1. [10] (a) Calculate the total time required to transfer a 1000 KB file on a 1.5 Mbps link with an RTT of 50 ms, a packet size of 1 KB, and an initial 100 ms of handshaking delay before the data is sent. Repeat the computation for the case when there is a waiting time of one RTT after each packet is sent.

Solution:

a **Part1.**

Initial Handshaking = 2 * R.T.T. = 2*50 ms = 100 ms

we can send continuously :

to send 1000KB file, & BandWidth is 1.5 Mbps..

So, time to transmit is 1000KB / 1.5 Mbps = 5461.33 ms

propagation time = 1/2RTT = 25ms

Total Time = **T.T.** + **Initial Handshaking time + propagation time**

= 5461.33 + 100 + 25 = **5586.33 ms**

Part2.

` Initial Handshaking = 2 * R.T.T. = 2*50 ms = 100 ms

time to send 1 packet = 1KB / 1.5 Mbps

Inter packet gap = 1 R.T.T. = 50 ms

Total Time = **Initial Handshaking time + 1000 packets T.T. + 999 * R.T.T. (waiting time) + propagation time**

= 100 + 1000(1KB / 1.5 Mbps) + 999(50)+25

= **55.536 sec**

(b) Compare the delay in sending a message of length X bits over a path of length k in both a circuit switched network and a lightly loaded packet switched network. Assume that the circuit setup time is t sec, the propagation delay per hop is d sec, the packet size is P bits, and the data rate is B bps. What is the condition to be satisfied for the packet switched network to have a lower transmission delay?

Solution:

Ans.1 (b) Given:- x = msg length in bits
t = ~~is~~ the circuit setup time in sec
d = Propagation delay per hop in sec.
K = path length in hops.
P = Packet size in bits
B = data rate in bits per second.

Over ~~su~~ circuit switched N/w -

Total delay = setup time (t) + total propagation delay (kd) + transmission time (x/B)

$$= t + Kd + x/B \quad \text{--- (1)}$$

Over packet-switched N/w-

Total delay = Propagation delay ($\cancel{k}d$) + Transmission delay for one pkt (P/B) + Queueing delay (if given).

$$= Kd + \frac{x}{B} + (K-1)\frac{P}{B} \quad \text{--- (11)}$$

Condition for packet switched n/w to have lower transmission delay —

Packet switched delay < circuit switched delay

$$\cancel{Kd} + \cancel{\frac{x}{B}} + (K-1)\frac{P}{B} < t + \cancel{Kd} + \cancel{\frac{x}{B}}$$

~~Simply~~ simplify above eq^n

$$\boxed{(K-1)\frac{P}{B} < t}$$

∴ When $(K-1)\frac{P}{B} < t$, packet switched n/w have a lower transmission delay.

2. (a) Consider an error-free 64 kbps satellite channel with a propagation delay of 270 ms being used to send 512-byte data frames in one direction, with very short ACKs in the other direction.

What is the maximum throughput when Go-Back-N protocol is used with window sizes of 1, 7, 15, and 127?

(b) How many bits do you need to represent the sequence numbers for this link (0 is a valid sequence number)?

(c) Using the maximum sequence number space you just computed above, what is the maximum throughput that can be achieved with the Selective Repeat protocol operating on the same link?

Answer)

(a)

Propagation delay: 270ms

Round-Trip Time: 2*270ms = 540ms

Transmission delay for 1 Frame: 512 Byte / (64/8) KBps = 64ms

Total time for 1-frame: 540ms + 64ms = 604ms

At t=604 msec, the ACK arrives at the earth.

The data rate here is 4096 bits in 604 msec, or about 6781 bps.

With a window size of 7 frames, transmission time is 448 msec for the full window, at which time the sender has to stop.

At 604 msec, the first ACK arrives and the cycle can start again.

Here we have 7 ×4096 =28,672 bits in 604 msec. The data rate is 47,470.2 bps.

Continuous transmission can only occur if the transmitter is still sending when the first ACK gets back at t=604 msec.

In other words, if the window size is greater than 604 msec worth of transmission, it can run at full speed.

For a window size of 10 or greater this condition is met, so for any window size of 10 or greater (e.g., 15 or 127) the data rate is 64 kbps

(b) With max. window size of 10 which gives maximum throughput, only 4-bits are required.

Use max window size of part a to calculate bits. 4 bits would be enough for window size of 10.

(c)

2)

c) Sequence bits $= 4$

Max no. of seq nos :- $2^4 = 16$

for Selective Repeat,

No. of Sequence Nos $\geq$ 2. Window Size

$16 \geq 2 \cdot W$

$W \leq 8$

$\therefore$ Total RTT for 1 frame :- $2 * 270 + 64$ ms
(Propagation Time) (Transmission Time)

$= 604$ ms.

After sending 8 frames, it receives ACK.
and after that it sends the next one,

$\therefore$ Max Data Rate.

$(8 \times 512 \times 8)$ bits in 604 ms.

$\therefore$ Throughput is $\frac{2^{15}}{604} \times 10^3$ bps.

$= 54251.655$ bps.

4. [8] (a) An ISP is allocated a block of consecutive IP address 198.16.0.0/16. Suppose that four organizations, A, B, C, and D, request 4000, 2000, 4000, and 8000 addresses, respectively, and in that order. For each of these organizations, give the IP address blocks assigned, with the first and last IP address and the prefix in the CIDR notation.

- **Ans :** Organization A requests 4000 addresses. The largest power of 2 that's less than or equal to 4000 is 4096 (2^12). So, we need a /20 prefix (32 - 12 = 20 bits for the network portion).
    - First IP address: 198.16.0.0
    - Last IP address: 198.16.15.255
    - CIDR notation: 198.16.0.0/20
    - Address range: 198.16.0.0 - 198.16.15.255 (4096 addresses)
- Organization B requests 2000 addresses. The largest power of 2 that's less than or equal to 2000 is 2048 (2^11). So, we need a /21 prefix (32 - 11 = 21 bits for the network portion). Since the previous block of 4096 addresses was assigned to A, we need to continue from where A's block ends. The next available address is 198.16.16.0.
    - First IP address: 198.16.16.0
    - Last IP address: 198.16.23.255
    - CIDR notation: 198.16.16.0/21
    - Address range: 198.16.16.0 - 198.16.23.255 (2048 addresses)
- Organization C requests 4000 addresses. The largest power of 2 that's less than or equal to 4000 is 4096 (2^12). Since the previous block of 2048 addresses was assigned to B, we need to continue from where B's block ends. The next available address is 198.16.24.0.
    - First IP address: 198.16.24.0
    - Last IP address: 198.16.31.255
    - CIDR notation: 198.16.24.0/20
    - Address range: 198.16.24.0 - 198.16.31.255 (4096 addresses)
- Organization D requests 8000 addresses. The largest power of 2 that's less than or equal to 8000 is 8192 (2^13). Since the previous block of 4096 addresses was assigned to C, we need to continue from where C's block ends. The next available address is 198.16.32.0.
    - First IP address: 198.16.32.0
    - Last IP address: 198.16.63.255
    - CIDR notation: 198.16.32.0/19
    - Address range: 198.16.32.0 - 198.16.63.255 (8192 addresses)

(b) Suppose that a university is using addresses in the range 198.16.40.0- 198.16.63.255. How would a router in the provider ISP aggregate the addresses and announce them in CIDR notation without jeopardizing the traffic to other organizations?

Answer :

By aggregating the university's IP addresses into a single CIDR block (198.16.40.0/21), the ISP can efficiently announce this block without impacting traffic to other organizations.

198.16.40.0/21: This covers the range from 198.16.40.0 to 198.16.47.255.

198.16.48.0/20: This covers the range from 198.16.48.0 to 198.16.63.255.

5. [10] (a) Consider sending a 3000-byte IP datagram on a path of two links; first link has an MTU size of 1500 bytes, followed by the second link with an MTU size of 500 bytes. Show the IP datagram fragments received at the destination (after the second link), with the header fields relevant for reassembly. Suppose the original datagram is stamped with an identification number of 100. Assume 20-byte IP header without options.

Solution:

Before entering the first network, we have an IP datagram of 2980 (payload) + 20 (IP header) = 3000 bytes.

In the first network, packets have a room for 1500 – 20 = 1480 bytes of data. Therefore, the datagram needs to split as follows:

1st fragment: IP header of 20 bytes and data of 1480 bytes. Offset is 0.

2nd fragment: IP header of 20 bytes and data of 1480 bytes. Offset is 1480/8 = 185.

3rd fragment: IP header of 20 bytes and data of (2980 - (2 ×148)) = 20 bytes.

Offset is 2960/8 = 370.

When 1st fragment goes through the second network, it needs to split into four fragments. Each fragment can contain at most 500-20 = 480 bytes of data. So, we have the following fragments:

4th fragment: IP header of 20 bytes and data of 480 bytes. Offset is 0.

5th fragment: IP header of 20 bytes and data of 480 bytes. Offset is 480/8 = 60.

6th fragment: IP header of 20 bytes and data of 480 bytes. Offset is 960/8 = 120.

7th fragment: IP header of 20 bytes and data of (1480-(3 × 480)) = 40 bytes.

Offset is 1440/8 = 180.

Similarly, when 2nd fragment goes through the second network, it needs to be split into four fragments.

8th fragment: IP header of 20 bytes and data of 480 bytes. Offset is 185.

9th fragment: IP header of 20 bytes and data of 480 bytes. Offset is 185+60 = 245.

10th fragment: IP header of 20 bytes and data of 480 bytes. Offset is 185+120 = 305.

11th fragment: IP header of 20 bytes and data of (1480-(3 × 480)) = 40 bytes.

Offset is 185+180 = 365.

Finally, the 3rd fragment goes as it is through the second network since it is of 40 bytes.

The fragments that reach the destination are 4, 5, 6, 7, 8, 9, 10, 11, 3. They are assembled in this order because their offsets are in increasing order (0, 60, 120, 180, 185, 245, 305, 365, 370). Also note that 4th to 11th fragments will have their more fragment bits set (i.e., MF = 1) while 3rd fragment will not (i.e., MF = 0), indicating it is the last fragment of the original datagram.

(b) For the network shown below, suppose the forwarding tables are all established with distance vector routing and then link C−E fails. Give the distance vector tables at:

(i) Nodes A,B,D,F after C and E have reported the news

(ii) Nodes A and D after their next mutual exchange and

(iii) Node C after A sends its update.

Solution: Use format Node, distance, next hop

(Removed forwarding tables obtained using distance vector routing)

(i) Nodes A,B,D,F after C and E have reported the news

A:

Node	**Distance**	**Next hop**
B	∞	-
C	3	C
D	∞	-
E	∞	-
F	9	C

B:

Node	**Distance**	**Next hop**
A	∞	-
C	∞	-
D	4	E
E	2	E
F	∞	-

D:

Node	**Distance**	**Next hop**
A	∞	-
B	4	E
C	∞	-
E	2	E
F	∞	-

F:

Node	**Distance**	**Next hop**
A	9	C
B	∞	-
C	6	C
D	∞	-
E	∞	-

(ii) Nodes A and D after their next mutual exchange

A:

Node	**Distance**	**Next hop**
B	12	D
C	3	C
D	8	D
E	10	D
F	9	C

D:

Node	**Distance**	**Next hop**
A	8	A
B	4	E

C	11	A
E	2	E
F	17	A

(iii) Node C after A sends its update

C:

Node	**Distance**	**Next hop**
A	3	A
B	15	A
D	11	A
E	13	A
F	6	F

3. [12](a) What is the fastest rate at which data can be sent out of a link layer interface when the application uses TCP without having the 32-bit sequence numbers wrap around? Assume that link layer frames can be sent continuously over the wire and that the MTU size of the interface is 1540 bytes. Assume the TCP+IP header to be 40 bytes, link layer header to be 26 bytes, and the maximum segment lifetime to be 120 seconds.

TCP uses a 32-bit sequence number field in its header. When the sequence numbers wrap around, it will take (2^32) - 1 sequence numbers to wrap from the highest value back to zero.

- To calculate the maximum rate at which TCP sequence numbers can be generated without wrapping, we need to divide the total number of possible sequence numbers by the maximum segment lifetime:
- Total number of sequence numbers = 2^32
- Maximum segment lifetime = 120 seconds.
- So, the maximum rate without wraparound is:
- (2^32) / 120 seconds= 35791394 bytes/sec
- MTU size = 1540 bytes
- TCP+IP header size = 40 bytes , Link layer header size = 26 bytes
- So, the maximum rate at which data can be sent out of the link layer interface is:
- (35,791394 bytes/second) / (MTU size - Total header size) (35,791394) / (1500 bytes - 66 bytes) (35,791394) / (1474 bytes)
- ≈ 23860 frames per second for each 1500 bytes
- The maximum rate now becomes =(23860 * 1566 * 8) bits/sec = 299 Mbps

b.) How many bits would be required to carry the advertised window field in TCP segment when operating over a link with 1Gbps capacity and 100ms RTT? Based on the line rate computed in case (a) and the advertised window size computed in this part, compute the utilization of the link with only one TCP flow

Solution:

The formula for calculating the Bandwidth Delay Product is:

BDP (in bits) = Link Capacity (in bits per second) * Round-Trip Time (in seconds)
Given that the link capacity is 1 Gbps (1 gigabit per second) and the round-trip time (RTT) is 100 ms (0.1 seconds), we can calculate the BDP:
BDP = 1,000,000,000 bits per second * 0.1 seconds = 100,000,000 bits
Now, let's consider the advertised window size. To fully utilize the network, the sender should not send more data than the BDP. Therefore, the advertised window size should be at least equal to the BDP.
In this case, the BDP is 100,000,000 bits, which can be represented using 27 bits (2^27 = 134,217,728).
Therefore, at least 27 bits in the advertised window field to fully utilize the link's capacity.

Utilization= 299 Mbps/ 1 Gbps = 30%.

C)Consider a TCP Reno connection running over a path with a maximum segment size of 1000 bytes and the receiver advertised window of 64,000 bytes. Assume that the average round trip time is almost constant at 100 ms and there are no losses or additional delays during the data transfer. How long does it take to transfer a file of 2,000,000 bytes over this connection? Of this total transfer time, how long does TCP stay in the slow start phase? What is the average throughput of the TCP connection? Compare the reduction in the link utilization due to TCP congestion control algorithm by comparing the throughput computed in part (a) and (c).

Solution:

Q3

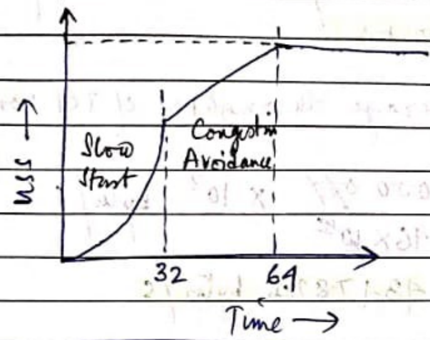
c) Given,

Receiver Advertised Window Size = 64000 bytes

MSS = 1000 bytes

Max. Receiver Window Size (in MSS) = $\frac{64000}{1000} = 64$

Slow Start Threshold = $\frac{\text{Window Size}}{2} = 32$



$\boxed{\text{Slow start Phase}}$

No. of RTT's to reach 32 MSS (Threshold)

= 6 $(1 \to 2 \to 4 \to 8 \to 16 \to 32)$

Total Time = 6×100 ms = 600 ms

No. of bytes Transferred = 63×1000 = 63K bytes

$\boxed{\text{Congestion Avoidance Phase}}$

No. of RTT's to reach 64 MSS = 32

$(33 \to 34 \to 35 - - - 63 \to 64)$

Total Time = 32×100 = 3200 m/s

No. of bytes transferred = $\left(33 \times 32 + \frac{31 \times 32}{2}\right) \times 1000$ bytes

= 1552×1000 bytes

Total bytes Transferred = $(1552 + 63)$ = 1615×1000 bytes

Total bytes left = $(2000 - 1615) \times 1000$ = 385×1000 bytes

No. of additional RTT's = $\frac{385}{64} = 6.0156 \approx 7$

Time Taken = 7×100 m/s = 700 ms.

Answers

→ How long does it take to transfer a file

$= (600 + 3200 + 700) = 4500$ ms

$+ 100$ ms (For connection Establishment)

$=$ 4600 ms

→ How long does TCP stay in slow start phase?

$=$ 600 ms

→ What is the average throughput of TCP connection?

$$\frac{2000\,\cancel{00}}{46 \times \cancel{10^2}} \times 10^3 \text{ bytes/s}$$

$=$ 434782.6 bytes/s