- We can optimize reccomendations using cookies

Drawbacks: privacy, security

CH:02 (completed)

CH:03 Transport Layer

- Application Layer handovers data to lower layer i.e transport layer.

- Transport layer provides a logical comunication end-to-end connection b/w different application processes running on different hosts.

- Multiplexing → have to send data from several paths/networks/links to a single destination / channel.

- TCP is complex as compared to UDP. The choice b/w them is made according to the requirements.

- Multiplexing /De-multiplexing → same concept used in FDM & TDM.

- address of process identified by port number, address of host identified by IP address.

- Socket have it's own unique port number. Uniqueness of port number is in the machine only. A machine has it's own IP address.

- UDP → User Datagram Protocol.

- Implementation of transport layer is fixed.

Foundation for Advancement of Science & Technology

Lecture - 11

Mid - 1 solution

Q2

A) i) $0.2(1-0.2)^{N-1}$. $(\hat{p}(1-\hat{p})^{N-1})$ transmitting, not transmitting

4) N = 11 users

$^{11}C_3 \times \hat{p}^3 \times (1-\hat{p})^8$ (any 3)

B) 64 kbps (1.536/24)

Lecture NO. 12

TCP sockets identified by:

- source IP address
- destination IP address
- source port number
- destination port number

- UDP is a protocol that provides unreliable data transfer. (connectionless: no handshake)

Checksum

Ex: → add two 16-bit integers

1 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 sender

1 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1

(1) 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 → carry

carry

add

1 0 1 1 1 0 1 1 1 0 1 1 1 1 0 0

1st complement

0 1 0 0 0 1 0 0 0 1 0 0 0 0 1 1

→ checksum

→ Transport layer provides reliability.

1 1 1 0 0 1 1 8 1 1 1 0 0 1 1 0

1 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 receiver

$\boxed{1}$ 1 0 1 1 1 1 1 0 0 0 0 1 1 1 1 0 1 1

1

1 0 1 1 1 1 1 0 0 0 0 0 1 1 1 1 1 0 0

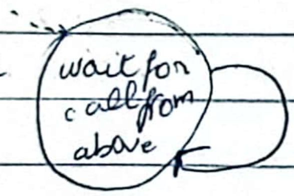
1st complement

0 1 0 0 0 0 0 1 1 1 1 1 0 0 0 0 1 1 → checksum 2

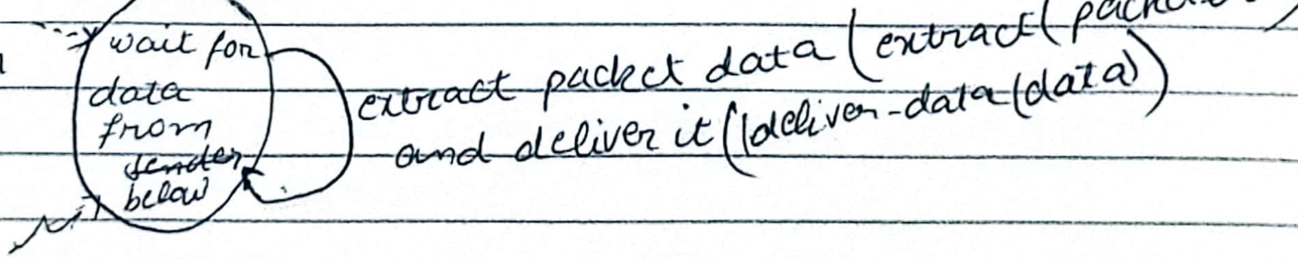
checksum 1 ≠ checksum 2 (indicating an error)

→ rdt 1.0

sender



receiver

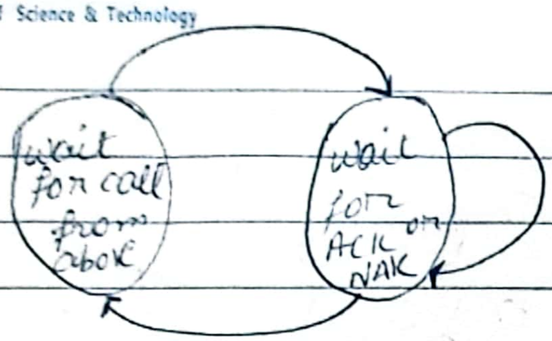


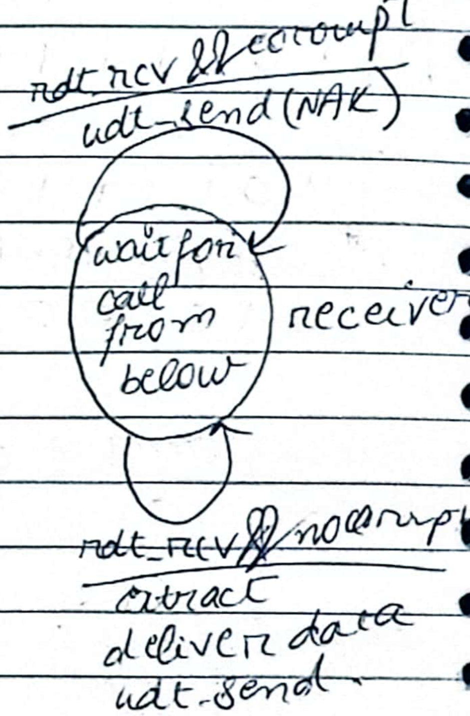
→ rdt 2.0 (channel with bit errors)

• ACKs • NAKs.

rdt 2.0: FSM specification

sender





receiver

Lecture - 13

rdt 3.0 (not a standard protocol)

↓

reliable data transfer, but the resource utilization is low

(stop-and-wait) → only 1 packet

Resource/User utilization = $\frac{L/R}{RTT + L/R}$

$U_{sender} = \frac{0.008}{30.008} =$

→ Pipelining: (more than 1 packet)
(increased utilization)

Ex:

$U_{sender} = \frac{3L/R}{RTT + L/R} = \frac{0.0024}{30.008} = 0.00081$ → 3 packet

pipelining increases utilization by a factor of 3!

- utilization is increased but it also increases the complexity
- Window Size: how many packets can be in a pipeline in 1 minute.
- Sequence Number Range: $0 \rightarrow 2^k - 1$
  ($\bmod 2^k$ arithmetic)
  ↳ simple words → wraparound

→ Go-Back-N has only 1 timer.
↳ if there is packet loss, then there are many overheads.

individual act

• To avoid overheads, follow selective repeat approach

have to maintain a logical timer for every packet

to repeat selective piece/ packet

Ex: $N = 3$, $K = 3 = 2^3 - 1 = 7$ (0-7)

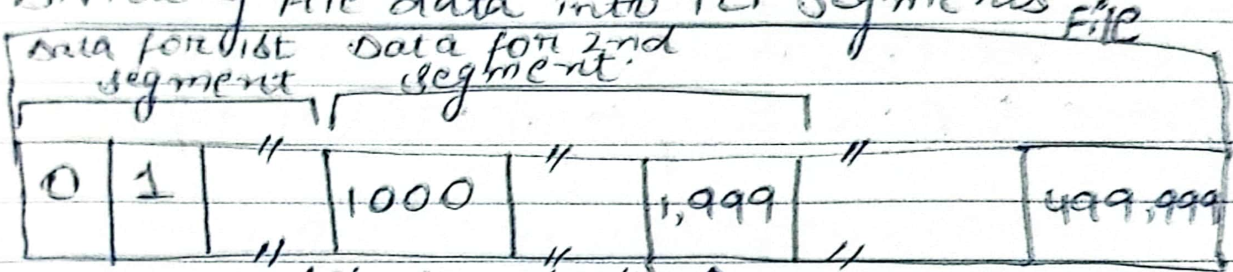
→ TCP is designed following all the above discussed guidelines.

Lecture - 11

→ TCP overview (basic features)

- Message from application layer is transferred to transport layer and a header (UDP/TCP) is appended.
- The IP address is not needed in the header

→ Dividing File data into TCP segments



(Fig from book)

- 500,000 data is divided into segments of data with each segment consisting of 1000 bytes

→ Sequence Number → Number of first byte in the segment's data

→ Acknowledgements → sequence no. of next byte expected from other side.

→ Receiver has an idea of what sequence number/ acknowledge will be sent by the sender.

•) Timeout should > RTT

- Sample RTT: measured time from segment transmission until ACK receipt. is not constant.

•) $\text{Estimated RTT} = (1-\alpha) * \text{EstimatedRTT} + \alpha * \text{Sample RTT}$.

- Exponential weighted Mean Average (EWMA)
- typical value $\alpha = 0.125$
- influence of past sample decreases exponentially fast.

•) $\text{Time Out Interval} = \text{EstimatedRTT} + 4 * \text{DevRTT}$

safety margin

•) $\text{DevRTT} = (1-\beta) * \text{DevRTT} + \beta * |\underbrace{\text{Sample RTT} - \text{Estimated RTT}}|$

initial will be given or assumed.

Absolute error

not constant

$\boxed{\text{typically: } \beta = 0.25}$

- Probabilistic Measures (It is possible that the sender's assumption is not correct).
- Fast retransmission is helpful in many cases.
-

CH: 03

P31

Sample RTT1 = 106 ms
" 2 = 120 ms
" 3 = 140 ms
" 4 = 90 ms
" 5 = 115 ms

$\alpha = 0.125$
Estimated RTT = 100 ms
$\beta = 0.25$
Dev RTT = 5 ms

For sample RTT 1 (106)

$$EstimatedRTT = (1-0.125)\times 100 + \cancel{0.250}\ 0.125\times 106$$
$$= 87.5 + 13.25$$
$$= 100.75$$

$$DevRTT = (1-0.25)\times 5 + 0.25\times |106 - 100.75|$$
$$= 5.0625$$

$$TimeOut\ Interval = 100.75 + 4\times 5.0625$$
$$= 121$$

For Sample RTT 2 (120)

$$EstimatedRTT = (1-0.125)\times 100.75 + 0.125\times 120$$
$$= 103.2$$

$$DevRTT = (1-0.25)\times 5.0625 + 0.25\times |(120 - 103.2)|$$
$$= 7.99 \approx 8$$

$$TimeOut\ Interval = 103.2 + 4\times 7.99$$
$$= 135.16$$

(Same process for next samples)