15. Sep. 2018

* Computer Networks :-

A —channel / Medium— B

- A: Sender, transmitter, Host, Source
- Medium: wired, wireless
- B: Receiver, Host, destination.

TCP/IP stack

- Application layer (Protocols) → PoP3, SMTP, HTTP
- Transport layer (protocols) → TCP, UDP
- Network layer → IP (Routers)
- Data link layer
- Physical layer → Binary Data (If the form of Digital signal)

sender ↑ Receiver ↓

www. Yahoo.com

- www → Host name
- Yahoo.com → Domain Name
- com → First level

www.jo.edu.jo

- edu → First level
- jo → Second level.
- jo.edu.jo → Domain Name

---

* Each Layer offers many interesting protocols.

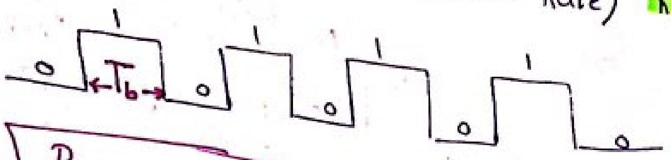
$\boxed{\text{* each Protocol} \longrightarrow \text{Service}}$

1. **Protocols:** Set of rules adheared by communicating devices (switch, Router) for the sake of having timely, recognizable reliable transfer of Information.
   - timely → synchronous
   - recognizable → Format
   - reliable → Accurancy of data.

2. **Computing:**
   - → Centralized —Ex→ Master / slave Model (old approach)
   - → Distributed
     - → Client / server Model
     - → Local Area Network (LANs)

3. **Performance:** (Network performance)
   - → Delay
     1. propagation Delay T_{prop}
     2. Transmission Delay T_{trans}
     3. Processing Delay T_{proc}
     4. Queuing Delay T_{Que}
   - → Throughput

1) T_{prop} → bit Rate (Data Rate) R_b



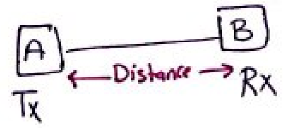
$$\boxed{R_b = \frac{1}{T_b}}$$ (bps) → bit per second

Note:

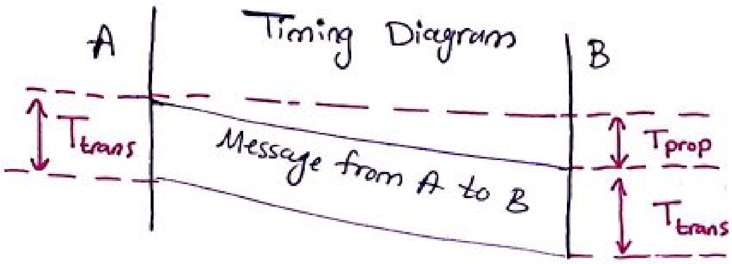
Raw bits (0's and 1's)

تتحول → signal → Digital / Analog

$$T_{prop} = \frac{d\,(m)}{\text{velocity}\,(m/s)}$$ (d → Distance)



velocity → Depends on channel



velocity

- Metal: 2.3×10^8 m/s, At the form of electric current, $\ll 3\times10^8$ m/s
- fiber: 3×10^8 m/s, At the form of light

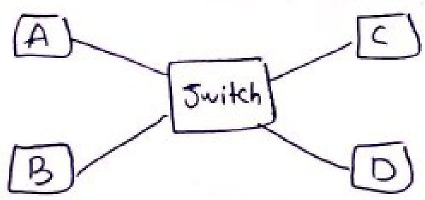
T_{prop} :- Time elapsed from the moment the first or any bit is sent till the moment it is received by destination.

2) T_{trans} : Time elapsed from the moment the first bit is transmitted till the moment the last bit of the message is transmitted

$$\boxed{T_{trans} = \frac{\text{length of the message (bits)}}{R_b\ (\text{bps})}}$$

** Note: propagation and transmission delays called <u>deterministic</u> → they have certein formula while the others called Non-deterministic

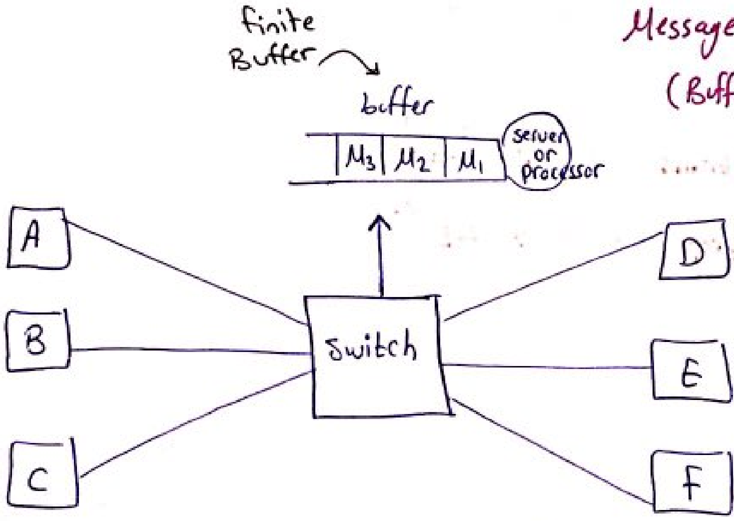
Note



18. Sep. 2018

Last lecture:

Delay
- → Propagation delay] deterministic → $T_{prop} = \frac{d(m)}{v(m/s)}$
- → Transmission delay] deterministic → $T_{trans} = \frac{L(bits)}{R_b(bps)}$
- → processing delay] Non-Deterministic → T_{proc} ? How fast switches or routers process receiving packets (messages)
- → Queuing delay] Non-Deterministic → T_{Que} ? Message waiting time (Buffers or Queue)

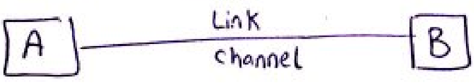


delay ↓ → performance

QoS → Quality of service

* Link effeciency (L_E)

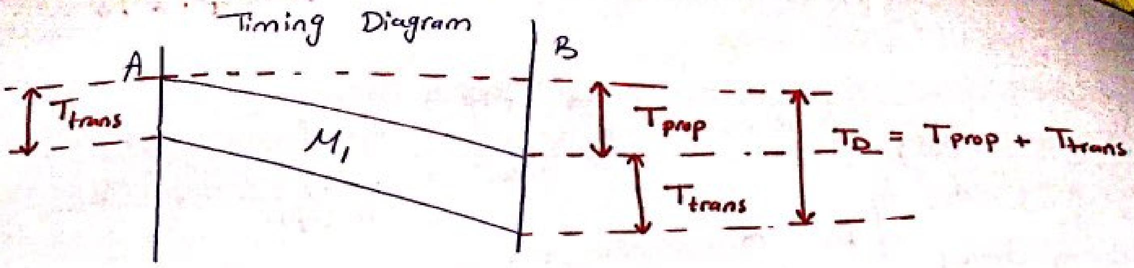
Case 1:



$L_E = \frac{T_{trans}}{T_D}$ → $T_D = T_{prop} + T_{trans} + T_{proc} + T_{Que}$

↳ End-to-End Delay

Timing Diagram



$$\rightarrow L_E\Big|_{\text{case ①}} = \frac{T_{trans}}{T_D} = \frac{T_{trans} \,/\, T_{trans}}{T_{prop} + T_{trans} \,/\, T_{trans}}$$

$$= \frac{1}{1 + \frac{T_{prop}}{T_{trans}}} = \frac{1}{1+\alpha} \quad , \quad \boxed{\alpha = \frac{T_{prop}}{T_{trans}}}$$

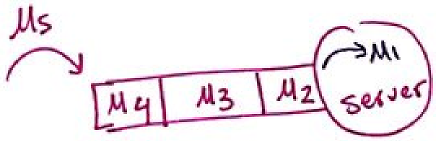
IF $T_{prop} \gg T_{trans} \longrightarrow L_E \downarrow$

$T_{trans} \ll T_{prop} \longrightarrow L_E \uparrow \rightarrow \underline{L_E = 1 \rightarrow 100\%}$

* T_{prop} has Direct effect on L_E

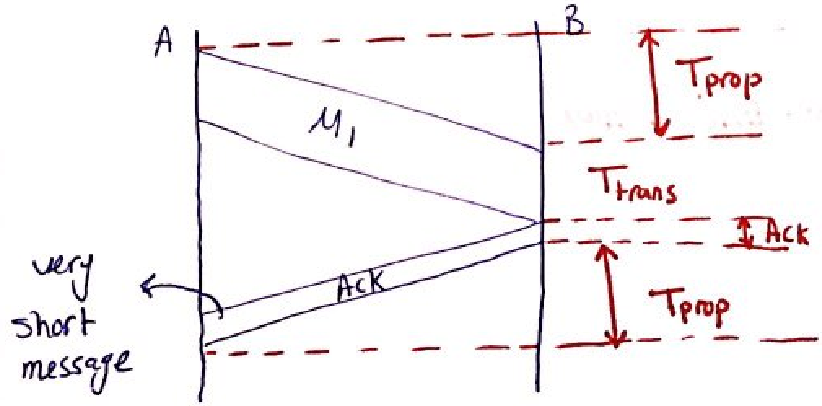
*** packet loss :-**

- Damage
  - Ex: affected by weather
- Drop
  - Buffer is finite (capacity is limited)



↳ the Buffer is full (there is no space) it will drop M5

Case 2:



* Does the Ack has an impact on the delay ?!

Note: Consider having

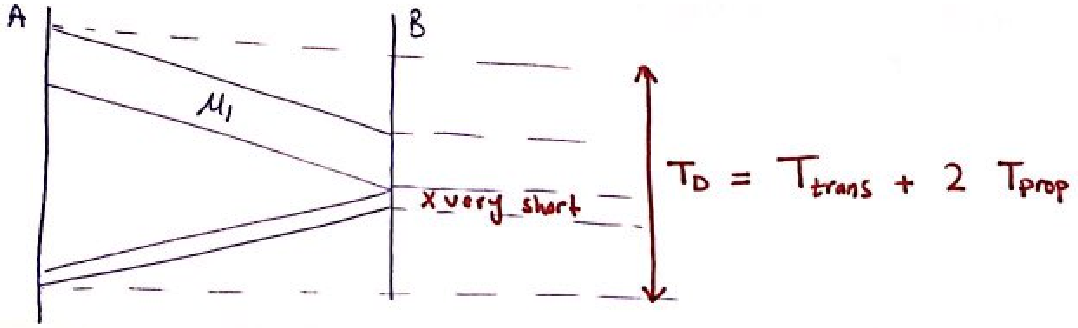
18. sep. 2018

Note :

Consider having very short message

$\rightarrow T_{Ack} = \frac{L \downarrow}{R_b}$

$\rightarrow T_{Ack} \approx 0$



$$L_E = \frac{T_{trans}}{T_D} = \frac{1}{1+2\alpha}, \quad \boxed{\alpha = \frac{T_{prop}}{T_{rans}}}$$

$$\underbrace{\frac{1}{1+2\alpha}}_{\text{case (2)}} \ll \underbrace{\frac{1}{1+\alpha}}_{\text{case (1)}}$$

So, Ack is an additional overhead

last lecture : 20.9.2018

→ Delay (T_{prop} , T_{trans} , T_{proc} , T_{Queue})

→ Link effeciency (L_E)

$$L_E = \frac{T_{trans}}{T_D}$$

→ Case ① (No Ack) → $L_E = \frac{1}{1+\alpha}$, $\alpha = \frac{T_{prop}}{T_{trans}}$

→ Case ② (Ack) → $L_E = \frac{1}{1+2\alpha} << \frac{1}{1+\alpha}$

Conclusion :- Ack is an additional overhead.

→ **Throughput ?!**

Is the rate at which the data is reliably delivered to the destination

Goal Is to maximize the throughput of the channel

Throughput is directly related to the channel capacity

$$* \text{ Throughput} = \frac{\text{Length of the message (bits)}}{T_D \text{ (End-to-End Delay) (sec)}}$$

Note $R_b = \frac{\text{length of message}}{T_{trans}}$

Source data rate (bit rate)

$$\text{Throughput}_{Case ②} < \text{Throughput}_{Case ①} < R_b$$

(ACK) (No Ack)

$$T_D \leftarrow \frac{L}{(T_{trans} + 2T_{prop})} \qquad \frac{L}{T_{trans} + T_{prop}} \qquad \frac{L}{T_{trans}}$$

Problems (Set #1)

Question#1: Consider a LAN with a maximum distance of 4km. At what bit rate would the propagation delay (at a propagation speed of 2.3×10^8 m/s) be equal to the transmission delay for 512-byte packets?

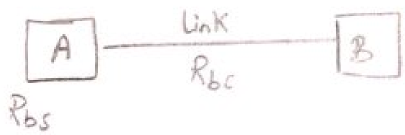
Question#2: Suppose there is a 100-Mbps point-to-point link between the earth and the moon. The distance from the moon to the earth is approximately 385,000 km, and data travels over the link at the speed of light (3×10^8 m/s). A camera on the moon takes pictures on the earth and saves them in digital format to disk. If the Mission Control on earth wants to download the most current image of size 5×10^6 bytes. What will be the amount of time that will elapse between the moment when the request goes out and the moment when transfer is completed?

Question#3: Hosts A and B are each connected to a switch S via 10-Mbps links as shown in the figure below. The propagation delay on each link is 20 μs. S is a store-and-forward device; it begins retransmitting a received packet 35 μs after it finished receiving it. Calculate the total time required to transmit 10,000 bits from A to B

(a) As a single packet;

(b) As two 5,000-bit packets sent one right after the other.

$\boxed{\text{Note :}}$



$R_{bs} \leq R_{bc}$

(Source) (Channel)

Question (1) :-

$$T_{trans} = T_{prop}$$

$$\frac{L\,(bits)}{R_b\,(bps)} = \frac{d\,(m)}{v\,(m/s)} \longrightarrow \frac{512 \times 8}{R_b} = \frac{4 \times 10^3\ m}{2.3 \times 10^8\ m/s}$$

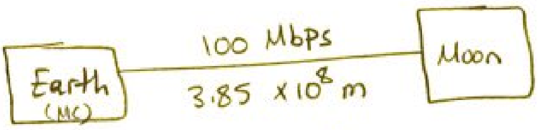
للتحويل لـ bits

$$R_b = 235.52\ Mbps$$

Question (2) :-

Note : Point-to-Point Link (Directly connected) without switch

A ——— B

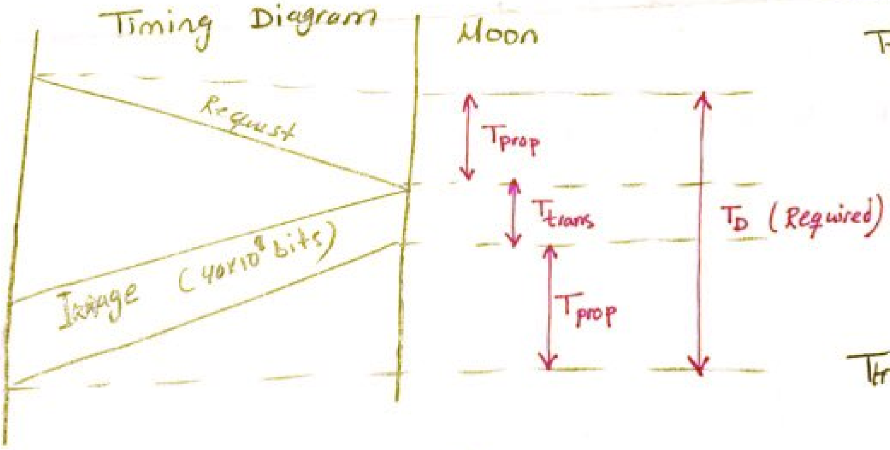


Propagation speed = 3×10^8 m/s

Message size = 5×10^6 bytes → 40×10^6 bits

Request

very short message so $T_{trans} \approx 0$



$$T_D = T_{trans} + 2\,T_{prop}$$

$$T_{prop} = \frac{d\,(m)}{v\,(m/s)} = \frac{3.85 \times 10^8\ m}{3 \times 10^8\ m/s}$$

$$\boxed{T_{prop} = 1.285\ sec}$$

$$T_{trans} = \frac{L}{R_b} = \frac{40 \times 10^6\ bits}{100 \times 10^6\ bps}$$

$$\boxed{T_{trans} = 0.4\ sec}$$

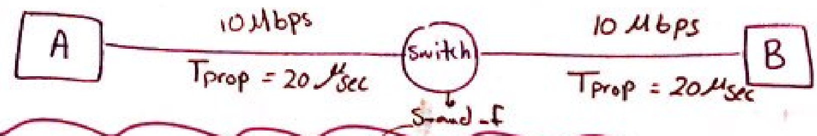
$$T_D = 0.4 + 2(1.285)$$

$$T_D = 2.97\ sec$$

Question (3) :-

* Question (3) :-

23. Sep. 2018



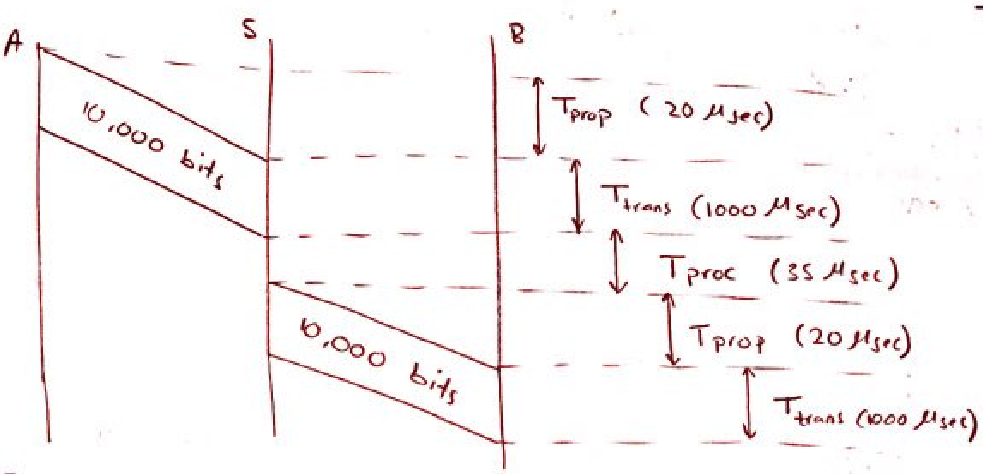
Switch → store-and-forward switching → بيستنى وصول ال packet كلها بعدين بيبلش processing

OR → Cut-through switching → بيستنى وصول ال Header

$T_{proc} = 35\ \mu sec$

Required :-

(a) T_D at the time of sending 10,000 bits together.

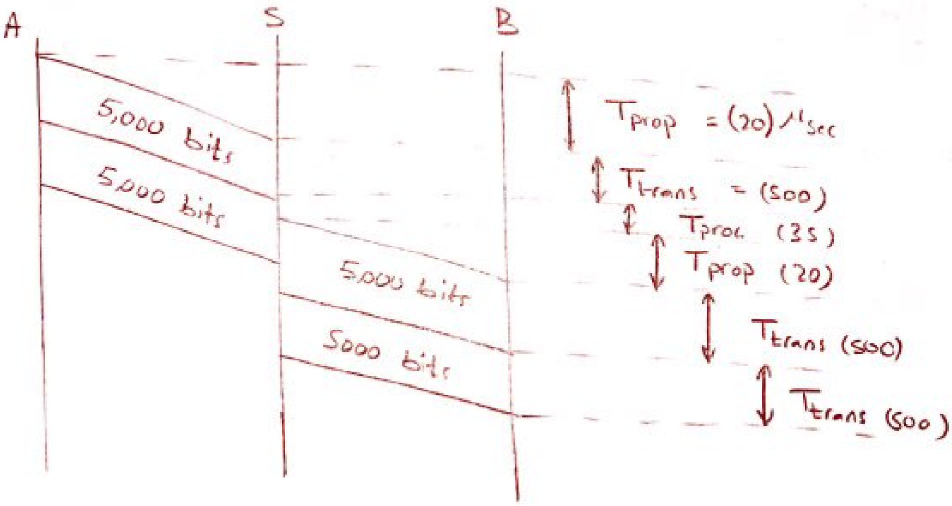


$$T_{trans} = \frac{10000 \text{ bits}}{10 \text{ Mbps}} = 1 \text{ msec} = 1000\ \mu sec$$

$$\boxed{T_D = 2075\ \mu sec}$$

(b) T_D when dividing the message into two equal messages ?!

Big Concern



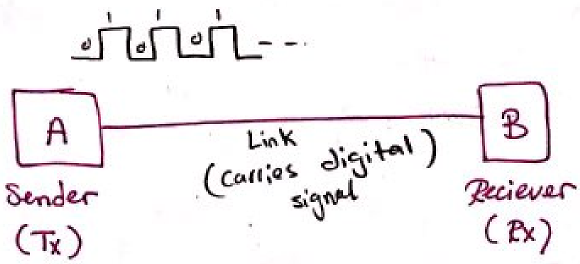
$$T_{trans} = \frac{5000 \text{ bits}}{10 \times 10^6 \text{ bps}} = 500\ \mu sec$$

$$\boxed{T_D = 1575\ \mu sec}$$

Great Reduction!

* splitting the message helps in reduce End-to-End Delay.

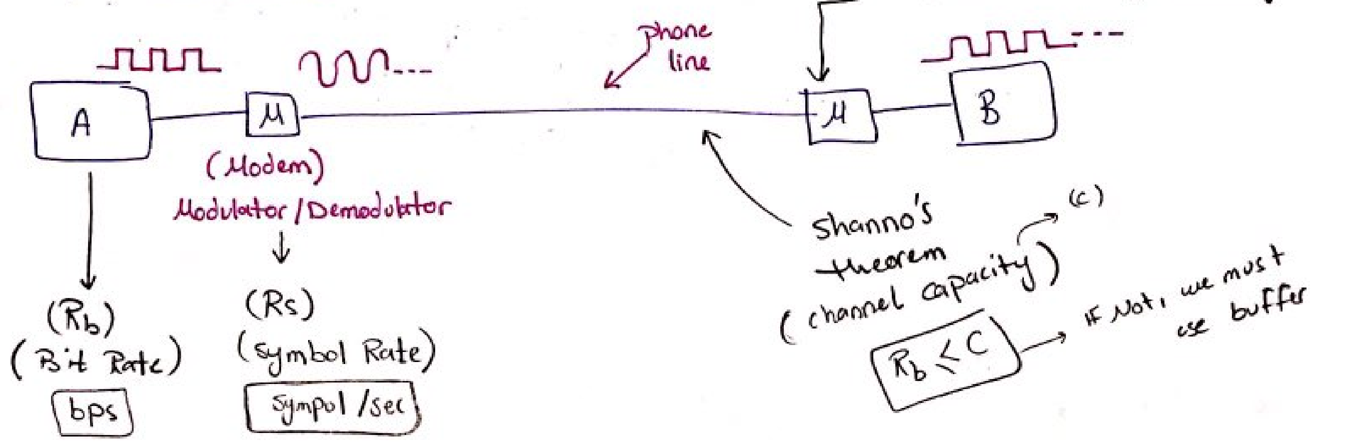
*** Physical Layer :-** Responsible of transmitting raw bits between two points.



In ADSL : link carries analog signal (phone Line)

Big picture :-

pulse code modulation (PCM) → Sampling (Nyquist theorem) → Quantaization → Encoding



$\boxed{\text{Problem}}$ In analog transmission, the bandwidth is small.

	Digital	Analog
① BW	High	low
② Error Rate	low	high
③ Device for Regenerating signal (weak signal)	Repeater	Amplifier

Ex: 3bits / symbol → حاصل الضرب لازم يساوي Data Rate

* x Symbols / sec — إذا طلع أكثر ح احتاج Buffer

$R_b = (3x \text{ bits / sec})$

Question:

How bits "0" and "1" are mapped into a digital signal ?!

* A lot of interesting schemes:

① Non-Return-to-Zero (NRZ)

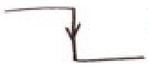
"0" :  "1" 

② NRZ-I (Inverted)

1: Transition of the current state

0: No transition.

③ Manchester :

1 :  High-to-low

0:  low-to-high

* Encoding schemes :-

25.9.2018

① Non-Return-to-Zero (NRZ)

1 → level high

0 → level low

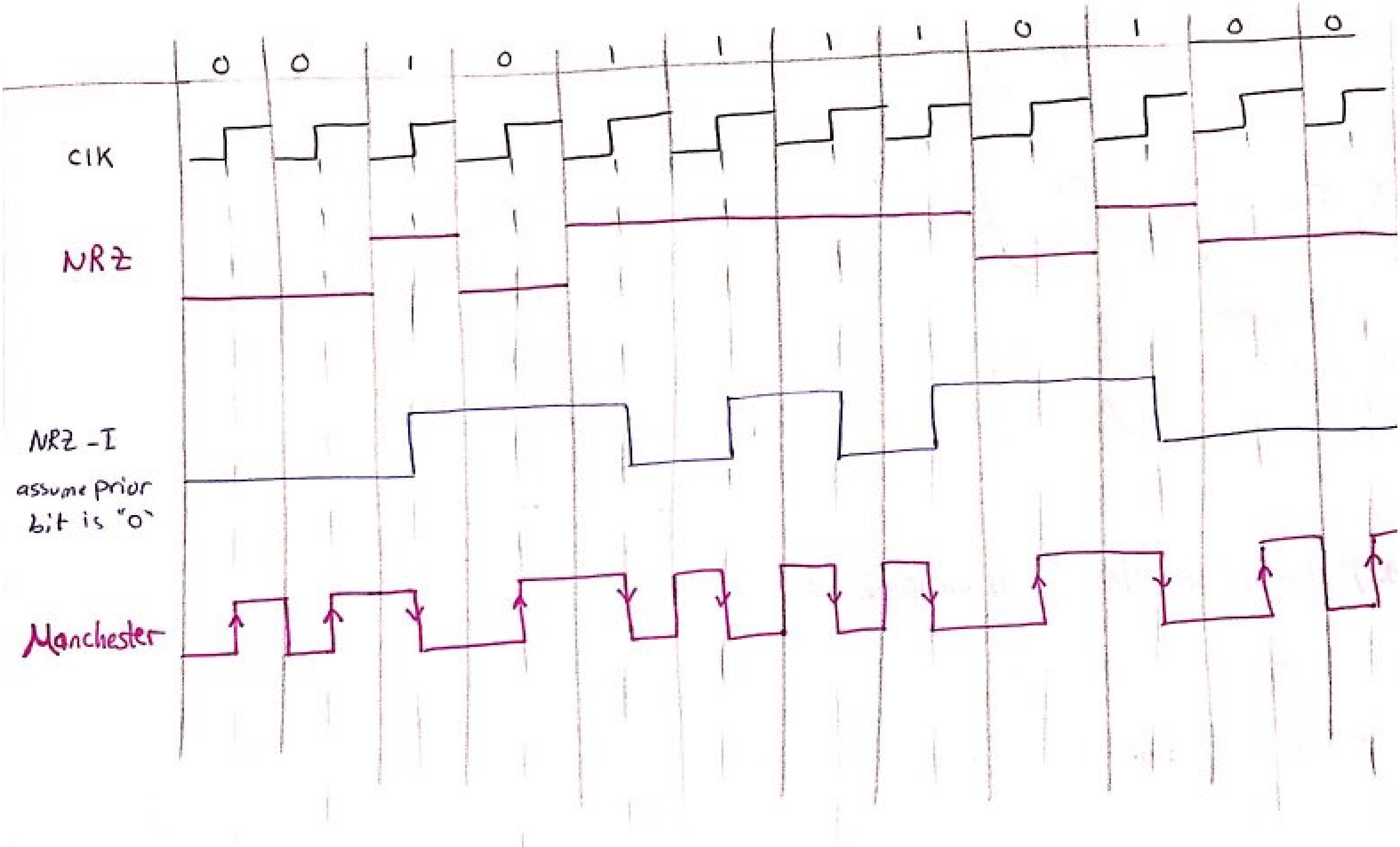
② NRZ-Inverted (NRZ-I)

"1" : Transition , 0 : No transition.

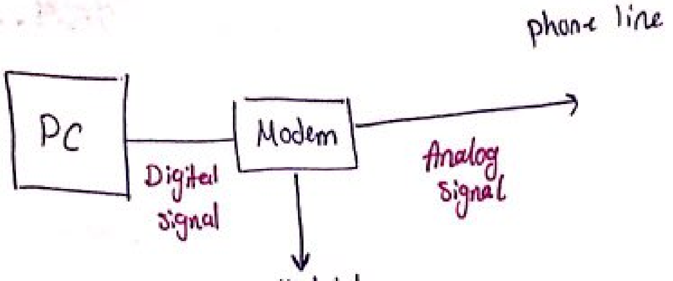
③ Manchester < 1 → ↓ ; 0 → ↑

Ex:- 001011110100 → Message to be sent :

* بتجي بالامتحان .

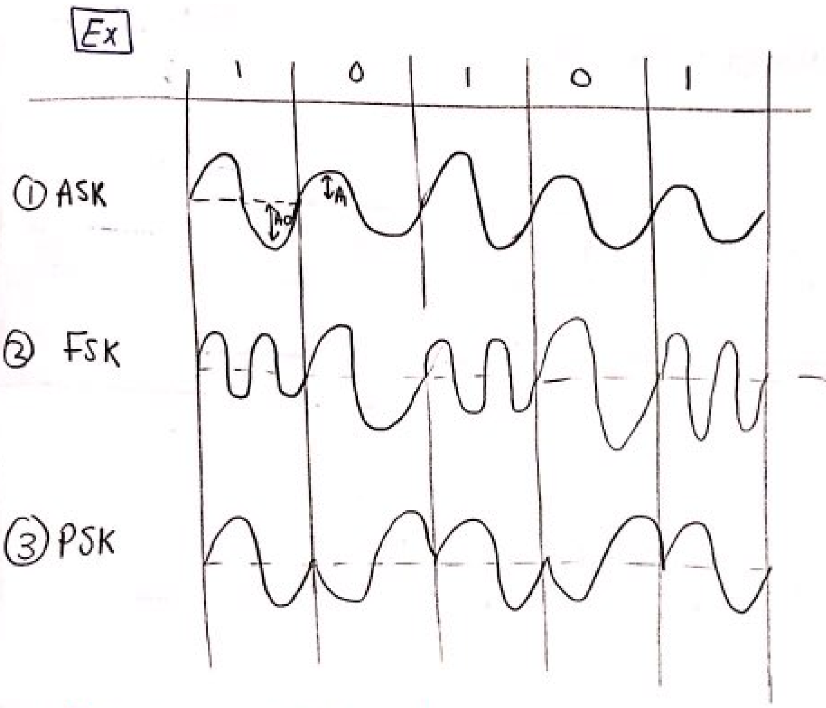


25. Sep. 2



Modulator.

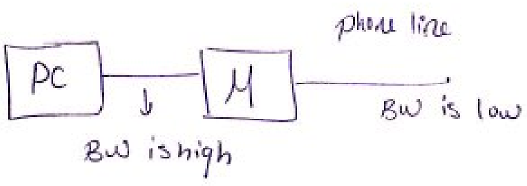
1. Amplitude shift Keying (ASK)
2. Frequency shift Keying (FSK)
3. phase shift Keying (PSK)



* Problem in Analog transmission :-

- low Bandwidth.

* we know that digital transmission offers very high bandwidth.



M

↓

Signaling Rate

(baud Rate)

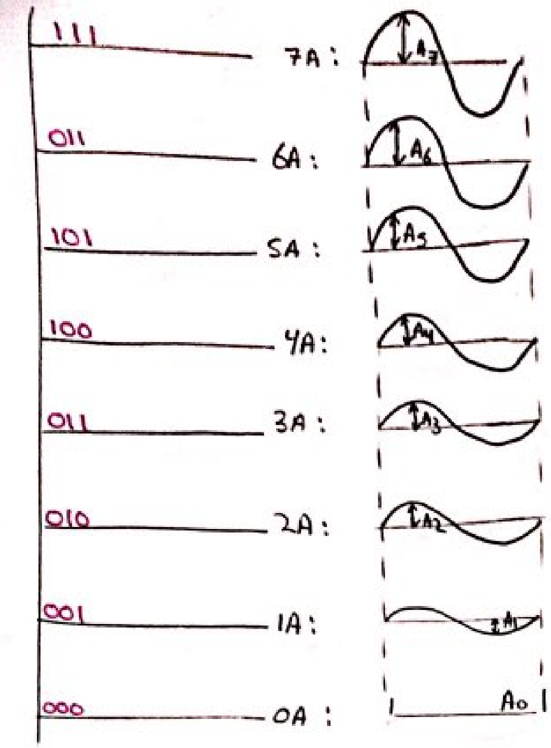
$R_s \rightarrow$ symbols/sec

$n \rightarrow$ bits/symbol

$$\boxed{R_b = n * R_s}$$

Ex:- Consider these signals:-

ASK →



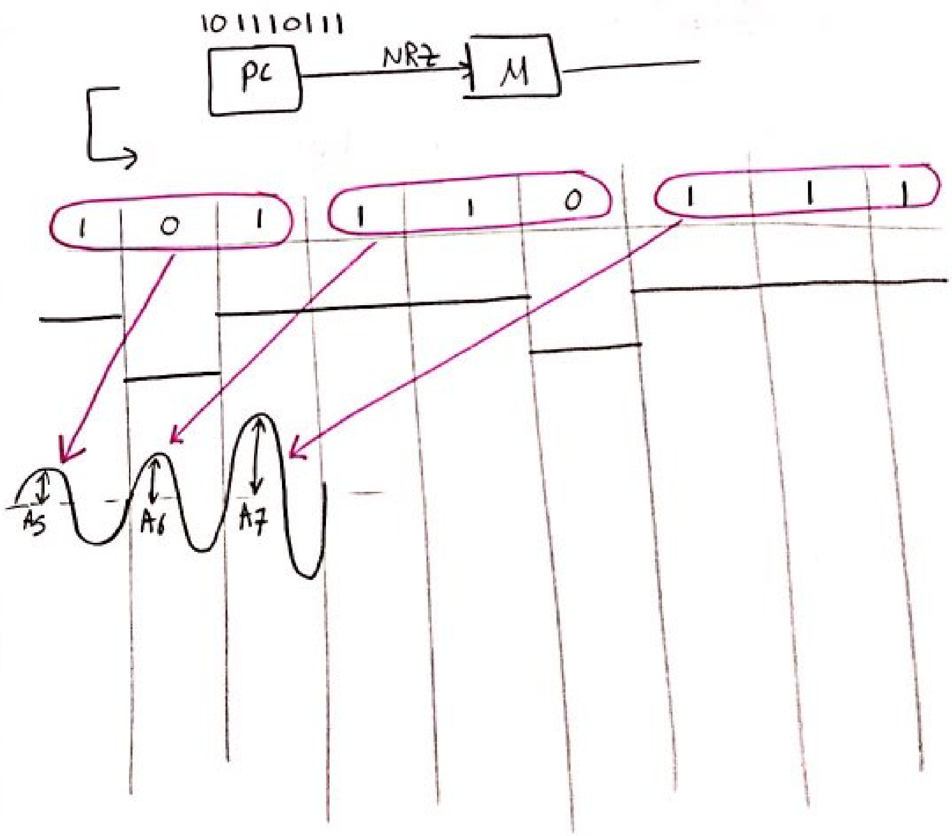
Question -

How many bits can be represented by these signals (symbols) ?! n ?

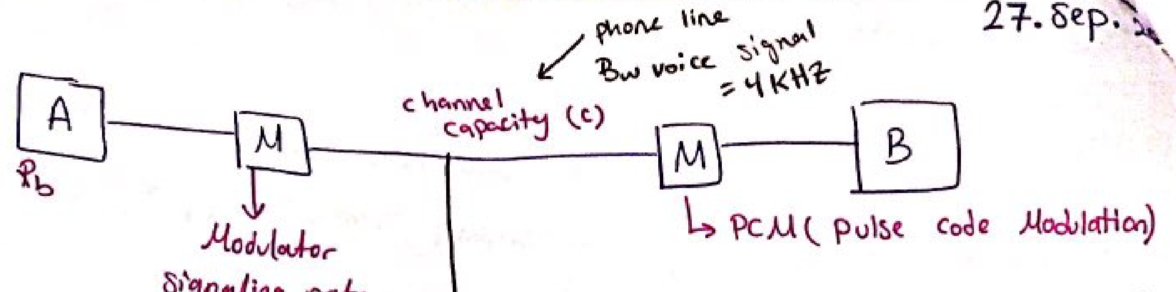
$n = 3$ bits/symbol

$2^3 = 8$ signals (symbols)

2 signals → 1 bit
4 signals → 2 bits



27. Sep.

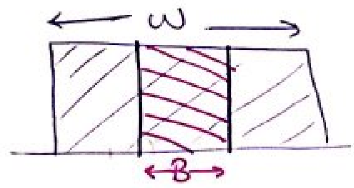


↳ channel capacity : Maximum data Rate that can be supported by channel.

$$R_b \leq C = W \log_2 (1 + SNR)$$ → signal-to-noise ratio

W: Bandwidth of the channel which mainly refers to the range of frequencies that can pass through the channel without degradation. (HZ)

B: Bandwidth of the signal " " " " available in the signal.



SNR Signal-to-Noise ratio = $\frac{\text{Signal power}}{\text{Noise power}}$

we need SNR to be **High**

<u>Example</u>:- IF we want to send data at a rate of 8000 bps (R_b) through a channel that has a bandwidth of 1000 HZ (W), what is the SNR required ?!

solution

$R_b = 8000$ bps

$W = 1000$ HZ

$$R_b \leq C = W \log_2 (1 + SNR)$$

$$8000 = 1000 \log_2 (1 + SNR)$$

$$1 + SNR = 2^8 = 256 \rightarrow \boxed{SNR = 255}$$

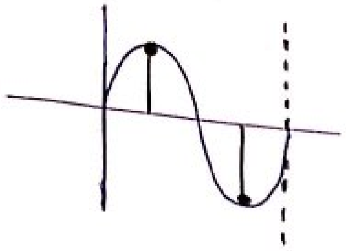
Scanned with CamScanner

27.sep.2018

(PCM) → Pulse code Modulation.

Sampling (Nyquist theorem) | Quantaization | Encoding

[1] Nyquist theorem :-



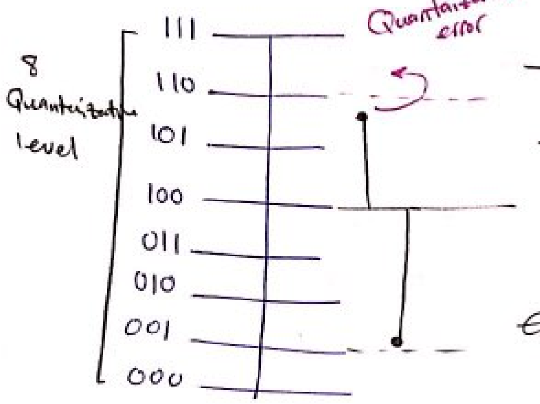
* Effecient signal recovery

→ $f_s \geq 2 f_m$

f_s ↙ Sampling rate (samples/sec)

f_m ↘ signal being sampled

Quantaization



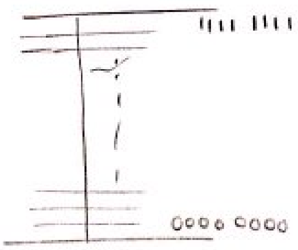
→ 3bits/sample

→ 2^3 → * of Quantaization level

Encoding : 110 001 - - -

Note :-

M ↓ A/D convertor (Analog to digital)



Nowadays : 8 bits/sample

→ $2^8 = 256$ Quantaization levels

- 8 bits/sample
- Voice signal BW = 4KHz
- Nyquist theorem → $f_s \geq 2 f_m$

f_s: sample/sec

f_m: ↳ frequency of the signal being sampled

- f_s = 8000 samples/sec

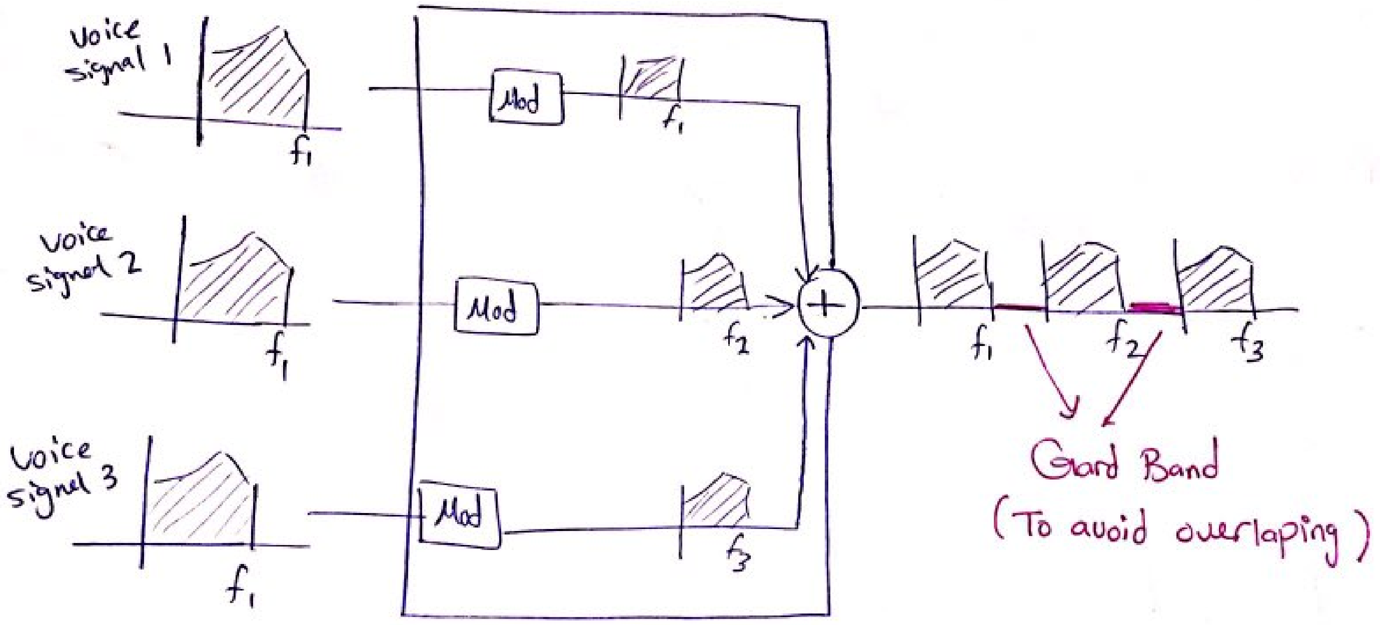
Now! 8 bits/sample * 8000 sample/sec = 64 K bps

This is the basic data rate of public switched Telephone Network (PSTN)

→ Multiplexing :—

- → Frequency division multiplexing (FDM)
- → Time division multiplexing (TDM)

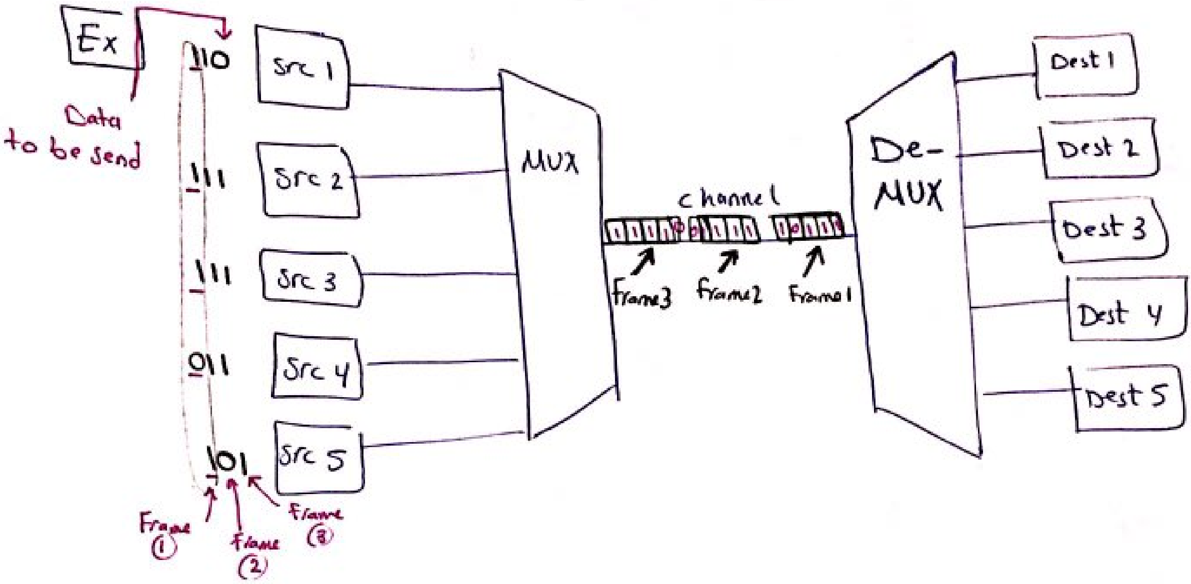
* FDM: used when the bandwidth of the channel exceeds the bandwidth of signal. This is mainly done through modulating each signal to a different carrier signal.



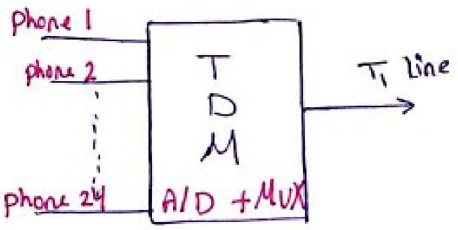
30. sep. 2018

Multiplexing
- → Frequency division multiplexing (FDM) (Analog transmission)
- → Time division multiplexing (TDM) (Digital transmission)
  - ↳ used when the data rate of the medium (channel) exceeds the data rate of the signal to be transmitted.

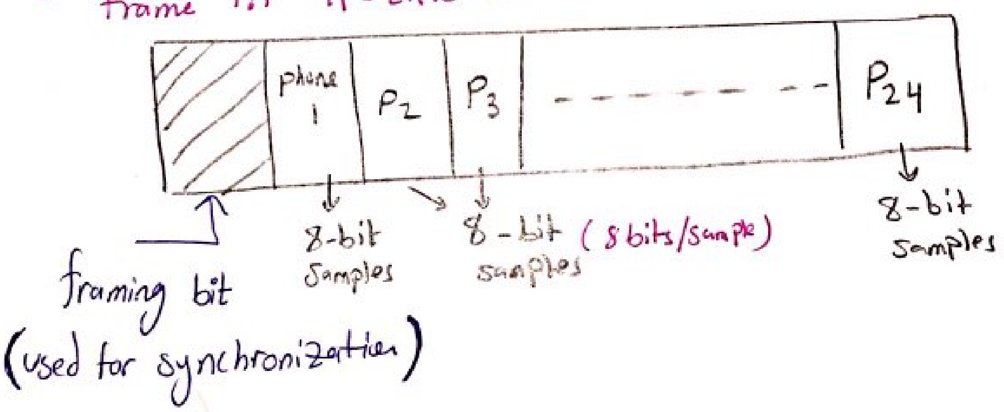
Ex



- Digital carrier system :- (Ex: T_1-Line)
  (DCS)



- Frame in T_1-Line :-



→ How can we find the data rate of T_1 line ?!

8 bits samples * 24 + 1 (1 bit sample) framing bit

= 193 bits / sample

→ 193 bits / sample * $\boxed{8000}$ sample / sec = 1.544 Mbps

2 * 4K (2fm)

	DCS	# of T_1 s	# of channels	Data Rate
T_1	1	1	24	1.544 Mbps ← line ال احفظ أرقام
T_2	2	4 $\xrightarrow{*24}$	96	6.312 Mbps
T_3	3	28 $\xrightarrow{*24}$	672	44.736 Mbps
T_4	4	168	4032	274.176 Mbps

$\boxed{\text{Example 1}}$ Suppose we are to design a TDM carrier to support 30 voice channels assuming 16-bit samples and 8000 sample / sec Determine the required data rate ?!

→ **Solution:** 30 * 16-bit / sample + 1 bit / sample = 481 bits / samples

481 bits / sample * 8000 samples / sec

= 3.8 Mbps

$\boxed{\text{Example :}}$ Find the number of the following devices that can be accomodated by a T_1 TDM line # when 1% of the line capacity is used for synchronization?

→ (2400 - bps computer terminals)

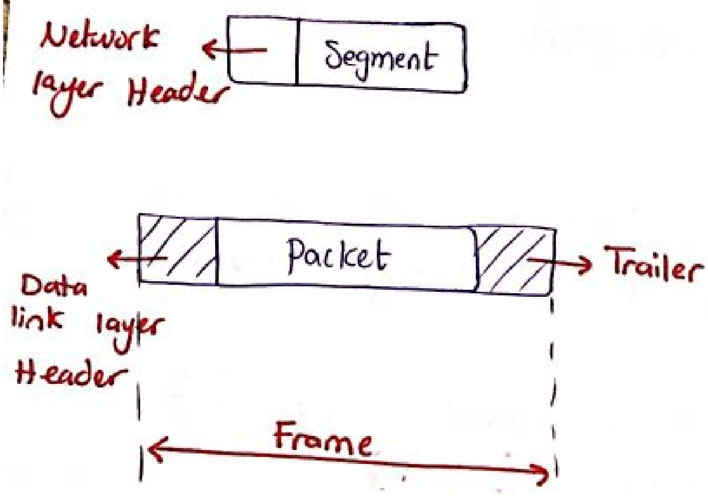
T_1 line capacity = 1.544 Mbps

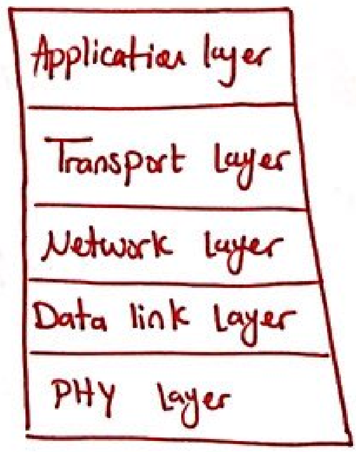
$$\left\lfloor \frac{99\% * 1.544 \text{ Mbps}}{2400 \text{ bps}} \right\rfloor \rightarrow 1\% \text{ for synch.}$$

= 636 devices

*** Chapter (7) → Data link layer** 2.Oct. 2018

The unit of data in (Data link layer) → frame





*** links Types :-**

[1] Simplex → One signal in one direction.

[2] Half - Duplex → Two directions are possible, but one signal at a time.

[3] Full - Duplex → Two directions are possible, and 2 signals at atime (Send / Recieve)

* Chapter (6) :

4.Oct.2018

Public Switch telephone Network :

[1] PSTN

- Dial-up ⟶ 56 Kbps
- DSL ⟶ ADSL, SDSL ⟶ 256 Kbps — 100 M bps (16 bps trials)

[2] ISDN

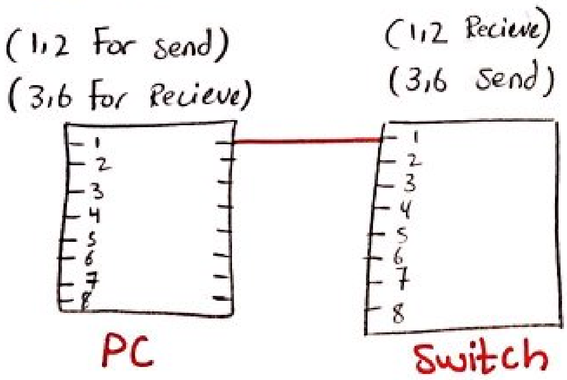
① Basic Interface Rate ⟶ 2 B channels (64 Kbps), one D channel to carry control and signaling Information. (128 Kbps)

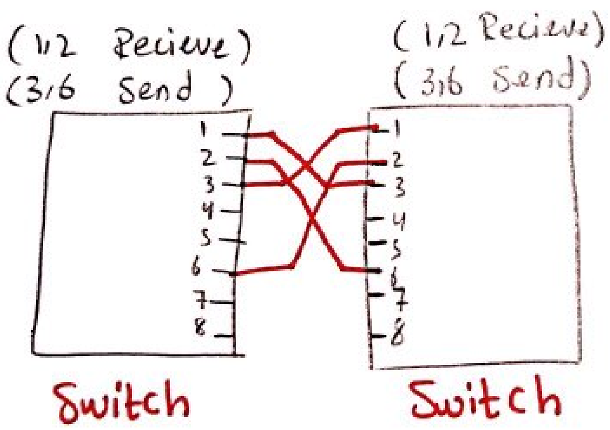
② Primary Interface Rate ⟶ 23 B channel (1.472 Mbps), one D channel (64 Kbps) [USA]

* 30 B channel (1.92 Mbps) [UK]

* Chapter (5) :

Classes of transmission media :





* UTP Cable Types :-

[1] Straight - through

[2] Cross - over (PC-PC OR switch - switch)

7.Oct.2018

* Ethernet :

A set of networking technologies available or working at both Physical and Data link layer.

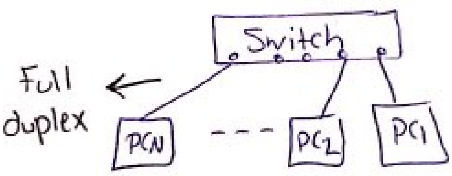
Coaxial cable ↓ First cable used in Ethernet

- **Physical layer**
  (Cabling system)
- **Data link layer**
  ↓
  Channel access control
  ↓
  (CSMA / CD) → used in old ethernet.
  ↳ Colision Detection

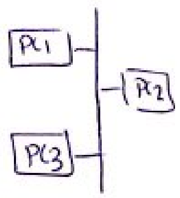
Ethernet

- **Old**
  ↓
  10 Mbps
  ↓
  Bus Topology
  ↓
  Coaxide Cable
  ↓
  500 m → Thick ethernet
  200 m → Thin
- **Fast**
  ↓
  100 Mbps
  ↓
  Start topology
  (Switch)
  ↳ No Bus
- **Giga**
  ↓
  1 Gbps
  ↓
  Start topology
  (Switch)

Fast and Giga



Old



CSMA/CD ← Solved by — Colision ← Causes — Half Duplex

* Big picture :-

Internet Access

- **Individual**
  - Internet Service Provider
    - ① SLIP
    - ② PPP
    - To communicate with Internet service Provider
- **Group**
  - Local Area Network (LAN)
    - Ethernet.

9.Oct. 2018

Flow Control

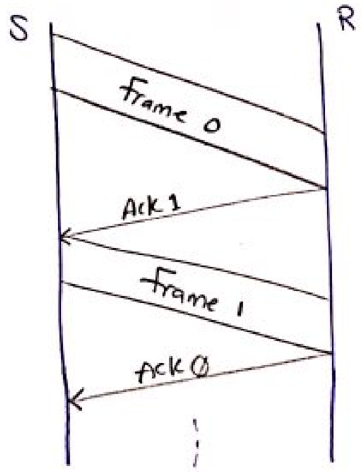
- Stop-and-wait (Send-and-wait)
- Sliding window

Error Control

(Two phases)

- Error detection (2 types)
  - parity check
  - cyclic Redundancy check (CRC)
- Error Correction ↓ Retransmission

* Stop-and-wait timing Diagram :

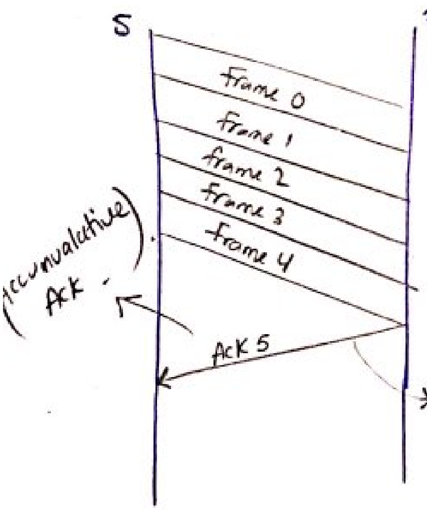


poor performance

- Higher Delay
- link in effeciency
- frame sequence : 0, 1, 0, 1, ---

Physical Addressing

* Sliding-Window Timing Diagram :



- Frame sequence : 0, 1, 2, 3, 4 ---
  It depends on the window size
- Lower delay
- More link effecient.

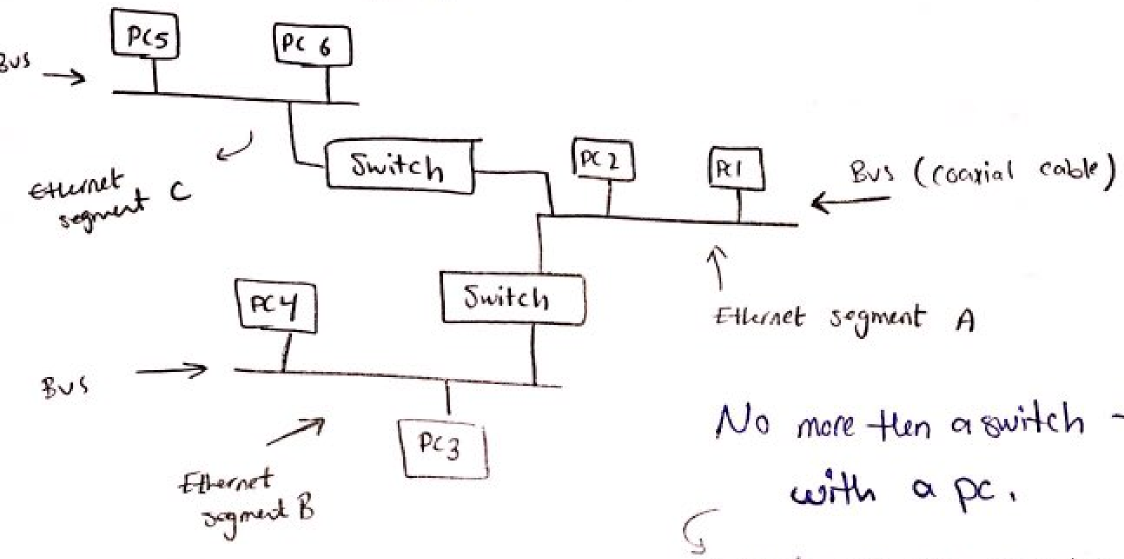
→ It means we have already recieved frame 4, please send frame 5

[forward / Broadcast]

* forward → Known MAC Address
* Broadcast → unknown dest. MAC address

Old Ethernet (10 M bps)

- Coaxile cable
  - 500 m (thick)
  - 200 m (thin)
- Half duplex links
- Bus Topology



No more then a switch to be connected with a pc.

So that we use spanning tree protocol to guarantee that ~~there~~ it's loop free

OSI	TCP/IP Stack
7 layers	5 layers
(HDLC)	(Ethernet)

11.Oct.2018

* Local Area Network :

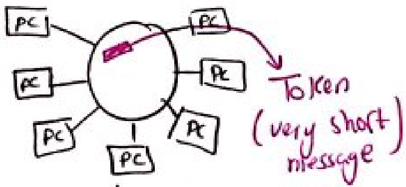
* 3 Protocols :

Ethernet

1. old (10 Mbps)
   - Half duplex
   - Bus topology

   problem ↓ Collision ↓ solution CSMA/CD

Token Ring



↓ Half-duplex (16 Mbps)

عشان الـ pc يبعت لازم ياخد الـ Token

get the token first, Then transmit

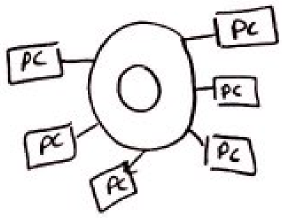
(No collision)

FDDI - dual Ring → Fiber - Distribution - Data Interface

→ 100 Mbps

→ Dual - Ring

* using fiber
* Full duplex



2. Fast Ethernet
3. Giga Ethernet

Best effort

Un-Reliable ← UDP (User Datagram protocol) ← **Transport layer** → TCP → Reliable (Transmission control protocol)

Asynchronous Balanced mode

* The side effect of flow and error control is :- High Delay

* Real Time streaming ↓ ex (Audio and video transmission)

video is split into frames ($\underbrace{\text{I-frame}}_{\text{message}}$, P-frame, B-frame)

UDP ⟵ Transport layer ⟶ TCP

Note:

TCP
LLC → F and EC have to work

UDP
LLC → No F and EC

Data link layer → Ethernet

Given by : ISP ⟵ IP layer (Packet) ⟶ IP address ⟶ IPV4 → 32 bits (version)
IPV6 → 128 bits

= Company given by ⟵ Data link (Frame) ⟶ MAC address ⟶ 48 bits

packet Header
Src IP address
dest IP address

Data link Header | Data link trailer

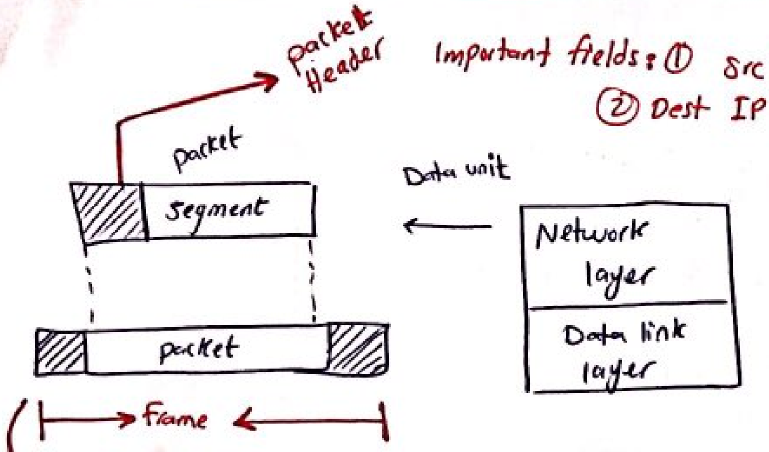
Header	Packet	trailer

⟵ Frame ⟶

{ Src: MAC Address
Dest : " " }

14.oct.2018

Important fields: ① Src IP address
② Dest IP address



Important fields : ① Src MAC Address
② Dest MAC Address

Notes:

① Src IP address
② Src MAC address } Known

NIC → MAC Address
↳ Network Interface Card

③ Dest IP address → Unknown

URL address : www.yahoo.com
Dest. IP address

site	IP address
www.yahoo.com	100.0.0.1

Hold this

رح نشرح هاد بس

IPaddress
- IPV4 → 32-bit
- IPV6 → 128-bit

32-bit ≡ 4 bytes

1-byte . 1-byte . 1-byte . 1-byte

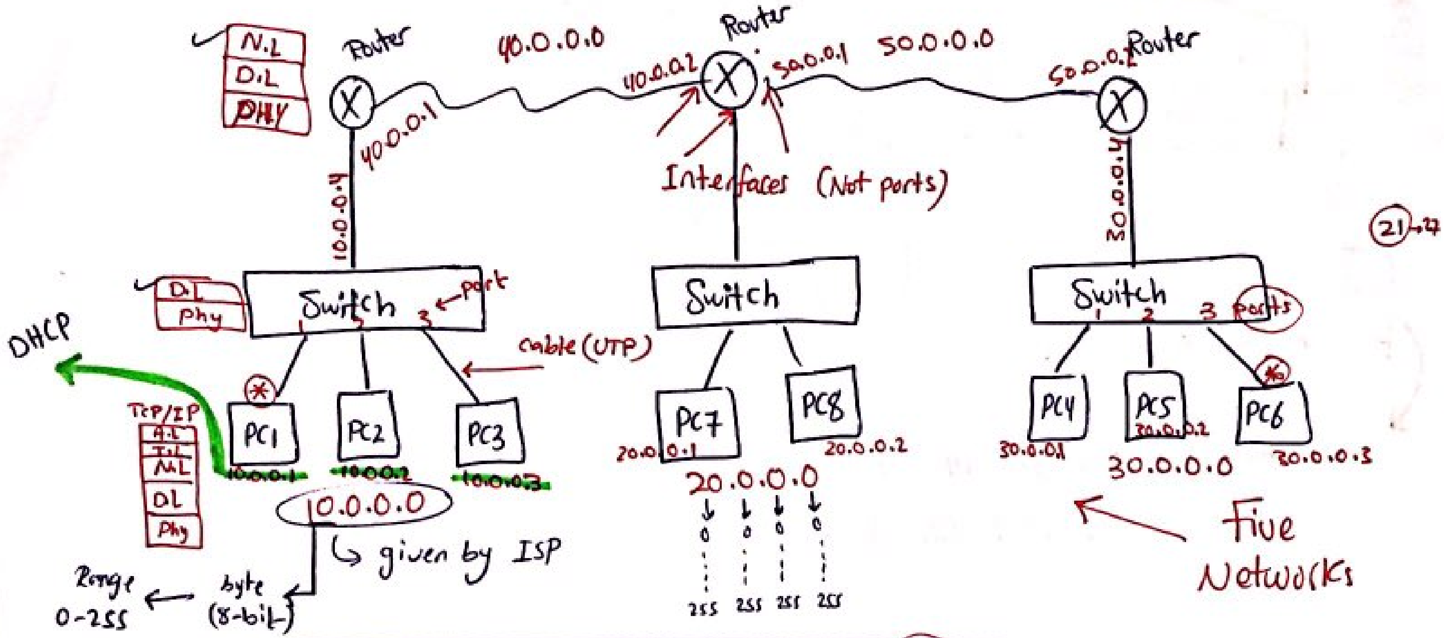
to make it easier (simple)

* Every Domain Name has own servers — very Important Note.

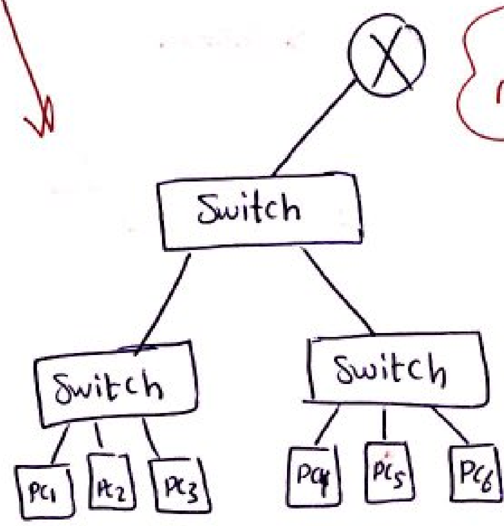
Domain name Server (DNS)

ARP Protocol :

IP Address	MAC Address
100.0.0.1	10.20.30.40.50.60



One Network

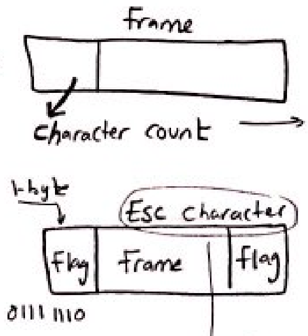


Every Interface represents a separate Network *

Every Interface has own IP address

Framing

- Frame Format
- Synchronization
  - ① Charachter count
  - ② Byte stuffing
  - ③ bit stuffing
  - ④ phy layer coding violation



Character count → * of characters of bytes in the frame

Byte stuffing → بنحط flags بالبداية والنهاية عشان نميز بين ال frames

→ بنحط قبال ال flag ال content عشان نفهم انها content مش flag

Byte stuffing

→ Example page 2

Data to be send

	ESC		Flag	

	ESC	ESC		ESC	Flag	

Data Recieved

	ESC		Flag	

Bit stuffing

→ used in HDLC (already gone)

Physical layer Coding violation

→ (Framing in Ethernet)

فيه حركات ال bits بيبعتها ال sender عشان يفهم ال reciever شو ال clock تبعت ال sender
وإذا فهم ال clock بيقدر يفهم ال start, end تبع ال frame

18.oct.2018

* Ethernet :

Is a family of computer networking technologies for local Area (LAN) and Larger Networks.

Ethernet cabling system ⟶ PHY Layer

* It was introduced In 1980 while it was first standarized in 1983

Old Ethernet ⟶ CSMA/CD, Shared Media, Half Duplex links

Fast ethernet ⟶ No CSMA/CD, Dedicated Media, full-duplex links.

* the Table on page 2,3,4 ⟶ Self reading.

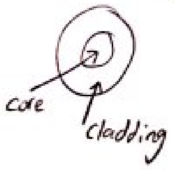
page 4 (* Ethernet Media standards and distances)

Fiber
- Single Mode (SMF) (used in long Distances)
- Multiple Mode (MMF) (used for short distances)

Optical Fiber
- OM1 (62.5/125 μm) — optical Multi-mode
  - 62.5 ← core size
  - 125 ← cladding size
- OM2 (50/125 μm) — Range (10Mbps – 1Gbps) ← 1Giga
- OM3 (50/125 μm) — Range (10Gbps – 100Gbps)
- OM4 (50/125 μm)

* less dence is better

OM3, OM4 ⟶ (laser optimized) ⟶ Higher rate



(laser optimized)
- Short Range (SR)
  - OM3 (100 M)
  - OM4 (150 m)
  - wavelength = 850 nm
- Long Range (LR) ↓ 10 Km ↓ wavelength = 1310 nm
- Extra long Range (ER) ↓ 40 Km ↓ wave length = 1550 nm

Scanned with CamScanner

18.oct.2018

Page 5

Available Rates (widely used)

* 40 Gbps

SR LR ER

4 lanes

every lane → 10 Giga

* 100 Gbps

SR LR ER

4 lanes

4 lanes 10 lanes

↓ 25 giga/lane $\boxed{100/4}$

↓ 1 giga/lane $\boxed{100/10}$

* 100 GBASE - CR10 cat-7 10 m

Rate

100GBase - SR10

Short Range

10 lanes

100GBase - CR10

copper (UTP)

Page 5:

100 G Port types

→ study first 2 types + the last 2 types Page 6

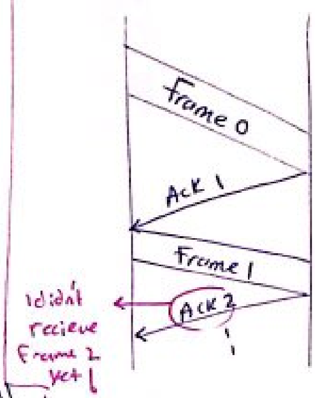
21.Oct.2018

Flow and Error Control → L.L.C sublayers

Flow control
- stop-and-wait
- sliding window
  - Go-back-N
  - Selective reject

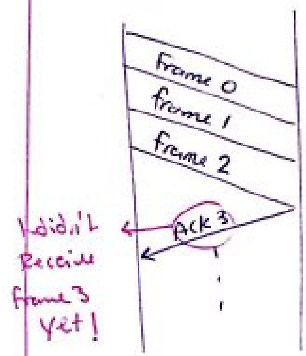
Note

Stop-and-wait timing Diagram:



(*) message → segment → packet → frame → Raw bits → Digital signal

Sliding-window



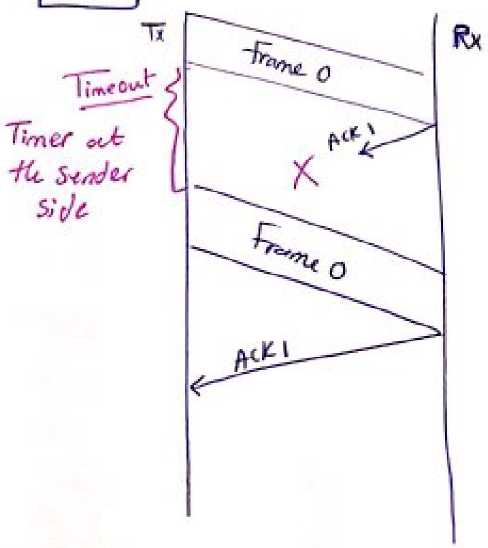
Error
- → Damage
- → Drop

* Automatic Repeat reQuest (ARQ) slide 6

→ Error Control
- Error detection
  - Acknowledgement
    - Positive (PACK)
    - Negative (NACK)
- Error correction
  - Retransmission (NACK)

Loss
- Damage — High error rate
- Drop — Buffer-overflow

Note

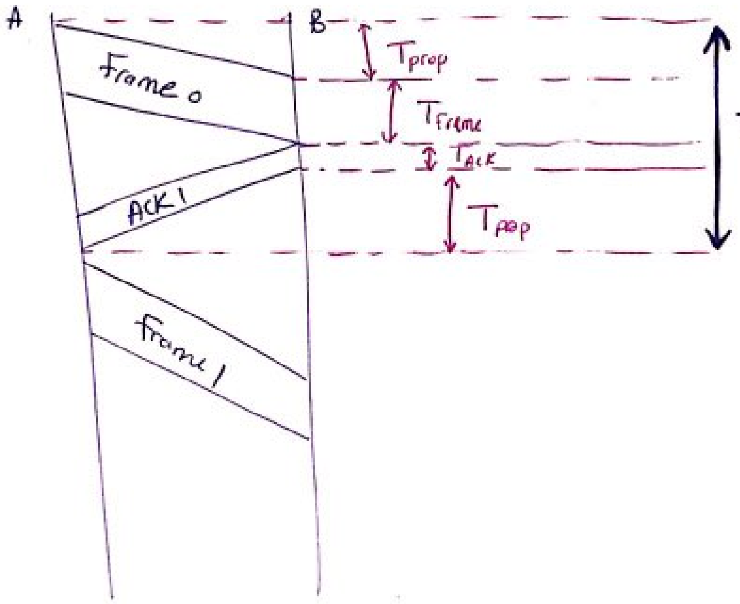


→ slide 9

slide 10 → $T_{trans} \equiv T_{frame}$

$\boxed{\text{* Slide 10}}$ using Timing Diagram ...

pic



$$T_D = T_{frame} + 2T_{prop} + T_{ACK}$$

$\boxed{\text{Slide 14}}$

$$\frac{50}{100} = \frac{T_{trans}}{T_D} = \frac{T_{trans} \,/\, T_{trans}}{T_{trans} + 2T_{prop} \,/\, T_{trans}}$$

$$0.5 = \frac{1}{1 + 2\frac{T_{prop}}{T_{trans}}} \qquad T_{trans} = 2T_{prop}$$

$$0.5 = \frac{1}{1 + \frac{2\cancel{T_{prop}}}{2\cancel{T_{prop}}}} = 0.5 \checkmark$$

$$T_{trans} = 40\ ms$$

$$\frac{L}{R_b} = 40ms$$

$$L = 40 \times \cancel{10^{-3}} \times 4 \times \cancel{10^{3}}$$

$$= 160\ \text{bits}$$

23.Oct.2018

Link effeciency = $\frac{T_{trans}}{T_D}$

As far as stop-and-wait protocol is concerned, Link effeciency = $\frac{T_{trans}}{T_{trans} + 2T_{prop} + \cancel{T_{ACK}}}$ 0 very short message

Fragmentation

slide 15

large block of data may be split in small frames.

stop and wait with errors → نفس ماطلوب

Sliding window

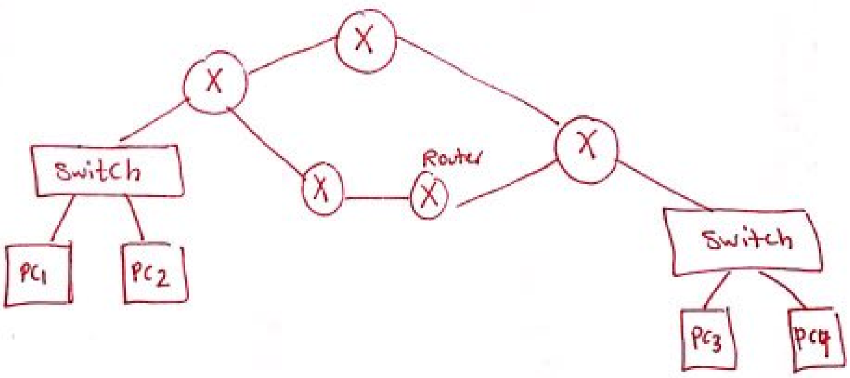
slide 21

1 N-bit sequence number → 3-bit → frame sequence # → $f_0, f_1, --- f_7$

2 Sliding window size

Slide 23: RR_x: Reciever is Ready for frame x

* The window is gonna expand / depends on the number of frames acknowledged.



Slide 25 frame sequence # = 3-bit

window size = 7-bit

Scanned with CamScanner

GO-Back-N ARQ Protocol :

↳ Better then selective reject

slide 28 ✓

Example Slide 31 ✓

stop and wait Acks :

1. PAck → Ack
2. NAck

sliding window Acks :

1- RR
2. REJ

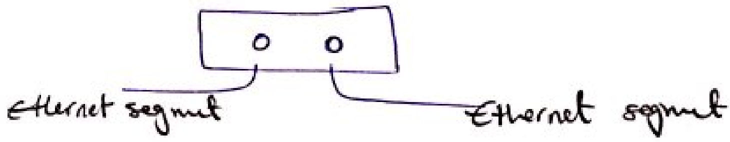
REJ 4 : Means that there is an error in frame 4 or hasn't been recieved.

RR_x → Next frame

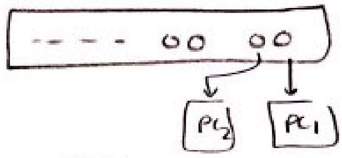
* Interworking device :— (chapter 7 – part C)

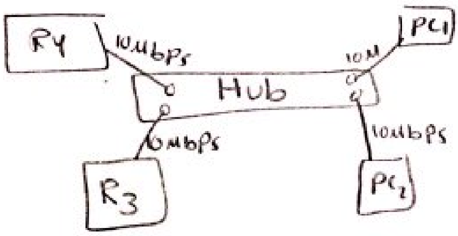
25.Oct.2018

Repeater → Regeneration for signal (Physical layer device)
two ports for two signals (1-core)



Hubs → multiple ports (Physical layer device)
↳ looks like a star topology



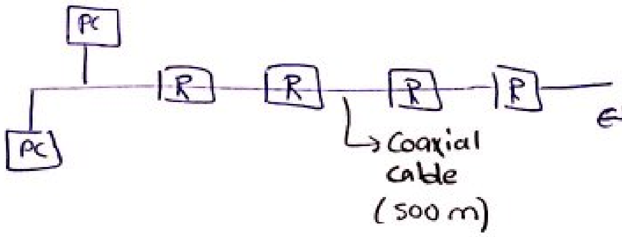


(Hub is not a store-and-forward device)
↓
(No Buffer
No processing)

*** limitation :—**

1- Single collision domain
2 – All links should have same rate.

Tip : old Ethernet



Ethernet segment
↖ Bus topology
(Half duplex)

four ports available

Bridge
Switches

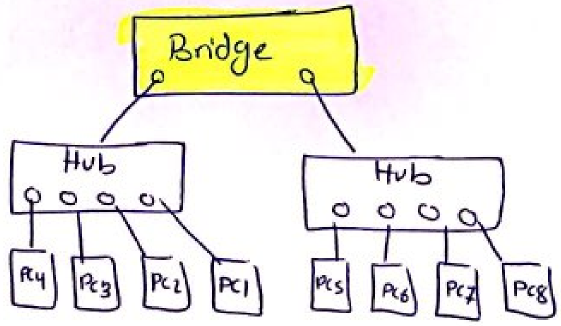
→ layer 2 devices

multiple ports available

(1- they understand MAC address
2- they are store-and-forward devices
3- They isolate the Collision.
4- they contain switching tables.

MAC Add	Port
10:20:30:	2
⋮	⋮

Tip:



Question: How many collision domain available? 2

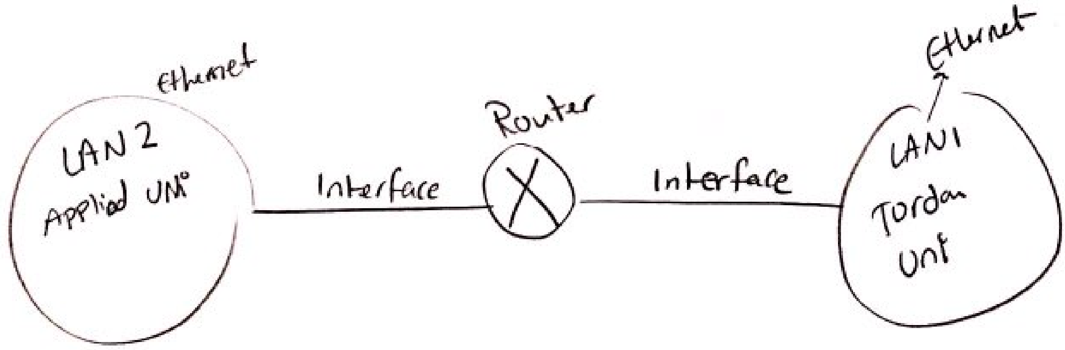
Router layer 3 devices:

- they understand IP-addresses.
- they contain routing tables.
- As a result of employing contain routing protocol
- they connect local area networks.

IP	Interface
10.0.0	Se0/0/0
|	|

Note Gateways: Layer 3 devices → connect dissimiller net
different.

→ Similler Network:



30. Oct. 2018

* Local Area Networks :- (Ethernet)

LAN :- is a network provides connectivity of computers, mainframes, storage devices.

802 Standards (slide 9) [Ex. ->]

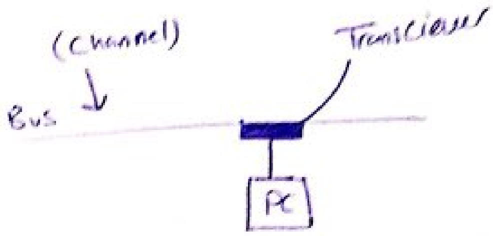
802.3 → MAC + physical layers

802.5 → Token Ring

802.8 → FDDI

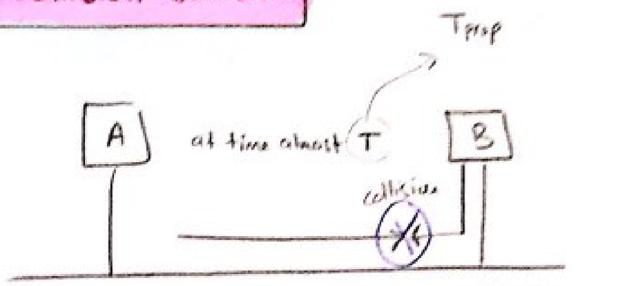
802.11 → Wireless

[Slide 11] **CSMA/CD** → carrier sense, multiple access with collision detection.



* collision affects the destination.

collision detection



Node A Start transmission at time 0

Node B starts transmission at Time T

* A knows that a collision has taken place only if :

1. A's message reaches B at time T
2. B's message reaches A at time 2T
3. So, A must still transmitting at 2T

* let's take the worst-case scenario :-

$2\tau \rightarrow T_{prop} = \frac{d}{v}$

worst-case destination

in old Ethernet $\longrightarrow 2 * \frac{2.5 \times 10^3 \, m}{1 \times 10^8 \, m/s} \approx 51.2 \, \mu sec$

0.98

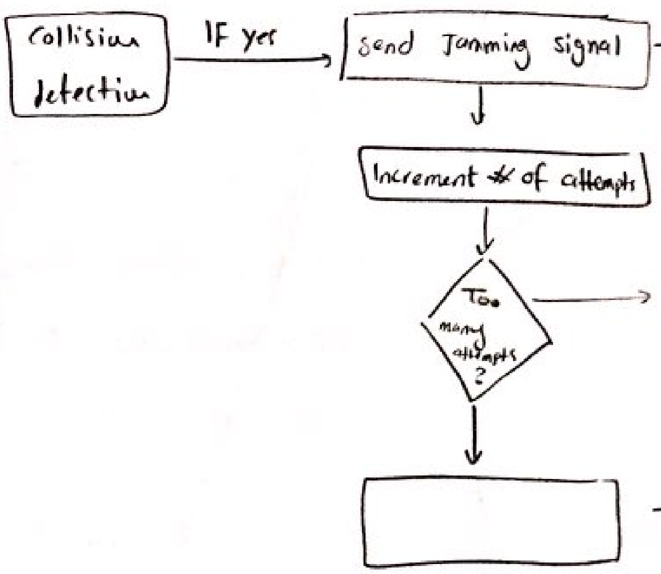
$51.2 \, \mu s * 10 \, M \, bps$

$\boxed{= 512 \text{ bits}}$

*** Note :** Frame size has to be of length 512 bits in order to detect the collision.

*** Transmit process** . slide 12

Channel busy? → بنشيك عليها من خلال Transceiver



بنبعتها لكل PC's +
الموجودة عال Bus

usually 5 attempts

Backoff-time depends on the # of attempts .

first waiting time → 51.2 μsec

Back-off Alg

Attempt	waiting time
1	0 , 51.2 μsec
2	0 , 51.2 μ , 102.4 μ , 153.6 μ
3	⋮

$K = (0 - (2^n - 1)) * 51.2 \, \mu sec$

# of attempts ↑ ⟵
waiting time ↑

2

30. oct. 2018

Token Ring Operation :-

$\boxed{\text{No Possible collision}}$

* One limitation of the Token:-
that we need one pc to monitor
the (Token)

Token Holding Time (THT)

Token Rotation Time (TRT)

---

4. Nov. 2018

Slide $\boxed{21}$ → أكيد بيجي عليه سؤال

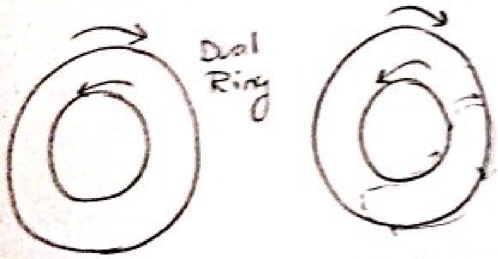
↳ what is the minimum time to report that the token is missed :

Token Release → early release (token ال release بحل frame ال transmission لما اخلص)
→ delayed release

كيف ممكن يضبط إذا صار collision؟!

↳ بحل Direction with movement.

$\boxed{\text{Slide 24}}$ FDDI protocol



Dual Ring

In the case of link breakage.

4.Nov.2018

* Switching Nodes:

Cognitive radio Google.

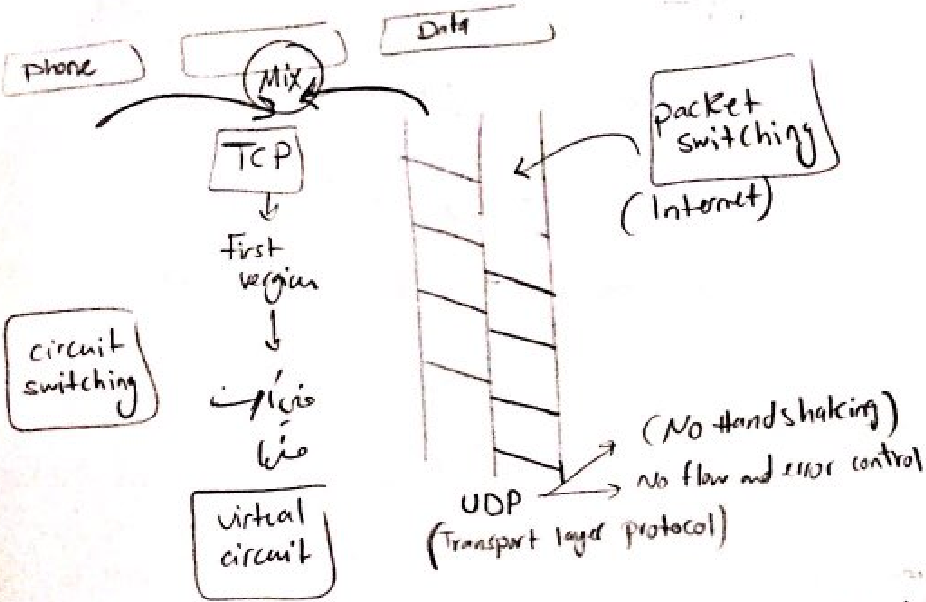
IOT Google
↳ Internet of things.

Cloud computing

Switching Technologies
- Circuit switching
- packet switching
  - Datagram
  - Virtual circuits
    ↳ No routing decisions to make (كدا بيعرف الطريق اللي لازم يمشي بيها)

slide 12

٣ رسمات



slide 7 Connection-less ⟶ No handshaking.

slide 8 → packet loss كل بشيك بين كل نقطتين (Point-to-point)
Drop مش Damage بسبب الـ

* فقط الـ destination بشيك كل damage, drop (End-to-End)

Slide 14

store and forward ⟶ processing يعني مش

IP The Internet Protocol. → Layer 3 — 6.Nov.2018

* IP is a connection-less, unreliable → (No Handshaking with the destination)

Slide ② [
* IP provides Best effort services:
  - No guarantee of delivery of error-free packets
  - No guarantee of ordered delivery of packets
  - No guarantee of delivery of packets.
]

Transport layer → Majority : TCP, UDP

IP Datagram format (slide 3)

Type of service → related to QoS (quality of service)

أهم شغلتين → IP addressing, Routing Protocol

Version	Header length	ToS	Total length
Identification (packet رقم ال)		F (3 bit)	Fragmentation
TTL (Time to live)	Protocol (TCP UDP)	Header check (Not contents check)	
Source IP			
Dest IP			

Note Considering UDP transport layer protocol, ToS field is filled out by zeros.

the maximum size of the packet is 1500 bytes.

Slide 9 very Important :-

4000 bytes datagram → 20 bytes Header, 3980 data

ICMP : Internet control management protocol
→ بيستعين فيه (IP) عندما يصير أي مشكلة

Header length = 20 bytes ← حفظ

	length = 4000	ID = x	frag flag = 1	offset = 0

offset = → 1480 ÷ 8 = 185 (1500 - 20 Header; ÷ 8 To save space)

(حركة افتراضية (آخر بايت ساوة بالحافة

	length = 1500	ID = x	frag flag = 1	offset = 0
	1500	ID = x	frag flag = 1	offset = 185
	1040	ID = x	frag flag = 1	offset = 370

→ $\frac{1480}{8} = 185$

→ $\frac{\text{length1} + \text{length 2} - 2(20)}{8}$ → $\frac{1500 + 1500 - 40}{8}$

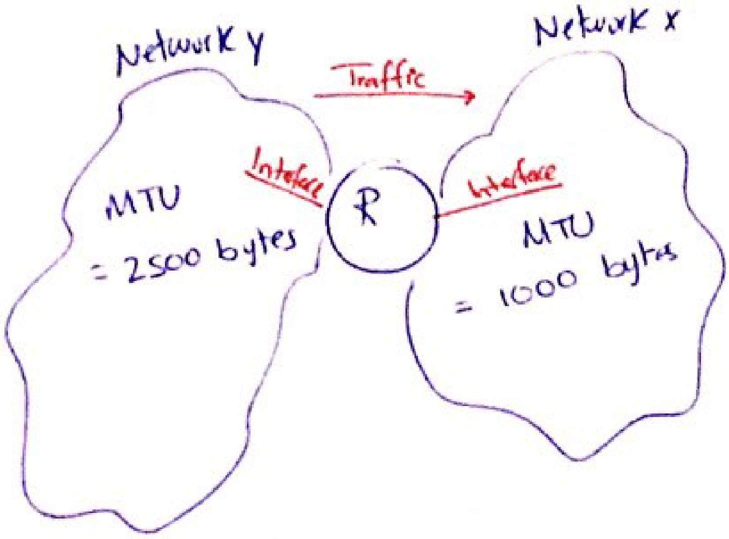
↳ لو بدي احطها 1000 بيطلع في نقص بالداتا.

→ 2 more packets → each takes 20 bytes for Header → we need more bytes to avoid loss of data

Scanned with CamScanner

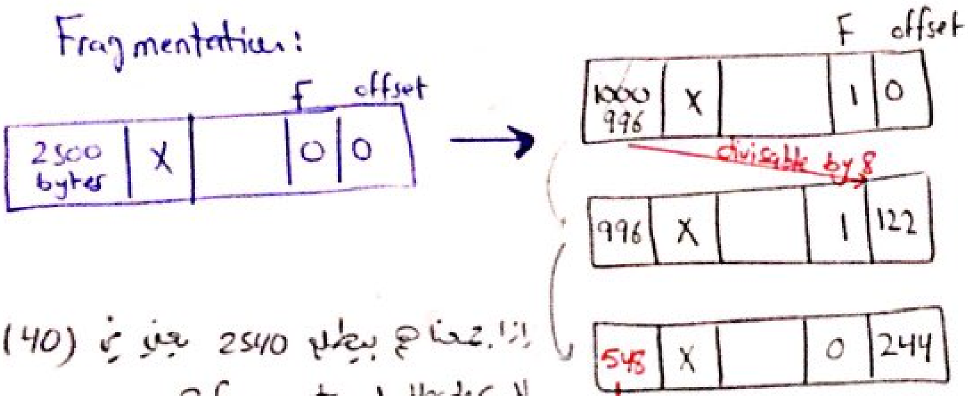
* IP Addressing :-

8.Nov.2018



سؤال جاي بالامتحان أكيد

Fragmentation:



$\frac{980}{8} = 122.5$ X

إذا جمعنا بيطلع 2540 يعني في (40) زيادة
الـ Header لـ 2 fragments

Data*

$2480 - (976 + 976) = 528$

+ (20 Header)

548

* verification :-

$996 + 996 + 548 = 2540$ Two more packets.

IP Addressing

0-255 . 0-255 . 0-255 . 0-255

→ For organization (Easy to memorize)

* Major Parts:-

?	?
Network ID	Host ID

Sol:- Classful IP Addressing

Class A (0-127)

1 byte	3 bytes
0 Net ID	Host ID

→ Large-sized organizations

Class B (128-191)

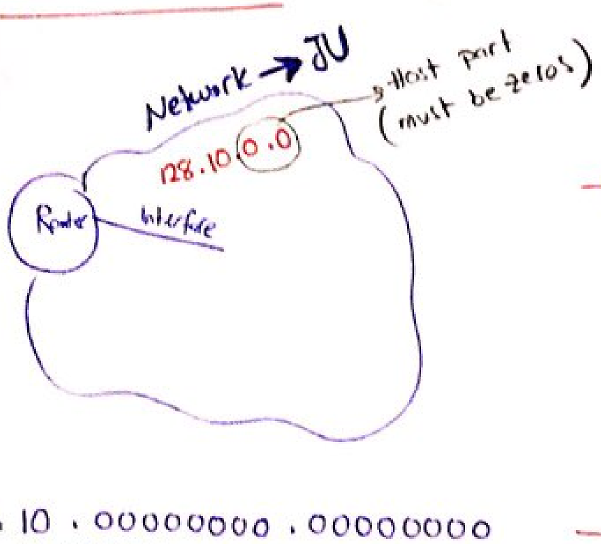
2 bytes	2 bytes
1 0 Network ID	Host ID

→ Medium-sized organizations.

Class C (192-223)

3 bytes	1 byte
Network ID	Host ID

→ Small-sized organizations



* to represent Network ID, the Host ID should be Zero

128 . 10 . 00000000 . 00000000 → Reserved (to represent Network ID)
128 . 10 . 00000000 . 00000001 → First Host PC 128.10.0.1
128 . 10 . 00000000 . 00000010 → Second Host 128.10.0.2
128 . 10 ⋮ ⋮

128 . 10 . 11111111 . 11111110 → Last PC
128 . 10 . 11111111 . 11111111 → Reserved for Broadcast Address

Notes:

8 . Nov . 2018

Addressing : ① Unicast (one-to-one)

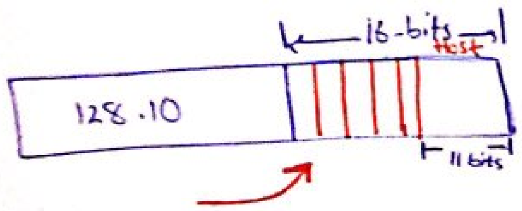
② Multicast (one-to-many)

③ Broadcast (one-to-All)

\# of PC's incorporated into this Network ?! $2^{16} - 2$ → Reserved (أول واحد وآخر واحد)

2 bytes .

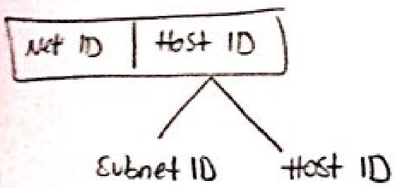
Smart Solution :- Subnetting (1985)



JU → 24 Faculties → Fs → How many bits to represent? → Each faculty can have up to

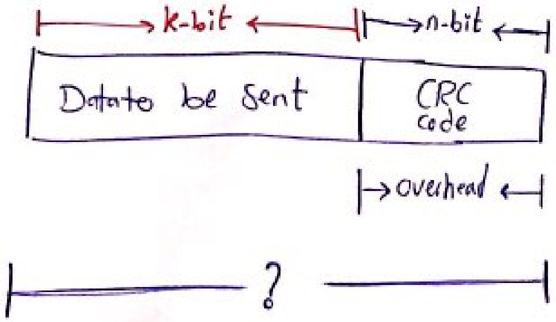
$2^{11} - 2 = 2046$ PC's

Final version



Scanned with CamScanner

Idea :



→ (k+n) bits have to be sent.

Concern :- How to generate CRC code ?

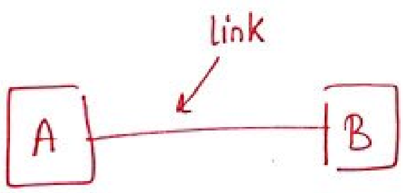
Generator Polynomial → will work at both sender and reciever

⇒ The highest power of this polynomial will be the length of CRC code.

For example, let $G(x) = x^3 + x^2 + 1$, then CRC length = 3 (bits.)

x^{3} power [illegible]

Example



Message: 101110
$G(x) = x^3 + 1$

$G(x) = x^3 + 1$

Question :- what is the generated CRC code ?!

CRC length = 3 bits

∴ Codeword (message to be sent = 101110 _?_ _?_ _?_)

→ use long division.

→

(11)

polynomial
coeffecient exist = 1
coeffeicient not exist = 0

x^3
x^2
x
Constant

```
              101001
      1001 ) 101110000   ← initially
        XOR ⊕1001
              001010
             ⊕1001
               001100
                 1001
                 01010
                  1001
                  0|011  CRC (3-bit)
```

leading zero's (ignored)

CRC = 011

Codeword = 10111 0011

Case I :- when the length is pure (perfect) → (No error exists)

- Recieved Codeword = 10111 0011
- B? will check if there is an error in the recieved codeword.

```
              10101
      1001 ) 1011 10011
             ⊕1001
              001010
                1001
                001101
                  1001
                  01001
                   1001
                   0000
```

ignored

Completely devisable.

Note
Ideal CRC length = 32 bits

18. Nov. 2018

Case II — Having Noisy Channel.

- Assume that the error patterns is 000000100

→ error pattern: 0 0 0 0 0 0 1 0 0

→ Recieved Codeword: 1 0 1 1 1 0 0 1 1

→ Recieved codeword (correct): 1 0 1 1 1 0 1 1 1

There is an error in this bit (must be flipped)

bit in error: 1 → 0, 0 → 1

→ long division :-

```
        101011
      _________
1001 | 101110111
     ⊕ 1001
      --------
     x 001010
     ⊕  1001
      --------
     x   001111
           1001
      --------
     x     01101
            1001
      --------
     x      0100
```

At this case, B declares that an error has been detected.

Solution → Retransmission

27. Nov. 2018

* Dual Computations :-

EIGRP Routing Metric

Query and Reply $\xrightarrow{\text{تشترك}}$ Dual Re-computation

Feasible distance → Computations

Remember :- RIP Routing Metric
↓
Hop count

- Successor → Next hop → Routing table
- Feasible successor → Next (hop) → (backup path) → Topology table.
- Reported Distance
- Feasibility Condition → Feasible successor

(R.D < F.D) → Advantage → having loop-free paths

How to calculate the cost ?? مهم

$$\text{EIGRP Metric} = \left[\frac{10{,}000{,}000}{\text{Slowest B.W (Kbps)}}\right] * 256 + \left(\frac{\text{Sum of delay}}{10\,\mu}\right) * 256$$

Example Slide 40

$$\left[\frac{10{,}000{,}000}{1024}\right] * 256 + \frac{(100 + 20000)}{10} * 256$$

$$= 9765 * 256 + 514560 = \boxed{3014400}$$ Feasible Distance ≡ Cost

Reported Distance R_1 → $\left[\frac{10000000}{1544}\right] * 256 + \text{- - -}$

أقل ↗

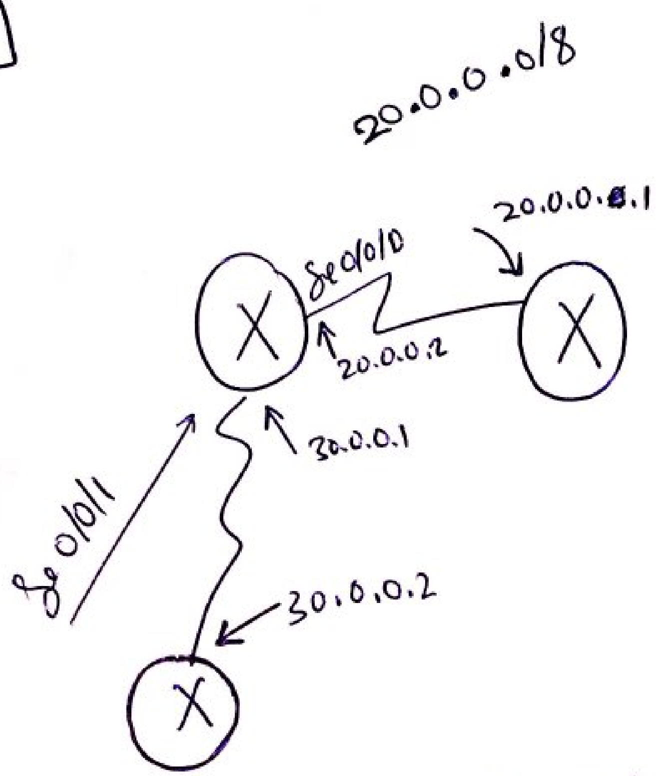
$$= 1657856 + 514560 = \boxed{2172416}$$ → Next hop (backup Root)

$\boxed{R_3 \rightarrow \text{successor}}$ $\boxed{R_1 \rightarrow \text{feasible successor}}$

2.Dec.2018

* show ip EIGRP neighbors :-

slide 13



Smooth - Round trip time → (2-ways)

without Noise

$\boxed{RTO = 6 \times \text{Round-Trip time}}$

Record 15:00

1	2	3	------------	16
↓	↓	↓		↓
200 ms	300 ms	450 ms		15000 ms
	↓			
	1.5(200 ms)	1.5(300)		

Computer Networks

Dynamic Routing Protocols (2)

Link-State Protocols → OSPF

→ ISIS (the first link-state protocol)

* الفكرة إنه بدل ما أخبر بس جاري ← بخبر الجميع بنفس الوقت

* Link-state packet → It gonna be flooded all over the Network.

Link State Packets

- The link state packets consist of the following information:
  1. The ***address*** of the node creating the LSP
  2. A ***list*** of directly connected neighbors to that node with the cost of the link to each neighbor
  3. A ***sequence number*** to make sure it is the most recent one
  4. A ***time-to-live*** to insure that an LSP doesn't circulate indefinitely
- A node (router) will only send an LSP if there is a (1) change of status to some of its links or if a timer (2) expires

LS Protocols: Dijkstra's algorithm

- ***Link-State (LS) Protocols***
  - Based on an algorithm by Dijkstra Not Bellman-ford
  - Each router on the network is assumed to know the state of the links to all its neighbors
  - Each router will disseminate (via reliable flooding of ***link state packets***, LSPs) the information about its link states to all routers in the network.
  - (*) In this case, every router will have enough information to build a complete map of the network and therefore is able to construct a ***Shortest Path Spanning Tree*** from itself to every other router

* wide flooding is not recommended :-

① Control over-head.

② Causes Network congestion.

* المفروض احنا نبعث داتا مش control

SPT (Shortest Path Tree) algorithm (Dijkstra)

- SPT = $\{a\}$
- for all nodes v
  - if v is adjacent to a then $D(v)$ = **cost (a, v)**
  - else $D(v) = infinity$
- Loop
  - *find* w not in SPT, where $D(w)$ is min
  - add w in SPT
  - for all v adjacent to w and not in SPT
    - $D(v) = min\ (D(v), D(w) + C(w, v))$
- until all nodes are in SPT

* dijkstra → wide flooding

* Distance vector كرحــدا تخبر جارك بس .

* ال conversion time بالـ (dijkstra) أسرع من DV

EIGRP → Distance vector
لـ بس ما بيعتمد كلى Bellman-ford
بيعتمد على dual Algorithm

---

* Routing Metric :-

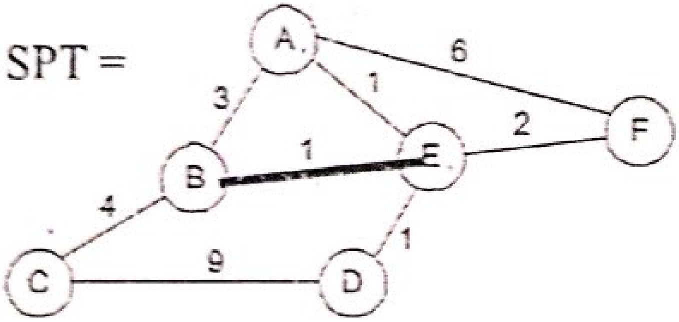
In EIGRP → Routing metric depends on BW, delay, $\boxed{\text{load , Reliability .}}$ → excluded due to the difficulty in finding these metrics.

* OSPF Routing Metric :- $\frac{10^8}{\text{Bandwidth}}$ bit/sec

↳ will be found for each link →
كل link ح يكون عليه cost خاصة, لو كان عندنا أكثر من link
ح يعطينا accomulative cost .

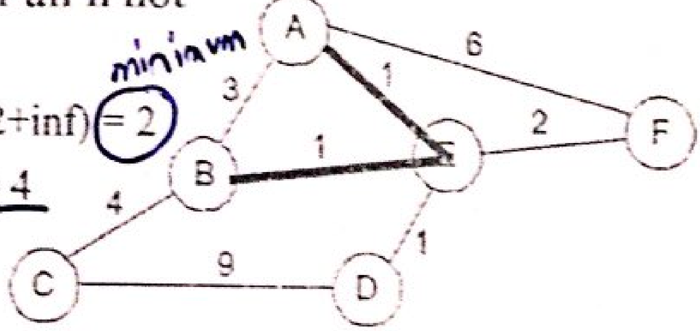
Dijkstra's Shortest Path Algorithm (1)

- Initialize shortest path tree SPT = {B} cost
- For each n not in SPT, C(n) = l(s,n)
  - C(E) = 1, C(A) = 3, C(C) = 4, C(others) = infinity
- Add closest node to the tree: SPT = SPT U {E} since C(E) is minimum for all w not in SPT. weight
  - No shorter path to E can ever be found via some other roundabout path.
- Shortest path tree SPT = {B, E}



Dijkstra's Shortest Path Algorithm (3)

- Loop again, select node with the lowest cost path:
  - C(A) = 2, C(D) = 2, C(F) = 3, C(C) = 4
  - SPT = SPT U {A} = {B, E, A}
  - No shorter path can be found from B to A, regardless of any new nodes added to tree
- Recalc: C(n) = MIN (C(n), C(A) + l(A,n)) for all n not yet in SPT
  - C(D) = MIN(2, 2+inf) = 2 minimum
  - C(F) = 3, C(C) = 4

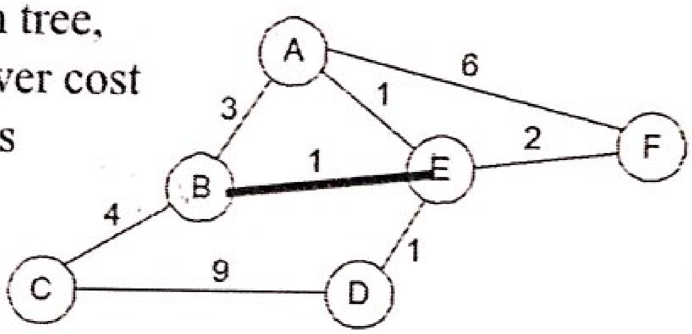


Dijkstra's Shortest Path Algorithm (2)

- Recalculate C(n) = MIN (C(n), C(E) + l(E,n)) for all nodes n not yet in SPT
  - C(A) = MIN(C(A)=3, 1 + 1) = MIN(3,2) = 2
  - C(D) = MIN(infinity, 1 + 1) = 2
  - C(F) = MIN(infinity, 1 + 2) = 3
  - C(C) = MIN(4, 1 + infinity) = 4

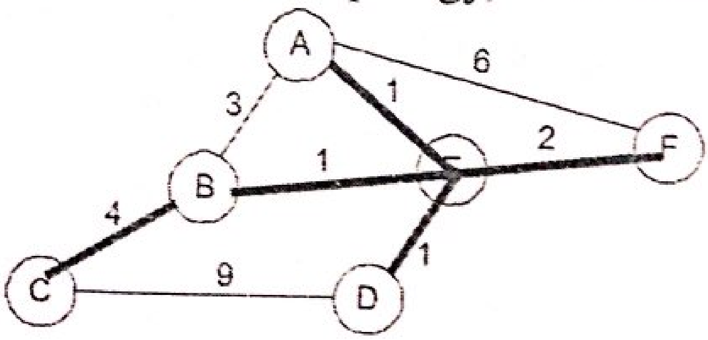
sp ، إذا ، لكن ، *
مهم

- Each new node in tree, could create a lower cost path, so redo costs



Dijkstra's Shortest Path Algorithm (4)

- Continue to loop, adding lowest cost node at each step and updating costs
- SPT crawls outward
- Remember to store the links in SPT as they are added (each node's predecessor is stored)
- Each node has to store the entire topology, or database of link costs



Scanned with CamScanner

TCP ⟨ Financial applications / Military applications ← معلومات مهمة بتحتاج ضمان وصول بستخدم (ACK) (TCP)

Transport Layer Protocols

أهم 2 protocols في هاي الـ layer

(Handshaking) Reliable ← TCP

Unreliable ← UDP

$\boxed{\text{TCP}}$ Transmission control Protocol → Reliable protocol

↳ يوفر في flow and error control

$\boxed{\text{Real-time streaming}}$ → Audio or video transmission

بنستخدم protocol خ UDP

Jargons

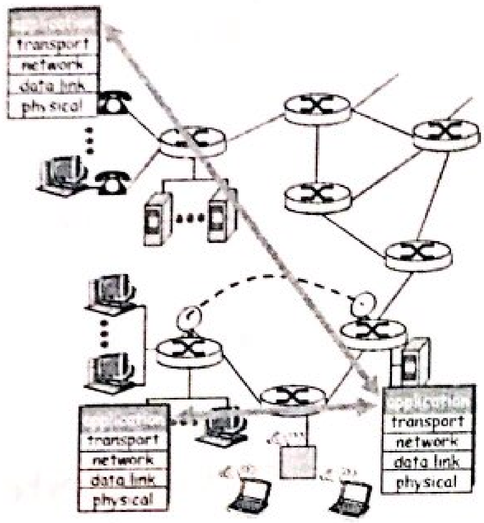
Process: program running within a host.

- within same host, two processes communicate using inter-process communication
- processes running in different hosts communicate with an application-layer protocol

Application-layer Protocols

Application-layer protocols

- one "piece" of an app
- define messages exchanged by apps and actions taken
- use communication services provided by lower layer protocols (TCP, UDP)



Client-Server Paradigm

معظم الناس بحكوا
مع بعض من خلالها

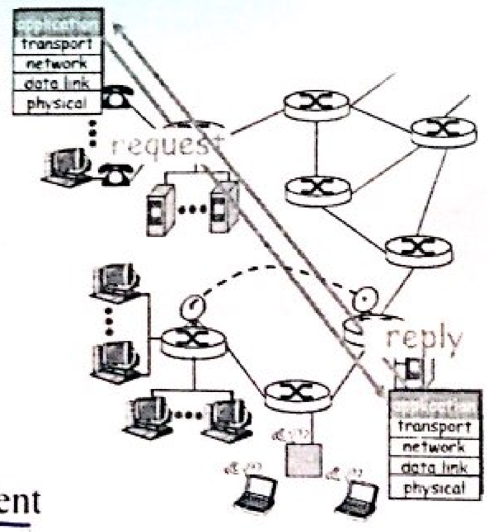
Typical network app has two pieces: *client* and *server*

Client:

- initiates contact with server
- typically requests service from server

Server:

- provides requested service to client



* Compression Process basically composed of
3 types of frames : I / P / B

Background
(I-frame)
when losing I-Frame ,
we will lose the whole
Frames (video)

* To see smooth video we need $\boxed{\text{30 frames/sec}}$

picturing rate to see smooth video

<u>Ack</u> لمتاش ملك مستقبل يكون هنا

* Point-to-Point Check → بتعمل check على damage مش ع drop

بتشغل error detection technique
(CRC)
وتصلح ال error (ما بتشتغلش عال drop)

* packets will be transfered over differet Roots, It's not necesserly to follow the same root, for that reason; It's impossible to check the drop problem.

* سـؤال مهم:- ليش عنا flow and error control بال Data link و transport layer

؟

الجواب
- Point-to-point → check (damage)
- End-to-End → check (Both damage and drop)

---

www. yahoo. com

Host ← www, com → domain Name

* Application layer protocol

بتحدد شو نوع ال application تبحثي، ومنها بتحدد ال Transport layer

وشو مصحح الأخطاء فيها مثلاً هيك ما بنستخدم TCP في transport.

Application-layer Protocols

Q: how does a process "identify" the other process with which it wants to communicate?

إذا في جهازين بينهم - بحكو مع بعض بحتاج IP address

- IP address
  - of host running other process

بتنظم الـ requests الي بتوصلنا

- Port number
  - allows receiving host to determine to which local process the message should be delivered

موجود بـ transport layer

Application layer protocols
↓
server الكل protocol
مثلاً FTP protocol → FTP server
HTTP protocol → HTTP server

5

Transport Service Requirements

3 major concerns:

Application	Data loss	Bandwidth	Time Sensitive
file transfer	no loss	elastic	no
e-mail	no loss	elastic	no
Web documents	loss-tolerant	elastic	no
real-time audio/video	loss-tolerant	audio: 5Kb-1Mb video:10Kb-5Mb	yes, 100's msec
stored audio/video	loss-tolerant	same as above	yes, few secs
interactive games	loss-tolerant	few Kbps up	yes, 100's msec
financial apps	no loss	elastic	

بستخدم TCP لأنه Time sensitive

لبعض الأبلكيشن تتحمل data loss لما نحصل بستخدم UDP

إذا Time كان مهم بستخدم TCP

مهم يكون عالي BW utilization

في شي اسمه CRC بالداتا كانت نستخدم 4-bytes مشان أتحقق

الداتا في Noise بستخدم لللا نفسها

7

Transport Services

Data loss
- some apps can tolerate some loss
- other apps require 100% reliable data transfer

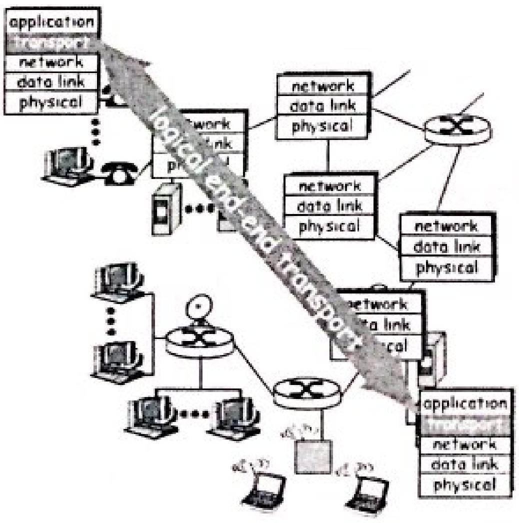
Timing
- some apps require low delay to be "effective"

Bandwidth
- some apps require minimum amount of bandwidth to be "effective"
- other apps ("elastic apps") make use of whatever bandwidth they get

Transport-Layer Protocols

Services:

- Reliable, in-order delivery (TCP)
- Unreliable ("best-effort"), unordered delivery: UDP
- Services not available:
  - real-time
  - bandwidth guarantees



TCP Services

- *Connection-oriented:* setup required between client and server → لازم يكون فيه Handshaking بين ال client وال server.
- *Reliable transport* between sending and receiving processes
- *Flow control:* sender won't overwhelm receiver
- *Congestion control:* throttle sender when network overloaded
- *No*
  - timing
  - minimum bandwidth guarantees

Internet Application and Transport Protocols

Application	Application layer protocol	Underlying transport protocol
e-mail	smtp [RFC 821]	TCP
remote terminal access	telnet [RFC 854]	TCP
Web	http [RFC 2068]	TCP
file transfer	ftp [RFC 959]	TCP
streaming multimedia	proprietary (e.g. RealNetworks)	TCP or UDP
Internet telephony	proprietary	typically UDP

ال video عبارة عن frames ((I,P,B)) نفترض إنه كل وحدة منهم عبارة عن packet لو كذا فى مسح ودي بيتقسمو

ف ال packet header نستخدم فيه TCP لأنه فيها معلومات عن source و Des
عشان لازم تضيع

أما الباقي بنستخدم فيها UDP

UDP Services

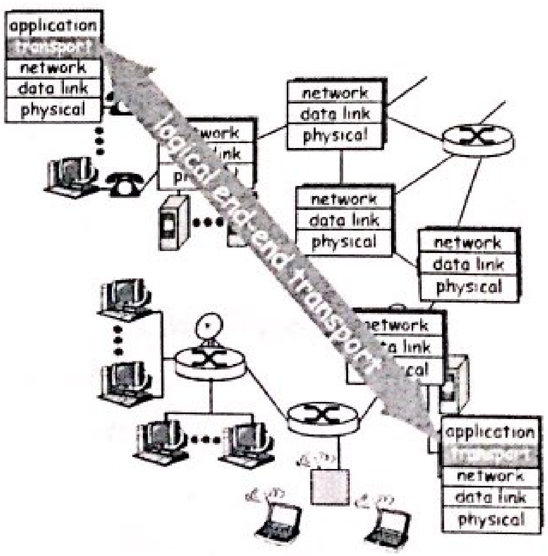
connectionless

- unreliable data transfer between sending and receiving processes
- Does not provide
  - connection setup
  - reliability
  - flow control
  - congestion control
  - timing guarantee
  - bandwidth guarantee

Q: Why is there a UDP?

Transport Services and Protocols

- Provide *logical communication* between processes running on different hosts
- Transport protocols run in end systems
- *Network layer*
  - data transfer between end systems
- *Transport layer*
  - data transfer between processes

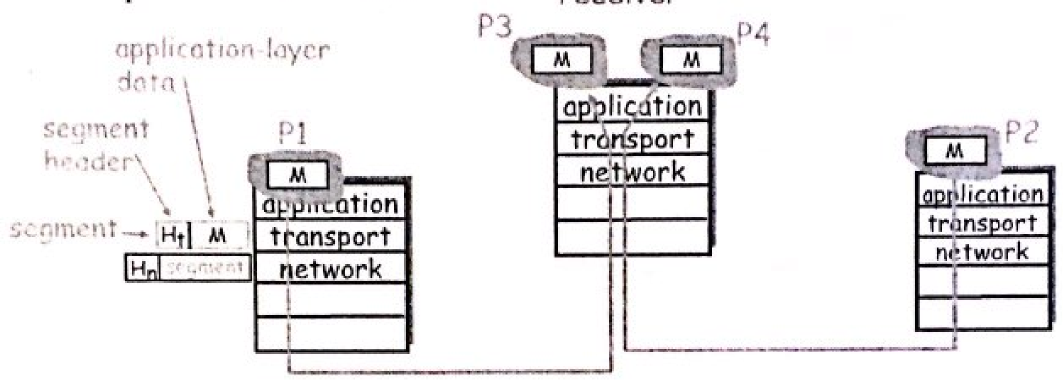


* ال Reciever اله اكتر من server و كل server
بيستقبل على port معين

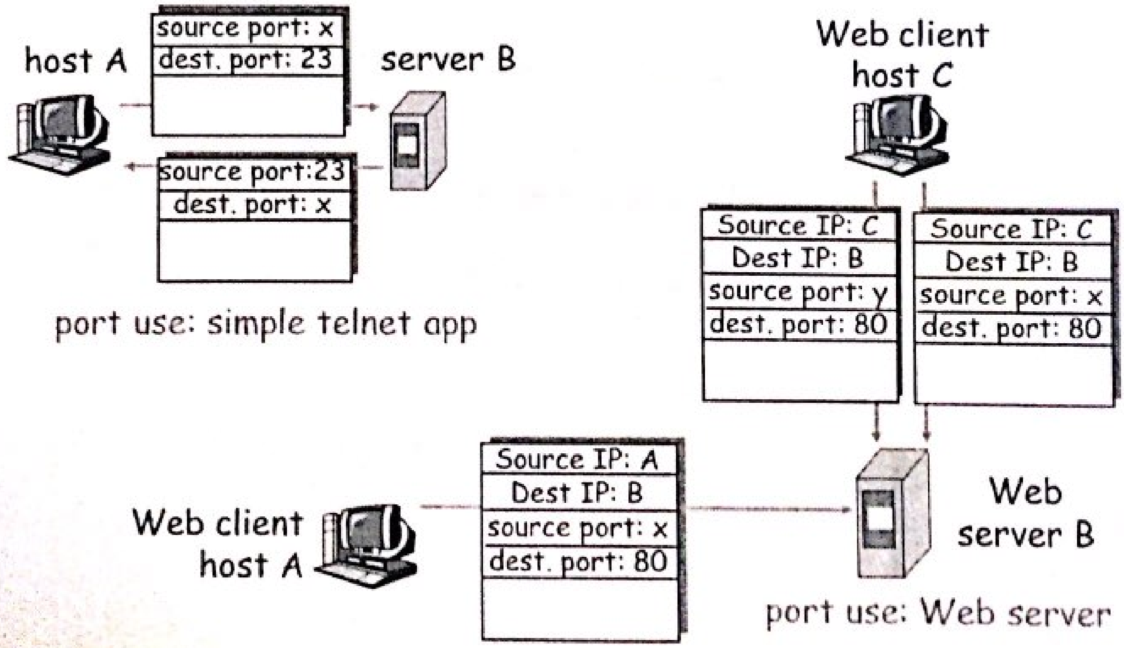
Demultiplexing

Recall: *segment* - unit of data exchanged between transport layer entities
- aka TPDU: transport protocol data unit

Demultiplexing: delivering received segments to correct app layer processes



Examples



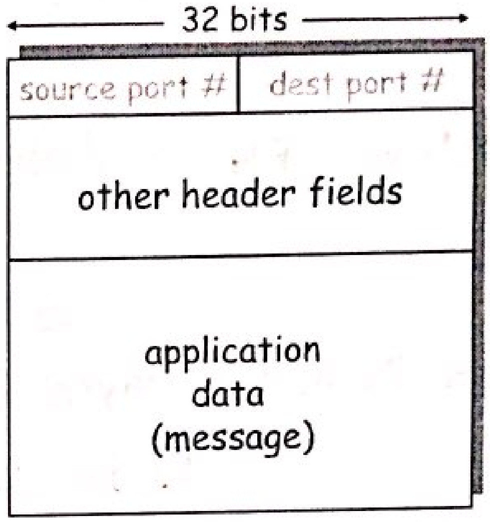
Multiplexing

Multiplexing:

gathering data from multiple app processes, enveloping data with header (later used for demultiplexing)

- Well-know port numbers defined in RFC 1700, e.g.,

  HTTP: 80
  FTP: 21
  Telnet: 23



TCP/UDP segment format

UDP: User Datagram Protocol

- "Bare Bones" Internet transport protocol
- "Best effort" service, UDP segments may be:
  - lost
  - delivered out of order to app
- *Connectionless:*
  - no handshaking between UDP sender, receiver
  - each UDP segment handled independently of others

Why is there a UDP?

- no connection establishment (which can add delay)
- simple: no connection state at sender, receiver
- Often used for streaming multimedia apps
  - loss tolerant
  - rate sensitive

* Transport layer Protocols :-

- main protocols : (1) TCP (Transmission control protocol) → Reliable
  (2) UDP (User Datagram protocol) → Unreliable

* there is Flow and error control in { Data link layer / Transport layer } why ?!

→ in Data link layer (point-to-Point check)
↳ for damage only

in Transport layer (End-to-End check)
↳ for damage and drop .

أزمة

TCP Congestion Control

- TCP has a mechanism for congestion control. The mechanism is implemented at the sender
- The window size at the sender is set as follows:

Send Window = MIN (flow control window, congestion window)

where

- flow control window is advertised by the receiver
- congestion window is adjusted based on feedback from the network

Point-to-point → check for damage problem
End-to-End → check for both damage and drop + congestion control

85 جزء → Additive Increase multiplicative decrease Congestion avoidance and slow-start

TCP Variation: TCP Tahoe

- 1^{st} improvement was TCP Tahoe (1988)
  - Adjusts sending window as congestion increases or decreases (AIMD congestion avoidance & slow-start)
  - Improved retransmission policy (Fast Retransmit)

congestion protocol → transport layer في الموجود → congestion window
Flow and error control → link layer → sliding window

غير أكثر موجود فيهم

What Are TCP Variations?

- Implementations of TCP that use different algorithms to achieve end-to-end congestion control.
  - Tahoe
  - Reno
  - NewReno

ive

TCP Tahoe Window Control

- *TCP sender* maintains two new variables:
  - cwnd – congestion window

    cwnd is inferred from the level of congestion in the network.
  - ssthresh – slow-start threshold

    ssthresh can be thought of as an estimate of the level below which congestion is not expected.

Slow Start Phase

(cwnd < ssthresh)

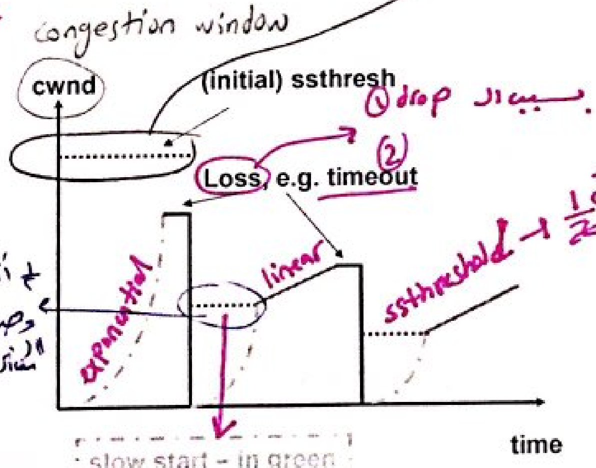
- Initially:
  - cwnd = 1*MSS (Maximum Segment Size)
  - ssthresh is very large.
- If no loss:
  - cwnd += 1*MSS (after each new ACK)
  - *(This gives exponential growth of cwnd)*
- If loss (timeout):
  - ssthresh = max(flight size/2, 2*MSS)
  - cwnd = 1*MSS

اول شي بيبعت 1 وإذا
إجى ACK بيبعت 2 وإذا
إجى ACK بيبعت 4 وبعدين 8
(exponential) 2^n

Slow Start & Congestion Avoidance

بيلش ب MSS = 1

- Initally: Maximum segment size
  - $\boxed{\text{cwnd = 1*MSS}}$
  - ssthresh = very high (65535)



- If a new ACK comes:
  - if cwnd < ssthresh → update cwnd according to slow start
  - if cwnd ≥ ssthresh → update cwnd according to congestion avoidance
  - If cwnd = ssthresh → either

← أكبر قيمة وصلها "conversion"

(*) إذا إجى ACK السهم كلهم ما بيرجعوا و 180 رك

- If timeout (i.e. loss) :
  - ssthresh = flight size/2;
  - cwnd = 1*MSS

maximum segment size

لوكنت ب slow-start وبعتت 16 وبعدين إجى ACK بيصيروا (32) أما بال avoidance بيصيروا 17 بشرط يجي ACK الهم كلهم)

* بيلش slow-start لغاية حد معين إذا وصلناه معناها ما بنقدر نكمل بنفس ال exponential ← فمنشان هيك صار linear

Congestion Avoidance Phase
(cwnd > ssthresh)

- If no loss:
  - increase cwnd *at most* 1*MSS per RTT (additive increase)
  - cwnd += (MSS*MSS / cwnd) on every ACK (*approximation* to increasing cwnd by 1*MSS per RTT)

- If loss:
  - ssthresh = max (flight size/2, 2*MSS) (multiplicative decrease)
  - cwnd = 1*MSS.

سؤال جاي

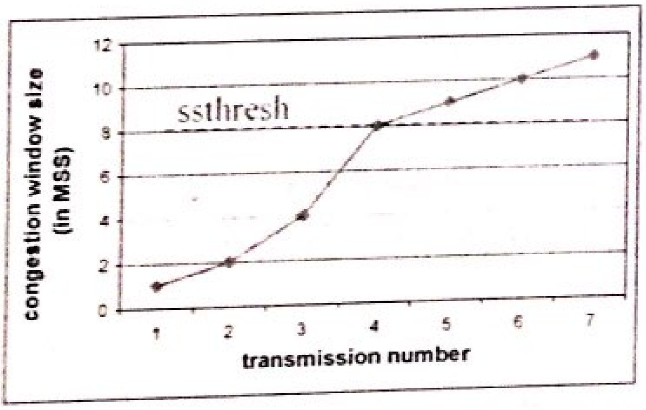
* ليش عنا flow control بـ transport layer بالرغم من وجودها بـ Data link layer

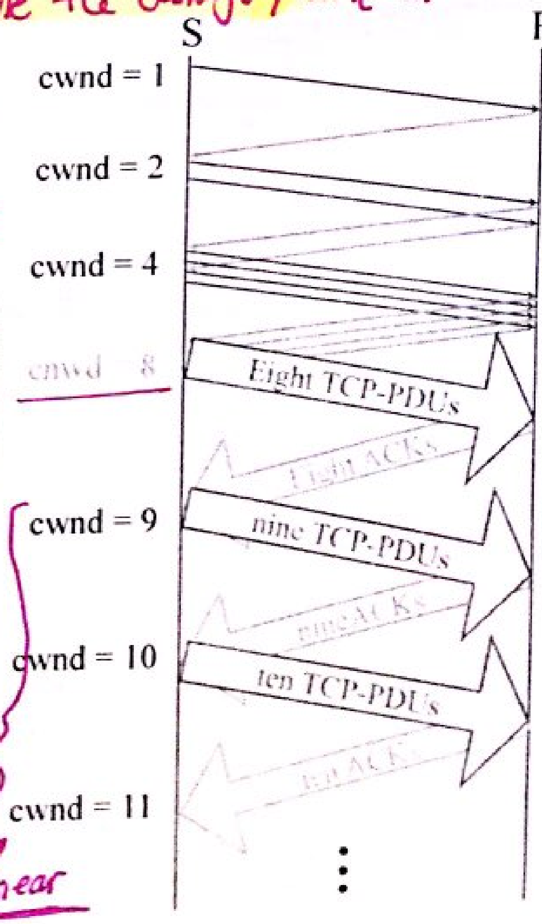
packet loss → drop / damage

we employ flow and error control in data-link layer to handle the damage, while in transport layer to handle both damage and drop

Example: Slow Start/Congestion Avoidance

assume ssthresh = 8*MSS → 8*1 = 8 segments





flight size → أعلى قيمة وصلتها الـ congestion window

Scanned with CamScanner

TCP Tahoe's Retransmission Policy

- When a segment is lost, original TCP waits for an ACK that's not coming and eventually times-out.
- Often, many, if not all, of the segments sent after the lost segment arrive at the receiver.
- For each segment received, the receiver sends a duplicate ACK, notifying the sender that the receiver is waiting for the missing segment.
- TCP Tahoe interprets duplicate ACK's as an indication that a segment was lost.

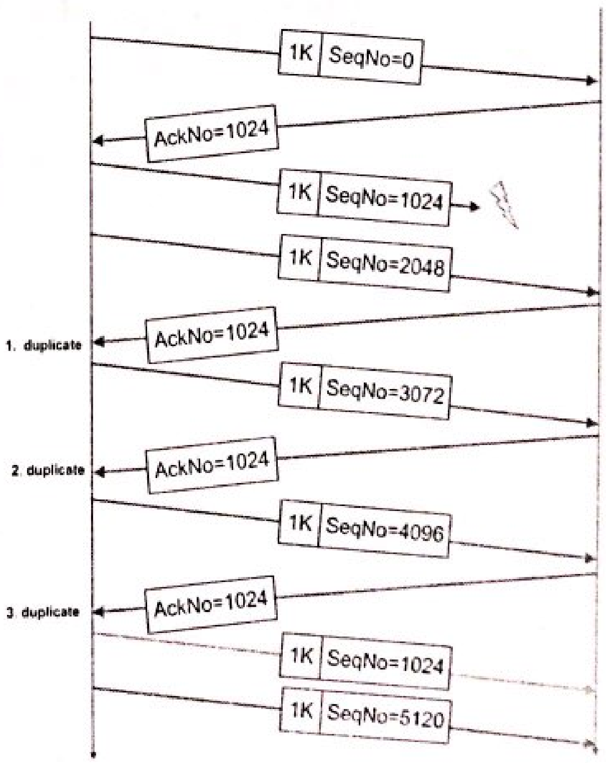
Fast Retransmit (Cont.)

IF

ما بستنى الـ meout

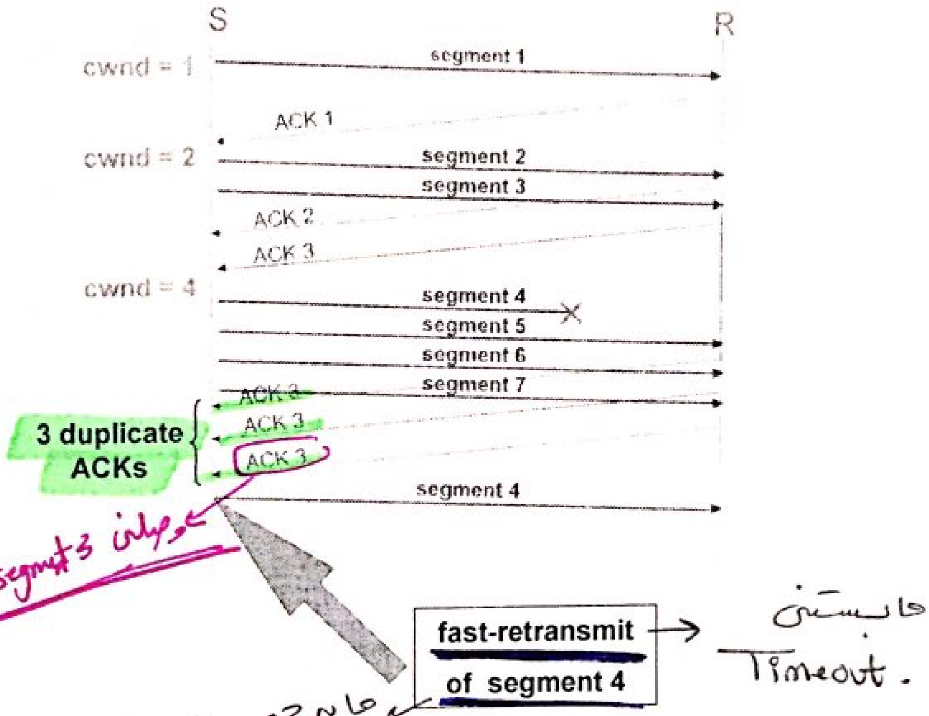
If three or more duplicate ACKs are received in a row, the TCP sender believes that a segment has been lost.

- Then TCP performs a retransmission of what seems to be the missing segment, without waiting for a timeout to happen.
- Enter slow start:
  ssthresh = cwnd/2
  cwnd = 1



TCP Tahoe's Fast Retransmit

1. Sender receives 3 dupACKS.
2. Sender infers that the segment is lost.
3. Sender re-sends the segment immediately!
4. Sender returns to slow-start.



ما برجع يبعث 5,6,7 كمان مرة

* لازم يكون خزانـــــش عند Reciever ليرجع يعمل Re-order للsegments

TCP Variation: *TCP Reno*

- 2nd Improvement was TCP Reno (1990)
  - From Tahoe:
    - AIMD congestion avoidance with slow-start
    - Fast retransmit
  - New to Reno:
    - Fast recovery

* → Tahoe → Fast Retransmission

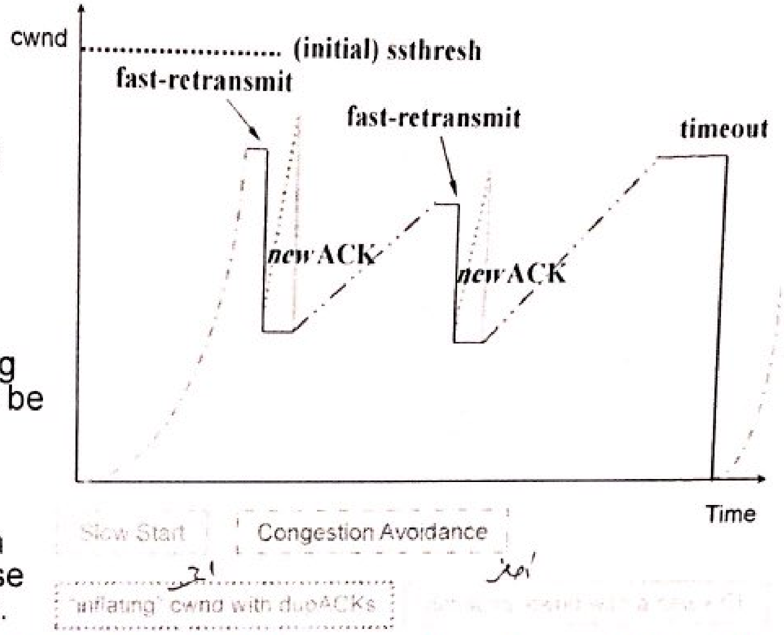
Fast Recovery

Concept:

- After fast retransmit, reduce cwnd by half, and continue sending segments at this reduced level.

Observations:

- Receiver is still getting T-PDUs. There can't be overwhelming congestion.
- How does sender transmit T-PDUs on a dupACK? Need to use a "trick" - inflate cwnd.



Fast Recovery ← بمش slowstart لغاية نقطة معينة لما يصير Timeout او 3dupACKs
بعمل Retransmition او Fastretransmit
في حالة Timeout ، في حالة 3dup ACK's (ما خلص Timeout)
← ما بينزل للـ(1) بصير هنا اشي اسمه inflating و deflating

TCP Reno (Revisited)

* لما يوصلني 3dupAck مباشرة بعمل Retransmit
بس بعد هيك بروح على fast-recovery يعني :-
اذا اجى New duplicate غير اول 3
بعمل inflating للـ congestion window (بتزيد)

- Duplicate ACKs:
  - Fast retransmit
  - Fast recovery

  → Fast Recovery avoids slow start

- Timeout:
  - Retransmit
  - Slow Start

- TCP Reno improves upon TCP Tahoe when a single packet is dropped in a round-trip time.

Fast Recovery (Cont.)

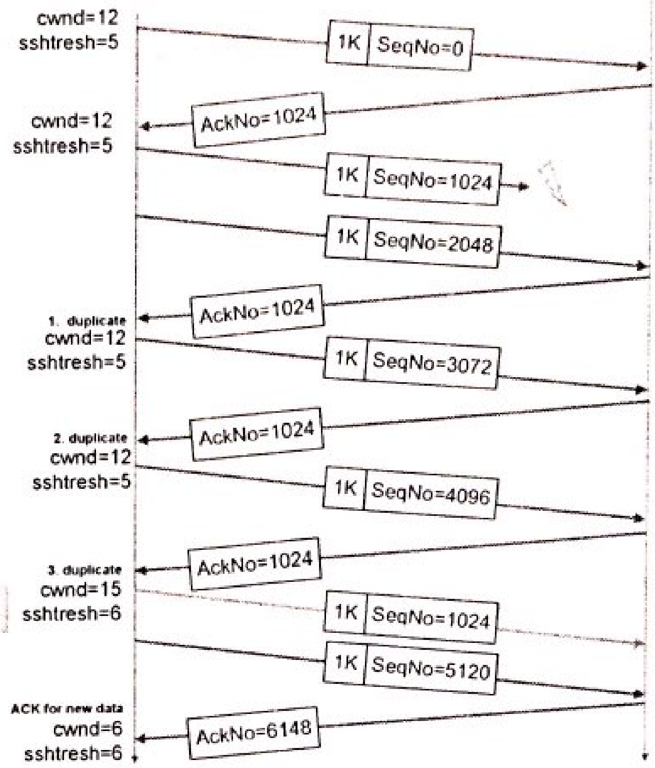
- Fast recovery avoids slow start after a fast retransmit
- **Intuition:** Duplicate ACKs indicate that data is getting through
- After three duplicate ACKs set
  - Retransmit packet that is presumed lost
  - ssthresh = cwnd/2
  - cwnd = (cwnd/2)+3
  - (note the order of operations)
  - Increment cwnd by one for each additional duplicate ACK

cwnd=9 (not 15)

- When ACK arrives that acknowledges "new data" (here: AckNo=6148), set

  cwnd=ssthresh

  enter congestion avoidance



Fast Retransmit / Fast Recovery in TCP Reno

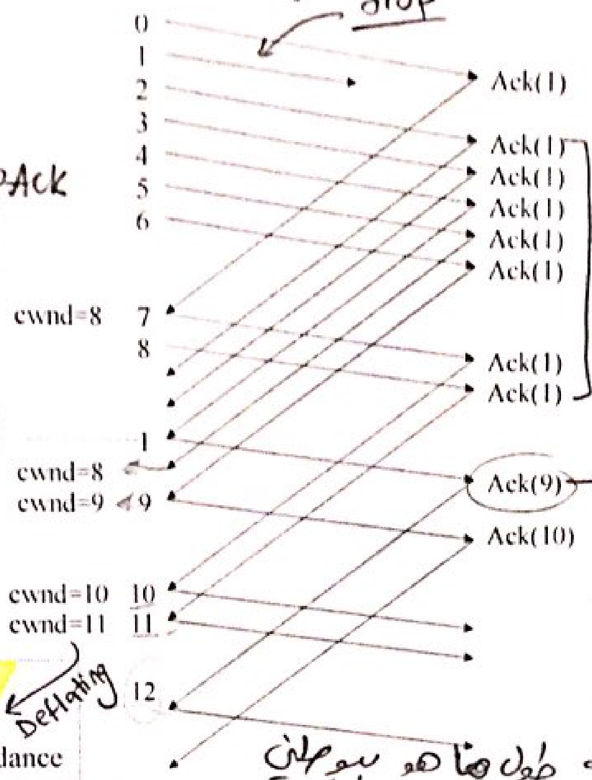
- A sender uses fast retransmit / fast recovery algorithm to improve throughput of TCP
  - "Fast" – because it doesn't wait for *time out* when not getting an ACK for a segment
- **Fast Retransmit** – after 3 "**duplicative ACKs**", the sender assumes that the segment was lost, retransmits the segment and moves to Fast Recovery phase
- **Fast Recovery** – the sender decreases Congestion Window (cwnd) twice of its original size, adds 3 (3 packets have left the network and buffered by the receiver) and continue to send new segments (if allowed by the cwnd value) until receiving new different ACK, which should acknowledge receiving all segments sent till moving to Fast Recovery phase (assuming that no more segments were lost).
  - For each additional duplicated ACK received, increment cwnd by 1

مهم جداً للفاينل

Example

Initial state
cwnd=7
Slow start

آخر Segment   3 dupACK
8/2 + 3



duplicate Ack's

Not duplicate
يعمل Exit

مباشرة. يروح عليها
ما يروح على slow start

Fast Retransmit
cwnd=8/2+3=7
ssthresh=8/2=4
=> Fast Recovery

Exit Fast Recovery
cwnd=ssthresh=4
=> Congestion Avoidance

بعمل Inflating لأنه طول ما هو بوصلني Ack بزيد الـ congestion، لما يجي أول Ack لـ New Frame بعمل deflating لل congestion window

35:00

Limitation of TCP Reno algorithm

- If **cwnd** size is too small (smaller than 4 packets) then it's not possible to get 3 duplicate acks and run the algorithm
- The algorithm can not manage a loss of multiple packets from a single window of data
  - It will cause a use of retransmission time out
- The algorithm doesn't manage a loss of packets during the Fast Recovery stage
  - Not a loss of the retransmitted packet
  - There is no recursive run of the Fast Retransmit