12/1/19

PAGE NO.:
DATE: / /

① point to point communic^n
② end to end "

→ Wireless connec^n : possible only when devices are in the vicinity of each other.

→ Medium only allows analog signal
→ Comp. provides o/p in 0's & 1's
So, to convert digital → analog : we need a Modem
(vice-versa : Demodulation)

↓ How to convert

→ Content - based Networking
→ Info. - centric N/W : take very quick decisions & transmit your packets at dest^n.

cable TV • changing channel ? requesting STB to switch to another frequency

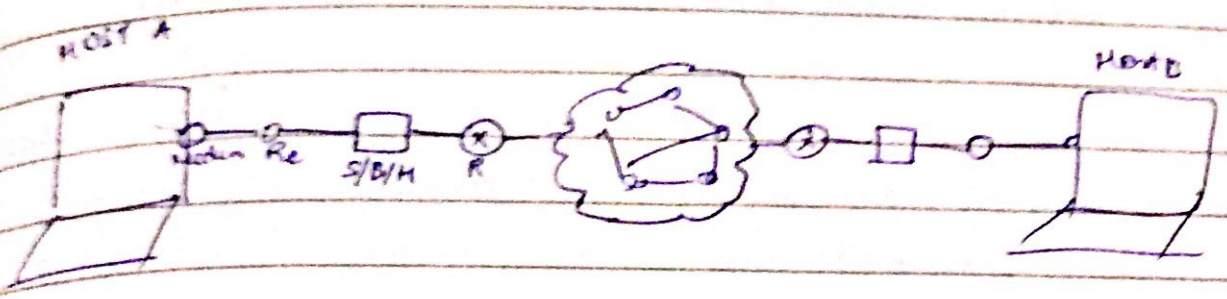
→ cable connection : Had booster to have good quality of video
↓ is a
Repeater (Re) → amplifies the signal (improves quality)
Physical level layer device

• For point - to - point connec^n : we require
Switch / Bridge / Hub (Data link level layer devices)
↓
connected to Router

Teacher's Signature.....

1) Centralized
2) DC
3) Distributed

- Right now, we only use switches — layer 2 (works for DL layer)
  layer 3 (works for NW layer)



App" layer : end-to-end connec".

- Original msg : Text form

→ Appl" layer : Convert text to hypertext form { have to match interpretes in all OSs }
(so that it works on all OS)

(need

Protocols : HTTP / FTP / HTTPS / SMTP / SNMP etc.

Transfer protocols

Trp : layers in TCP/IP : end-to-end connec" (logical connec"s .....)

→ ① Transfer msg to some form so that it works on our n/w.

→ URLs (can remember) mapped to addresses (N/W needs this Add.)
Many servers are available When you write a URL, whichever server is available will provide connec" to you.

DNS

→ Hierarchial add. scheme exists

.in
.ac.in .gov.in

DNS :→ . URL request reaches DNS server, checks db & forwards corresponding IP add to your system.

Teacher's Signature .....

Wireshark

PAGE NO.:
DATE: / /

→ Till now, we don't know dest^n.

1st, logical connec^n needs to be established by Xport layers

TCP ✓ UDP

Objective of Xport layer : (logical connec^n)

① To establish an end-to-end connec^n — connection oriented (Ack received) TCP; Connectionless (No Ack) UDP

Accessing Wi-fi : UDP

" Bank ... : TCP

② Flow control → Goback N, selective Repeat, etc...

↓

May have different bandwidths

Otherwise, data may be lost

③ Congestion control

Intermediate nodes may be connected to other n/w also (↑ traffic) ⇒ Congestion may occur.

④ Reliable and Unreliable Connection

Reliable → Banking Services (TCP)

Unreliable → e-mail (UDP)

* → Port Add (depends on app^{n}s) : 16 bits

~~Network~~

* 3 add is n/w

TL : Port Add : 16 bits

HTTP : 80
HTTPS : 483

Network layer : Takes care to forward data

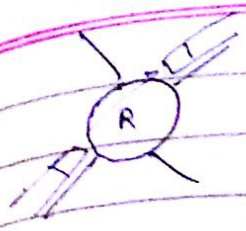
Objectives :

① Router : Has multiple interfaces

Func^{n}s at NL : 1) Send 2) Receive 3) Forward 4) Drop

Router :

Teacher's Signature.....



Each interface generally maintains 2 queues.
→ Have 3 interfaces to fwd the packet
↓
select acc. to packet add

delays
* 1) processing
* 2) Propog"
* 3) Transmission
4) Queuing

③ Quality of Service (QoS)
→ Talk about Wired & Wireless at N/w layer.

Appl" Layer

Services : DNS (Name service), HTTP (Web), SMTP, POP, IMAP (E Mail), FTP (File Transfer)

HTTP (Port No. 80 / 8080)

Data Center : Many servers
Problem : How to replicate data from 1 server to another

Cookies → can't find if it is http or not (if we use IP add → can identify IPV4/V6)

HTTP by default is a stateless protocol
↳ nothing is stored. (No state info is stored)

We Cookies to store state info. Client forwards the add"al info along with Request msg by reading cookie

Gmail : if you don't log out, next time you'll directly see your inbox.

• Gmail sends 1 cookie to you. 1 cookie stores

generally, social media ac's cookies : active till 15 days.

Teacher's Signature

Anonymyzer. ✕

PAGE NO.:
DATE: / /

After 15 days, if you log in, you're to enter username and pswd.

→ Session Hijacking Attack : If you're active session, he may steal the info from cookie & further communicate with server

↓ SOT"

One Time Password (OTP)

↳ Now, people are attacking on this also.

HTTP Proxy

→ Use of Proxy server : To access some blocked site, you bypass the proxy server.

DNS

↳ Assign unique names to an IP add.

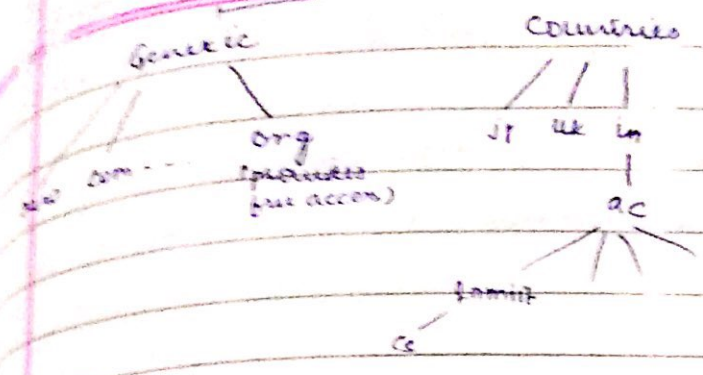
→ DHCP assigns IP add. to host system.
↓
dynamic host config" protocol

↳ 2 hosts have same IP add. connected to WLAN.
They won't be connected to Internet (because server will be confused) → IP conflict.
↓
That's why, have to maintain unique IP names.

Teacher's Signature.....

PAGE NO.:
DATE: / /

DNS



→ maintain hierarchical addressing

Domain Resource Records

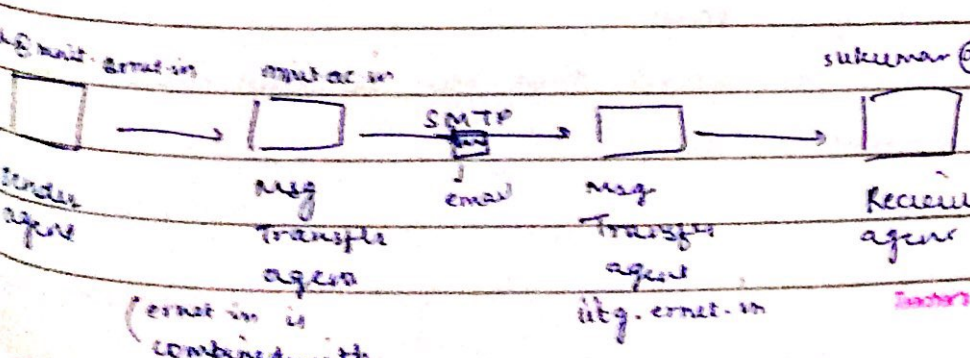
- domain_name : case insensitive comp name ≤ 63 characters & full path ≤ 255 char.
- Time_to_live : Time for which server is active (if inactive, has to contact → ac → in> if active directly committ will give response)
- Class : Normally IN - Internet Resources
- Type : Type of record
- Value : Value of record (IP add for A record type)
  - A record → IPV4 add.

Why DNS uses UDP ?

- UDP is much faster. TCP req. handshake time. It'll create traffic
- DNS request & responses are very small & fits well within 1 UDP segment.
- UDP is unreliable. In DNS reliability is ensured at Appl" layer.

E-mail

- 3 agents
  - User
  - Transfer



Teacher's Signature

[codec] → email header info

to see origin of e-mail (shows server agent)

PAGE NO.:
DATE: / /

SMTP (Simple Mail Transfer Protocol) → RFC 821, RFC 5321
↳ (Port No. : 25)

MIME → Multipurpose Internet Mail Extension
↳ Use multi-language & multimedia contents inside a mail.

↳ ~~content driven~~

↳ need to maintain 2 sockets. At your side, at the server's side
(Port no.: any random no.) <1024 (Port no. : 25)

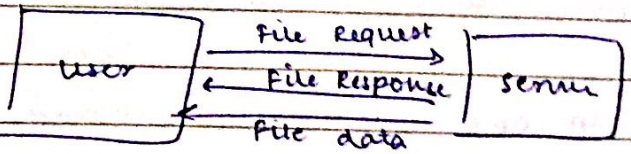
stateful protocols

POP 3 (Post Office Protocol V3) → earlier protocol for email delivery

IMAP V4 (Internet Message Access Protocol)
↳ uses port no. : 143

FTP (File Transfer Protocol) [IP Messenger : send upto 1GB]
↳ Port no. : 21

→ built on client-server model



Performance Metrics in CN

- Bandwidth (Throughput)
  - ↳ Width of freq. band
  - ↳ No of bits per second that can be transmitted over a communication link

Teacher's Signature

many users ~~using~~ listening RedFM at a time. Have to allot diff bandwidth to each user.

BOOK : CN by Kurose & Ross

PAGE NO.:
DATE: / /

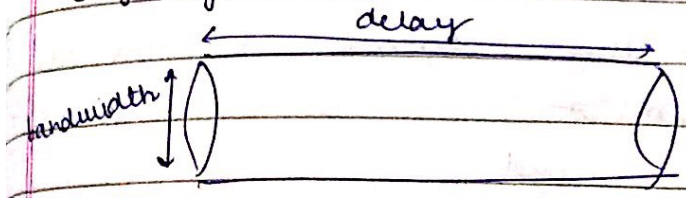
1 Mbps = 10^6 bits / sec = 1×2^{20} bits / sec

10^{-6} sec to transmit each bit or each bit occupies ~~for~~ 1 μs space.

- Latency = Propogation + transmit + queue (3 delays)
  - ↓ RTT/2 (generally)
- Propogⁿ = distance / speed of light
- Transmit = size / bandwidth

1 bit transmission : propgⁿ imp

large bytes : bandwidth imp.



BDP :

large file transfer : ~~cattle~~ bandwidth is critical

small : latency " "

Infinite bandwidth

- Throughput = Transfer size / time
- Transfer time = RTT + 1 / (bw x Transfer size)
- RTT dominates

Date : Signature

PAGE NO.:
DATE: / /

Q. Calculate total time to transfer a 1.5 MB file in following cases, assuming a RTT of 80ms, a pkt size of 1 KB data & an initial 2×RTT of "handshaking" before data is sent.

(a) The bandwidth is 10 Mbps, & data pkts can be sent continuously

(b) " " " " , but after we finish sending each data pkt we must wait 1 RTT before sending next

(c) The link allows infinitely fast transmit, but limits bandwidth s.t. only 20 pkts can be sent per RTT

(a) Before data is sent : $2 \times 80 = 160$ ms

Total no. of pkts : $\frac{1.5 \times 1024 \text{ KB}}{1 \text{ KB}} = 1536$

~~Time~~ Propogation time : 40 ms (1 side ka)

~~Total Time = 160 ms~~

bandwidth ↑ ↓

10 Mb → 1 second

1.5 MB → 0.15 second = 150 ms

Total Time = 160 + 150 = 310 ms

(b) 160 + 150 + 80 (1536) = 123190 ms

= 123.2 s

(c) $\frac{1536}{20} = 76.8$

77 × 80 ms = 6160 ms = 6 sec

(a) $160 + \frac{1258912}{10000000} + \frac{RTT}{2}$ (Propg) $\simeq 1.458$ sec.

(b) $1.458 + 122.8 = 124.258$

(c)

Teacher's Signature.....

PAGE NO.:
DATE: / /

Transport layer

HTTP header → has get & POST Method.
" data →

Protocols
- UDP
- TCP

1st obj : TO provide end-to-end communic" (for both)

UDP	TCP
↳ Unreliable	↳ Reliable
↳ Conn" less	↳ Conn" Oriented
	↳ Flow & congestion Control
	↳ Ordered Pkt Delivery.

↳ Ad Audio/Video : always require ordering of frames

Connec" Establishment : Req + Ack (2-way handshake)

⇓

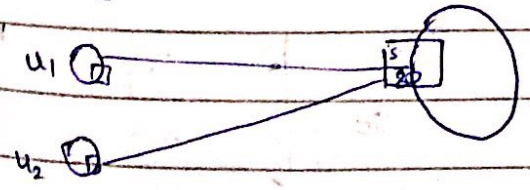
Problem : ↳ Some pkts get lost
↳ Whether ~~server~~ client is reponding or not : can't identify.

↳ How it server differentiate whether Req is new or duplicate of old one ?

Ans: Use Throwaway Transport Add (Port No)

↳ Don't use a port no. if it has been already - delayed duplicate pkts 'll never find their way to a tranport procss

↳ Is it feasible ?



all are connected to http. Same 80 will provide service to all users with diff. procs
We can't delete 80. as it is providing services to many users.

Teacher's Signature .....

PAGE NO.:
DATE: / /

Sol^n 2 : Give each conn^n a unique identifier chosen by the initiating party & put in each statement.

- Problem :

Sol^n 3 : Devise a mechanism to kill off aged pkts that 're still hobbling about, Restrict the pkts timeline.

3 ways

1.) Prevent pkts from looping

* 2.) Putting hop count in each pkt - decrement at each hop (initialize to max value)

3.) Time stamping : in real life, sync. is always a problem

9/2/19

We 've 2 types of conn's → that's why we 've 2 protocols

① Conn^n Oriented :

How to establish conn^n ?

1.) 3 - way handshaking

2.) 2 - way "

4 Most of appl^n services → HTTP (& TCP)

4 First you type URL → sent to DNS → returns IP add. to your system → contact the server. → server responds back.

Cases possible :

1.) Request is lost → How client will get his answers ?

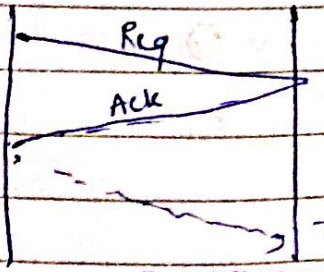
Sol^n

Timeout

doesn't recieve any response by server, wait upto RTT (sending + receiving time) & ~~wait~~ ~~opt~~ send request again

Acknowledgement

TO



Teacher's Signature

DOS attack : like TCP, 3-way handshaking

within timeout, you've to recieve ack. Else → send request again

2-way handshaking
↓ disadv:

how to authenticate really connected to the client ?

(how server will authenticate whether a frame is original or duplicate (sent again)) → (if ACK not sent within timeout / if ACK is lost, etc)

• server never maintains any timeout. Timeout is only maintain by clients.

→ ARQ (Automatic Requests)

2 things we want to achieve
1) Reliability
2) Flow control

5 Objectives (in general)
1) End-to-end comm"
2) Reliability
3) Flow Control
4) Congestion Control
5) Reliable Transmission

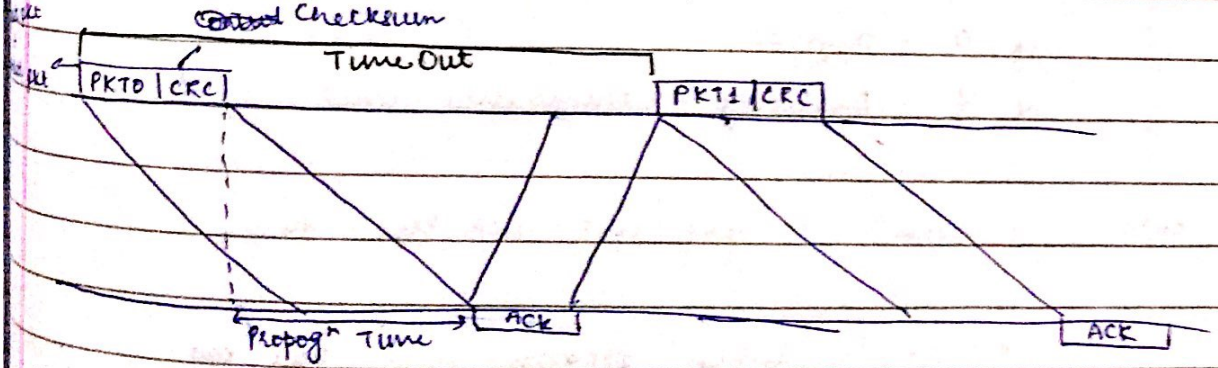
• Here, Flow Control & Reliable transmission are closely binded.

↳ For this, need to maintain 2 things.
• Ack
• Time Out

→ 3 Protocols in ARQs (Flow Control Mechanisms) → need to study same at DLL also.

1) Stop & wait (Binary sync mech) 2) Go back N (X Modem mechanism) 3) Selective Repeat (TCP Mech) → Used in this mechanism

1) Stop & Wait — wait for response from server side.



data pkt + control pkt
↓ ↓
payload header portion

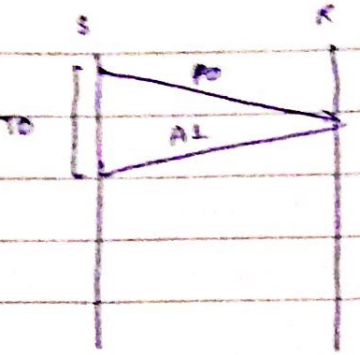
CRC : also of control pkt

Teacher's Signature _____

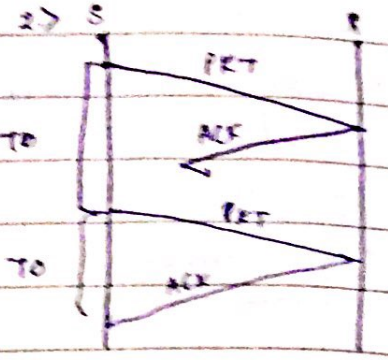
Flow control : also determines whether pkt is corrupted or correct

↓ correct : you get correct ACK else not

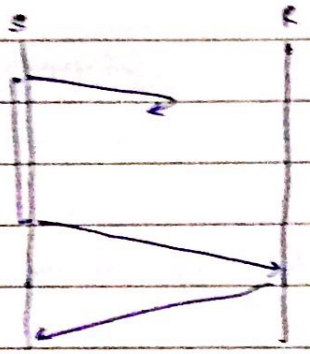
Diff scenarios Possible

1) 

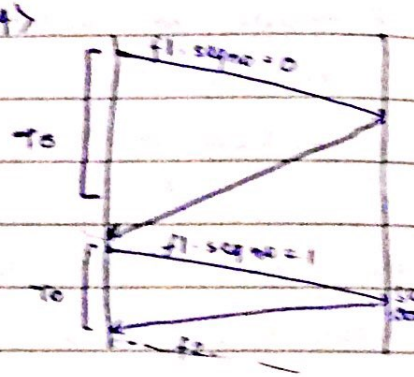
- You forwarded correct pkt.

2) 

- ACK lost, Resend the same pkt & get ack within TO

3) 

- Pkt lost, resend the same pkt & get ack within TO

4) 

→ ACK sent within TO, but received after TO.

Sender must think if he already forwarded ack, why to do it again?

That's why we need these.

→ Use 1 bit sequence no here:

if 0 : original

if 1 : Duplicate (retransmitted one)

3 cond" : pkt lost, ack lost, delay

in case : when resending, send pkt seq no. 1 ⇒ R will send ACK (R will think S has not received ack)

0 ⇒ R will reject pkt

PAGE NO.:
DATE: / /

drawbacks :

① Huge timeouts → very inefficient

Efficiency of stop & wait ;

Transmitted segment size = 1000 bits. Link speed = 1.5 Mbps segment transm^n & recep^h time is 40 ms. a) How many no. of bits can be propogated over this channel. b) Find efficiency of channel

a) 1.5 Mbps Time = 40 ms

40 ms → 1000 bits

1 s → $\frac{1000}{40} \times 1000 = 25000$ bits / sec

b) $\frac{25000 \text{ bits/sec}}{1.5 \times 1024 \times 1024 \text{ b/sec}} = 0.01589$

a.) $40 \times 10^{-3} \times 1.5 \times 10^{6} \Rightarrow$ bits transmitting in 1 sec

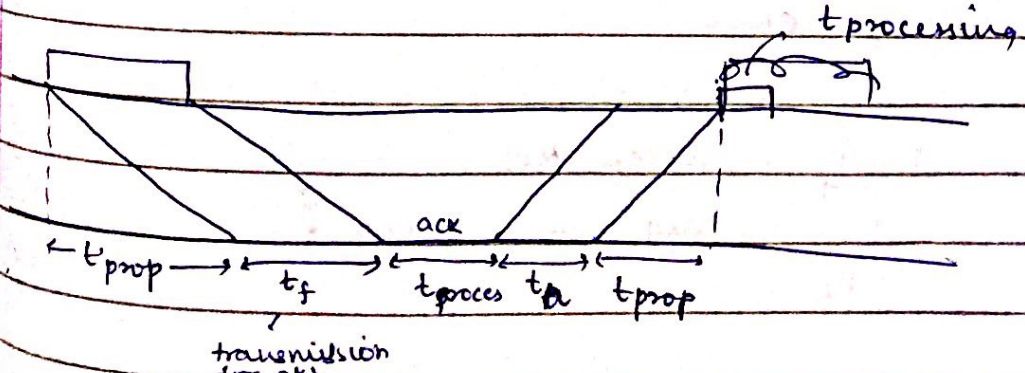
$= 60000$

b) efficiency :

we're transmitting only 1000 bits

$\text{eff} = \boxed{\frac{1000}{60000}}$

for Stop & Wait :



t_0 (delay time for pkt 0) $= 2 \times t_{prop} + t_f + t_a + 2 \times t_{proc}$

Teacher's Signature

Bandwidth delay efficiency

n_f → transmitting this much
R → have this much capacity

no. of bits in f_0

PAGE NO.:
DATE: no. of bits in ack

$$t_0 = 2(t_{prop} + t_{proc}) + \frac{n_f}{R} + \frac{n_a}{R}$$

(t_f, t_a; R → bit rate of the channel)

$$R^{\circ}_{eff} = \frac{\text{No of bits delivered to dest}^n}{\text{Total time req to deliver the info bits}}$$

→ only need data bits (not overhead wala part) → n_0

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

(n_0 : overhead bits)

* Transmission effeciency of : Ratio of R°_{eff} to R.

$$\eta_{sw} = \frac{R^{\circ}_{eff}}{R} = \frac{(n_f - n_0)/t_0}{R} = \left(\frac{n_f - n_0}{R}\right)\left[\frac{1}{2(t_{prop}+t_{proc}) + \frac{n_f}{R} + \frac{n_0}{R}}\right]$$

(η_{sw} (effeciency of sliding window))

$$= \left(\frac{n_f - n_0}{R}\right)\left[\frac{R}{2R(t_{prop}+t_{proc}) + n_f + n_a}\right]$$

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

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

($\frac{n_0}{n_f}$ → loss of transm^n effeciency; denominator → loss of)

↳ We need both η_{sw} & R°_{eff}

↳ Here, we've not considered any error.

How to introduce error ?

P_f ← error rate prob.

$1 - P_f$: success prob.

$\frac{1}{1-P_f}$: avg. success prob.

Teacher's Signature.....

[6.0 MCQs]

PAGE NO.:
DATE: / /

n_f

Q. segment sizes are : 1250 bytes & 25 byte overhead & 25 byte ack R = 1 Mbps $2 \times (t_{prop} + t_{proc})$ = i) 1 ms, ii) 10 ms, iii) 100 ms. Try for R = 1 Gbps also. Find efficiency of stop & wait protocol.

i) R_{eff} = $\frac{n_f - n_o}{2 \times (t_{prop} + t_{proc}) + \frac{n_f}{R} + \frac{n_a}{R}}$

= $\frac{1250 - 25}{1\,ms + \frac{(1250 + 25) \times 8}{1 \times 10^6}}$ = $\frac{1225 \times 8}{1\,ms + 10.2\,ms}$

= $\frac{B75}{} \times 10^3$

η_{sw} = $\frac{1250 - 25}{1250 + 25 + 1\,ms\,(1 \times 10^6)}$ = $\frac{12.25 \times 8}{12.5 \times 8 + 1000}$

= $\frac{9800}{11200}$ = 0.875

ii) 10 ms

R_{eff}

⟹ Considering with errors :

- The effect of Transmission Errors on the efficiency of S & W ARQ

1) If a frame / segment incured errors during transm^n, the time out Mech will cause retransm^n of frame / segment. In that case, t_o is the transmi^n / retransm^n time (sec)

P_f → probability of segment transmitted has errors & it needs retransm^n

Then,

Avg. succesfull Retransm^n = $\frac{1}{1 - P_f}$ (1 in 10 segments)

t_{sw} = $\frac{t_o}{1 - P_f}$ = $t_o\left(\frac{1}{1 - P_f}\right)$ —— ①

time for stop & wait

$$\eta_{sw} = \frac{(n_f - n_o / t_{sw})}{R}$$

Teacher's Signature.....

PAGE NO.
DATE: / /

same values as prev ques^n, for 1 ms only.
$P_f = 10^{-6}, 10^{-5}, 10^{-4}$ → Bit error rate

If $P_f = 10^{-6}$

$$t_{sw} = \frac{t_o}{1-10^{-6}} = 2 \times (t_{prop} + t_{proc}) + n_f/R + n_o/R \left[\frac{1}{1-10^{-6}}\right]$$

To calculate the Transm^n efficiency for S & W ARQ:
We need to calculate an avg. total time req. to deliver correct frame

Let n_t → i transm^n's are req. to deliver correct segment/pkt
→ (i-1) retransm^n's are req. → (i-1) TO are req.

$$P[n_t = i] = (1-P_f) \cdot P_f^{\,i-1} \quad \text{for } i = 1, 2, 3 \ldots$$

↳ i-1 times, need timeout

$$E[t_{sw}] = t_o + \sum_{i=1}^{\infty} (i-1)\, P[n_t = i]\, t_{out}$$

expected t_{sw} time

$$= t_o + \sum_{i=1}^{\infty} (i-1)\, t_{out}\, (1-P_f)\, P_f^{\,i-1}$$

$$= t_o + \frac{t_{out}\, P_f}{1-P_f}$$

If $t_{out} = t_o$:

$$E[t_{sw}] = \frac{t_o}{1-P_f}$$ → that's how we get t_{sw} in ①

Teacher's Signature.....

Concurrent & parallel Computing → Pipeline

PAGE NO.:
DATE: / /

A system approach : CN by Peterson & David

Ques: Bandwidth = 50 Kbps
1 way transmn time = 250 ms. Find out Bandwith delay product (BDP).
Window size = W = 2 (Bandwidth delay) + 1
Find link utilization

delay = 250 ms +

↳ Drawbacks:
- For single pkt, we're waiting
- TO very huge

↓ sol^n

Use pepelining concept

⇓

need not to wait till a task is completed. can be introduced while a task is going on.
Need to choose a buffer which will be transmitted

- Sliding window Protocol : used in GO Back N & Selective Repeat

↓ Problem

Whenever error, whatever successful pkt have been sent needs to be retransmitted

↓

send ack for pkt which is lost.

Teacher's Signature