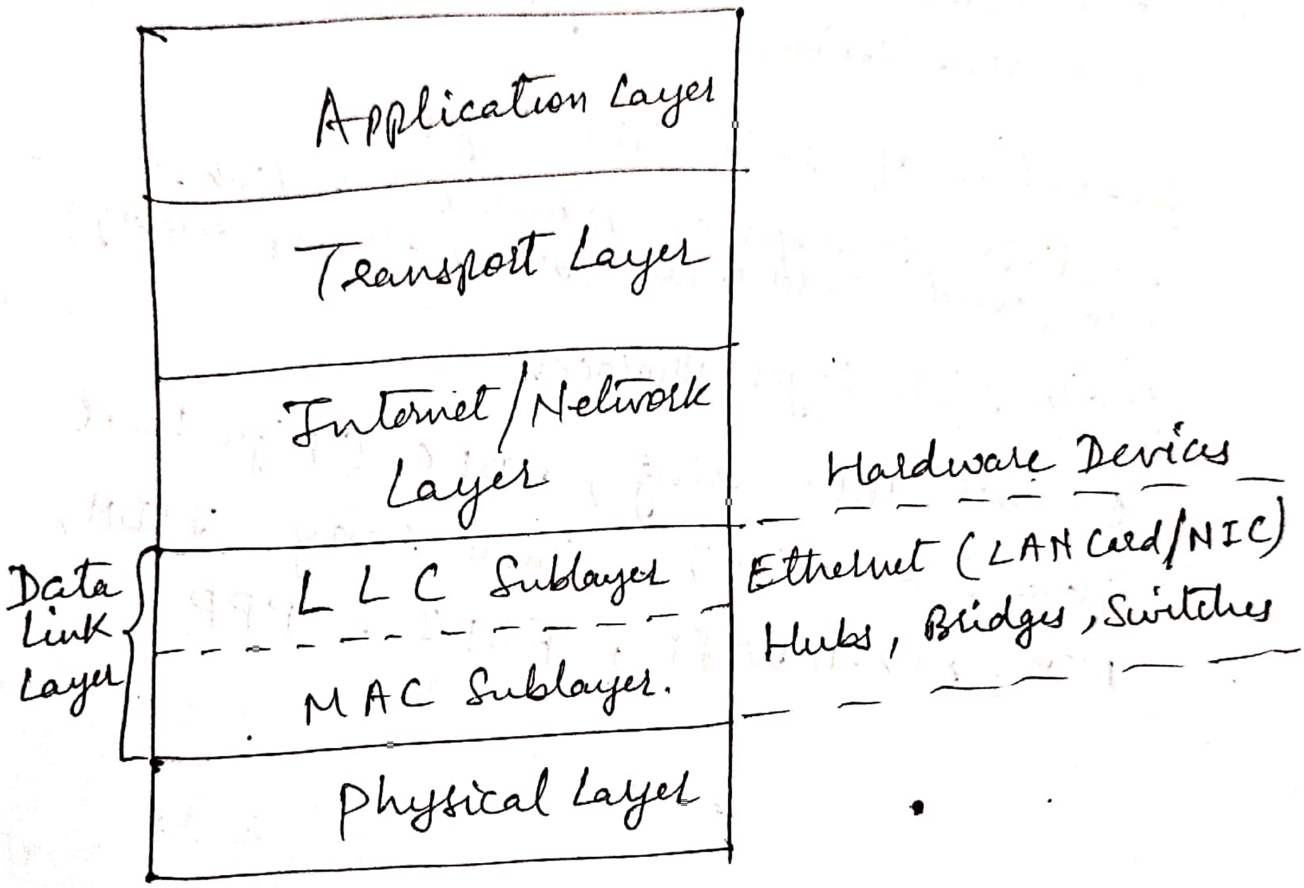
Scanned with CamScanner

Chapter 3 Data Link Layer Services and Protocols:

Chapter Note

3.1: Link Layer and its Services, Ethernet, hubs, bridges and Switches.

Link Layer and its Services:



TCP/IP Architecture

Data Link Layer:

- Layer 2 (L2) of OSI model & TCP/IP Stack.
- It has two sublayers LLC and MAC sublayer.
- LLC provides access between upper & lower layers.
- MAC sublayer provides access to various media (Ethernet, FDDI, ATM etc)
- Unit of communication is frame. • Supports Node to Node or Hop by Hop Communication

Functions of Data Link Layer

- Framing
- Flow control and Error control
- Access control
- Hardware addressing (MAC address/ physical address)
- Switching
- Retransmission

Services of Data Link Layer:

- Reliable transfer of frames over a link.
- To provide node to node (or hop-to-hop delivery)

Data Link Layer Protocols

Ethernet, Token ring, HDLC (High Level Data Link control), Frame relay, ISDN, ATM, 802.11 WiFi, FDDI, PPP

~~Wireless~~

Wired LANs : Ethernet

LAN (Local Area Network)

- computer Network restricted by geographic area such as a building or a campus (radius upto 4Km)
- used for resource sharing
- LANs can be connected to WANs (Wide Area Networks) or the Internet.

- LAN Technologies are :
  - → Ethernet
  - → Token Ring
  - → Token Bus
  - → FDDI
  - → ATM LAN.

* Ethernet is the most dominant Technology.
- Ethernet belongs to IEEE 802 standard.

Data Link Layer

LLC Sublayer

- provides access between higher & lower layers in TCP/IP or OSI stack
- provides framing
- provides flow control, error control

MAC Sublayer

- provides access to various media using various protocols..

LLC Sublayer PDU (protocol Data Unit)

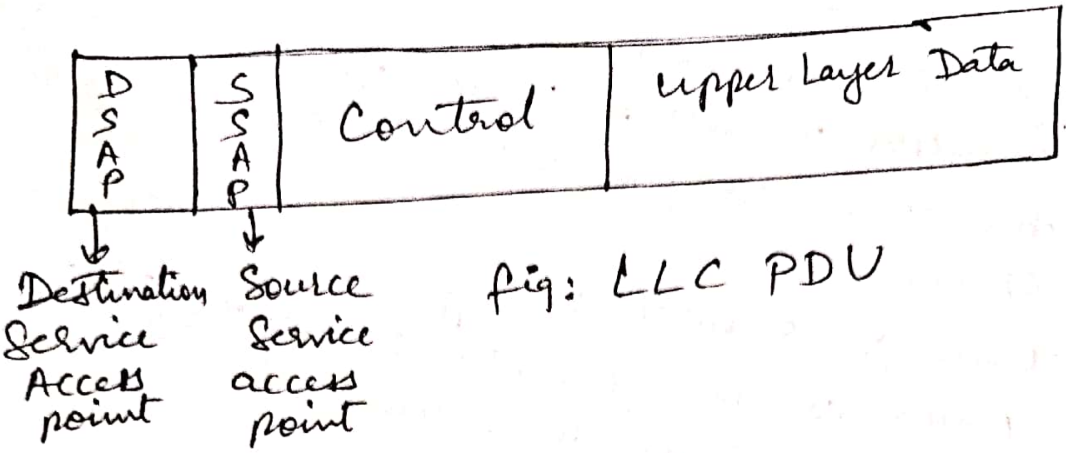


fig: LLC PDU

MAC Sublayer PDU

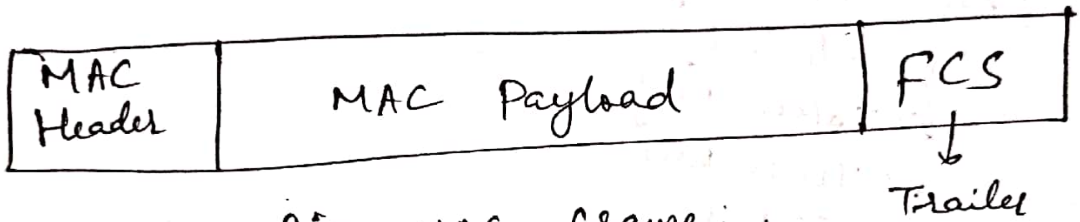
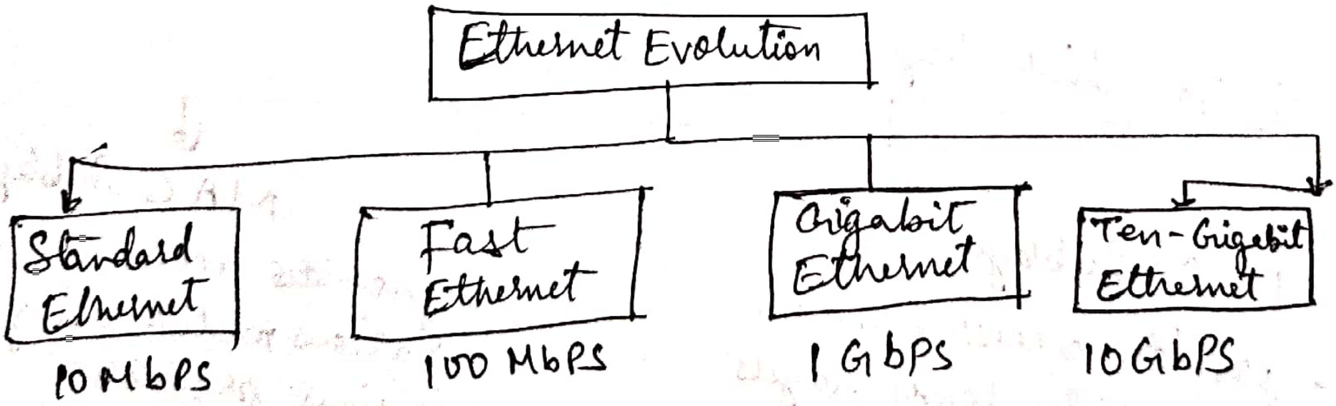


fig: MAC frame.

Standard Ethernet

Ethernet Evolution through four generations



Mac Sublayer:

- MAC sublayer governs the operation of the access Method.
- It also frames data received from the upper layer and passes them to the physical layer.

Frame Format

802. MAC frame.

Preamble (56 bits of alternating 1's & 0's)	SFD 10101011	Destination address	Source address	Length or Type	Data & Padding	CRC
7 Bytes	1 Byte	6 Bytes	6 Bytes	2 Bytes		4 Bytes (Trailer)

next field is the dst addr.

14 Bytes Header

4 Bytes Trailer

Total 18 Bytes overhead.

Ethernet:

unreliable medium as it does not have any mechanism for acknowledging received frames.

- Preamble
  - not part of a frame, added at physical layer.
  - 7 bytes (56 Bits)
  - Added to synchronize frames.

- Start Frame Delimiter (SFD)
  - 1 Byte (10101011); signals the beginning of a frame.

- SFD warns the station (device) that this is the last chance for synchronization.
- Last Two bits are 11 and alerts the receiver that the next field is the destination address.

- <u>Destination Address</u> (DA, 6 Bytes)
  - 6 Bytes
  - contains physical address (MAC addr, H/w addr) or link address) of the destination station or stations to receive the packet.
  - ~~Source~~.

- Source Address (SA, 6 Bytes)
  - 6 Bytes
  - contains physical address of the sender of the packet.
  - Length or Type

- <u>Length or Type</u>
  - used as Type by Ethernet to specify the upper layer protocol using the MAC frame
  - used as Length to define number of bytes in the data frame.

- <u>Data</u> → carries data encapsulated from the upper layer protocols.
  - min ~~46~~ 46 Bytes & maximum 1500 bytes of data.

- CRC

CRC-32 is used for error detection & correction.

Ethernet Addressing

- Each station (computer or device) on an Ethernet n/w (such as PC, workstation or printer) has its own network Interface Card (NIC).
- The NIC fits inside the station and provides the station with a 6-byte physical address.
- The Ethernet address is 6 Bytes (48 bits) normally written in hexadecimal notation, with a colon between the bytes.

e.g. Ethernet address is

$$\underbrace{AB:12:C4:F1:23:E5}$$

6 Bytes = 12 Hexadecimal digits
= 48 bits.

- Unicast, Multicast and Broadcast addresses

↳ Source (Sender) address is always unicast

↳ Destination (Receiver) address can be unicast, multicast or broadcast.

↳ If the LSB of the first byte in a destination address is 0, the address is unicast; otherwise, it is multicast.

→ If all bits in a destination address are 1 then it is a broadcast address

Unicast → one to one

Multicast → one to a group of computers (stations) in LAN.

Broadcast → one to all stations (computers) in LAN.

Q. Define the type of the following destination addresses:

(a) 4A : 30 : 10 : 21 : 10 : 1A

Ans. ~~A in Binary~~
consider first Byte ; 4A

4 A
↳ 1010 ↳ Last bit is '0'
So the address is unicast

(b) 47 : 20 : 1B : 2E : 08 : EE

Ans consider first Byte; 47

4 7
↳ 0111 ↳ Last bit is '1'

So the address is multicast

(c) FF : FF : FF : FF : FF : FF

↳ All Bits are 1's

∴ The address is Broadcast.

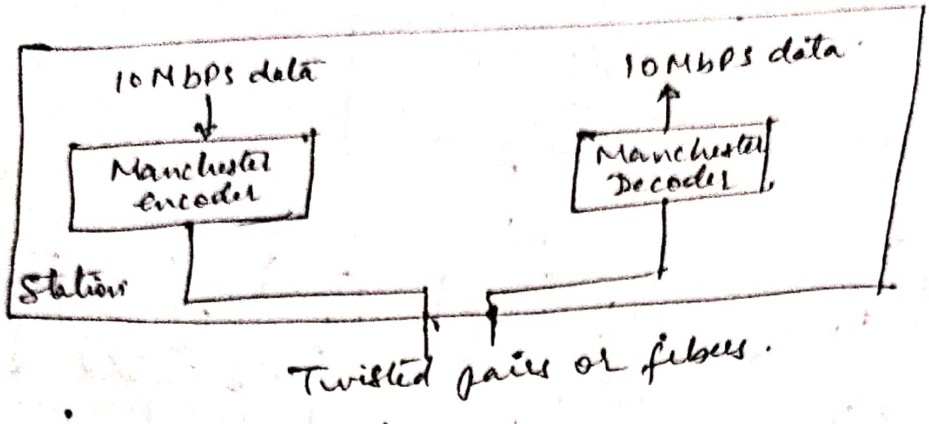
Physical Layer of Standard Ethernet

Standard Ethernet Common Implementations

- 10 Base5
  - 500m
  - 10 Mbps
  - Baseband Signalling
- 10 Base2
  - 10Mbps
  - 200m
  - Baseband
- 10 Base-T
  - 10Mbps
  - Baseband
  - Twisted Pair
- 10 Base-F
  - Baseband
  - 10Mbps
  - Fiber

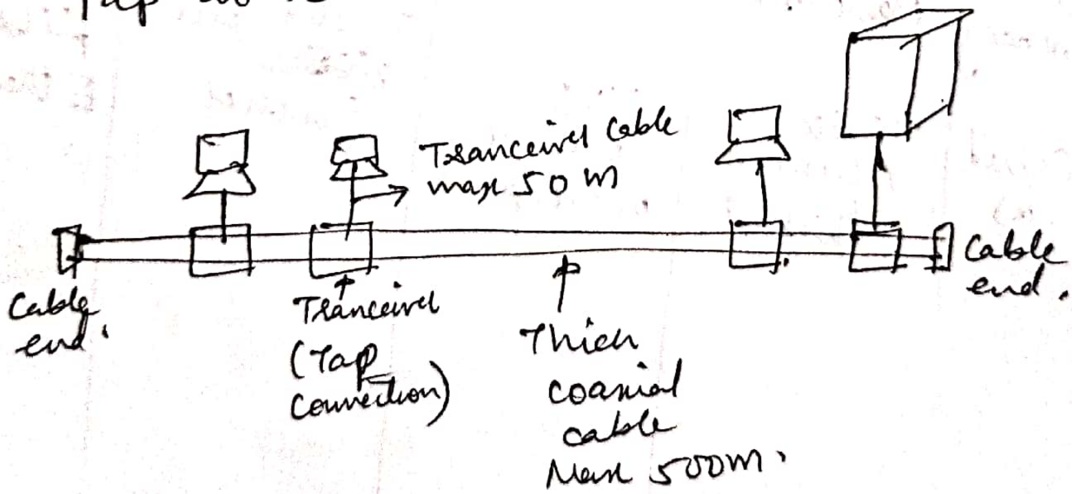
Topology	Bus	Bus	Star	Star
Physical Media	Thick coaxial	Thin coaxial	UTP (2 pairs)	Fiber (2 in number)
Maximum Length	500 m	185 m	100 m	2000 m
Line encoding	Manchester	Manchester	Manchester	Manchester
Called as	Thick Ethernet or Thicknet	Thin Ethernet or cheapernet	Twisted pair Ethernet	Fiber Ethernet

Encoding in Standard Ethernet:



10 Base 5: Thick Ethernet

- 10 Base 5, thick Ethernet or Thicknet:
- Cable size of Garden hose & too stiff to Bend with hands.
- Uses Bus topology; wherein devices are connected via transceivers (TX/RX) using Tap to a thick coaxial cable.



10 Base 5 → 500 m.

10 → 10MbPS

Base → Baseband (digital)

LAN Design

- As a Network Engineer, you may be required to select and produce hardware for the network.
- Improper n/w device (equipments) may lead to slow LAN & decreased user productivity.
- A Hierarchical Designed LAN is designed into different functional layers.

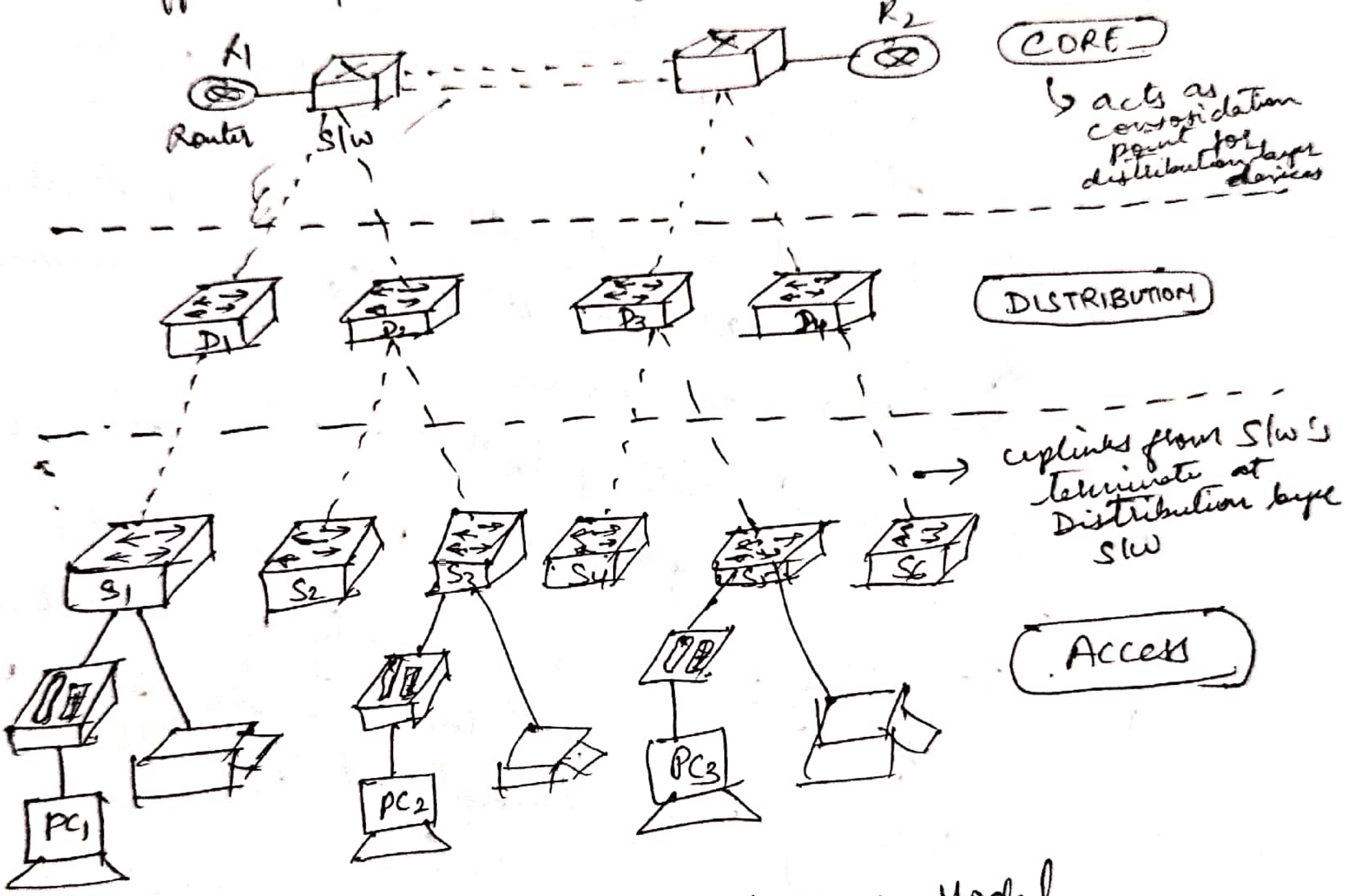


fig: The Hierarchical Network Model.

Access layer: • It enables users to access the n/w
• Extends from users work area to the intermediate Distribution facility (or IDF)

IDF: Includes switching, the horizontal cabling is avoided

Distribution: It serves as the consolidation point for the access layer switches.
• Responsible for power enforcement in VLAN Routing.

CORE : Its a high speed redundant n/w backbone
• Ties together multiple distribution layer components

Benefits of a Hierarchical Network

Scalability : Hierarchical n/w's can be expanded easily

Redundancy : Redundancy at the core & distribution level ensure path availability.

Performance : Link aggregation betⁿ levels & high performance core & distribution level switches allow for near wire speed throughout the n/w.

Security : port security at the access level & policies at the distribution level make the n/w more secure

Manageability : Consistency betⁿ switches at each level makes mgmt more simple.

Maintainability : The modularity of hierarchical design allows for the n/w to scale w/o becoming overly complicated.

Switch Performance

- Port Density ⟹ no of ports available on a single switch
- Forwarding Rates ⟹ Amount of Data a s/w comprises per second.
- Link Aggregation

Port density ⟹ 24 port s/w, 48 port s/w, modular s/w upto to 1000+ ports

Forwarding rates ⟹ the amount of data a switch can process per second.

How will You Select Switch for Access Layer, Distribution Layer & Core Layer.

Access Layer Switch Features [How to Select Access layer S/w]

1) Check whether POE is required or not to power IP phone or WAP

2) QoS should be maintained if IP Telephony is used.

3) Access layer Switches should include VLAN Capability

4) Switches should have sufficient port density as users are directly connected to the access layer.

5) Link aggregation should be considered if N/w has consistently high volume of traffic.

Distribution Layer Switch Features [How to Select Distribution Layer S/w]

1) Port Density requirement at distribution layer is not high as much that reqd at Access layer.

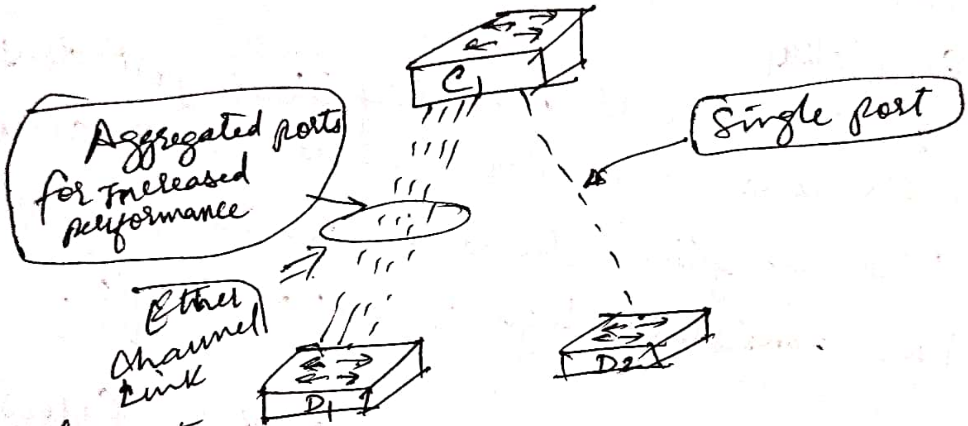
2) Layer 3 Switching & link aggregation should also be considered at the distribution layer.

3) If QoS in, VLAN is implemented at the access layer, Then They should be continued at the distribution layer.

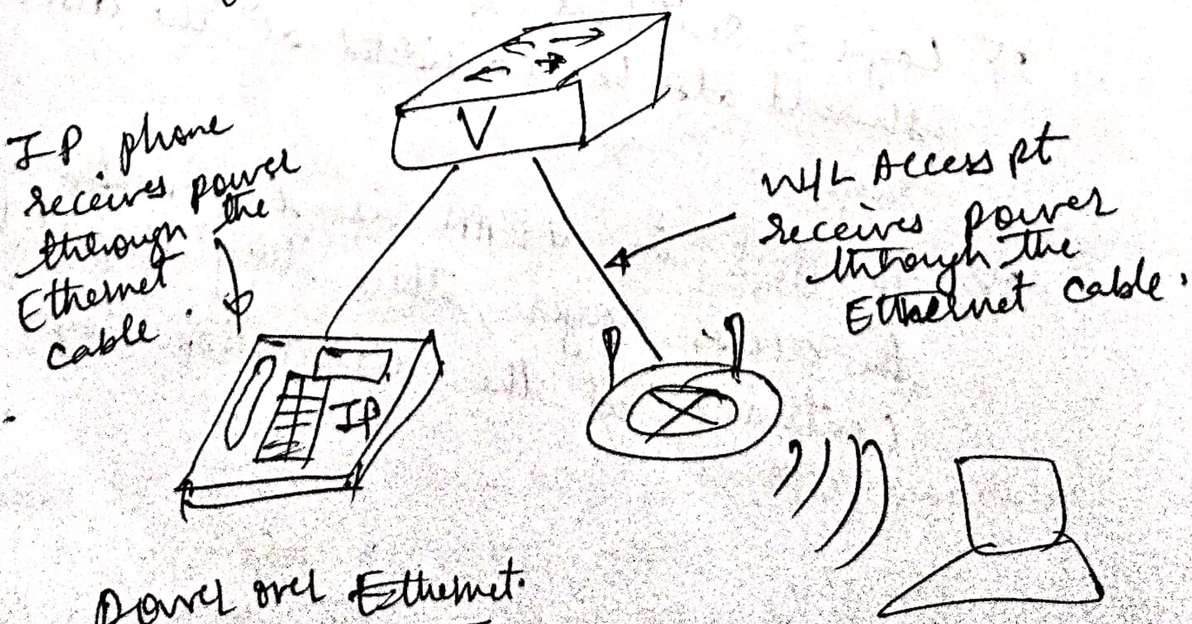
Wire Speed → Data rate of each port on the Switch (100 Mbps or 1000 Mbps)
100 Mbps ↓ Fast Ethernet
1000 Mbps ↳ Gigabit Ethernet

As Bandwidth requirements increase, Switches with higher forwarding rates are required at ~~D~~ Distribution & core layers & thus cost of S/w increases.

Link Aggregation



(4 port links are aggregated to form a single Ether channel → More BW Link from C_1 to D_1



Power over Ethernet.

Layer 3 functions

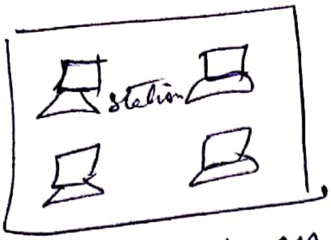
Core Layer Swich features

1) CL S/W should include 1 GBPS or even 10 GBPS to accomodate higher volumes of traffic coming towards the backbone.
2) Layer 3 Support, link aggregation capability, redundant components & very high forwarding rate should also be implemented at the Core layer.

Cisco provides all 3 types of Switches (with access, distribution or core capabilities).

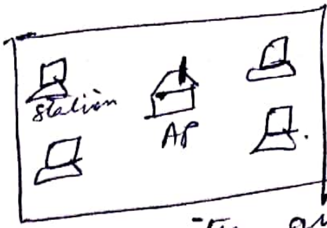
Wireless LANs

IEEE 802.11 → DLL
→ Phy layer.



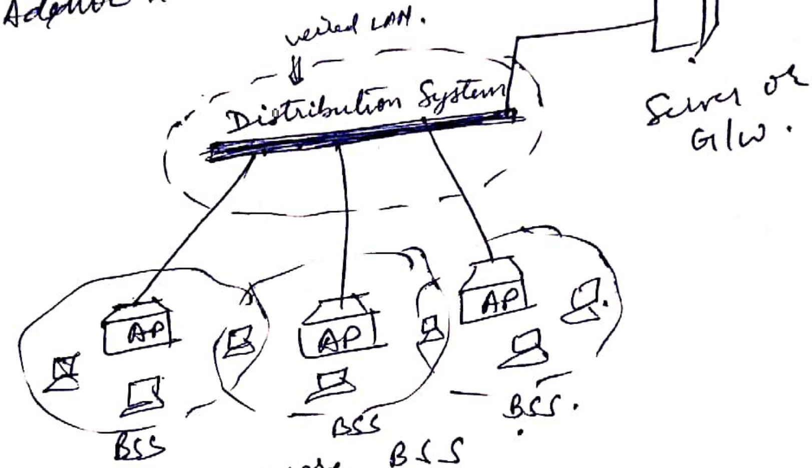
BSS w/o an AP

ⓐ Adhoc n/w



BSS with an AP

ⓑ Infrastructure n/w.



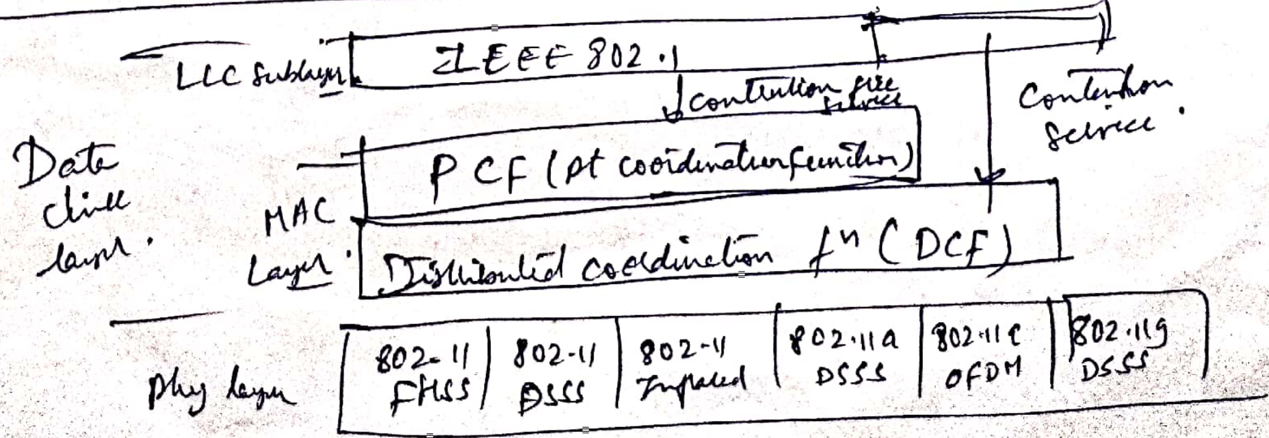
Two or more BSS

ⓒ Extended service sets (ESS)

↓ 2 stations → mobile & stationary stations
↓ within BSS
↓ e.g AP connected to LAN.

IEEE 802.11 Station

- no transition
- BSS - transition
- ESS - transition mobility.



DATA LINK CONTROL PROTOCOLS (DLC)

MU Question

Primary responsibility of DLC protocols is to manage & control the flow of frames between two stations.

HIGH-LEVEL DATA LINK CONTROL (HDLC)

MU Question
↳ Bit stuffing

HDLC

- Bit oriented protocol
- Supports both half duplex and full duplex communications.
- Bit oriented means protocol treats frames as bit streams
- It Implements the ARQ Mechanism
- Three types of stations run the HDLC protocol
  - → Primary station
  - → Secondary station
  - → Combined station
- Stations running HDLC can communicate in one of three modes:
  - → Normal response mode (NRM)
  - → Asynchronous response mode (ARM)
  - → Asynchronous balanced Mode (ABM). (used today commonly)

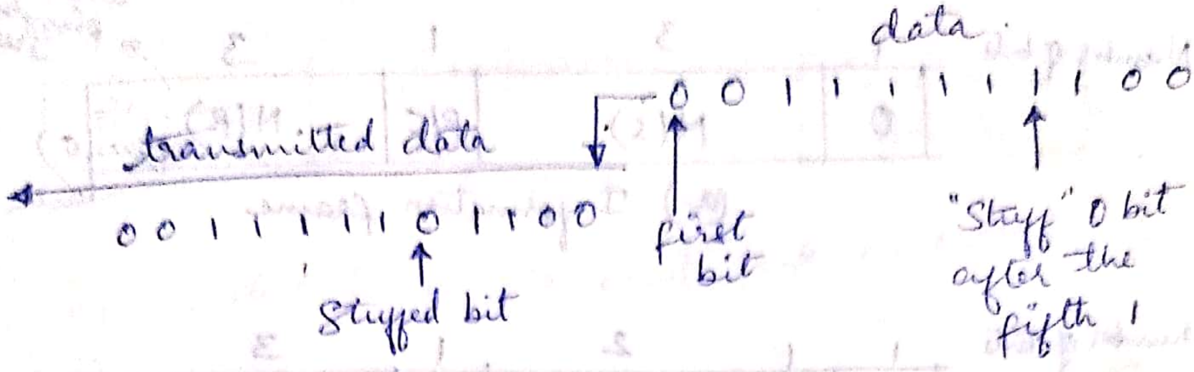
No. of bits	8	8 or 16	8 or 16 (used for flow & error control)	Variable	16 or 32	8
	Flag	Address (last bit)	Control	--- Data ----	FCS	Flag

fig: HDLC Frame Format.

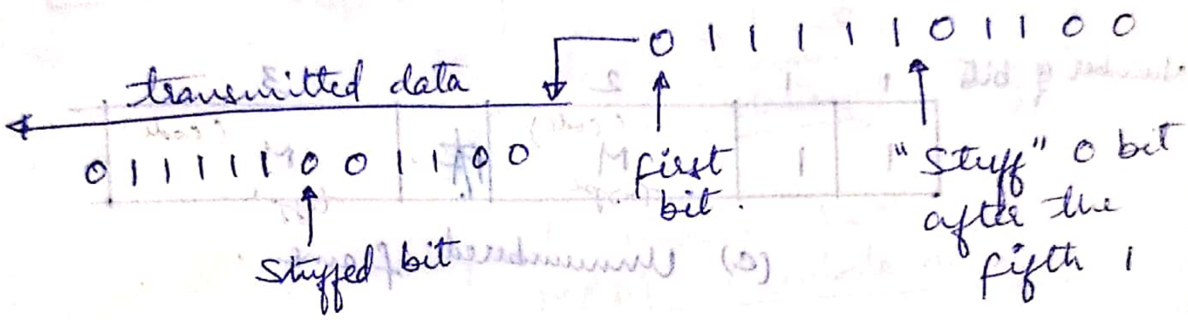
One Byte address ends with a 1
$2^7 = 128$ address
8^{th} bit used for some other purpose.
Addr more than 1 byte, all bytes end with 1 except last byte ends with 0

- Standard format
- Extended format

Shubhangi Khalche,



(a) More than five consecutive 1s.



(b) five consecutive 1s.

fig: Bit stuffing — MU Question

Types of frames:

- Information frames
- Supervisory frames
- unnumbered frames.

Flag	Address	control	user information	FCS	Flag

I-frame.

Flag	Address	control	FCS	flag

S-frame.

Flag	Address	control	Mgt Information	FCS	flag

U-frame.

fig HDLC frames.

N(S) = 0 to 7

ack. when piggybacking used

Number of bits	1	3	1	3
	0	N(S)	P/F	N(R) (next ack)

(a) Information frame

Number of bits	1	1	2	1	3
	1	0	code S Type	P/F	N(R) (Next ack)

(b) Supervisory frame

Number of bits	1	1	2	1	3
	1	1	(code) M Type	P/F	(code) M Type

(c) Unnumbered frame.

fig: control fields for HDLC frames.

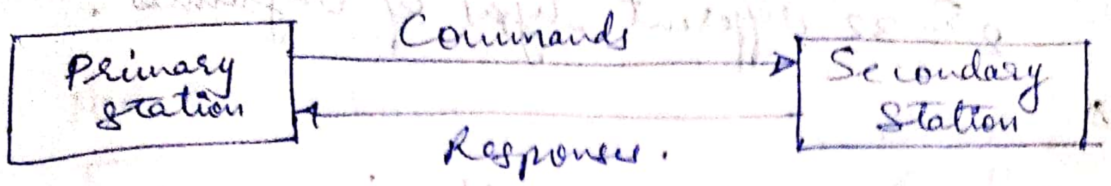
S field of Supervisory frame.	Operation
0 0	RR (Receive Ready)
0 1	REJ (Reject) (Neg. ack) Go Back N
1 0	RNR (Receive not ready)
1 1	SREJ (Selective Reject) Selective repeat

0 0 → Next frame expected to be retransmitted

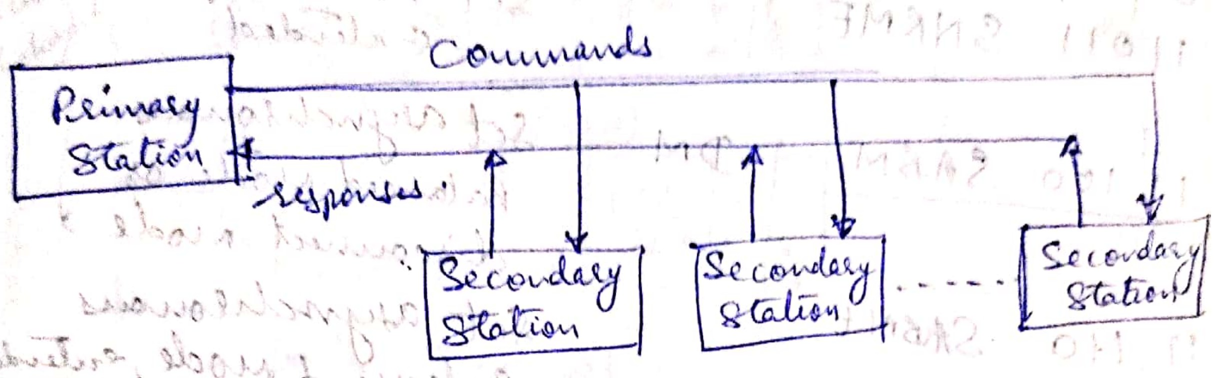
0 1 → N(R) frame

1 0 → Temporary problems to accept

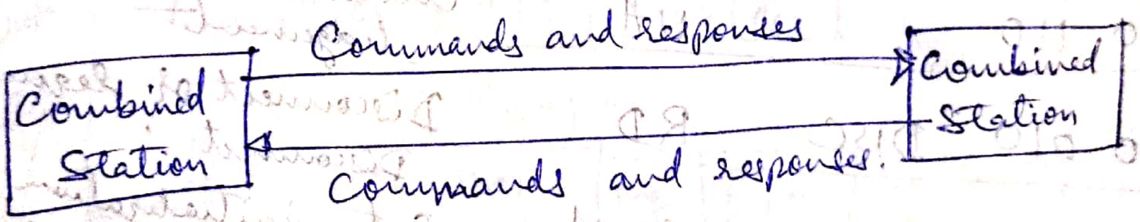
1 1 → Retransmit only specific frames



(a) Point-to-Point link. (NRM)



(b) Multipoint link. ARM
~~NRM~~ (unbalanced).



flow, error

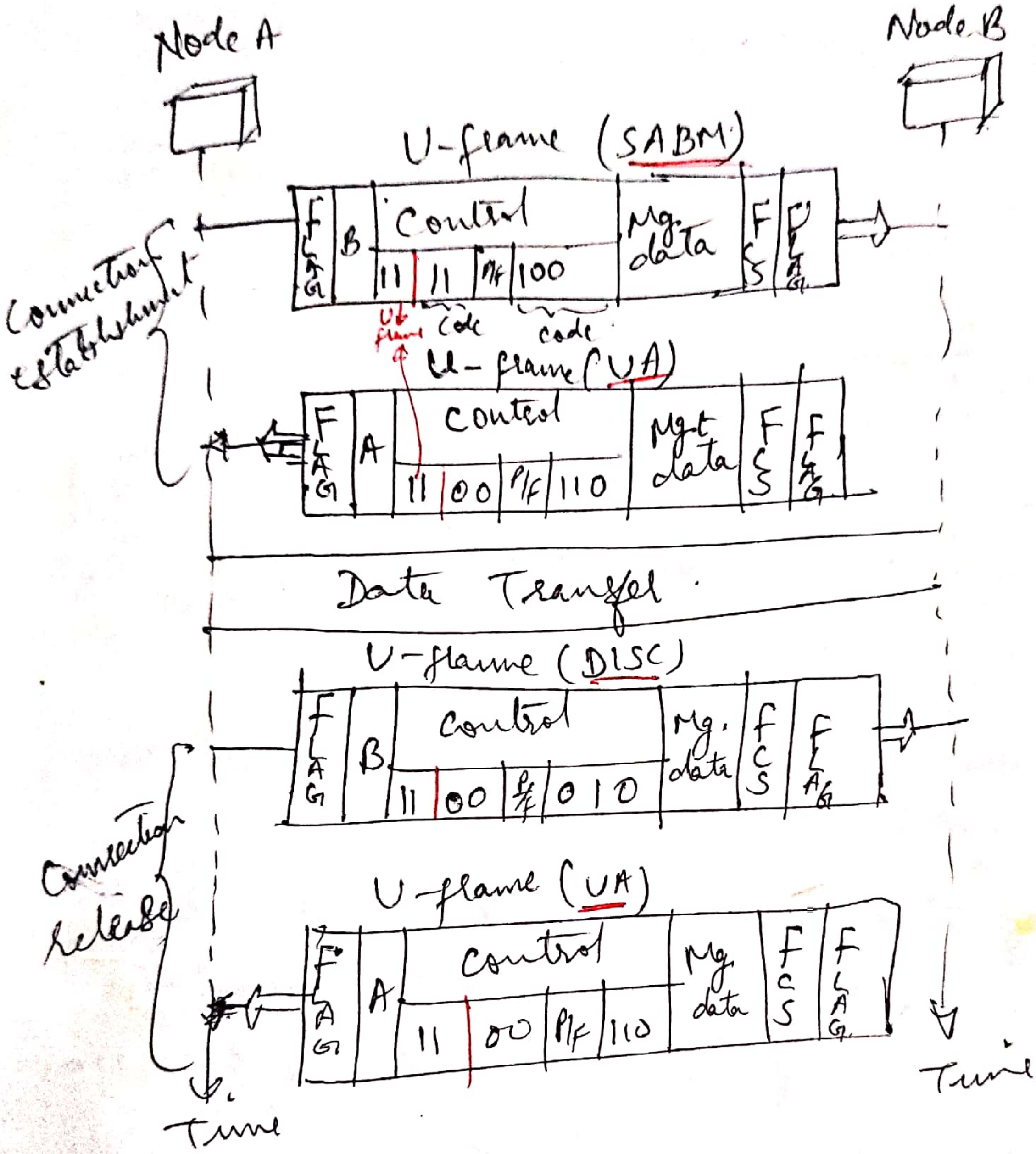
(C) Point-to-point link between combined stations.
ABM.

$2^5 = 32$ different types of U-frames.

Code	Command	Response	Meaning
00 001	SNRM		Set normal Response Mode
11 011	SNRME		Set normal Response mode Extended
11 100	SABM	DM	Set asynchronous Balanced mode or disconnect mode.
11 110	SABME		Set asynchronous Balanced mode, extended
00 000	UI	UI	Unnumbered Information
00 110		UA	unnumbered acknowledgement
00 010	DISC	RD	Disconnect or request Disconnect.
10 000	SIM	RIM	Set initialization model or request information mode.
00 100	UP		unnumbered poll
11 001	RSET		Reset
11 101	XID	XID	Exchange ID
10 001	FRMR	FRMR	Frame reject.

Table: U-frame Control Command & response.

fig e.g. Connection / Disconnection



Shubhangi Khuche

POINT TO POINT PROTOCOL (PPP)

Ref: Behrouz A Forouzan.

- PPP operates only at the physical and data link layers.
- At the data link layer, PPP employs a version of HDLC.

	01111110	11111111	11000000				
	Flag	Address	Control	Protocol	Data & Padding	FCS	Flag
Bytes	1 byte	1 byte	1 byte	1 or 2 bytes	Variable	2 or 4 bytes	1 byte

Fig: PPP frame.

The descriptions of the fields are as follows:

- **Flag Field**: The flag field, like the one in HDLC, identifies the boundaries of a PPP frame. Its value is 01111110.
- **Address Field**: Because PPP is used for a point-to-point connection, it uses the broadcast address of HDLC, 11111111, to avoid a data link address in the protocol.
- **Control field**: The control field uses the format of the Unnumbered Frame (U-frame) in HDLC. The value is 11000000 to show that the frame does not contain

any sequence numbers and that there is no flow and error control.

- Protocol field: The protocol field defines what is being carried in the data field. User data or other information.

- Data field: This field carries either the user data or other information.

- FCS: The frame check sequence field, again HDLC, is simply a two-byte or four-byte CRC.

POINT TO POINT PROTOCOL — PPP (Ref: Tanenbaum)

The Internet needs a point-to-point protocol for a variety of purposes, including
- router-to-router traffic &
- home user-to-ISP traffic.

Functions of PPP:

- PPP handles error detection.
- Supports multiple protocols.
- Allows IP addresses to be negotiated at connection time.
- Permits authentication.

PPP Provides three features:

1) A framing method that unambiguously delineates the end of one frame and the start of the next one. The frame format also handles error detection.

2) A Link Control Protocol (LCP) for bringing lines up, testing them, negotiating options and bringing them down gracefully when they are no longer needed.

LCP supports synchronous and asynchronous circuits and byte oriented and bit-oriented encodings.

3) A way to negotiate network-layer options in a way that is independent of the network layer protocol to be used. The method chosen is to have a different NCP (Network Control Protocol) for each network layer supported.

To see how these pieces work together, let us consider the typical scenario of a home user calling up an Internet Service provider to make a home PC a temporary Internet host.

Shubhangi Khorche

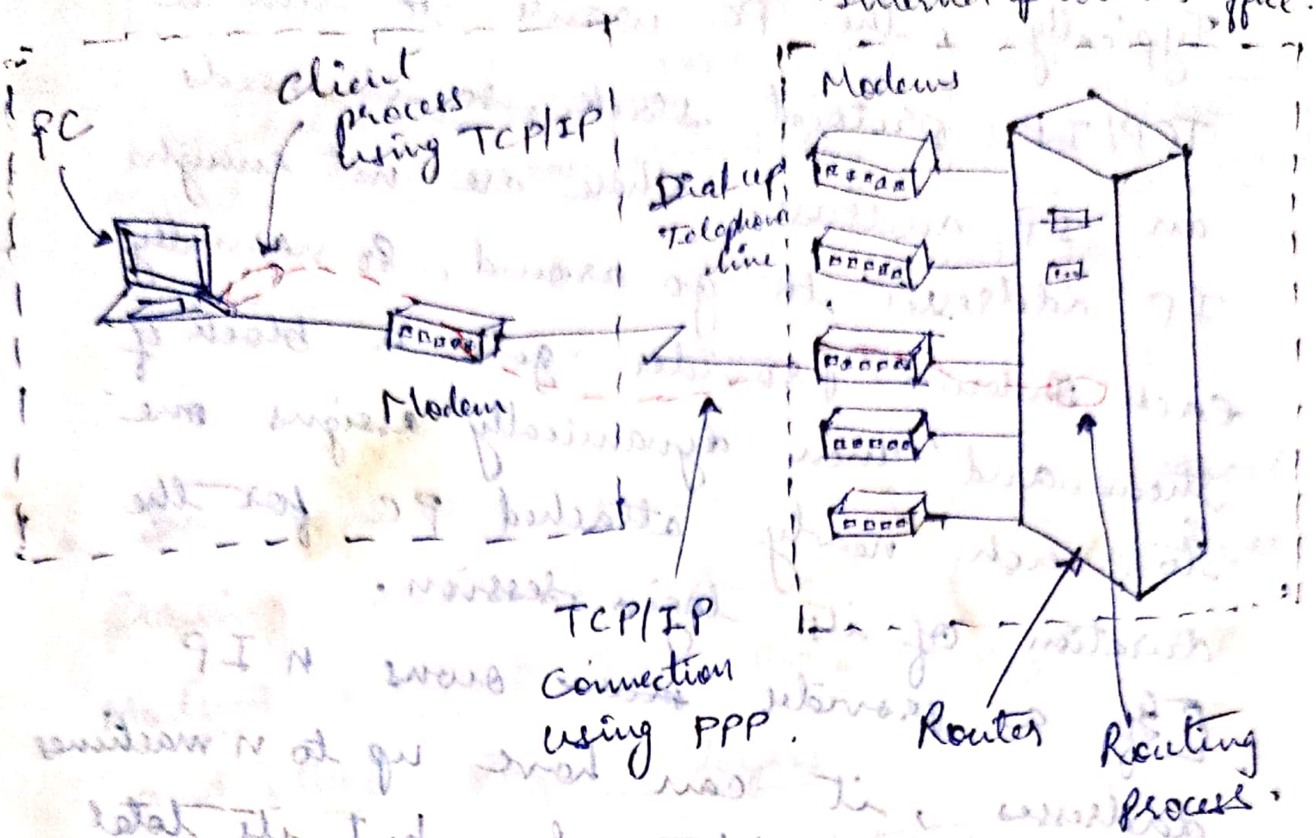


fig: A home personal computer acting as an Internet host.

The PC first calls the provider's router via a modem. After the router's modem has answered the phone and established a physical connection, the PC sends the router a series of LCP packets in the payload field of one or more PPP frames. These packets and their responses select the PPP parameters to be used.

Once the parameters have been agreed upon, a series of NCP packets are sent to configure the network layer.

Shubhangi Kharche

Data link Control

Ref: Behrouz A Forouzan

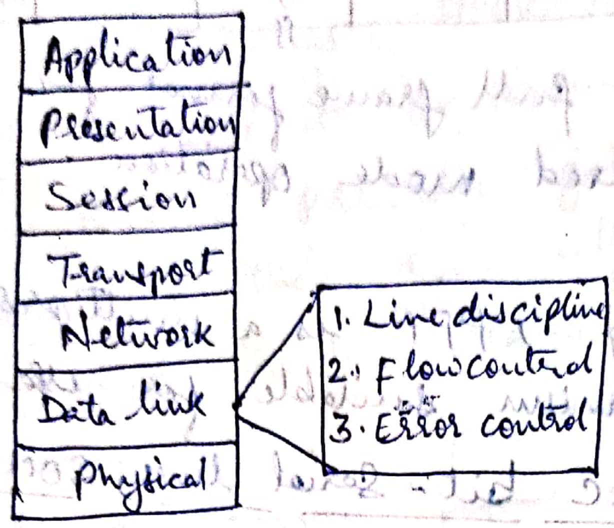


fig 1: Data link layer

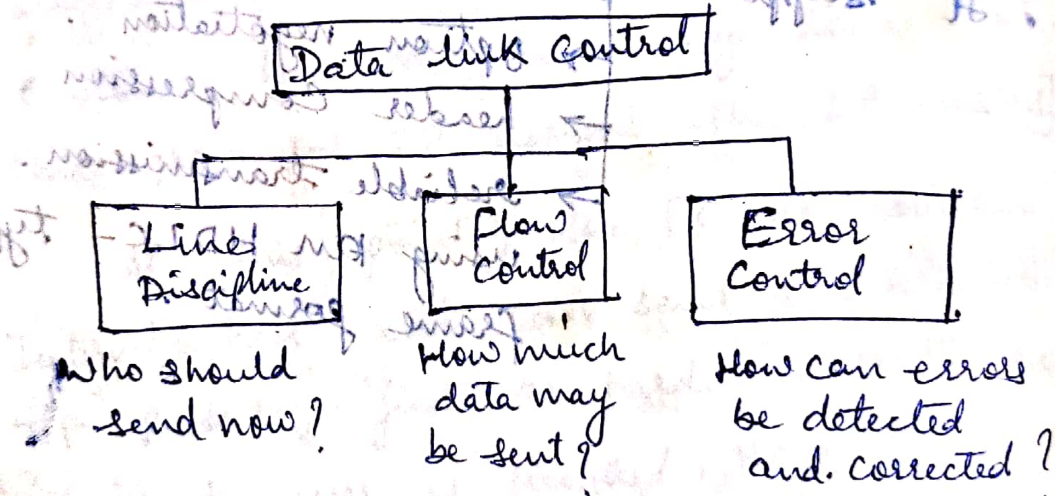


fig 2: Data link layer functions.

FLOW CONTROL:

- Flow control is a set of procedures that tells the sender how much data it can transmit before it must wait for an acknowledgment from the receiver.

The flow of data must not overwhelm the receiver.

Shubhangi Khuche.

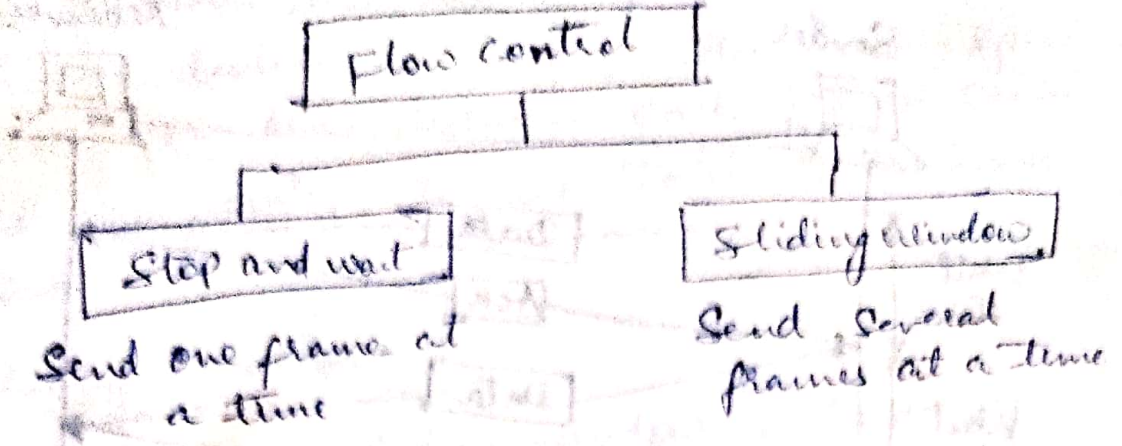


fig 3: Categories of flow control.

Stop-and-Wait:

In a stop-and-wait method of flow control, the sender waits for an acknowledgement after every frame it sends. Only when an acknowledgement has been received is the next frame sent. This process of alternately sending and waiting repeats until the sender transmits an end of transmission (EOT) frame.

Stop-and-wait can be compared to a picky executive giving dictation: She says a word, her assistant says "OK", She says another word, her assistant says, "OK", and so on.

In the Stop-and-wait method of flow control, the sender sends one frame and waits for an acknowledgement before sending the next frame.

Frames may be acknowledged at any point without waiting for the window to fill up and may be transmitted as long as the window is not yet full.

To keep track of which frames have been transmitted and which received, sliding window introduces an identification scheme based on the size of the window.

The frames are numbered modulo-n, which means they are numbered from 0 to n-1. For example, if n = 8, the frames are numbered 0, 1, 2, 3, 4, 5, 6, 0, 1, 2, 3, 4, 5, 6, ~~7~~, 0, 1, ... The size of the window is n-1 (in this case, 7). In other words, the window cannot cover the whole module (8 frames); it covers one frame less.

When the receiver sends an ACK, it includes the number of the next frame it expects to receive. In other words, to acknowledge the receipt of a string of frames ending in frame 4, the receiver sends an ACK containing the number 5. When the sender sees an ACK with the number 5, it knows that all frames up through number 4 have been received.

The window can hold n-1 frames at either end, therefore, a maximum of n-1 frames may be sent before an acknowledgement is required. Fig 5 shows the relationship of a window to the main buffer.

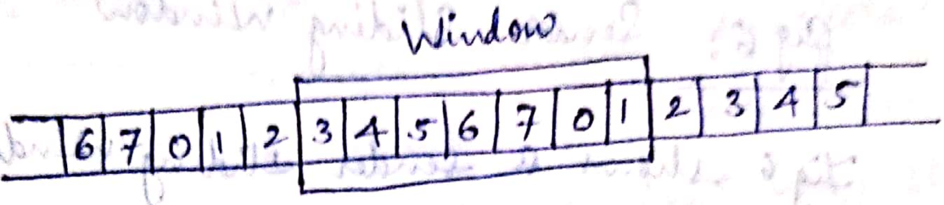


fig 5: Sliding window

<u>Sender Window</u>: At the beginning of a transmission, the sender's window contains n-1 frames. As frames are sent out, the left boundary of the window moves inward, shrinking the size of the window. Given a window of size W, if three frames have been transmitted since the last acknowledgement, then the number of frames left in the window is W-3. Once an ACK arrives, the window expands to allow in a number of new frames equal to the number of frames acknowledged by that ACK.

Sender window

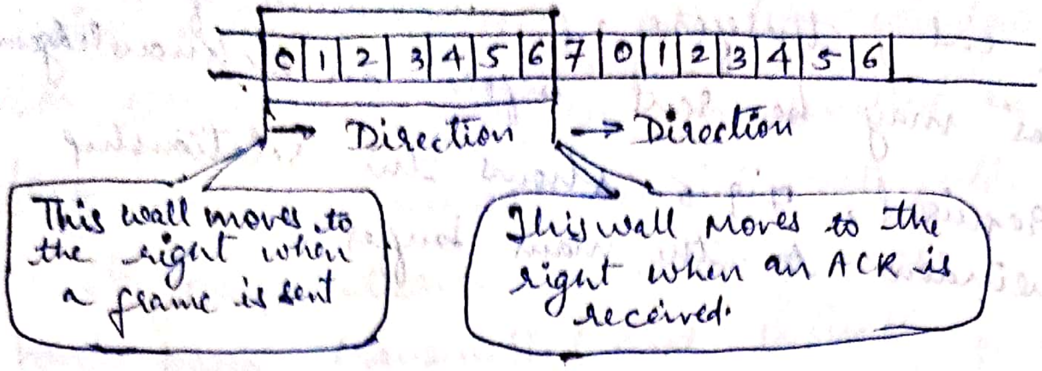


fig 6: Sender Sliding window

Fig 6 shows a sender sliding window of size 7. Given a window of size 7, if frames 0 through 4 have been sent and no acknowledgement has been received, the sender's window contains two frames (numbers 5 & 6). Now, if an ACK numbered 4 is received, four frames (0 through 3) are known to have arrived undamaged and the sender's window expands to include the next four frames in its buffer. At this point, the sender's window contains six frames (numbers 5, 6, 7, 0, 1, 2). If the received ACK had been numbered 2, the sender's window would have expanded by only two frames, to contain a total of four.

Conceptually, the sliding window of the sender shrinks from the left when frames of data are sent. The sliding window of the sender expands to the right when acknowledgements are received.

Receiver Window :

At the beginning of transmission, the receiver window does not contain n-1 frames but n-1 spaces for frames. As new frames come in, the size of the receiver window shrinks. The receiver window therefore represents not the number of frames received but the number of frames that may still be received before an ACK must be sent.

Given a window of size W, if three frames are received without an acknowledgement being returned, the number of spaces in the window is W-3. As soon as an acknowledgement is sent, the window expands to include places for a number of frames equal to the number of frames acknowledged.

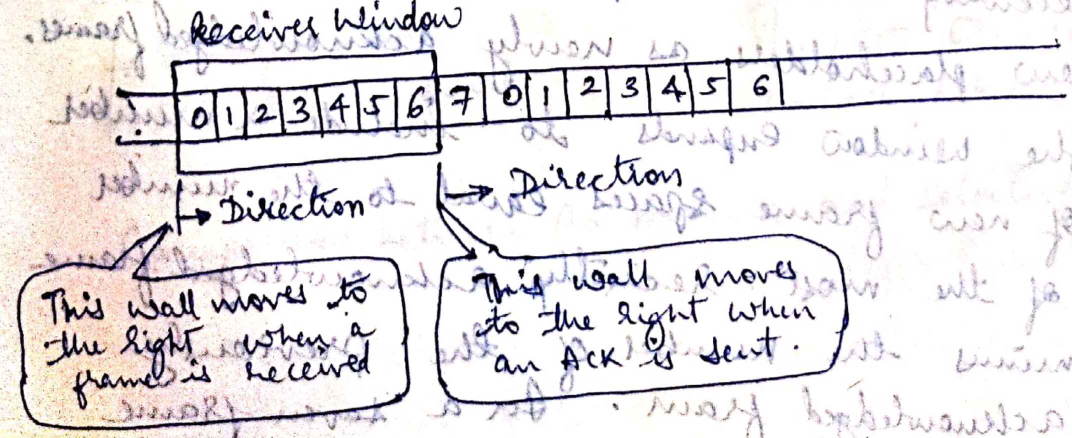


fig 7: Receiver Sliding window

Fig 7 shows a receiving window of size 7. In the figure, the window contains spaces for seven frames, meaning that seven frames may be received before an ACK must be sent. With the arrival of the first frame, the receiving window shrinks, moving the boundary from space 0 to 1. The window has shrunk by one, so the receiver may now accept six frames before it is required to send an ACK. If frames 0 through 3 have arrived but have not been acknowledged, the window will contain three frame spaces.

Conceptually, the sliding window of the receiver shrinks from the left when frames of data are received. The sliding window of the receiver expands to the right when acknowledgements are sent.

As each ACK is sent out, the receiving window expands to include as many new placeholders as newly acknowledged frames. The window expands to include a number of new frame spaces equal to the number of the most recently acknowledged frame minus the number of the previously acknowledged frame. In a seven-frame window, if the prior ACK was for frame 2 & the current ACK is for frame 5, the window expands by three (5−2); if the prior ACK was for frame 3 & the current ACK is for frame 1, the window expands by six (1+8−3).

ERROR CONTROL

- Error control
  - Stop-and-wait ARQ
  - Sliding Window ARQ.
    - Go-back-n
    - Selective-reject

fig 1: Categories of error control.

- ARQ
- Stop-and-wait ARQ.
  - → Damaged Frames
  - → Lost Frames.
  - → Lost Data Frame.
  - → Lost Acknowledgement.

- Sliding Window ARQ (MU Question)
  - Go-Back-n ARQ (MU Question)
    - → Damaged Frame
    - → Lost Acknowledgement.
  - Selective-Reject ARQ. (MU Question)
    - → Damaged Frames
    - → Lost Frames
    - → Lost Acknowledgement.

- Comparison between Go-Back-n and Selective-Reject (MU Question)

Automatic Repeat Request (ARQ)

Error correction in the data link layer is implemented simply: anytime an error is detected in an exchange, a negative acknowledgement (NAK) is returned and the specified frames are retransmitted. This process is called automatic repeat request (ARQ).

Error control in the data link layer is based on automatic repeat request (ARQ), which means retransmission of data in three cases: damaged frame, lost frame, and lost acknowledgement.

It sometimes happens that a frame is so damaged by noise during transmission that the receiver does not recognize it as a frame at all. In those cases, ARQ allows us to say that the frame has been lost. A second function of ARQ is the automatic retransmission of lost frames, including lost ACK and NAK frames (where the loss is detected by the sender instead of the receiver)

Stop-and-Wait ARQ:

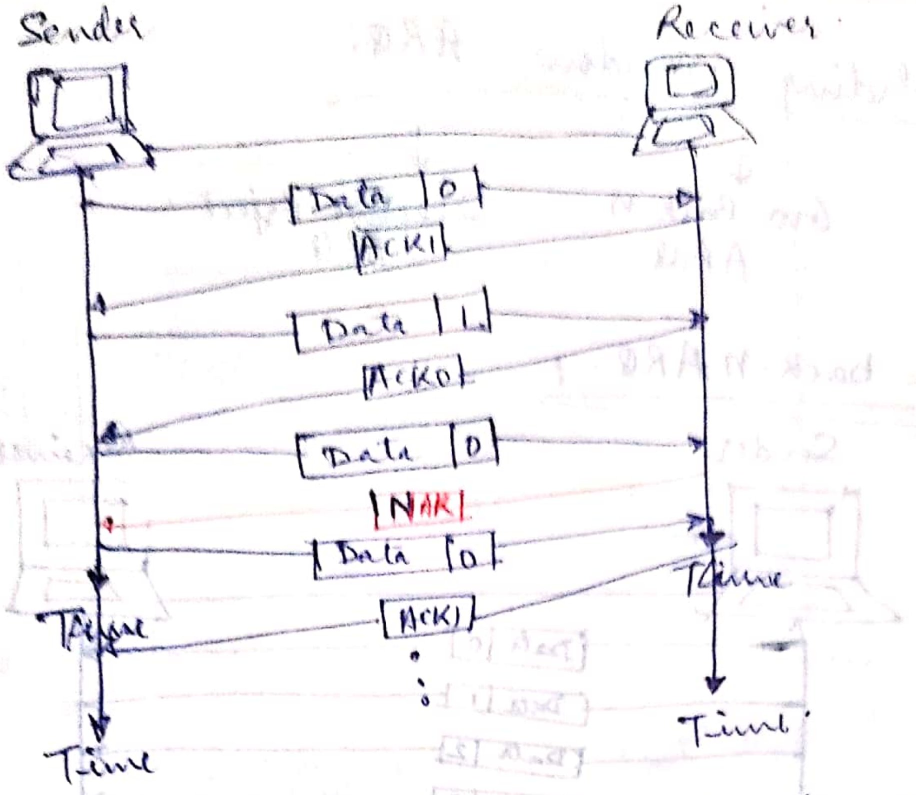


fig 2: Stop-and-wait ARQ, damaged frame.

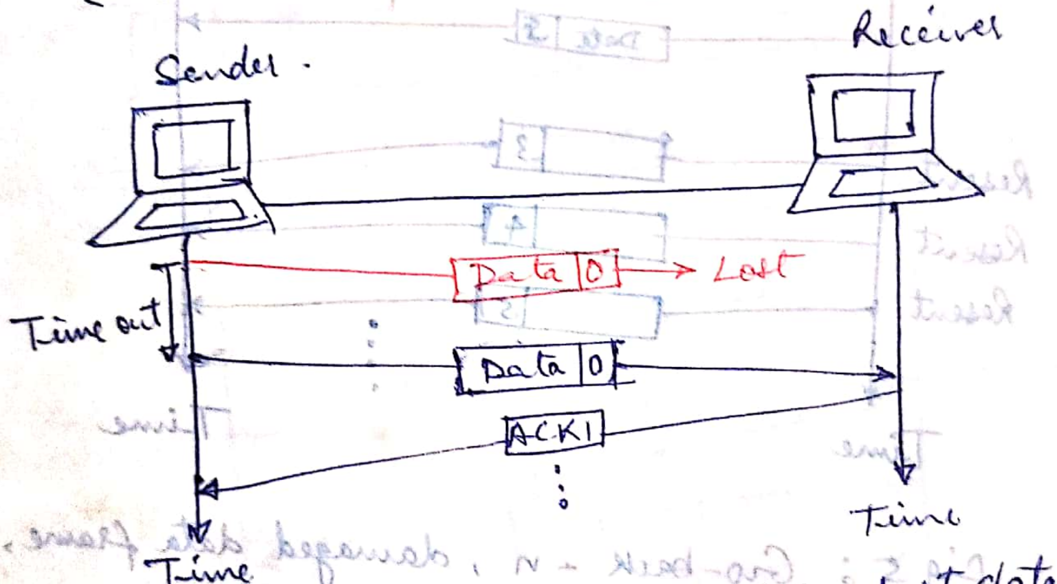


fig 3: Stop-and-wait ARQ, lost data frame.

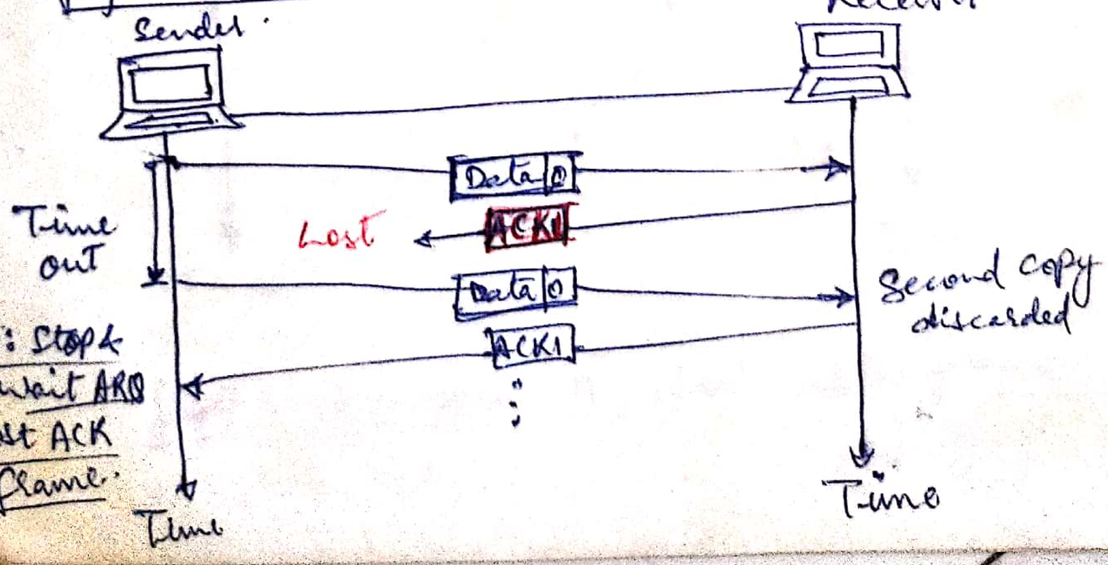


fig 4: Stop & wait ARQ Lost ACK frame.

Sliding Window ARQ.

- Go-Back-n ARQ
- Selective Reject ARQ.

Go-back-n ARQ.

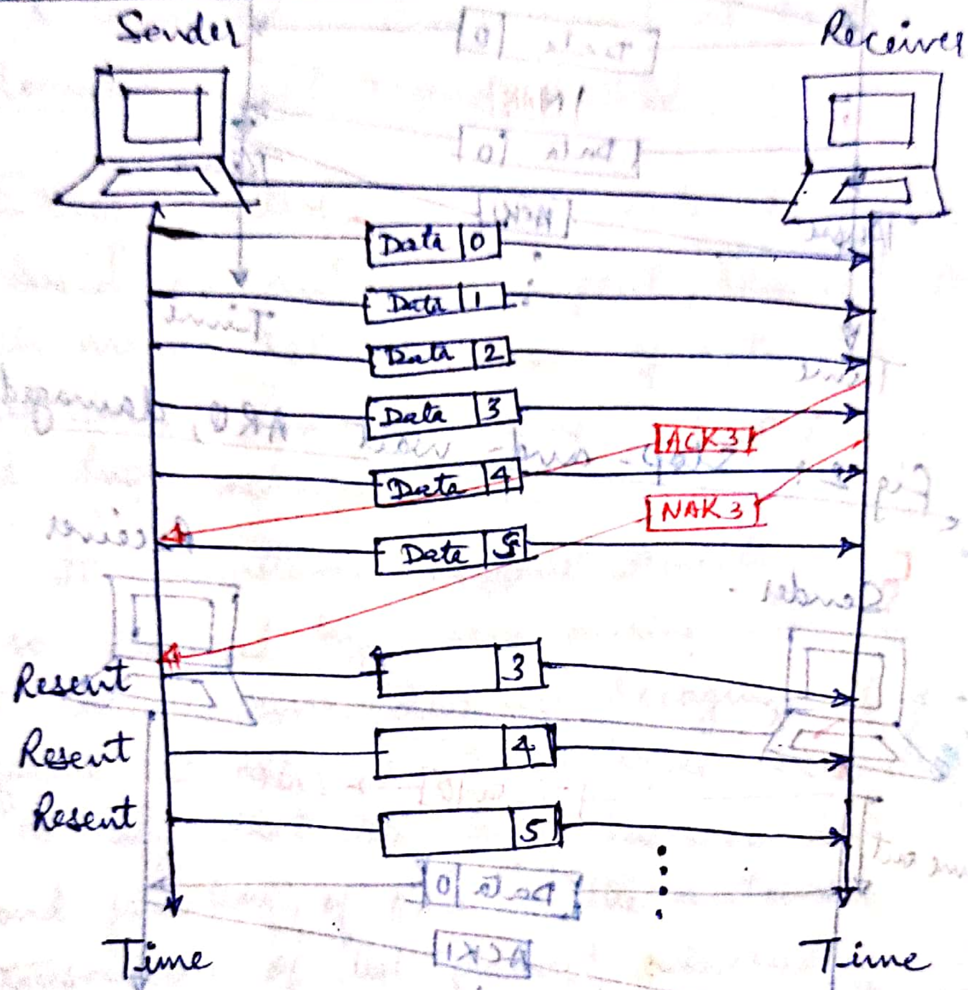


fig 5: Go-back-n, damaged data frame.

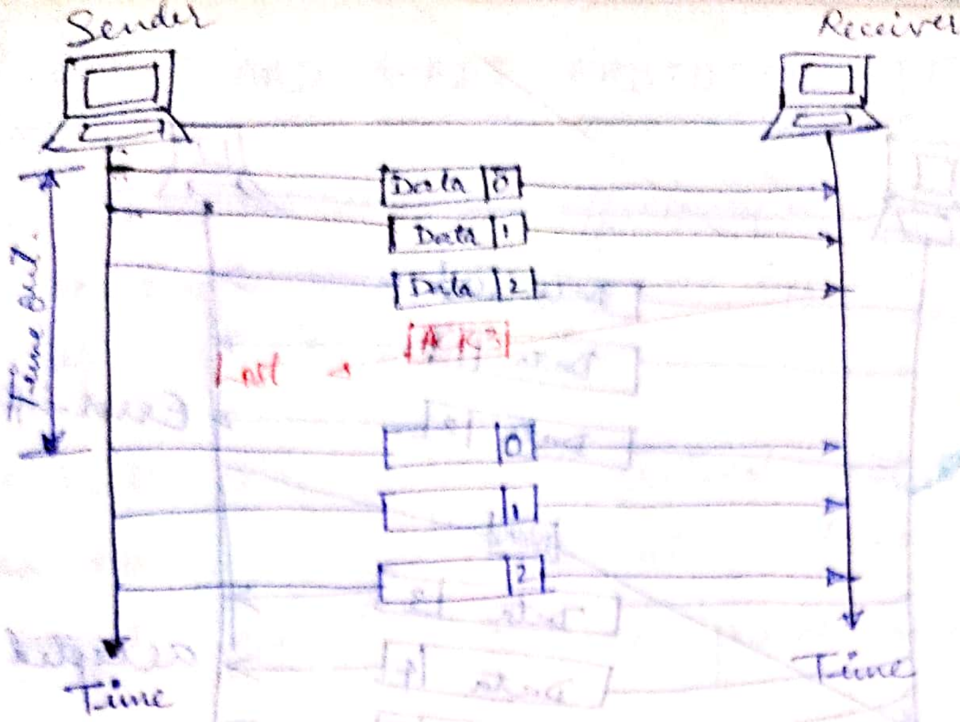


fig 6: Go-back-n, lost ACK.

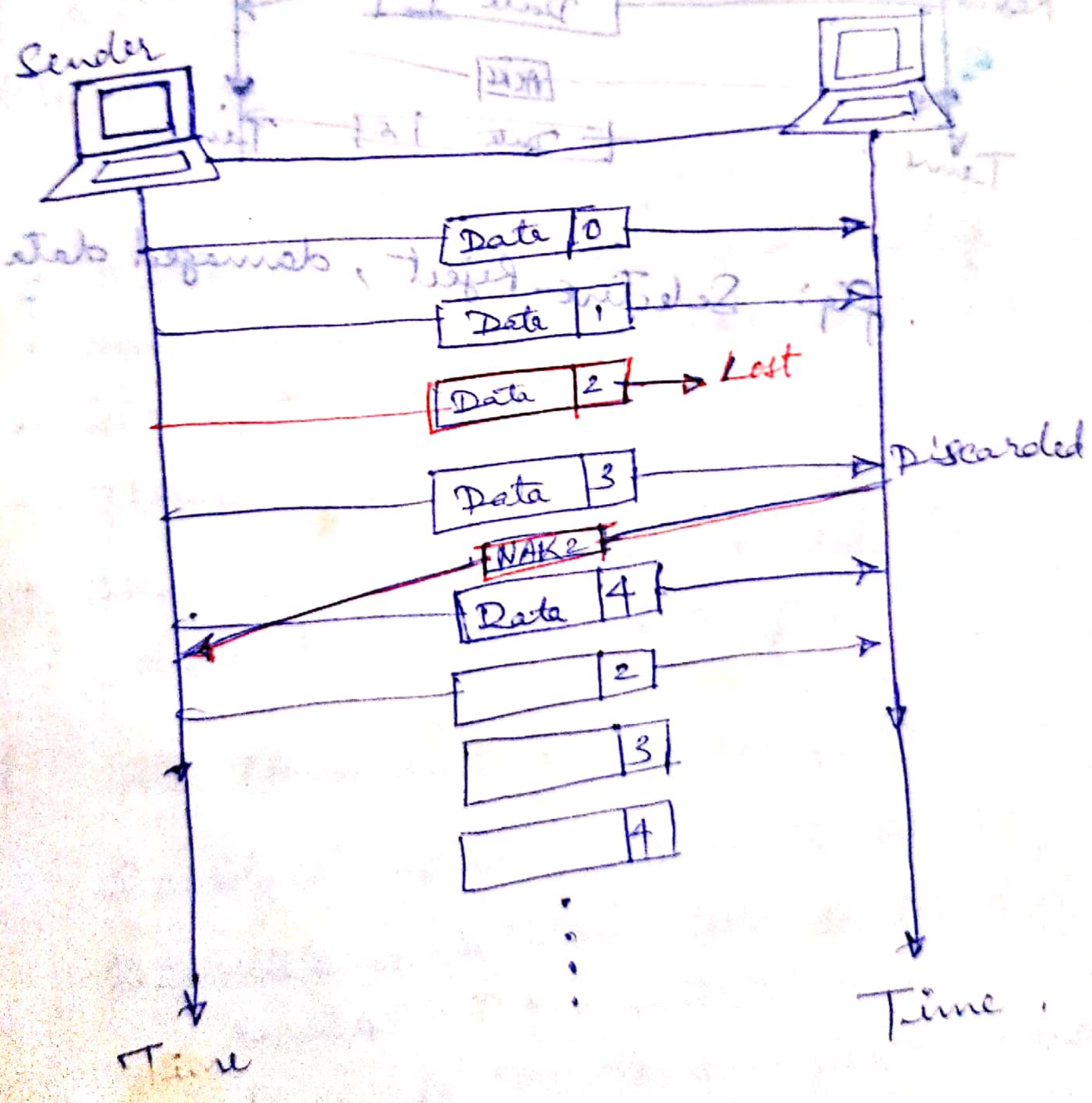


fig 7: Go-back-n, lost data frame.

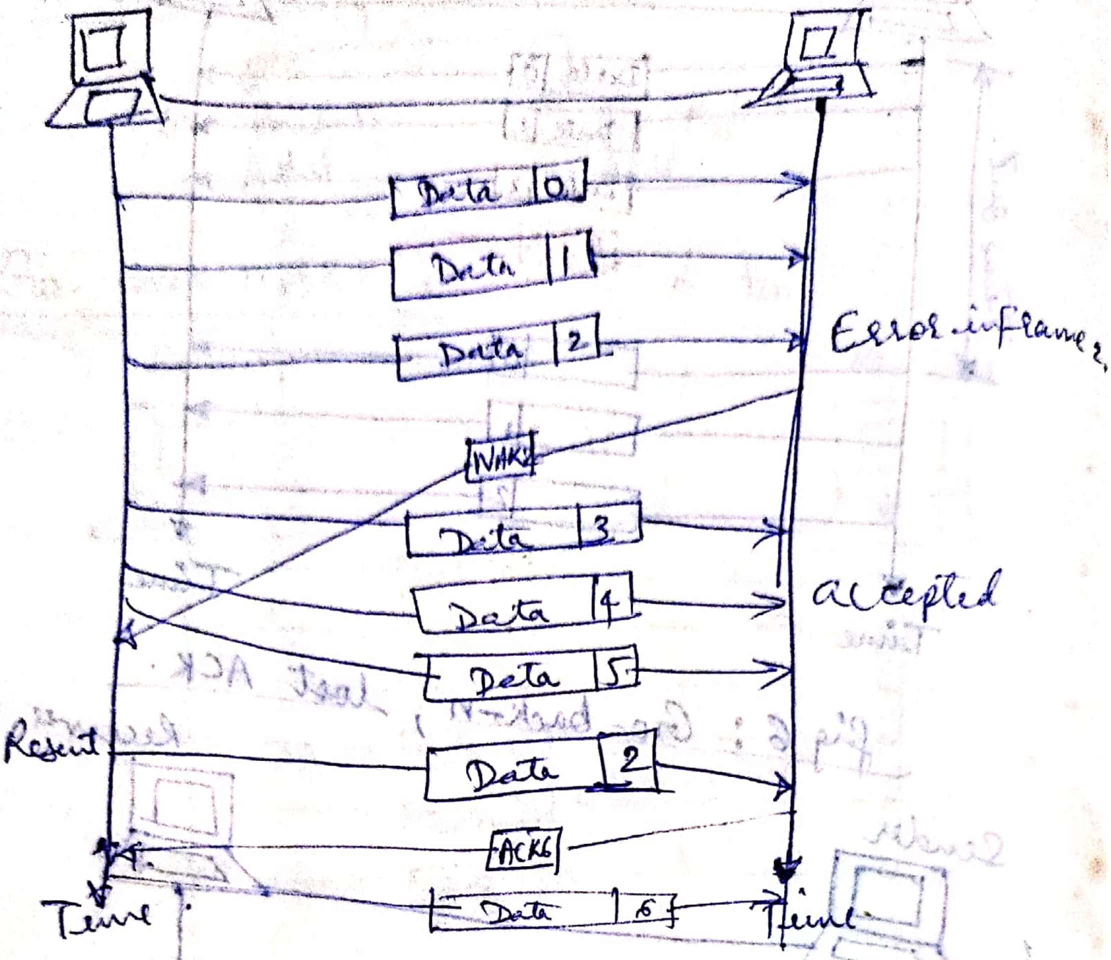


fig: Selective Reject, damaged data frame

Ref: William Shay

STOP AND WAIT PROTOCOL EFFICIENCY

MU Question

Three measures of efficiency are :

1) Buffer Space.
2) Channel utilization.
3) Effective data rate.

With Stop & wait protocol not more than one frame is sent at a time, so a buffer capacity of one frame is sufficient.

Channel utilization means the percentage of time for which the channel transfers data frames.

Channel utilization depends on:

- Distance between the sender and receiver
- Signal speed over the channel.
- The bit rate
- Frame size &
- The amount of time needed to construct and send a frame

Let, R = transmission rate (10 Mbps or 10 bits per μsec)

S = Signal speed (200 meters per μsec).

D = distance between the sender and receiver (200 meters)

T = time to create one frame (1 μsec).

F = number of bits in a frame (200)

N = Number of data bits in a frame (160)

A = number of bits in an acknowledgement (40

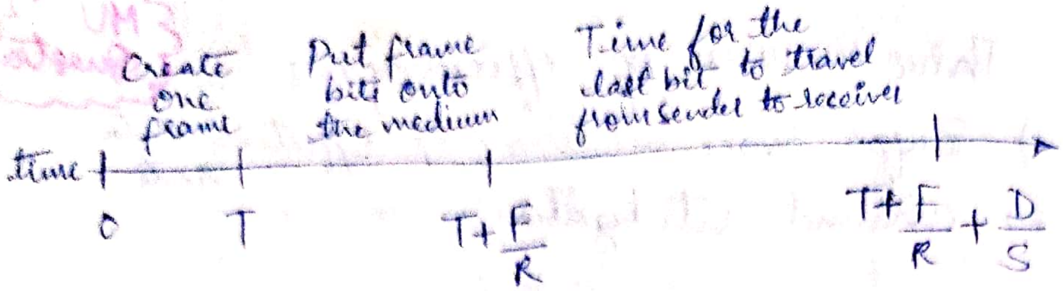


fig : Time Required to send a frame to a receiver.

Time required to send a frame to a receiver is calculated in following 3 steps:

1. Determine the time needed to create one frame (T)
2. Determine the time needed to put frame bits onto the medium (F/R)
3. Determine the time needed for the bits to travel from sender to receiver (D/S)

Thus, the time required to send a frame to a receiver

$$= T + \frac{F}{R} + \frac{D}{S}$$

Similarly, Time required to send an acknowledgement to the sender

$$= T + \frac{A}{R} + \frac{D}{S}$$

Now, Time elapsed between sending two consecutive frames is

$$time = \left(T + \frac{F}{R} + \frac{D}{S}\right) + \left(T + \frac{A}{R} + \frac{D}{S}\right)$$

$$= 2\left(T + \frac{D}{S}\right) + \frac{F+A}{R}$$

The amount of time a data frame is actually in transit is $F/R + D/S$.

Thus, if we define P as the percentage of time during which frame bits occupy the channel, we have

$$P\,(\text{stop \& wait protocol}) = 100 \times \frac{\text{Transit Time}}{\text{Time elapsed between sending two consecutive frames}}$$

$$= 100 \times \frac{\frac{F}{R} + \frac{D}{S}}{2\left(T + \frac{D}{S}\right) + \frac{F+A}{R}}$$

$$= 100 \times \frac{\frac{200\text{ bits}}{10\text{ bits}/\mu sec} + \frac{200\text{ meters}}{200\text{ meters}/\mu sec}}{2\left(1\,\mu sec + \frac{200\text{ mts}}{200\text{ mts}/\mu sec}\right) + \frac{200\text{ bits} + 40\text{ bits}}{10\text{ bits}/\mu sec}}$$

$$= 75\%$$

Thus, if we measure efficiency solely by channel utilization, then efficency of stop and wait Protocol is 75%.

Effective data rate is the actual number of data bits sent per unit of time.

To calculate the effective data rate, we divide the number of data bits (N) sent by the elapsed time between sending two frames.

$$\therefore \text{Effective data rate (Stop-and-wait protocol)} = \frac{N}{2\left(T+\frac{D}{S}\right)+\frac{F+A}{R}}$$

$$= \frac{160 \text{ bits}}{2\left[1\,\mu\text{sec} + \frac{200 \text{ meters}}{200 \text{ meters}/\mu\text{sec}}\right] + \frac{200 \text{ bits} + 40 \text{ bits}}{10 \text{ bits}/\mu\text{sec}}}$$

$$= 5.7 \text{ bits}/\mu\text{sec}$$

$$= 5.7 \text{ Mbps}$$

These measures provide only part of the total picture. Other factors to consider are the users the protocol serve, the amount of data to transfer, and the fact that the channel will be used by others.

MU Question SLIDING WINDOW PROTOCOL EFFICIENCY:

Let R = transmission rate (10Mbps or 10 bits/μsec).
S = Signal Speed (200 mts/sec)
D = Distance between the sender and receiver (200 metres)
T = Time to create one frame (1 μsec)
F = number of bits in a frame (200)
N = number of data bits in a frame (160)
A = number of bits in an acknowledgment (40)
W = Window Size (4 frames).

Consider that the protocol sends W frames, waits for the first acknowledgment, sends W more frames, waits for an acknowledgment and so on (as in fig below).

This protocol now resembles stop and wait. However, instead of sending and waiting for individual frames, it sends and waits for a window full of frames

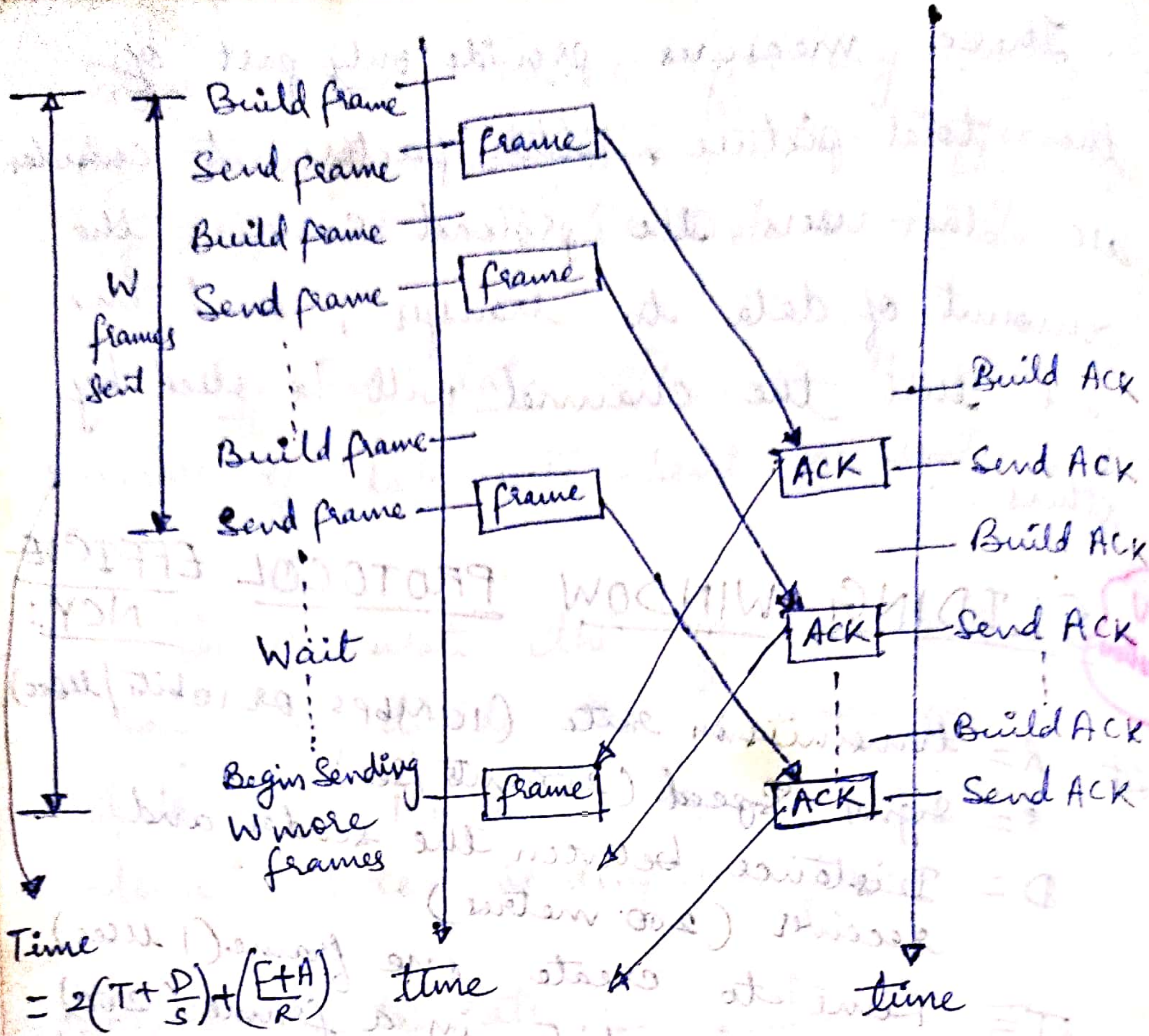


fig : Sending All windowed Frames and waiting.

Mathematically, these two cases can be distinguished by comparing the time to send W frames with the time to send one frame and receive an acknowledgement.

The time to build and send one frame is $T + F/R$. Thus, the time to build and send W frames is

$W \times (T + F/R)$

Time to send a frame and receive an acknowledgment is

$$2\left(T + \frac{D}{S}\right) + \left(\frac{F+A}{R}\right)$$

Let $F = A$ because acknowledgments arrive piggybacked on data frames (of size F) instead of via separate ACK frames (of size A)

$\therefore$ Time to send a frame & receive an acknowledgment is

$$2\left(T + \frac{D}{S} + \frac{F}{R}\right).$$

Since W frames are sent in the same amount of time,

Effective data rate (window-oriented stop & wait version) $= \dfrac{W \times N}{2\left(T + \frac{D}{S}\right) + \frac{2F}{R}}.$

$$= \frac{4 \times 160 \text{ bits}}{2\left(1\,\mu sec + \frac{5000 \text{ mts}}{200 \text{ mts}/\mu sec}\right) + \frac{2 \times 200 \text{ bits}}{10 \text{ bits}/\mu sec}}$$

Here $D = 5000$ mts.

$$= 0.615 \times 10^{6} \text{ bits/sec.}$$

Shubhangi Kharche

Point-To-Point Protocol (Leon Garcia).

	Flag	Address	Control	Protocol	Information	CRC	Flag
No of Bytes	1	1	1	1 or 2		2 or 4	1
No of bits	8	8	8	8	variable	16 or 32	8
	01111110	11111111	11000000				01111110
	7E Hex	All stations are to accept the frame	Unnumbered frame	Specifies what kind of packet is contained in the payload e.g. LCP, NCP, OSI CLNP, IP, IPX.			7E Hex

fig 1: Frame structure of PPP frame: PPP uses same flag as HDLC and inserts a protocol field that specifies the type of packet contained in the payload.

- PPP is byte oriented
- Flag field is used to indicate begining and End of a frame.
- Byte stuffing is used to remove the ambiguity whenever there is occurance of flag in other fields of PPP frame.
- A Control Escape character (octet) is defined as binary 01111101 ie 7D hex.
- Any occurence of Flag or escape character in the PPP frame is replaced by a two character sequence consisting of control Escape character followed by the original octet exclusive-ORed with 20 Hex.
- After the exclusive-OR operation the characters 7E and 7D are replaced by 7D, 5E and 7D, 5D respectively.

Data to be sent

41	7D	42	7E	50	70	46

Control escape character → flag

After stuffing and framing

7E	41	7D	5D	42	7D	5E	50	70	46	7E

flag — flag

fig 2: Example of byte stuffing in PPP

XORed
0 1 1 1 1 1 0 1 — 7DH
0 0 1 0 0 0 0 0 — 20H
0 1 0 1 1 1 0 1 = 5DH

XORed
0 1 1 1 1 1 1 0 — 7EH
0 0 1 0 0 0 0 0 — 20H
0 1 0 1 1 1 1 0 = 5EH

- The receiver ~~[illegible]~~ unstuffs the control escape characters & then computes the CRC.
- while destuffing it exclusive OR's the character following the control escape character with 20H.

0 1 0 1 1 1 0 1 — 5DH
0 0 1 0 0 0 0 0 — 20H
0 1 1 1 1 1 0 1 = 7DH

0 1 0 1 1 1 1 0 — 5EH
0 0 1 0 0 0 0 0 — 20H
0 1 1 1 1 1 1 0 — 7EH

- If the EXORing results are 7DH or 7EH then the receiver removes those characters.

Applications of PPP:

- PPP framing can be used over asynchronous, bit synchronous, or octet synchronous transmission systems.
- PPP is used extensively in dialup Modems.
- PPP also provides the framing in Packet-over-SONET (POS) to carry packet streams over high-speed SONET digital transmission systems.

Shubhangi Khelekar

A Typical Scenario

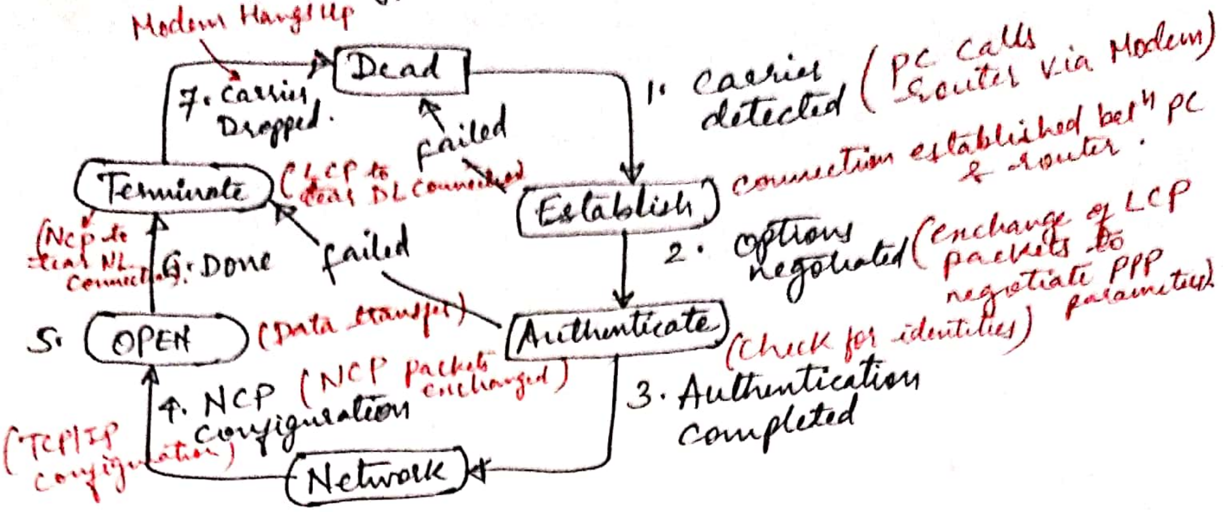


fig 3: PPP phase diagram

Home PC to Internet Service provider

1. PC calls router via Modem
2. PC and router exchange LCP packets to negotiate PPP parameters
3. Check of identities
4. NCP packets exchanged to configure the network layer, for example, TCP/IP (requires IP address assignment)
5. Data transport, for example, Send/receive IP packets.
6. NCP used to tear down the network layer connection (free up IP address); LCP used to shut down data link connection.
7. Modem hangs up.

Home Telephone to Telephone Exchange.

- Subscriber lifts the H/S
- Exchange sends dial tone
- Subscriber dials the no.
- Exchange verifies identities
- Connection is established
- conversation starts
- connection released after conversation ends.

Scanned with CamScanner

Authentication protocols

PAP

Password Authentication protocol

Initiator — Responder (peer process)

Link is established using LCP

Sends ID & password.

Authentication is successfull or has failed

System allows a few retries

If the request has failed, PAP instructs LCP to terminate the link.

- PAP is vulnerable to many different types of security attacks

CHAP (provides greater security)

Challenge Handshake Authentication protocol

Initiator — Responder (peer process)

- Exchange challenge response sequence.
- Establish a shared secret key.

Link is established using LCP

Challenge (Random number and an ID)

Cryptographic checksum of the challenge value that makes use of shared secret key.

- The authenticator verifies the cryptographic checksum by using the shared secret key.
- If the two checksums agree, then the authenticator sends an authentication msg.
- The CHAP allows an authenticator to reissue the challenge to periodically re-authenticate the process.

Scanned with CamScanner

Peer-to-Peer protocols

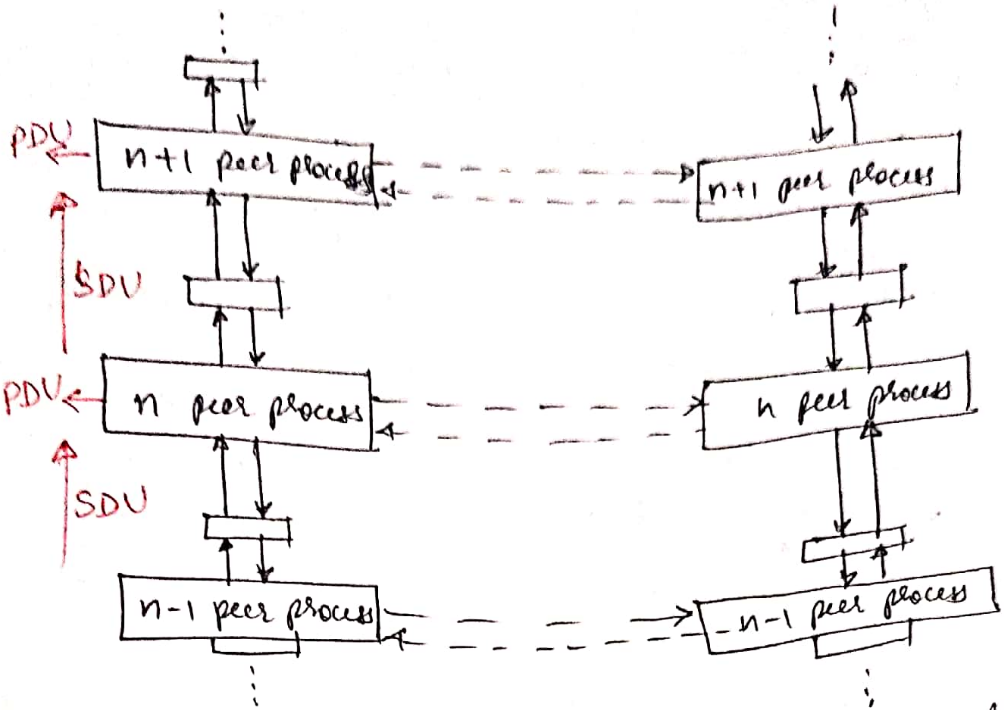
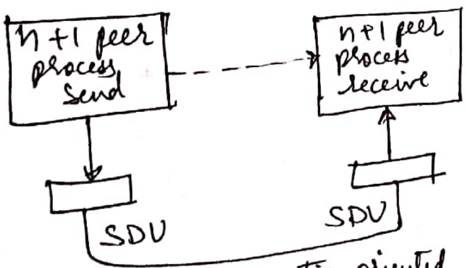


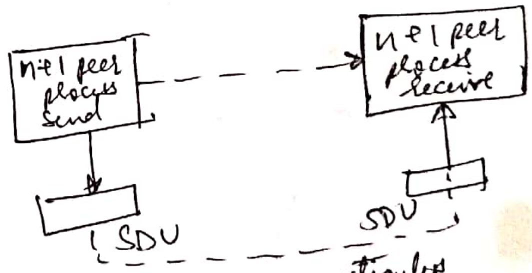
fig: Layer-n peer processes carry out a protocol to provide service to layer n+1. Layer n protocol uses services of layer n-1

Service Models



Layer n connection oriented service

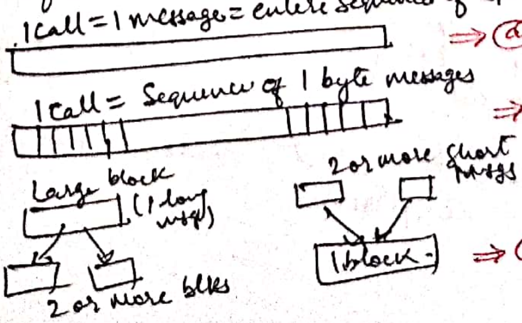
fig: Connection-oriented transfer service



Layer n connectionless service

fig: Connectionless transfer service.

Examples of services: Arbitrary Message size or structure.



⇒ (a) Message

⇒ (b) Stream

⇒ (c) Sequence of blocks.

End to End versus Hop by Hop

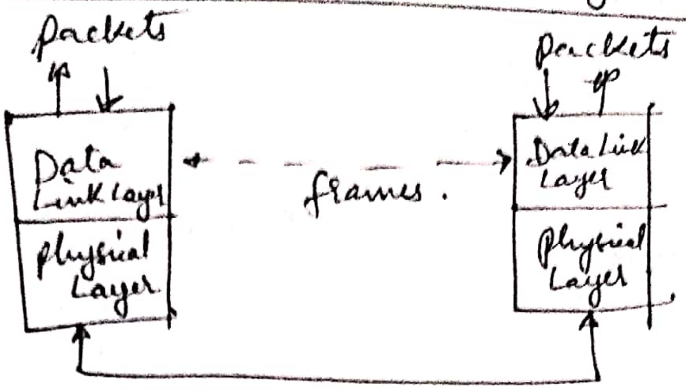


fig: peer-to-peer protocol across a single hop.

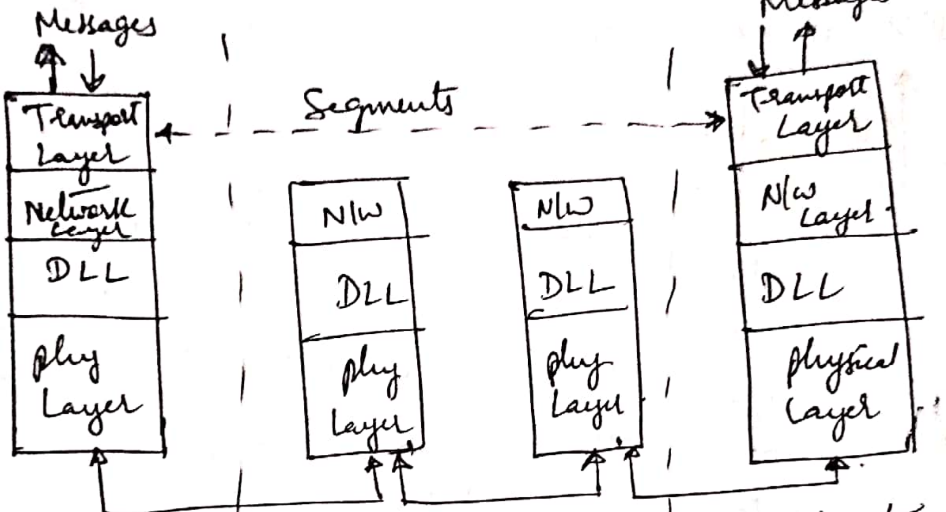
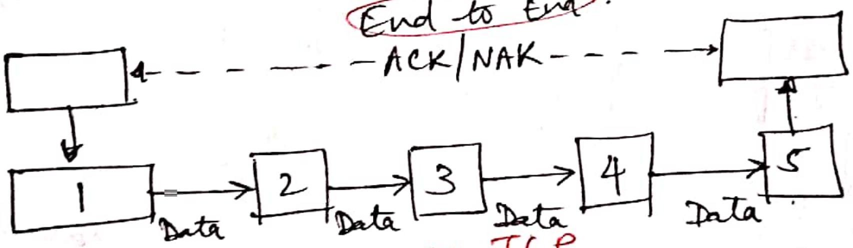
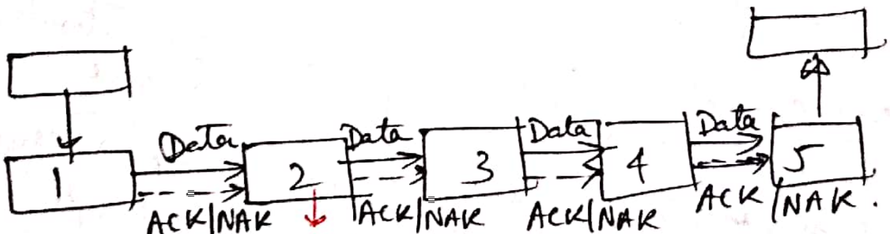


fig: peer to peer protocols operating end-to-end across a n/w-protocol stack view.

End to End.



eg TCP



- error control
- flow ctrl
- congestion ctrl

complex security

Hop by Hop eg HDLK

fig: End to End versus Hop by Hop approaches.

Shubhangi ...

ARQ Protocols and Reliable Data transfer Service.

3

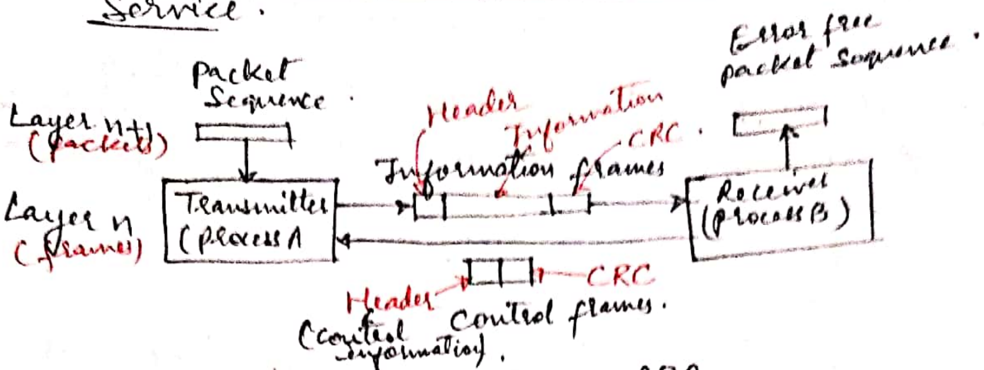


fig: Basic elements of ARQ.

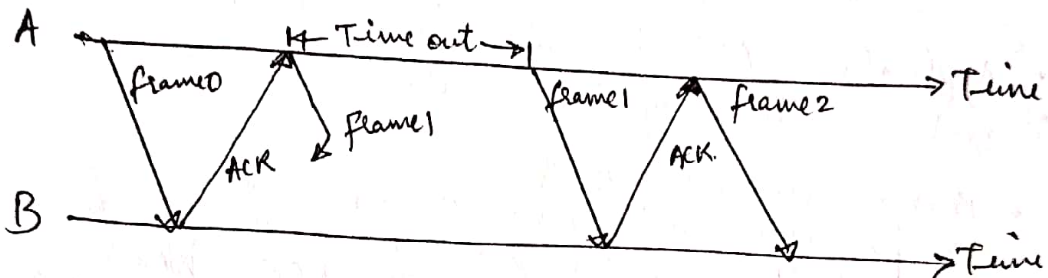
- What is ARQ? ARQ combines error detection & retransmission to ensure that data is delivered accurately to the user despite errors that occur during transmission.
- ARQ → Stop & wait
  - → Go-Back-N ARQ
  - → Selective Repeat ARQ.
- Application of ARQ in Data link controls that operate over a single noisy communication channel.
- Basic elements of ARQ
  - → Information frame (carries information packets) control info in header & CRC
  - → Control frames → consists ACK & NAK's → particular frame transmission; correct receipt of a frame or Group of frames
  - → Timeout mechanisms
- Header has a field which tells whether the frame is information, control, ACK or NAK
- Timeout mechanisms are required to prompt certain actions to maintain the flow of frames.

Basic operation of the packet transfer protocol:

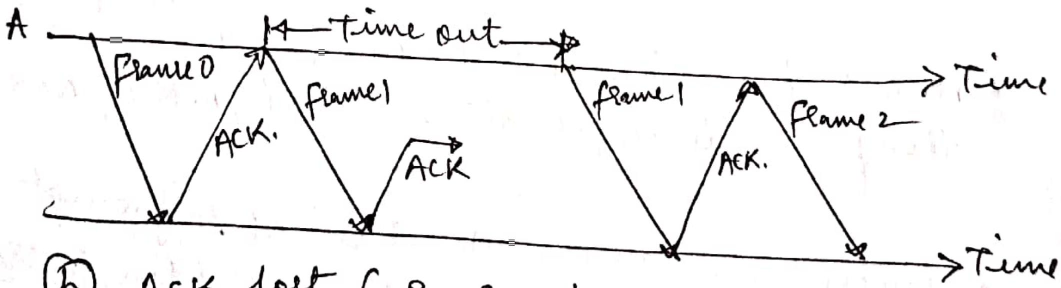
- Layer (n+1) prepares layer (n+1) PDU with the help of layer (n+1) protocol & asks for layer n SDU.
- layer n prepares layer n PDU with the help of layer n protocol & asks for layer (n-1) SDU
  } At Source
- Reverse at Destination.

Stop-&-Wait ARQ

- One frame is delivered at a time.
- A sends info frame to B, wait for an ACK, if A does not receive ACK within a particular time then it resends the info frame to B & again waits for an ack.
- → A sends I-frame to B
  → Sets I-frame timer which expires after a certain time out period.
  → Time out period is selected such that it is larger than the time ~~time~~ required to receive ack frame from B.



(a) frame 1 lost (A does not know what has happend)



(b) ACK lost (B receives frame 1 twice)
(Delivery of duplicate packet)

- Explanation of fig (a) & (b):

In above both cases A transmits frame 1 twice. A is ambiguous about what has happend. B receives frame 1 twice when ACK is in error & thus there is duplication of frame.

Need for Sequence Numbers

- Above ambiguity can be removed by including sequence number in the header of each I-frame.
- Process B would then recognize that the second transmission of frame 1 was a duplicate, discard the frame & resend the ACK for frame 1.
- A second type of ambiguity arises if the ACKs do not contain a sequence number.

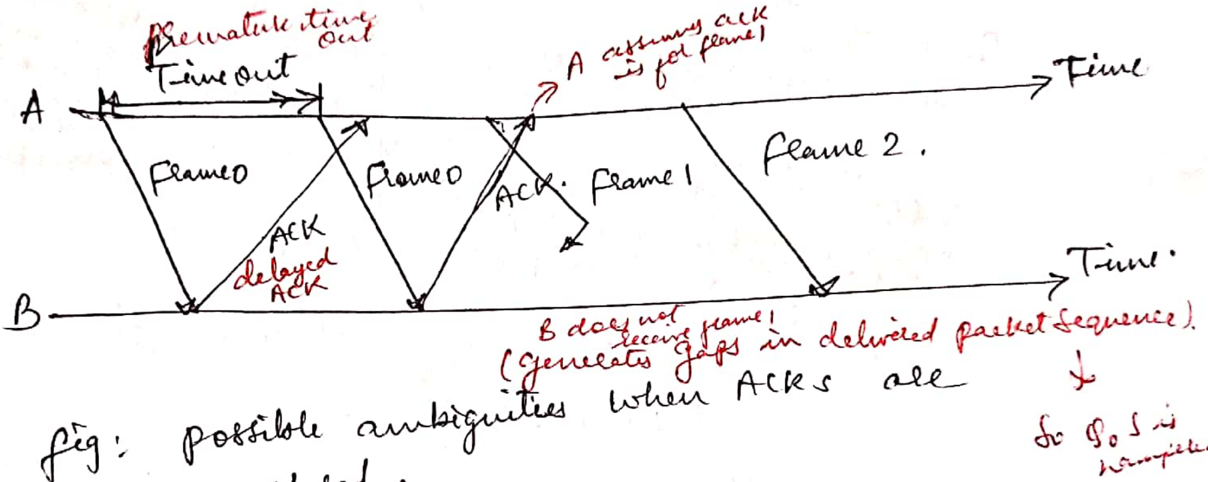
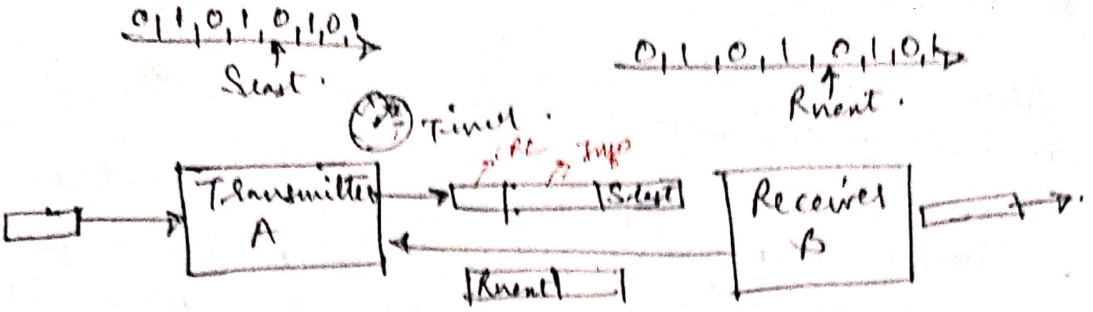


fig: possible ambiguities when ACKs are unnumbered.



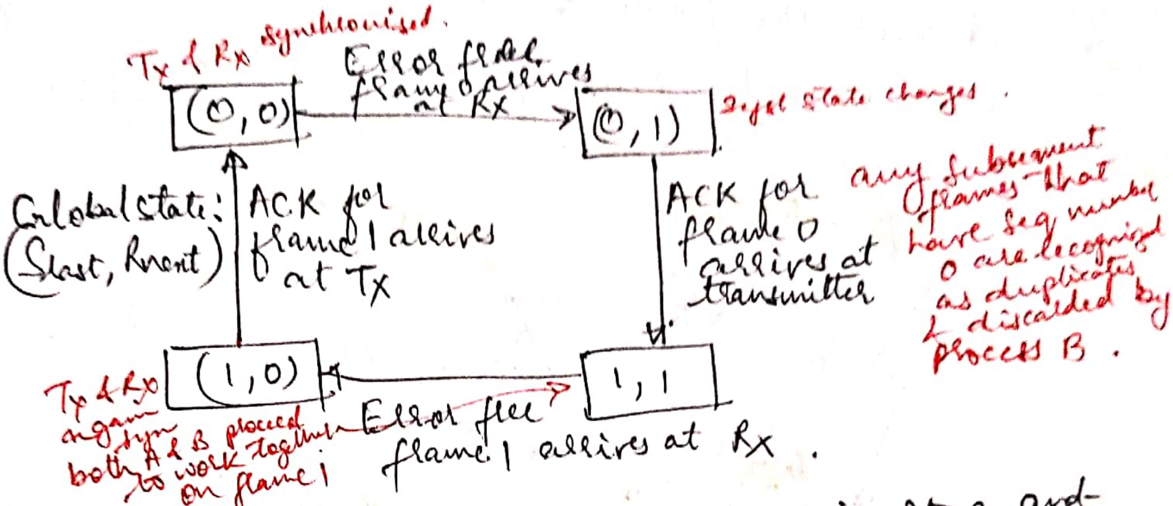


fig: System State Information in Stop-and-Wait ARQ.

Tx: Ack with $R_{next} = (S_{last} + 1) \bmod 2$

Rx → Received seq no $S_{last} = R_{next}$.

received seq no updated to

$$\boxed{S_{last} = (R_{next} + 1) \bmod 2.}$$

☆ Frame Sequence Number as well as ack frame Number both are one-bit & included in the Header field.

Performance Issues (Stop-and-Wait ARQ) → Efficiency
→ Delay BW product.

Stop-and-wait ARQ

5

Efficient when channels have low propagation delay

Inefficient when channels have much greater propagation delay than the time to transmit the frame.

Let frame size = 1000 bits

~~(Prop. delay)~~ Channel Speed = 1.5 Megabits / sec.

Time to transmit a frame & receive an acknowledgement = 40 ms. (propagation delay)

Thus, No of bits that can be transmitted over this channel = 40ms × 1.5 Mbits/s.
= 60000 bits

However, Stop-and-wait ARQ can transmit only 1000 bits in 40ms period.

Thus the efficiency is only $\frac{1000}{60000}$ = 1.6%

O/P — I/P.

Delay-bandwidth product

Delay-bandwidth product = Bit rate × delay.

= 1.5 Mbits/sec × 40 ms

= 60,000 bits

If we visualize the communication channel as a pipe, then the delay-bandwidth product can be interpreted as the size of the pipe, that is, the maximum number of bits that can be in transit at any given time. In Stop-and-wait ARQ the delay-bandwidth product can be viewed as a measure of lost opportunity in terms of transmitted bits.

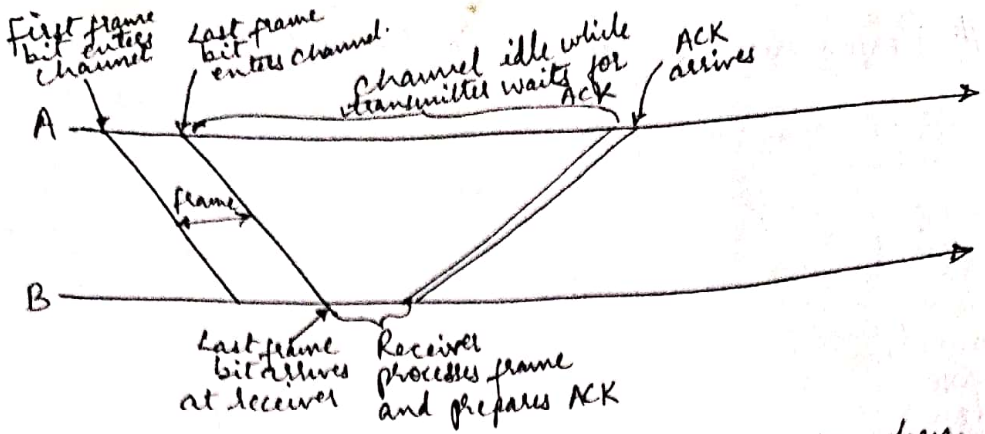


fig: Stop-and-wait ARQ is inefficient when time to receive an ACK (propagation delay) is large compared to the frame transmission time.

Transmission Efficiency of Stop-and-Wait ARQ.

Suppose that all information frames have the same length and that the transmitter has frames to transmit to the receiver.

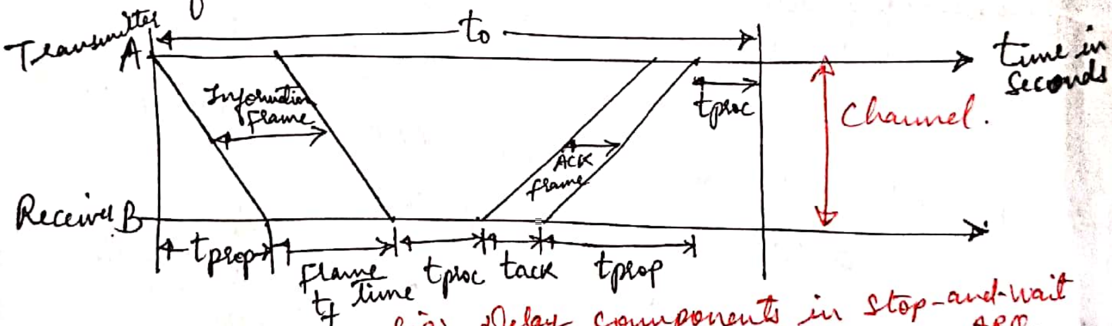


fig: Delay components in Stop-and-wait ARQ.

Let t_0 be the total Time delay incurred from the instant when I-frame is transmitted & acknoledgement is received.

Referring above fig from left to right;

t_{prop} be the Time taken by the first bit of I-frame to get thro' the channel & reach at receiver B.

t_f be the I-frame transmission Time.

t_{proc} be the time taken by the receiver to prepare an acknowledgement.

t_{ack} be the ack-frame transmission time.

t_{prop} be the time taken by the last bit of ack frame to get thro' the channel & reach at transmitter A.

t_{proc} be the time taken by CRC check. 6

Then, the basic time to send a frame and receive an ack, in the absence of errors, is given by

$$t_0 = t_{prop} + t_f + t_{proc} + t_{ack} + t_{prop} + t_{proc}$$

$$\boxed{t_0 = 2t_{prop} + 2t_{proc} + t_f + t_{ack}}$$

Now, let n_f = number of bits in information frame

n_a = number of bits in acknowledgement frame (or NAK frame)

& R = bit rate of transmission channel

Then $R = \frac{n_f}{t_f} = \frac{n_a}{t_{ack}}$

$\therefore t_f = \frac{n_f}{R}$ & $t_{ack} = \frac{n_a}{R}$

$$\therefore \boxed{t_0 = 2t_{prop} + 2t_{proc} + \frac{n_f}{R} + \frac{n_a}{R}}$$

Let n_0 be the number of overhead bits in a frame

i.e. n_0 = number of bits in header (Control bits, Sequence no bits, addr bits etc) (address)

+

number of bits in CRC check.

Then, number of information bits delivered to destination $= \underline{n_f - n_0}$ (Total no of bits in I-frame − overhead bits)

Thus, the effective transmission rate of the protocol in the absence of errors is,

$$R^0_{eff} = \frac{\text{No of information bits delivered to the destination}}{\text{Total time required to deliver the information bits}}$$

$$\boxed{R^0_{eff} = \frac{n_f - n_0}{t_0}}$$

Let η_o be the ~~transmission~~ efficiency of stop-and-wait ARQ and is given by

$$\eta_o = \frac{\underset{\text{(effective efficiency)}}{\text{O/P efficiency}}}{\underset{\text{(channel efficiency)}}{\text{I/P efficiency}}} = \frac{R_{eff}^{o}}{R}$$

$$\therefore \eta_o = \frac{\frac{n_f - n_o}{t_o}}{R}$$

$$= \frac{\frac{n_f - n_o}{2t_{prop} + 2t_{proc} + \frac{n_f}{R} + \frac{n_a}{R}}}{R} \quad ; \text{Substitute for } t_o$$

$$= \frac{1}{R}\left(\frac{n_f - n_o}{\frac{n_f + n_a + 2(t_{prop} + t_{proc})R}{R}}\right) \quad ; \text{take LCM}$$

$$\boxed{\eta_o = \frac{1 - \frac{n_o}{n_f}}{1 + \frac{n_a}{n_f} + \frac{2(t_{prop} + t_{proc})R}{n_f}}} \quad ; \text{Divide } N^r \text{ \& } D^r \text{ by } n_f$$

$\frac{n_o}{n_f} \Rightarrow$ loss in transmission efficiency due to the need of headers & CRC.

$\frac{n_a}{n_f} \Rightarrow$ loss in efficiency due to the time needed to prepare ACK/NAK frame.

$\underbrace{2(t_{prop} + t_{proc})}_{\text{reaction time}}R \Rightarrow$ Delay-Bandwidth product (of reaction time)

(Delay time X Bit rate)

~~unit is~~ = secs X bits/secs = bits

Thus unit is bits

Effect of Delay-Bandwidth product

Calculate the transmission efficiencies of the Stop-and-Wait ARQ in a system that transmits I-frames of 1250 bytes including 25 bytes of overhead, ack frames of 25 bytes, transmission rate of 1Mbps & reaction times of 1ms, 10ms, 100ms & 1 sec. Repeat the same example for transmission rate of 1Gbps. Comment on the effect of Delay-BW product as well as the effect of transmission rate on transmission efficiency.

Solution:

Given that:

(i) $n_f = 1250 \text{ bytes} = 1250 \times 8 = 10{,}000 \text{ bits}$

$n_o = 25 \text{ bytes} = 25 \times 8 = 200 \text{ bits}$

$n_a = 25 \text{ bytes} = 25 \times 8 = 200 \text{ bits}$

$R = 1\text{Mbps} = 10^6 \text{ bps}$

$2(t_{prop}+t_{proc}) = 1\text{ms}, 10\text{ms}, 100\text{ms} \,\&\, 1\text{s}$

$= 10^{-3}\text{s}, 10^{-2}\text{s}, 10^{-1}\text{s} \,\&\, 1\text{s}$

(ii) $R = 1\text{Gbps} = 10^9 \text{ bps}$.

We have, $$\eta_o = \frac{1 - \frac{n_o}{n_f}}{1 + \frac{n_a}{n_f} + \underbrace{\frac{2(t_{prop}+t_{proc})R}{n_f}}_{\text{delay BW product}}}$$

(i) (a) $$\eta_o = \frac{1 - \frac{200}{10000}}{1 + \frac{200}{10000} + \frac{10^{-3} \times 10^6}{10000}} = \frac{0.98}{1.02}$$

~~$9.79 \times 10^{-4} = 0.000979$~~

$= \cancel{0.9598}$

$\therefore \cancel{\eta_o = 95.98\%}$

$$\eta_o = \frac{1 - 0.02}{1 + 0.02 + 0.1} = \frac{0.98}{1.12} = 0.875$$

$\boxed{\eta_o = 87.5\%} \Rightarrow 88\%$

(b) $$\eta_o = \frac{1 - \frac{n_a}{n_f}}{1 + \frac{n_a}{n_f} + \frac{2(t_{prop} + t_{proc})R}{n_f}}$$

$$= \frac{1 - 0.02}{1 + 0.02 + \frac{10 \times 10^{-3} \times 10^6}{10^4}}$$

$$= \frac{0.98}{1 + 0.02 + 1} = \frac{0.98}{2.02}$$

$\boxed{\eta_o = 48.51\%} \simeq 49\%$

(c) $$\eta_o = \frac{1 - 0.02}{1 + 0.02 + \frac{100 \times 10^{-3} \times 10^6}{10^4}}$$

$$= \frac{0.98}{1 + 0.02 + 10} = \frac{0.98}{11.02}$$

$\boxed{\eta_o = 8.89\%} \simeq 9\%$

(d) $$\eta_o = \frac{1 - 0.02}{1 + 0.02 + \frac{1 \times 10^6}{10^4}}$$

$$= \frac{0.98}{1 + 0.02 + 100} = \frac{0.98}{101.02}$$

$\boxed{\eta_o = 0.97\%} \simeq 1\%$

Thus it is clear that as prop delay ↑ i.e. delay BW product ↑ the transmission efficiency decreases.

(ii) (a) $$\eta_o = \frac{1-\frac{n_o}{\eta_f}}{1+\frac{n_a}{\eta_f}+\frac{2(t_{prop}+t_{proc})R}{n_f}}$$

$$= \frac{1-0.02}{1+0.02+\frac{10^{-3}\times10^{9}}{10000}}$$

$$= \frac{0.98}{1+0.02+100} = \frac{0.98}{101.02}$$

$$\boxed{\eta_o = 0.97\% \simeq 1\%}$$

(b) $$\eta_o = \frac{1-0.02}{1+0.02+\frac{10\times10^{-3}\times10^{9}}{10^4}}$$

$$= \frac{0.98}{1+0.02+1000} = \frac{0.98}{1001.02}$$

$$\boxed{\eta_o = 0.097\% \simeq 0.1\%}$$

Thus, transmission efficiency decreases with the increase in Delay-Bandwidth product as well as with the increase in transmission rate.

(c) $$\eta_o = \frac{1-0.02}{1+0.02+\frac{100\times10^{-3}\times10^{9}}{10^4}}$$

$$= \frac{0.98}{1+0.02+10000} = \frac{0.98}{10001.02}$$

$$\boxed{\eta_o = 0.00979\% \simeq 0.01\%}$$

(d) $$\eta_o = \frac{1-0.02}{1+0.02+\frac{1\times10^{9}}{10^4}}$$

$$= \frac{0.98}{1+0.02+100000} = \frac{0.98}{100001.02}$$

$$\boxed{\eta_o = 0.000979\% \simeq 0.001\%}$$

Thus, when transmission rate is increased to 10^9 bits/s (ie 1 Gbits/s) the Transmission efficiencies drop down to the range from 1% to 0.001% for the corresponding propagation delays of 1 ms to 1 s.

Effect of transmission Errors on the efficiency of Stop-and-wait ARQ:

If a frame incurs errors during transmission, the time-out mechanism will cause retransmission of the frame.

Each transmission or retransmission will take upto t_0 seconds.

Let P_f be the probability that a frame transmission has errors & needs to be retransmitted.

Then $(1-P_f)$ will be the probability that a frame gets thro' without any errors

Suppose that 1 in 10 frame transmissions get through without error.

Therefore, $\boxed{1-P_f = \frac{1}{10} = 0.1}$

In general, $\frac{1}{(1-P_f)}$ frame transmissions are required for general values of P_f if frame transmissions errors are independent.

Let t_{SW} be the time in seconds required in stop-and-wait ARQ to transmit ~~the~~ frames.

Then $\boxed{t_{SW} = t_0 \cdot \frac{1}{1-P_f}}$

$$\eta_{SW} = \frac{R_{eff}^{error}}{R} = \frac{n_f - n_o}{t_{SW}} \cdot \frac{1}{R}$$

$$\& \; t_{SW} = t_0 \cdot \frac{1}{1-P_f}$$

Thus, the transmission efficiency of stop-and-wait ARQ in case of errors;

$$\eta_{SW} = \frac{\frac{n_f - n_o}{t_{SW}}}{R}$$

Scanned with CamScanner

$$\therefore \eta_{sw} = \frac{\frac{n_f - n_o}{\frac{t_o}{(1-P_f)}}}{R}$$

$$= \frac{\frac{n_f - n_o}{t_o} \cdot (1-P_f)}{R}$$

$$\boxed{\eta_{sw} = \left(\frac{1 - \frac{n_o}{n_f}}{1 + \frac{n_a}{n_f} + \frac{2(t_{prop} + t_{proc})R}{n_f}} \right) (1-P_f)}$$

$$\boxed{\eta_{sw} = \eta_o (1-P_f)}$$

Effect of Bit Error Rate (P)

Calculate the transmission efficiencies of Stop-and-Wait ARQ in a system that transmits I-frames of 1250 bytes including overhead of 25 bytes, ack frames of 25 bytes, transmission rate of 1 Mbps & reaction time of 1 ms for channels with a bit error rate of 10^{-6}, 10^{-5} and 10^{-4}. Comment on the effect of bit error rate on transmission efficiencies.

Solution:

Given that

$n_f = 1250$ bytes $= 1250 \times 8 = 10{,}000$ bits

$n_a = 25$ bytes $= 25 \times 8 = 200$ bits

$n_o = 25$ bytes $= 25 \times 8 = 200$ bits

$R = 1$ Mbps $= 10^6$ bps

$2(t_{prop} + t_{proc}) = 1\text{ms} = 10^{-3}$ s

ⓐ $P = 10^{-6}$ ⓑ $P = 10^{-5}$ ⓒ $P = 10^{-4}$.

Let P = Bit error rate ~~transmission~~ of error.
= probability of ~~transmission~~ single bit error.

Then $(1-P)$ = probability of transmission of single bit without error.

We have P_f = probability that a frame transmission has errors

& $1-P_f$ = probability that a frame transmission has no errors.

Since n_f is the no of bits in I-frame, the relation between $(1-P_f)$ & $(1-P)$ is given by;

$$\boxed{(1-P_f) = (1-P)^{n_f}}$$

no of bits in I-frame.

probability that a frame transmission is without errors = (probability of transmission of single bit without error)

We also have,

$$\boxed{\eta_{SW} = \eta_0 (1-P_f)}$$

Now let us calculate transmission efficiencies η_{SW} for different bit error rates, P when

$$\eta_0 = \frac{1-\frac{n_o}{n_f}}{1+\frac{n_a}{n_f} + \frac{2(t_{prop}+t_{proc})R}{n_f}}$$

$$= \frac{1-\frac{200}{10000}}{1+\frac{200}{10000} + \frac{10^{-3}\times 10^6}{10^4}}$$

$$= \frac{1-0.02}{1+0.02+0.1} = \frac{0.98}{1.12} = 0.875$$

$$\boxed{\eta_0 = 87.5\%}$$

(a) $P = 10^{-6}$, $\eta_o = 87.5\% = 0.875$

$(1-P_f) = (1-P)^{n_f} = (1-10^{-6})^{10000}$ 10

$\therefore \boxed{1-P_f = 0.99}$

& $\eta_{sw} = \eta_o (1-P_f)$

$= 0.875(0.99)$

$= 0.866$

$\therefore \boxed{\eta_{sw} = 86.6\%}$

(b) $P = 10^{-5}$, $\eta_o = 87.5\% = 0.875$

$(1-P_f) = (1-P)^{n_f} = (1-10^{-5})^{10000}$

$\boxed{1-P_f = 0.905}$

& $\eta_{sw} = \eta_o (1-P_f)$

$= 0.875(0.905)$

$\therefore = 0.7917$

$\boxed{\eta_{sw} = 79.2\%}$

(c) $P = 10^{-4}$, $\eta_o = 87.5\% = 0.875$

$(1-P_f) = (1-P)^{n_f} = (1-10^{-4})^{10000}$

$\boxed{1-P_f = 0.3678}$

& $\eta_{sw} = \eta_o (1-P_f)$

$= 0.875(0.3678)$

$= 0.3218$

$\boxed{\eta_{sw} = 32.2\%}$

Thus, it is clear that as bit error rate increases the efficiency of stop-and-wait ARQ decreases.

SUMMARY (Stop & Wait ARQ)

- Stop & wait ARQ transmits a single frame at a time.
- The frame types are I-frame, Control frame, ACK/NAK frames
- I-frame has header followed by Information followed by CRC field.
- Control frame has header followed by CRC field
- Stop & wait ARQ is ambiguous when I-frames & ACK/NAK frames are not given sequence numbers.
- To remove this ambiguity one bit in header is used as sequence number from each of I-frame (Seq no Slast) and ACK/NAK frame (Seq no Rnext)
- When the ambiguity exists there may be duplication of I-frame or ACK/NAK frame or ~~there m~~ gaps may be generated in the transmission.
- Basic components of ARQ are I-frames, control frames, ACK/NAK frames & time out mechanism.
- <u>Advantage</u> : • Stop & wait ARQ is simple
  • Bandwidth is wasted since most of the times channel is Idle.
- <u>Disadvantage</u> : Stop & wait ARQ is less efficient
- Stop-and-Wait ARQ is efficient when ~~pro~~ channel offers low propagation delay.
- Stop-and-Wait ARQ is less efficient when channel offers much greater propagation delay than the time to transmit a frame.

Shubhangi Khede

11

- Transmission Efficiency of Stop-and-Wait ARQ in the absence of errors is given by

$$\boxed{\eta_0 = \frac{1 - \frac{n_0}{n_f}}{1 + \frac{n_a}{n_f} + \frac{2(t_{prop} + t_{proc})R}{n_f}}}$$

where $2(t_{prop} + t_{proc})$ = reaction time = propagation delay

R = transmission rate

& $2(t_{prop} + t_{proc}) \times R$ = delay Bandwidth product.

Thus as propagation delay increases transmission efficiency decreases; as transmission rate increases transmission efficiency decreases & thus as Delay-Bandwidth product increases transmission efficiency decreases.

- Transmission efficiency of Stop-and-Wait ARQ with errors is given by

$$\boxed{\eta_{SW} = \eta_0 (1 - P_f)}$$

where P_f = probability that frame has transmission errors.

$1 - P_f$ = probability that frame has no transmission errors.

- $(1 - P_f) = (1 - P)^{n_f}$

where P = probability that single bit is in error = bit error rate.

$1 - P$ = probability that single bit transmission has no errors.

n_f = no of bits in I-frame.

- As Bit error rate increases the efficiency of Stop-and-Wait ARQ decreases.

12

Shubhangi Kharche -

Go-Back-N ARQ. [Sliding Window ARQ]

- Go-Back-N ARQ overcomes the inefficiency of Stop-and wait ARQ by pipelining the frames thereby keeping the channel busy most of the times.
- Go-Back-N ARQ sends frames numbered from 0 to W_s-1 i.e. if $W_s=8$ then it sends frames from 0 to 7.
- W_s is the window size which is chosen larger than the delay bandwidth product to ensure that the channel or pipe is kept full.

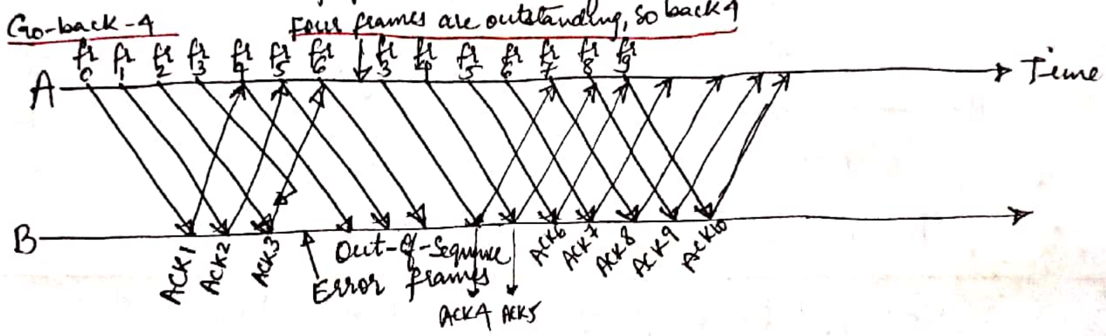
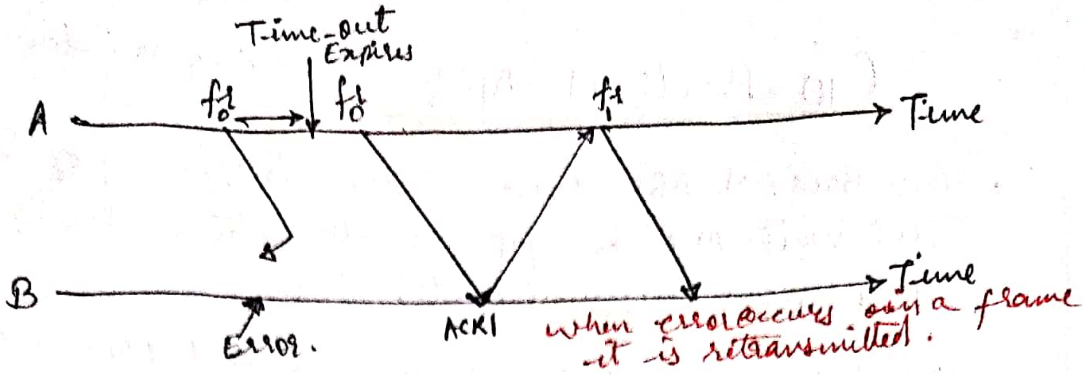


fig: Basic Go-Back-N. ARQ.

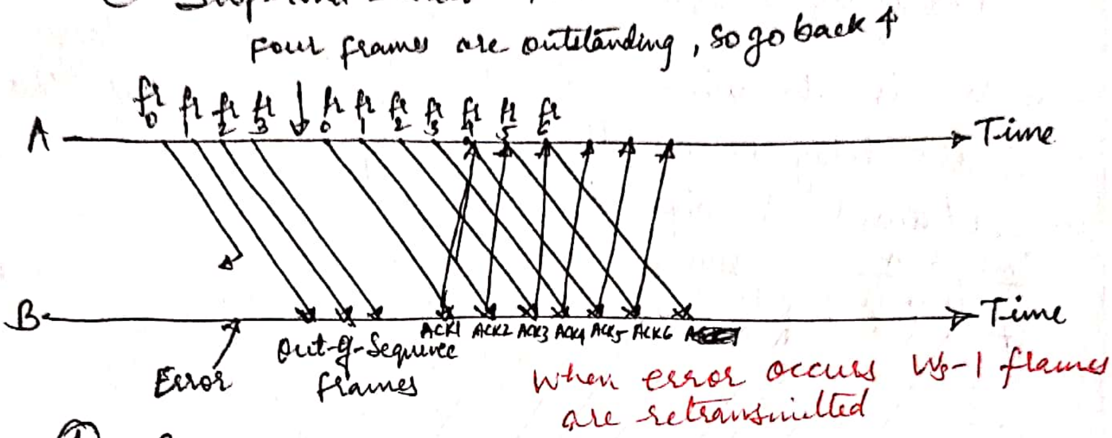
Refer above fig;

Transmitter A transmits frame 0 & continues transmission of subsequent frames till W_s i.e. transmitter A pipelines the transmission of frames. Meanwhile receiver sends ACK frames for correctly received frames. In the process receiver B i.e. fr 3 is in error. But Tx A does not receives frame 3 & does not receive ACK 4, already sends frames till fr 6 & starts retransmitting. Thus transmitter A has to Go-Back-4 frames from fr 3 to W_s.

Since Go-Back-N ARQ pipelines the frames to be transmitted the channel is busy most of the times, therefore the bandwidth is not wasted.



(a) Stop-and-Wait ARQ.



(b) Go-Back-N ARQ.

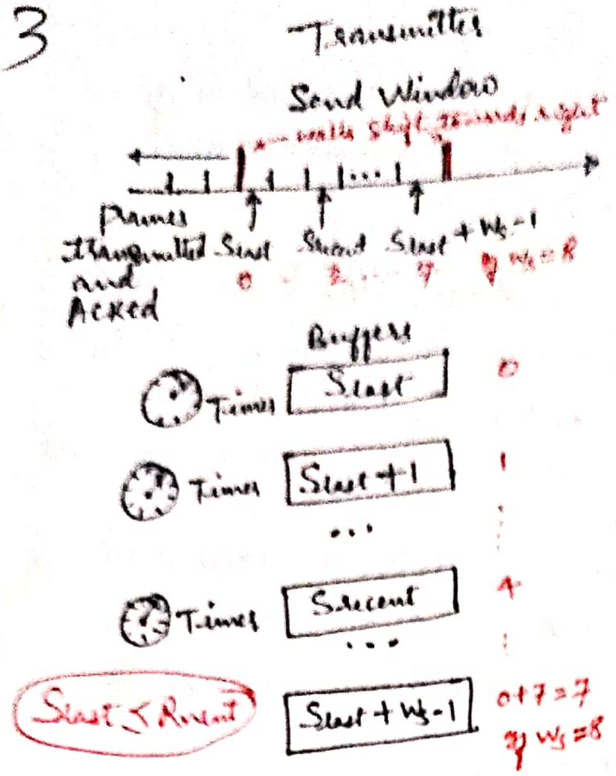
fig 2: Relationship of Stop-and-Wait ARQ and Go-Back-N ARQ.

Stop-and-Wait ARQ	Go-Back-N ARQ
In Stop-and-Wait ARQ, the occurrence of a frame transmission error results in the loss of transmission time ~~corresponding to W_s frames~~ equal to the duration of the time-out period.	In Go-Back-N ARQ the occurrence of a frame transmission error results in the loss of transmission time equal to the duration of W_s frames.

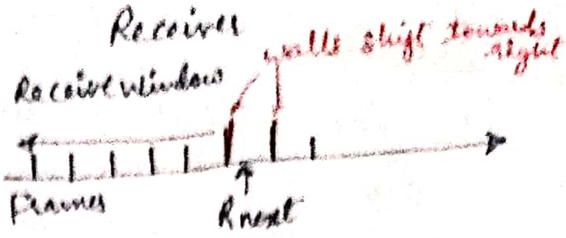
- Go-Back-N ARQ works correctly as long as the transmitter has an unlimited supply of packets that needs to be transmitted.
- In situations where packets arrive sporadically, there may not be W_s-1 subsequent transmissions. In this case retransmissions are not triggered, since the window is not exhausted.
- This problem is easily resolved by modifying Go-Back-N ARQ such that a timer is associated with each transmitted frame.

13

Send window size is W_s



Received window size is 1



The receiver will only accept a frame that is error free and that has sequence number Rnext.

Rnext ≤ Srecent

Fig 3: Go-Back-N ARQ [Modified since timer is associated with each frame]

Transmitter:

- The transmitter maintains a list of the frames it is processing.
- S_{last} = no of the last transmitted frame that is unacknowledged
- S_{recent} = no of the most recently transmitted frames.
- Transmitter maintains a timer for each transmitted frame. It also buffers all frames that have been transmitted but have not yet been acknowledged.
- The transmitter has a Send Window of available sequence numbers [i.e. Send window size is W_s.
  i.e. Sequence no's 0 to W_s-1]
- The lower ~~end~~ limit of the window is given by S_{last}, and the upper limit of the transmitter window is $S_{last}+W_s-1$ [i.e. If $W_s=8$ then $S_{last}=0$ & $S_{last}+W_s-1=7$]
- Whenever the transmitter receives an acknowledgement for the transmitted frame, the walls of the window shifts towards right. (slides)

Receiver:

- The receiver maintains a receive window of size 1 that consists of sequence number R_{next} which it expects to receive next.

- If an arriving frame undergoes the CRC check and has the correct sequence number, that is, R_{next}, then it is accepted and R_{next} is incremented & the receiver window slides forward.

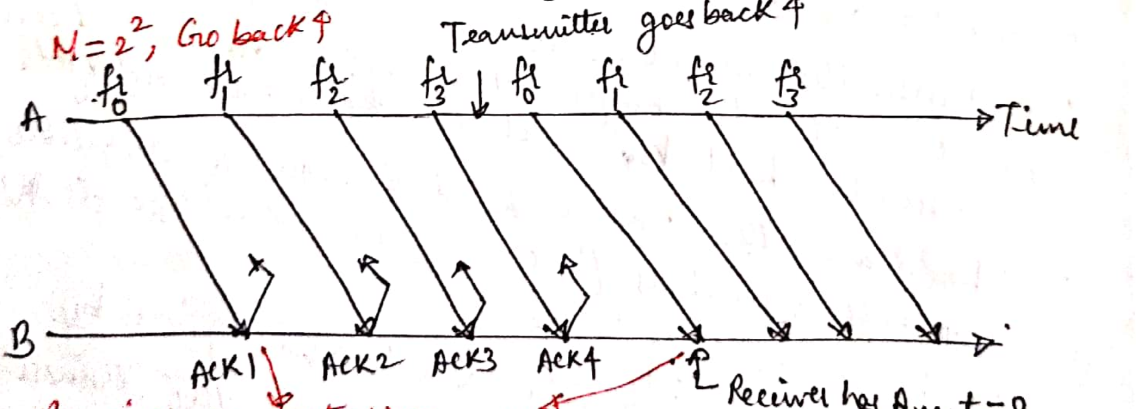
- The receiver then sends an acknowledgement containing the incremented sequence number R_{next}, which implicitly acknowledges receipt of all frames prior to R_{next}.

Note: $S_{last} \leq R_{next}$

$R_{next} \leq S_{recent}$

Maximum Window Size :

$M = 2^2$; Go back 4



Receiver cannot distinguish between old & new fr 0.

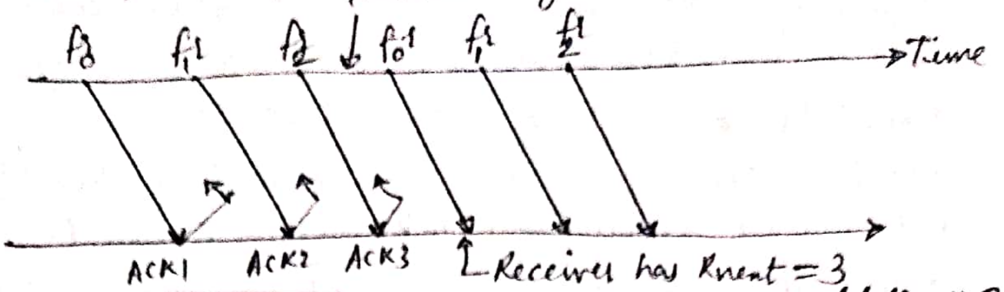
$\boxed{\text{Window size} = 2^m}$

↳ Results in duplication of frames

fig. 4 (a)

$M = 2^2 = 4$, Go-back-3 Transmitter goes back 3

14



$\boxed{\text{Window size} = 2^m - 1}$

No Duplication of frames

fig 4 (b)

fig 4: The window size should be less than 2^m.

- m-bits are reserved for sequence numbers within a header
- Thus, No of sequence numbers $= 2^m$
- If $m=3$ then $2^3 = 8$ sequence numbers are available i.e. 0, 1, 2, 3, 4, 5, 6, 7, 0, 1, 2, 3, ... This implies that the sequence numbers wrap around when the count reaches 2^m.
- Since the sequence numbers wrap around the receiver must be able to unambiguously determine which frame has been received.

* The receiver can determine the correct frame if the window size is less than 2^m. $\boxed{\text{i.e. } W_S < 2^m}$

- Fig 4 above shows the ambiguities that arise when the window size is not less than 2^m.

In fig 4 (a) the example uses $2^m = 2^2 = 4$ sequence numbers i.e. 0, 1, 2, 3. A window size $W_S = 4$. The transmitter transmits four frames fr0, fr1, fr2, fr3. The receiver receives the frames and sends the acknowledgements. But they are obliterated (lost) in the medium (reverse channel). So the transmitter sends again all frames. This time receiver has updated the ack no to Rnext to 0. Thus it receives again same fr0 & cannot distinguish between old and new fr0. Thus there is duplication of frames.

In fig (4) (b) the example uses $2^m = 2^2 = 4$ sequence numbers i.e. 0, 1, 2, 3 and window size $W_s = 3$. The transmitter sends three frames fr0, fr1, fr2. Receiver receives the frames and sends acknowledgements. But they are obliterated (lost) in reverse channel. Thus transmitter goes back 3 and resends the three frames. This time receiver updates ack no to $R_{next} = 3$. Thus when it receives fr0 again it can clearly ~~distinguish~~ understand ~~between~~ that old & new fr0 are same & discards the old frame. Thus there is not duplication of data when window size is $\boxed{W_s = 2^m - 1}$

Stop-and-Wait ARQ	Go-Back-N ARQ
• Transmits one frame at a time	• Transmits number of frames equal to the window size W_s
• If a frame is in error then it is retransmitted	• If a frame is in error then that frame & all subsequent frames are retransmitted
• Sequence number is one bit both for I-frame & ack frame.	• Sequence numbers m bits for I-frame & ~~one~~ m bit for ack frame.
• No of sequence numbers $= 2^1 = 2$ (i.e. 0,1)	• No of sequence numbers $= 2^m$. If m = 3 then No of sequence numbers $= 2^3 = 8$ i.e. 0, 1, 2, 3, 4, 5, 6, 7.

Shubhangi Kharche

Bidirectional Links [Modified Go Back-N]

- Here Information flow is bidirectional.
- Both Tx & Rx can transmit as well as receive the frame.
- The acknowledgements are piggybacked in the headers of the I-frames.
- Piggybacking increases efficiency of bandwidth usage.

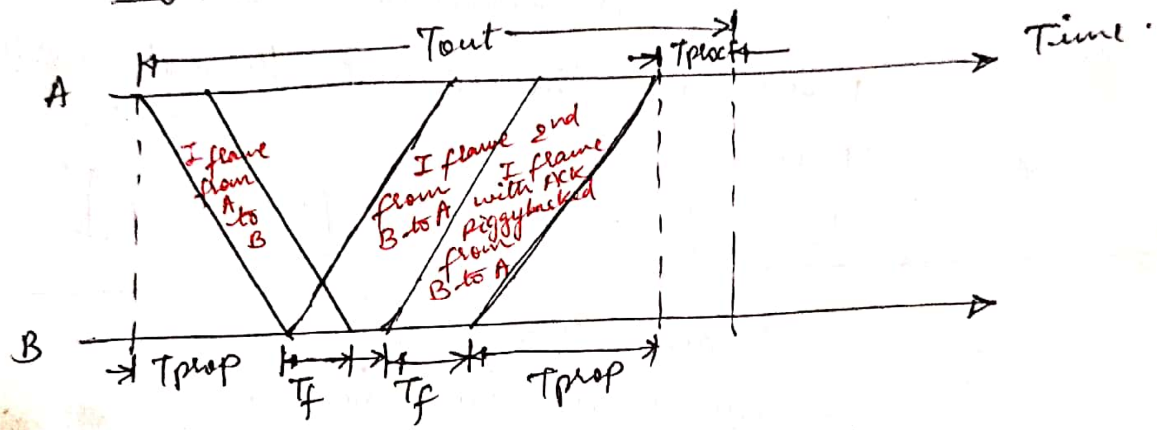


fig 5: Calculation of Time out Values.

- Here Timeout value, Tout is greater than the Time to transmit & receive an acknowledgement (similar as in Stop-and-Wait ARQ)
- Since links are bidirectional, here Both A & B can transmit as well as receive the I-frames.
- A sends I-frame to B which takes propagation time Tprop.
- B begins its I-frame transmission before receiving the I-frame from A. Thus Tf, time is required to transmit I-frames from A to B & B to A.
- B transmits 2nd I-frame by piggybacking the ack for received I-frame in the header of 2nd I-frame.
- 2nd I-frame from B to A takes another Tprop time.
- Thus the total round trip propagation is 2Tprop

Thus the time-out period is

$$\boxed{T_{out} = 2T_{prop} + 2T_f + T_{proc}}$$

Performance Issues

Let P_f = Probability that frames are transmitted with errors.

Then $1 - P_f$ = Probability that frames are transmitted without errors.

If 1 in 10 frames get through without error then

$$\boxed{1 - P_f = \frac{1}{10} = 0.1} \Rightarrow \boxed{P_f = 0.9}$$

If R = Channel transmission rate

n_f = no of bits in I-frame.

Then first transmission takes;

$t_f = \frac{n_f}{R}$ seconds.

With probability P_f (0.9) the first frame is in error, and so additional retransmissions are required.

Each time a retransmission is required, Go-Back-N transmits W_s frames, each of duration t_f, and the average no of retransmissions is $\frac{1}{(1-P_f)}$, i.e., 10 retransmissions

Therefore the total average time required to transmit a frame in Go-Back-N is

$$\boxed{t_{GBN} = t_f + P_f \frac{W_s t_f}{1-P_f}}$$

W_s frames each requiring t_f seconds.

$\frac{1}{1-P_f}$ no of transmissions.

probability that first frame is in error.

considering retransmission in case of errors.

put $P_f = 0.9$, $1-P_f = 0.1$, $\frac{1}{1-P_f} = 10$

$$\therefore t_{GBN} = t_f + \frac{0.9}{0.1} \cdot W_s t_f$$

$$\therefore \boxed{t_{GBN} = t_f + 9 W_s t_f}$$

Let n_o = no of overhead bits in I-frame.

Let R^o_{eff} = effective channel transmission rate

then $R^o_{eff} = \frac{n_f - n_o}{t_{GBN}}$

Thus, efficiency of Go-Back-N is given by:

$$\eta_{GBN} = \frac{R^o_{eff}}{R} = \frac{\frac{n_f - n_o}{t_{GBN}}}{R}$$

$$\therefore \eta_{GBN} = \frac{\frac{n_f - n_o}{\cancel{t_f} + 9W_s \cancel{t_f}}}{\frac{n_f}{\cancel{t_f}}} = \frac{\frac{n_f - n_o}{1+9W_s}}{n_f} = \frac{1 - \frac{n_o}{n_f}}{1+9W_s}$$

$$\eta_{GBN} = \frac{\frac{n_f - n_o}{t_{GBN}}}{R} \qquad \text{put } \boxed{t_{GBN} = t_f + P_f \frac{W_s t_f}{1-P_f}}$$

$$= \frac{\frac{n_f - n_o}{t_f + P_f \cdot \frac{W_s t_f}{1-P_f}}}{\frac{n_f}{\cancel{t_f}}} \qquad \because R = \frac{n_f}{t_f}$$

$$= \frac{n_f - n_o}{n_f\left[1 + \frac{P_f W_s}{1-P_f}\right]}$$

$$= \frac{1 - \frac{n_o}{n_f}}{(1-P_f) + P_f W_s}(1-P_f)$$

$$\boxed{\eta_{GBN} = \frac{1 - \frac{n_o}{n_f}}{1 + (W_s - 1) P_f}(1-P_f)}$$

Effect of Delay Bandwidth Product

Calculate the Transmission Efficiencies of Go-Back-N ARQ in a System that transmits I-frames of 1250 bytes including 25 bytes of overhead, ACK frames of 25 bytes, reaction time of 100ms, transmission rate of 1Mbps & bit error rates of 10^{-6}, 10^{-5}, and 10^{-4}. Comment on the Effect of Delay Bandwidth product and Bit error rate on the transmission efficiencies. Also compare the efficiencies with those of Stop-and-Wait ARQ.

Solution:

Given that

$n_f = 1250$ bytes $= 1250 \times 8 = 10000$ bits

$n_a = 25$ bytes $= 25 \times 8 = 200$ bits

$n_o = 25$ bytes $= 200$ bits

(i) $P = 10^{-6}$
(ii) $P = 10^{-5}$
(iii) $P = 10^{-4}$

$2(t_{prop} + t_{proc}) = 100\,ms = 0.1\,s$

$R = 1\,Mbps = 10^6\,bps.$

(I) Go-Back-N ARQ:

Delay-BW product $= 0.1 \times 10^6 = 10^5$ bits $= 1{,}00{,}000$ bits

1 frame $= 10000$ bits

$\therefore$ No of frames accomodated in Delay BW product of 100000 bits

is $= \frac{100000}{10000} = \underline{10 \text{ frames}}$ i.e 10 frames can pass thro the channel

For Go-Back-N ARQ frame size W_s must be greater than the delay-BW product.

$\therefore \boxed{W_s = 11 \text{ frames}}$

We have,

$$\eta_{GBN} = \frac{1 - \frac{n_o}{n_f}}{1 + (W_s - 1)P_f}(1 - P_f)$$

(i) $P = 10^{-6}$

$(1-P_f) = (1-P)^{n_f} = (1-10^{-6})^{10000}$

$\underline{1 - P_f = 0.99} \Rightarrow \boxed{P_f = 0.01}$

$$\therefore \eta_{GBN} = \frac{1 - \frac{200}{10000}}{1 + (11-1)0.01}(0.99)$$

$$= \frac{(0.98)(0.99)}{1.1}$$

$\boxed{\eta_{GBN} = 88.2\%}$

(ii) $P = 10^{-5}$

$(1-P_f) = (1-P)^{n_f} = (1-10^{-5})^{10000}$

$1 - P_f = 0.905 \Rightarrow \boxed{P_f = 0.095}$

$$\therefore \eta_{GBN} = \frac{1 - 0.02}{1 + (11-1)0.095}(0.905)$$

$\boxed{\eta_{GBN} = 45.5\%}$

(iii) $P = 10^{-4}$

$(1-P_f) = (1-P)^{n_f} = (1-10^{-4})^{10000} = \underline{0.3678}$

$\Rightarrow \boxed{P_f = 0.63}$

$$\therefore \eta_{GBN} = \frac{1 - 0.02}{1 + (11-1)0.63}(0.3678)$$

$\boxed{\eta_{GBN} = 4.9\%}$

Thus, it is clear that in GB-N ARQ as delay BW product & Bit error rate increases the transmission efficiency decreases.

(II) Stop & Wait ARQ.

We have:

$\eta_{SW} = \eta_0 (1 - P_f)$

$$\& \ \eta_0 = \frac{1 - \frac{n_o}{n_f}}{1 + \frac{n_a}{n_f} + \frac{2(t_{prop} + t_{proc})R}{n_f}}$$

(i) $\eta_0 = \frac{1 - 0.02}{1 + 0.02 + 10^5/10^4} = \frac{0.98}{11.02} = 0.0889$

$\& \ 1 - P_f = 0.99$

$\therefore \boxed{\eta_{SW} = 8.8\%}$

(ii) $\cancel{\eta_0 = \frac{1-0.02}{1+0.02+1}}$ $\eta_0 = \frac{0.98}{11.02} = 0.0889$

$\& \ \underline{1 - P_f = 0.905}$

$\therefore \boxed{\eta_{SW} = 0.0889 \times 0.905 = 8\%}$

(iii) $\eta_0 = 0.0889$ & $1 - P_f = 0.3678$

$\therefore \boxed{\eta_{SW} = 3.3\%}$

Thus, it is also clear that there is substantial improvement in transmission efficiencies of Go-Back-N ~~ARQ~~ relative to Stop & Wait ARQ for the first two bit error rates. For the third bit error rate there is marginal improvement.

Shubhangi Khelche

Selective Repeat ARQ

18

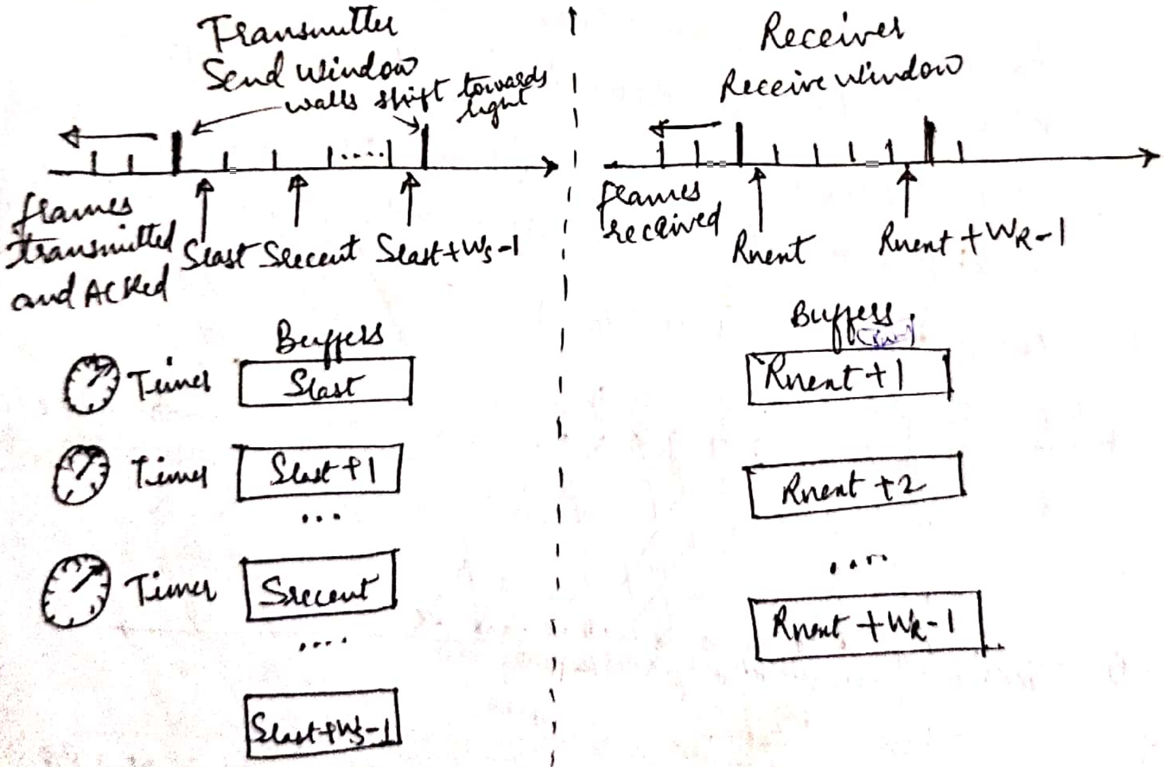
Disadvantage of Go-Back-N ARQ:

- In channels that have high error rates, the Go-Back-N ARQ protocol is inefficient because of the need to retransmit the frame in error and all the subsequent frames.

Selective Repeat ARQ is made more efficient than Go-Back-N ARQ by adding two features:

1) The receive window is made larger than one frame so that receiver can accept frames that are out of order & error free.

2) The retransmission mechanism is modified so that only individual frames are retransmitted.



fig①: Selective Repeat ARQ.

- In Selective repeat ARQ, the transmitter window size is W_s.
- The frames are transmitted with sequence numbers from 0 to $W_s - 1$
- whenever the frames are transmitted & acknowledged the walls of the window shift towards right.
- The Send Window advances the sequence numbers of the transmitted frames from S_{last} to $S_{last} + W_s - 1$.
- The receiver window size is W_R.
- The error free frames are received & buffered & acked with sequence numbers from ~~Rnext~~ 0 to $W_R - 1$.
- whenever the frames are received the walls of the window shift towards right.
- The receive window advances the ack frames sequence numbers from R_{next} to $R_{next} + W_R - 1$.

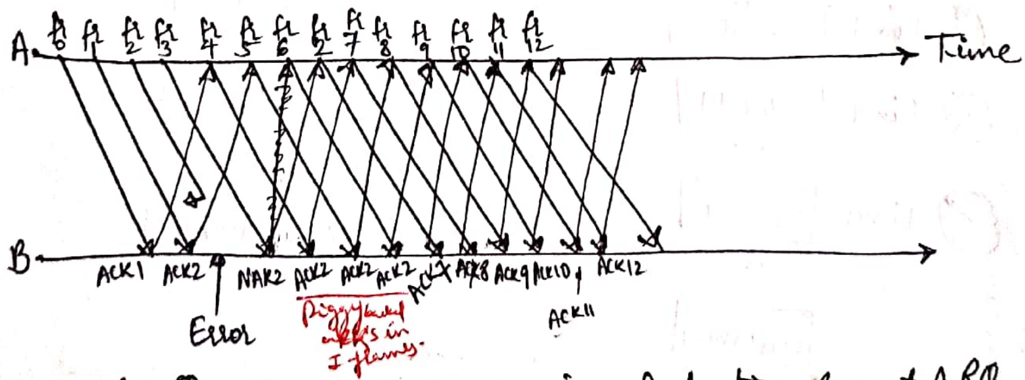


fig ②: Error recovery in Selective Repeat ARQ.

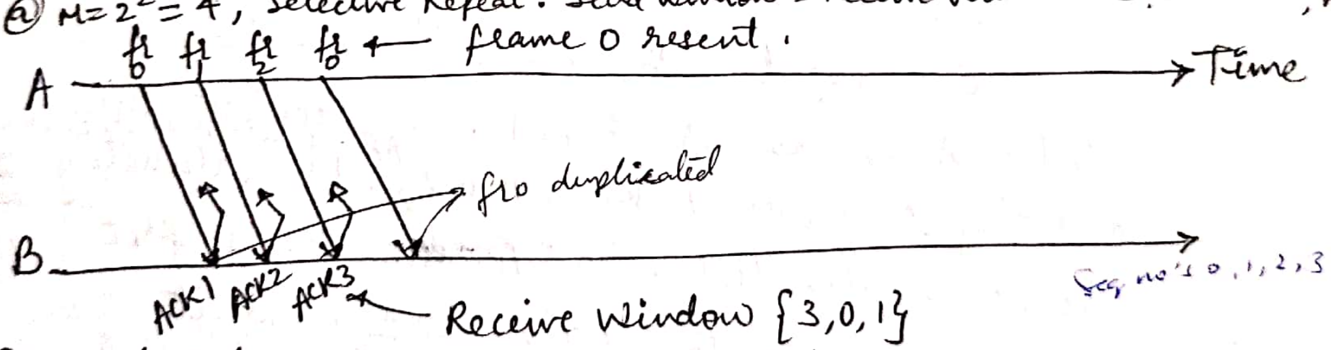
Scanned with CamScanner

In Selective Repeat ARQ, whenever a frame is in error & the timer expires then only the frame which is in error is retransmitted. whenever an out-of-sequence frame arrives at the receiver, a NAK frame is sent with sequence number R_{next}. when the transmitter receives such a NAK frame, it retransmits the specific frame, namely, R_{next}. The piggybacked acknowledgement in the information frames continues to carry R_{next}.

Again, when the frame with sequence number 2 finally arrives correctly at the receiver, frames 3, 4, 5 & 6 have already been received correctly. Consequently, the receipt of frame 2 results in sliding of the window forward by five frames.

Maximum Window Size

(a) $M = 2^2 = 4$, Selective Repeat: Send window = Receive window = 3 = $2^m - 1$; m=2



(b) Send window = receive window = 2 = 2^{m-1} ; m=2

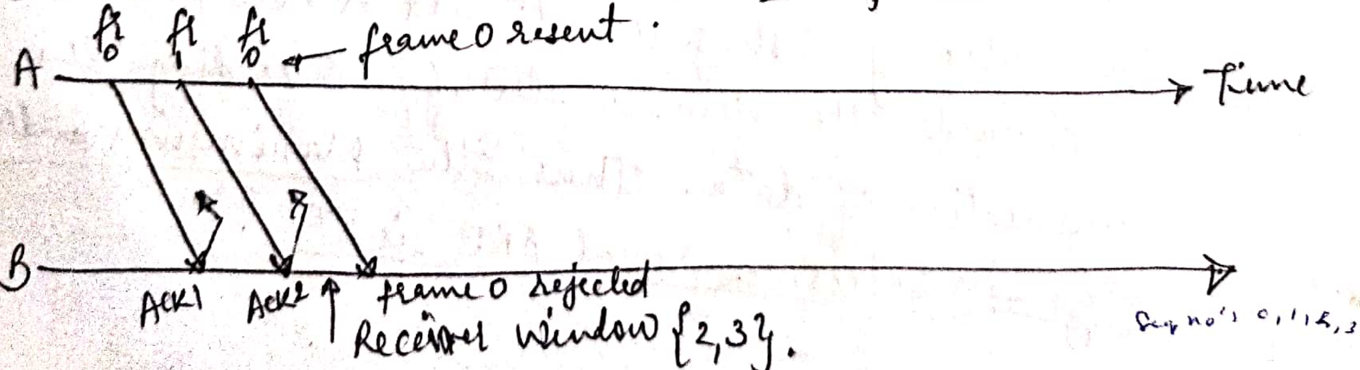


fig 3: Maximum Window Size in Selective Repeat ARQ.

~~In~~ Fig ③ ⓐ illustrates that there are 4 sequence numbers since m=2 bits are reserved in header for sequence numbering ($2^m = 2^2 = 4$). Hence the send window size is $2^m - 1 = 3$. Here receive window size is also 3. Station A transmits three frames fr0, fr1 & fr2. Station B receives all the three frames correctly. Unfortunately all the three acknowledgements are lost. ~~Thus station A see~~ After sending the ACK's the receive window is advanced to {3,0,1}

{3,0,1} $\because W_R = 3$
Rnext, Rnext+1, Rnext+2 Sequence no's 0,1,2,3
Rnext+WR-1

Since station A does not receive any acknowledgement it retransmits fr0 & thus there is duplication of fr0. On receiving fr0; B cannot distinguish between old frame 0 & new frame 0. Thus send & receive window size of $2^m - 1$ is too large.

~~In~~ Fig ③ ⓑ illustrates that for m=2; sequence no's $= 2^m = 2^2 = 4$ (i.e. 0,1,2,3). The send window & receive window size is $2^{m-1} = 2^{2-1} = 2^1 = 2$.

Station A sends fr0 & fr1. B receives them correctly & sends ACK1 & ACK2. After sending ACKs receive window advanced to {2,3}. Unfortunately the ACK's are lost. So A ~~re~~-sends fr0. But B expects to receive fr2. So it can clearly distinguish betn old & new frame 0 & discards the new fr0 thereby avoiding the duplication of data. Thus the Maximum Window size in Selective repeat ARQ is 2^{m-1}.

Performance Issues :

- Selective Repeat ARQ is the most efficient of the ARQ protocols as it uses the minimum number of retransmissions.

Let P_f = Probability of transmission with error.

Then $1-P_f$ = probability of transmission without error.

Average No of retransmissions required $= \frac{1}{1-P_f}$.

If t_f is the time in seconds to transmit the frame,

Then time to transmit the frame in selective repeat ARQ is

$$\boxed{t_{SR} = \frac{t_f}{1-P_f}}$$

Now let n_f = no of bits in I-frame

n_o = no of overhead bits due to header & CRC.

Then $R^o_{eff} = \frac{n_f - n_o}{t_{SR}} = \frac{n_f - n_o}{\frac{t_f}{1-P_f}}$

Let R be the channel transmission rate & $R = \frac{n_f}{t_f}$

∴ Efficiency of Selective repeat ARQ is

$$\eta_{SR} = \frac{R^o_{eff}}{R} = \frac{\frac{n_f - n_o}{\frac{\cancel{t_f}}{1-P_f}}}{\frac{n_f}{\cancel{t_f}}}$$

$$\therefore \eta_{SR} = \frac{n_f - n_o}{n_f} \cdot (1-P_f)$$

$$\boxed{\eta_{SR} = \left(1 - \frac{n_o}{n_f}\right)(1-P_f)}$$

If $P_f = 0$ i.e. probability of transmission with error is zero

then $\boxed{\eta_{SR} = 1 - \frac{n_o}{n_f}}$

Effect of Bit error Rate and Delay Bandwidth Product.

- Calculate the transmission efficiencies of Selective Repeat ARQ in a system that transmits I-frames of 1250 bytes including overhead of 25 bytes, ACK frames of 25 bytes, channel transmission rate of 1Mbps with reaction time of 100 ms for the bit error rates of 10^{-6}, 10^{-5} and 10^{-4}. Comment on the effect of bit error rate & delay BW product on the efficiency of Selective repeat ARQ. Also compare the efficiencies with those of GO-Back-N ARQ & Stop-and-wait ARQ.

[Ans (i) $\eta_{SR} = 97\%$ (ii) $\eta_{SR} = 89\%$ and (iii) $\eta_{SR} = 36\%$]

Solution:

Given that

$n_f = 1250$ bytes $= 10000$ bits

$n_o = 25$ bytes $= 200$ bits

$n_a = 25$ bytes $= 200$ bits.

(i) $p = 10^{-6}$ (ii) $p = 10^{-5}$ (iii) $p = 10^{-4}$

(i) $(1-P_f) = (1-p)^{n_f}$

$= (1-10^{-6})^{10000}$

$= 0.99$

$\eta_{SR} = \left(1 - \frac{n_o}{n_f}\right)(1-P_f)$

$= \left(1 - \frac{200}{10000}\right)(0.99)$

$= (0.98)(0.99)$

$\boxed{\eta_{SR} = 97.02\%}$

(ii) $(1-P_f) = (1-p)^{n_f}$

$= (1-10^{-5})^{10000}$

$\cancel{= 90.45\%}$

$= 0.905$

$\eta_{SR} = (0.98)(0.905)$

$\boxed{\eta_{SR} = 88.67\%}$

(iii) $\eta_{SR} = (0.98)(1-10^{-4})^{10000}$

$= (0.98)(0.3678)$

$\boxed{\eta_{SR} = 36.05\%}$

Thus, as Bit error rate increases efficiency of Selective repeat ARQ decreases.

Comparison of Efficiencies:

ARQ	Bit error rate $P=10^{-6}$	$P=10^{-5}$	$P=10^{-4}$
Stop-and-wait ARQ	8.8%	8%	3.3%
Go-Back-N ARQ	88.2%	45.5%	4.9%
Selective Repeat ARQ	97.02%	88.67%	36.05%

	Stop-and-Wait ARQ	Go-Back-N ARQ	Selective Repeat ARQ
Transmission Efficiency	$\eta_{SW} = \eta_0 \cdot (1-P_f) = \dfrac{(1-\frac{n_0}{n_f})\cdot(1-P_f)}{1+\frac{n_a}{n_f}+\frac{2(t_{prop}+t_{proc})R}{n_f}}$	$\eta_{GBN} = \dfrac{1-\frac{n_0}{n_f}}{1+(W_s-1)P_f}(1-P_f)$	$\eta_{SR} = \left(1-\frac{n_0}{n_f}\right)(1-P_f)$
	Neglecting header & CRC overhead	Neglecting header & CRC overhead	Neglecting header & CRC overhead
	$\eta_{SW} = \dfrac{(1-P_f)}{1+\frac{2(t_{prop}+t_{proc})R}{n_f}}$	$\eta_{GBN} = \dfrac{(1-P_f)}{1+(W_s-1)P_f}$	$\eta_{SR} = (1-P_f)$
	Let $L = \dfrac{2(t_{prop}+t_{proc})R}{n_f}$ = size of the pipe = multiply of frames = Delay BW product	In Go-Back-N $\boxed{W_s = L+1}$	
		$\therefore \eta_{GBN} = \dfrac{1-P_f}{1+(L+1-1)P_f}$	
	Then, $\boxed{\eta_{SW} = \dfrac{(1-P_f)}{1+L}}$	$\boxed{\eta_{GBN} = \dfrac{1-P_f}{1+LP_f}}$	$\boxed{\eta_{SR} = 1-P_f}$

Comments:
1) Go-Back-N ARQ is worse than Selective repeat ARQ by a factor of $(1+LP_f)$ & Stop-and-wait is worse than Selective repeat ARQ by a factor of $(1+L)$
2) If $LP_f << 1$ then $\boxed{\eta_{GBN} = \eta_{SR}}$
3) If P_f approaches 1 then LP_f becomes L & $\boxed{\eta_{GBN} = \eta_{SW}}$

ARQ Efficiency Comparison.

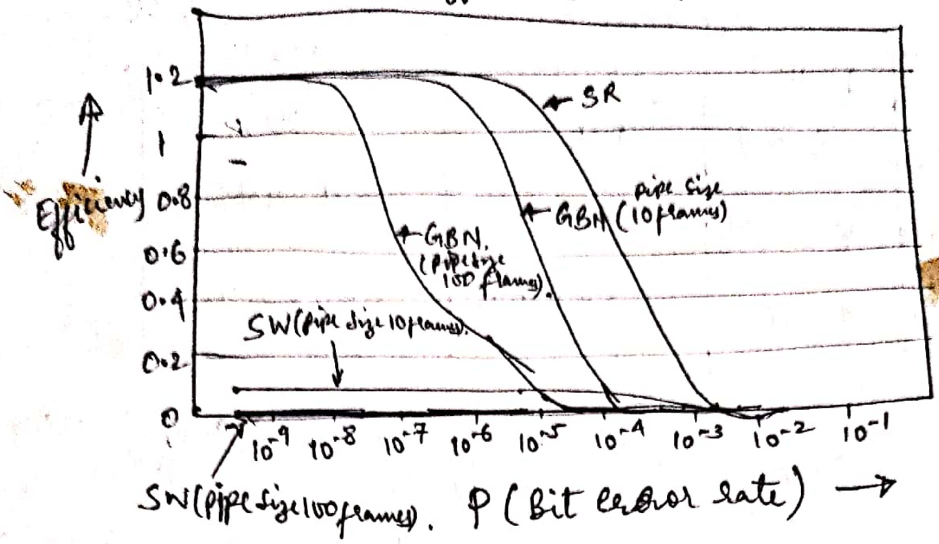


fig: Efficiency performance of ARQ protocols as a function of bit error rate for a frame size of 1250 bytes and for channels with delay-BW product of 10 & 100 frames.

Comparison of Stop-and-Wait, Go-Back-N and Selective repeat ARQ's.

Stop-and-Wait ARQ	GO-Back-N ARQ	Selective Repeat ARQ.
1) Only one frame is transmitted at a time.	Number of frames transmitted depend on the window size	Number of frames transmitted depend on the window size.
2) One bit is reserved in the header of I-frame & ack frames for sequence numbering.	m-bits are reserved in the header of I-frame & ack frames for sequence numbering.	m-bits are reserved in the header of I-frame & ack frame for sequence numbering
3) Not an example of Sliding Window protocol	Example of Sliding Window protocol. • Maximum Window size is $W_S = 2^m - 1$ • Transmitter Window size is W_S and receiver window size is only one frame.	Example of Sliding Window protocol. • Maximum Window size is $W_S = 2^{m-1}$ • Transmitter window size is W_S and receiver window size is W_R & they are equal. $\boxed{W_S = W_R}$
4) Whenever a frame is in error it is retransmitted.	4) Whenever a frame is in error, that frame & all subsequent frames are retransmitted	4) Whenever a frame is in error only that frame is retransmitted.
	$W_S >$ Delay BW product	$W_S >$ Delay BW. product.