Date:

NAME: ARUN (19K-1049)
COURSE: COMPUTER NETWORKS
SECTION: BSSE-5B

Chapter 3.4

Question 1 (a)

Following are the reasons due to which application designers use UDP over TCP:

(1) UDP is connectionless, therefore, no delay due to RTT.
(2) More clients can be serviced on UDP as compared to TCP as no congestion control is performed on UDP.
(3) Some applications donot require reliable data transfer. Therefore, developers use UDP in this case as it is simple to implement.

Question 1 (b)

It is possible to have reliable data transfer on UDP. This could be achieved at application layer. One way of implementing is checksum. Though, checksum is a weak protection technique but implementing it reduces the threat factor on UDP. Checksum is used with acknowledgement and retransmission.

Sanda

Date:

Question 2 (a)

Voice and video applications use TCP over UDP due to the following reasons.

(1) TCP is reliable than UDP

(2) No delay due to packet loss.

(3) ~~Its~~ Most firewalls support TCP, therefore no data is blocked when transfered using TCP.

Question 2 (b)

This situation cannot be handled on UDP as there is no congestion control algorithm applied on it. However, TCP can do it by using any of the congestion control algorithms; one such way is halving the transmission rate when a packet loss is detected. Other way of doing this, regardless of TCP and UDP, is applying congestion control at application layer.

Question 3 (a)

Yes, both the segments would be directed to the same socket at host C as it is a UDP connection. Host C will differentiate between these through the IP addresses with the sockets received as IP is assigned by the operating system.

Date:

Question 3(b)

Yes, host C will service requests from host A and B on different sockets as it is a persistent (TCP) connection. Both the sockets would have same port no. as they are on same host.

Question 4(a)

Sequence number is used by the receiver as it identifies/determines retransmitted data/packets, supports re-ordering, and provides information about dropped packets.

Question 4(b)

Timers were introduced to detect lost packets; if the ACK for a transmitted packet is not received within the set-time, the packet is retransmitted.

Question 4(c)

A timer would still be required to detect packet loss. Knowing the round trip time will help in detecting whether a packet is lost or its acknowledgement.

Question 5(a)

Data = 110 - 90
= 20

Question 5(b)

Host B will send an ACK for sequence number 90.

Sanda

Date:

Question 6 (a)

The sender reduces its transmission speed when a packet loss is detected, or when it reaches the maximum link capacity or when the link is congested.

Question 6 (b)

Sequence number is used to identify duplicate/retransmitted packets. Here, rdt3.0 does this by identifying duplicate acknowledgements. Therefore, a sequence number is not required here.

Question 7(a)

True, a sender can receive acknowledgement for a packet that falls outside its current window.

Example:

at t0:
sender: send Pkt1
send Pkt2
send Pkt3

at t1:
Receiver Ack1
Ack2
Ack3

at t2:
sender(timeout): send Pkt1
send Pkt2
send Pkt3

at t3:
Receiver Ack1
Ack2
Ack3

at t4:
sender receives acknowledgements sent at t1 and send Pkt4, Pkt5, Pkt6.

at t5:
sender receives acknowledgements sent at t3. These are outside its window.

Date:

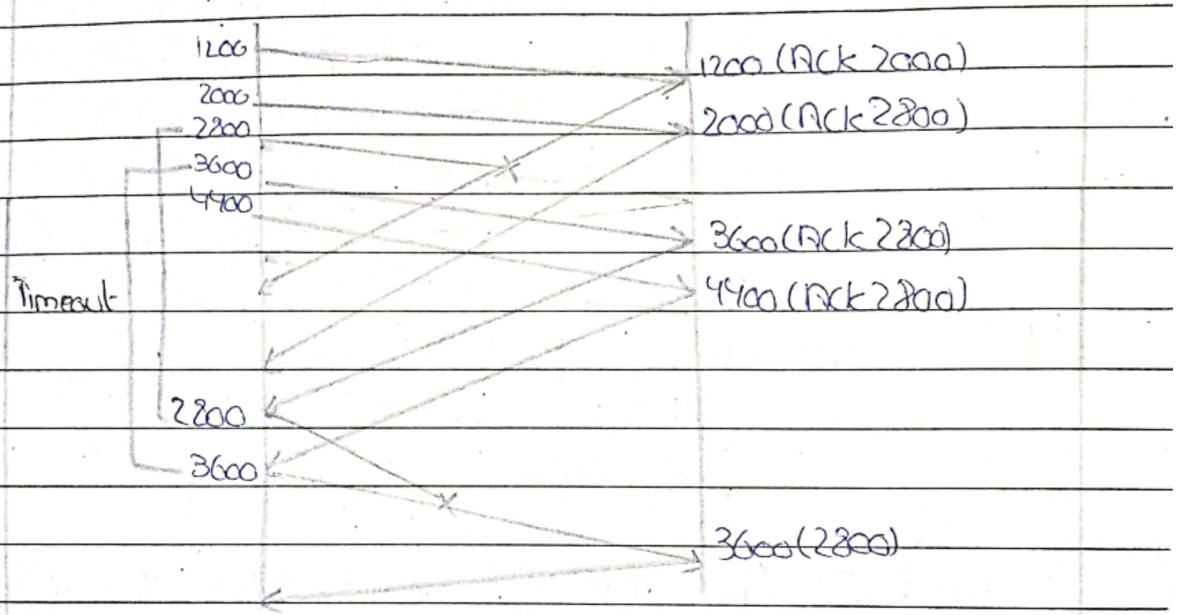
Question 7(b)

True, it is possible for the sender to ~~cache~~ receive an ACK for a packet that falls outside its current window.

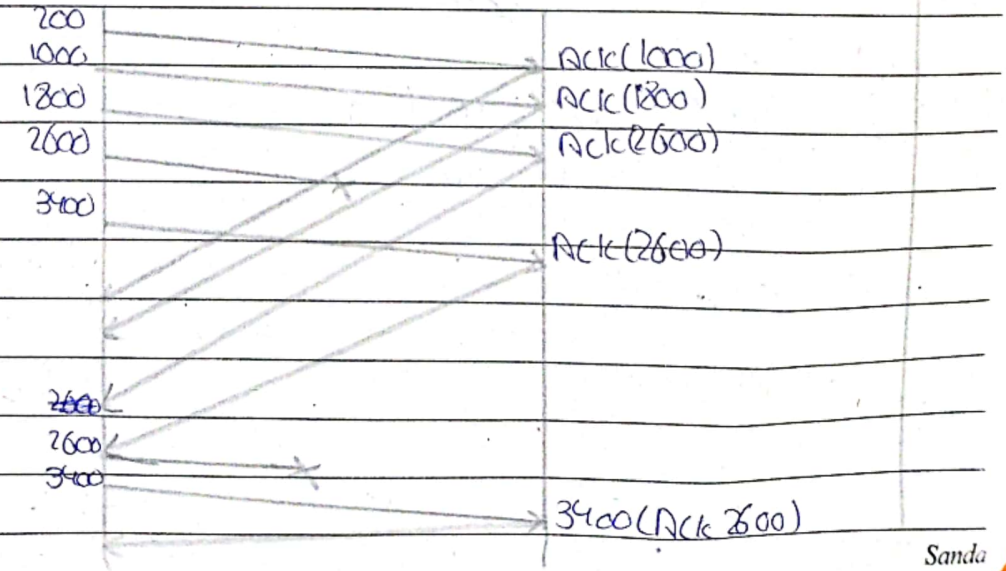
Example.

Same as in (a)

Question 8(a)



Question 8(b)



Sanda

Date:

Question 9(a)

Solution

EstimatedRTT: (1-0.125)*15 + (0.125)*20
= 15.625 ms

DevRTT: (1-0.25)*1.9 + 0.25*(20-15.625)
= 1.425 + 1.094
= 2.52 ms

RTO = 15.625 + (4*2.52)
= 25.7 ≈ 26 ms

Question 9(b)

Solution

EstimatedRTT: (1-0.125)*25 + (0.125)*30
= 21.875 + 3.75
= 25.625 ms

DevRTT: (1-0.25)*2.8 + (0.25)*(30-25.625)
= 2.1 + 1.094
= 3.2 ms

RTO: 25.625 + (4*3.2)
= 38.4 ≈ 38 ms

Pg No.

Sandal

Date:

Question 1a

(a) Between 1-6 and 23-26
(b) Between 6-16 and 17-22
(c) Triple duplicate ACK (Window size doesn't drop)
(d) Timeout (window size is dropped to 1)
(e) 32
(f) 21 (42 ÷ 2)
(g) 13 (26 ÷ 2)
(h) 7th transmission round
(i) The threshold and window size would be 4 (8÷2)

Question 11

Problem 1.

no. of needed subnets: 14
no. of needed usable hosts: 14
Network Address: 192.10.10.0

Address Class: C
Default subnet mask: 255.255.255.0
Custom subnet mask: 255.255.255.240
(128+64+32+16=240)
Total no. of subnets: $2^4 = 16$
Total no. of host address: $2^4 = 16$
No. of usable address: 16 - 2 = 14
(1 for network,
1 for broadcasting)
No. of bits borrowed: 4

Date:

Problem 2:

No. of needed subnets: 1000
no. of needed usable hosts: 60
Network Address = 165.100.0.0

Address class: B
Default subnet mask 255.255.0.0
Custom subnet mask 255.255.255.192
(128+64+32+16+8+4+2+1) (128+64)
Total number of subnets $2^{10} = 1024$
Total number of host addresses: $2^6 = 64$
Number of usable addresses: $64-2=62$
Number of bit borrowed: 10

Problem 3:

Network Address: 148.75.0.0/26

Address Class: B
Default subnet mask: 255.255.0.0
Custom subnet mask: 255.255.255.192
$26-16=10 = [128+64+32]+[16+8+4+2+1]$ $[128+64]$
Total no. of subnets: $2^{10} = 1024$
Total no. of host addresses: $2^6 = 64$
Total no. of usable addresses: $64-2=62$
No. of bits borrowed: 10

Date:

Question 12

IP pool: 192.168.1.0/24 (255.255.255.0)

(1) Netcom:

Host required: 50
Host address: $2^6 = 64$
Bits available for subnetting: 2
subnets: $2^2 = 4$

range:
0-63: 192.168.1.1 – 192.168.1.63
64-127 = (192.168.1.65 – 192.168.1.127)
128-191 = (192.168.1.129 – 192.168.1.191)
192-255 (192.168.1.193 – 192.168.1.254)

(2) Cyber Safe:

Host required: 48
Host address: $2^6 = 64$
Bits available for subnetting: 2
subnets: $2^2 = 4$

~~0-63 = (192.168.1.1 – 192.168.1.63)~~

range:
64-127 = 192.168.1.65 – 192.168.1.127
128-191 (192.168.1.129 – 192.168.1.191)
192-255 (192.168.1.193 – 192.168.1.254)

(3) CNSP-Zone:

Host required: 120
Host address: $2^7 = 128$
Bits available for subnetting: 1
Subnets: $2^1 = 2$

Range:
0-127 (192.168.1.1 – 192.168.1.127)
128-255 (192.168.1.129-192.168.1.254)

Sandal

Date:

Question 13

	Total length	Flag	Fragment's offset
Original Packet	1000+20	1	0
Frag#1	1000+20	1	1000/8
Frag#2	1000+20	1	2000/8
Frag#3	1000+20	1	3000/8
Frag#4	1000+20	1	4000/8
Frag#5	960	0	5000/8