COL334 Major Exam

Anish Banerjee

TOTAL POINTS

46.5 / 54

QUESTION 1

Multi-path router 10 pts

1.1 Structure of look-up table & forwarding operations 3 / 3

✓ **+ 1 pts** *Multiple next hops (Look-up table)*

✓ **+ 1 pts** *Weight for each option (Look-up table)*

✓ **+ 1 pts** *WRR over the next hop option*

+ 0 pts Unattempted/Incorrect

1.2 Weighted round-robin algorithm 1.5 / 2

✓ **+ 1 pts** *problem of hysteresis - unstable constantly changing network*

+ 1 pts solution - change only if current state is very different from previous state

+ 0 pts Unattempted/Incorrect

+ 0.5 *Point adjustment*

1.3 Suitability of TCP 1 / 1

✓ **+ 1 pts** *Yes, TCP handle packet reordering and loss*

+ 0 pts Unattempted/Incorrect/Missing reasoning

1.4 TCP performance: problems & consequences 1 / 2

+ 1 pts Triple dup ack parameter may need change

✓ **+ 1 pts** *Problems - spurious congestion detection, lower throughput*

+ 0 pts Unattempted/Incorrect/Missing reasoning

1.5 Comparison with packet trimming 1 / 2

✓ **+ 1 pts** *Packet trimming as solution with correct reasoning*

+ 1 pts Congestion window changes in response to the losses

+ 0 pts Unattempted/Incorrect/Missing reasoning

QUESTION 2

2 TCP throughput during congestion avoidance phase 5 / 5

✓ **+ 0.5 pts** *Steady-state congestion window (c->2c)*

✓ **+ 1 pts** *Related congestion window with N-RTTs*

✓ **+ 1 pts** *Solved for total data sent*

✓ **+ 1 pts** *Loss rate calculated*

✓ **+ 0.5 pts** *Throughput function correctly specified*

✓ **+ 1 pts** *Correct formula computed*

+ 0 pts Unattempted/Incorrect

QUESTION 3

Traceable encrypted messaging app 6 pts

3.1 Ensuring integrity of message delivery 1 / 2

✓ **+ 1 pts** *...B can check integrity of m*

+ 1 pts ...B can check message was signed by A

+ 0 pts Unattempted/Incorrect

This allows B to know the message was signed by the actor with public key Ka+. The certificate KaC is what allows B to validate that the identity of the actor with public key Ka+ is A.

3.2 Source traceability 1.5 / 2

+ 1 pts ...B encrypts further so that only C can read.

+ 1 pts ...forwards original signature for C to be able to verify integrity and original source

+ 0 pts Unattempted/Incorrect

+ 1.5 *Point adjustment*

You should have also encrypted it the message with KC+ so that only C could read it.

3.3 Full traceability 2 / 2

✓ **+ 1 pts** *Correct encryption*

✓ **+ 1 pts** *Correct signature*

+ 0 pts Unattempted/Incorrect

It should be K-B(K-A(H(m))) instead of K-B(H(m)), because that allows C to validate that B signed the message *after* A signed it (yours just allows C to validate that both A and B signed it, not the sequence).

QUESTION 4

Network topology 6 pts

4.1 Header information at source 3 / 3

✓ **+ 1 pts** *Correct LL Header:*

MAC_s

MAC_R1

✓ **+ 1 pts** *Correct IP Header:*

IP_s

IP_D

✓ **+ 1 pts** *Correct TCP Header:*

Random

80

+ 0 pts Unattempted/Incorrect

4.2 Header information at destination 1 / 1

✓ **+ 1 pts** *Correct LL Header:*

MAC_R2

MAC_D

- 1 pts Incorrect IP Header or TCP Header (anyone)

Both should be same as 4.1

+ 0 pts Unattempted/Incorrect

4.3 Gateway and network mask at source & destination 2 / 2

✓ **+ 0.5 pts** *Source Gateway: IP_R1 with explanation*

✓ **+ 0.5 pts** *Source Network mask: 102.100.10.0 with explanation*

✓ **+ 0.5 pts** *Destination Gateway: IP_R2*

✓ **+ 0.5 pts** *Destination Network mask: 219.62.0.0*

+ 0 pts Unattempted/Incorrect

QUESTION 5

4B/5B encoding 3 pts

5.1 Encode the given sequence 1 / 1

✓ **+ 1 pts** *10100 01111*

+ 0 pts Unattempted/Incorrect

5.2 Find a sequence with maximum zeroes 1 / 1

✓ **+ 0.5 pts** *code that ends in max zeroes (10100 -> 0010 or similar)*

✓ **+ 0.5 pts** *code that starts with max zeroes (01001 -> 0001 or similar)*

+ 0 pts Unattempted/Incorrect

5.3 Reason of using 4B/5B with NRZI 1 / 1

✓ **+ 0.5 pts** *NRZI ensures a transition for clock recovery at each 1*

✓ **+ 0.5 pts** *4B/5B guarantees to have at most 3 consecutive zeroes*

+ 0 pts Unattempted/Incorrect

QUESTION 6

Router configuration 6 pts

6.1 Interface addresses and count 5 / 5

✓ **+ 1 pts** *Interface 0: 10000--- => 8 addresses*

✓ **+ 1 pts** *Interface 0: 1001---- => 16 addresses*

✓ **+ 1 pts** *Interface 1: 10001--- => 8 addresses Interface 2: 101----- => 32 addresses*

✓ **+ 1 pts** *Interface 3: 1101---- => 16 addresses Interface 3: 111----- => 32 addresses*

✓ **+ 1 pts** *Interface 3: 0------- => 128 addresses*

+ 0 pts Unattempted/Incorrect

6.2 Prefix from unallocated block 1 / 1

✓ **+ 1 pts** *1100xx-- can accommodate 4 addresses*

+ 0 pts Unattempted/Incorrect

QUESTION 7

Short answer questions 18 pts

7.1 Communication efficiency improvement in persistent HTTP 1 / 1

✓ **+ 1 pts** *Avoids 1.5 RTT latency of new TCP connection*

+ 0.5 pts Correct idea but not full explanation

+ 0 pts Unattempted/Incorrect

7.2 Congestion control problem in UDP 1 / 1

✓ **+ 1 pts** *Can starve out TCP traffic*

+ 0 pts Unattempted/Incorrect

+ 0.5 pts Partial marks

7.3 Reverse path forwarding... T/F 0 / 1

+ 1 pts True

✓ **+ 0 pts** *Unattempted/Incorrect*

7.4 Intserve design for QoS... T/F 1 / 1

✓ **+ 1 pts** *True*

+ 0 pts Unattempted/incorrect

7.5 Traffic shaping filter... 1 / 1

✓ **+ 1 pts** *Leaky bucket*

+ 0 pts Unattempted/Incorrect

7.6 Define jitter 0.5 / 1

+ 1 pts Variability in latency, variance or std dev of RTT. / Queueing delay

+ 0 pts Unattempted/Incorrect

✓ **+ 0.5 pts** *Partial*

7.7 Routers for multicase addresses...T/F **1 / 1**

✓ **+ 1 pts** *True*

+ 0 pts Unattempted/Incorrect

7.8 Inter-packet v/s intra-packet FEC **1 / 2**

✓ **+ 1 pts** *Inter-packet (packet-loss recovery) Intra-packet (bit-corruption recovery)*

+ 1 pts Inter-packet (useful for congestion losses)
Intra-packet (useful for wireless links and such networks where bit corruption is possible)

✓ **+ 0 pts** *Unattempted/Incorrect*

7.9 Error checking for intra-packet FEC **1 / 1**

✓ **+ 1 pts** *CRC, Checksum, Parity (Anyone)*

+ 0 pts Unattempted/Incorrect

7.10 Type of DNS record for mail server **1 / 1**

✓ **+ 1 pts** *MX Record*

+ 0 pts Unattempted/Incorrect

7.11 Poison reverse ... T/F **1 / 1**

✓ **+ 1 pts** *True*

+ 0 pts Unattempted/Incorrect

7.12 Solution for head of line blocking...T/F **1 / 1**

✓ **+ 1 pts** *True*

+ 0 pts Unattempted/Incorrect

7.13 Cookies maintained by OS...T/F **1 / 1**

✓ **+ 1 pts** *False*

+ 0 pts Unattempted/Incorrect

7.14 Implementation of ad blocking by using HTTPS **0 / 1**

+ 1 pts Yes for ads from specific IP, No for ads served from content provider IP

✓ **+ 0 pts** *Unattempted/Incorrect*

7.15 Reason for bufferbloat **1 / 1**

✓ **+ 1 pts** *(2) Very large common buffer*

+ 0 pts Unattempted/Incorrect

7.16 How to run a server behind NAT? **1 / 1**

✓ **+ 1 pts** *Part forwarding (static mapping on a NAT)*

+ 0 pts Unattempted/Incorrect

7.17 Special set of IP addresses used by IP multicast...T/F **1 / 1**

✓ **+ 1 pts** *True*

+ 0 pts Unattempted/Incorrect

COL334/672: Semester 2023-24-1

Major exam: 120 minutes, closed-book.

Name: ANISH BANERJEE

Entry number: 2021CS10134

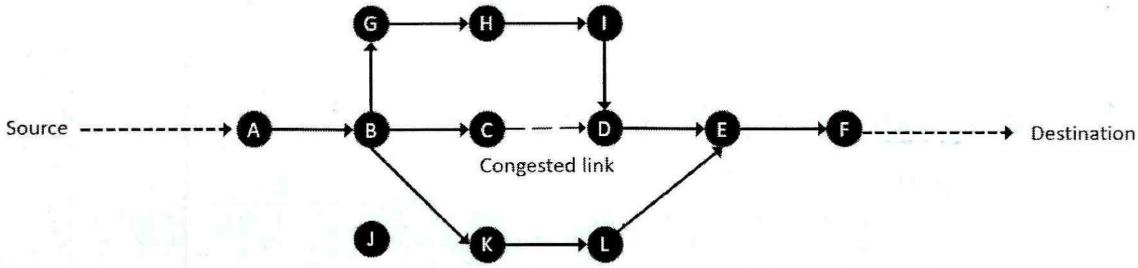
Needless to say, please explain your answers. Zero marks will be awarded if you just state an answer without any explanation. Use the roughwork pages to work out your answers and write them out neatly in the main answer sheets.

As a student of IIT Delhi, I will not give or receive aid in examinations. I will do my share and take an active part in seeing to it that others as well as myself uphold the spirit and letter of the Honour Code.

Signature: Anish Banerjee

Q1-conceptual (out of 10)	**Q6-easy (out of 6)**
Q2-medium (out of 5)	**Q7-scoring! (out of 18)**
Q3-medium (out of 6)	
Q4-easy (out of 6)	
Q5-easy (out of 3)	
Total (out of 54)	

1. This is a question about developing a new routing scheme to handle network congestion. An SDN network is assumed. Here, routers keep sending reports to the SDN controller about the congestion state in their outgoing link buffers. For example, routers could keep reporting on a regular basis to the controller about the space vacancy in their buffers. The controller uses this information to evolve a new multi-path routing: It informs routers upstream of a congested router of alternative next-hop routers and asks them to do a round-robin (or weighted round-robin) over these options. The figure below explains the setup. Initially, all packets for the destination router F were traveling along A->B->C->D->E->F. When the controller detected that the buffer behind the C->D link was getting congested, it informed upstream neighbours of C (in this case, only B), that it could also forward packets destined for F to G or K. Thus, packets for a particular flow may now take any of these paths: A->B->G->H->I->D->E->F, A->B->K->L->E->F, and the original A->B->C->D->E->F. Answer the following questions to build such a multi-path routing scheme.



a. Assume routers A..L belong to the same Autonomous System (AS) and BGP is used for sharing advertisements across ASes. Standard lookup tables in the AS routers are simply entries of (Destination IP prefix, Exit border gateway router for this prefix, Next hop interface to get to the exit border gateway router). To implement the proposed multi-path routing scheme, how would the structure of these lookup tables need to change? And what forwarding operations will need to happen in this new setup? [3]

* The lookup tables should store several next-hop interfaces to get to the exit border gateway router (BGR), along with their weights

(Dest. IP prefix, Exit BGR for this prefix, $((w_1, R_1), (w_2, R_2) \ldots (w_K, R_K)))$ — Next hop interfaces

* The SDN configures the weights of the ~~router~~ next-hop routers depending on the congestion situation in the various paths

* Forwarding: The routers implement round-robin or weighted fair queuing among the different routers ~~and the~~ according to the weights configured by the SDN. They also inform the SDN of a congested link so that SDN can adjust the weights.

b. Let us now think what routing algorithm the SDN controller can use to determine multiple paths. As stated above, assume all routers report to the controller about the congestion state on their outgoing links. The controller can then build a network graph with a weight assigned to each edge. And on this graph, the controller can run a single-source shortest path algorithm one by one for each source, a modified version which also produces second-most shortest paths, third-most shortest paths, and so on. The output can be used to inform each router of next hops to first/second/third... most shortest paths to various exit border gateway routers, and do a (weighted) round robin over these next hops. What problem do you think such an algorithm can run into? Give at least one step that could be taken to avoid or minimize such issues? [2]

* Since networks are dynamic in nature, the congestion situation may vary from time to time. So, the weights of the routers ~~also~~ should be changed by the SDN controller from time to time depending on the feedback from the routers

c. With this routing scheme operating at the network layer, will regular TCP running between a source and destination pair be able to function correctly without any changes? Here, correctly means that TCP will be able to provide reliability by ensuring retransmissions of lost data and reassembly of received data so that all the data gets across to the destination. [1]

Yes. Since TCP uses sequence numbers, reassembly of received data can be ensured. However, since the RTTs vary for different packets, timeouts ~~will need to~~ should be adjusted ~~to be max of the RTTs~~ suitably to avoid a large number of spurious retransmissions.

d. Clearly packet reordering will become quite common with this new routing scheme. How would TCP's performance be impacted because of this? Explain in terms of specific protocol parameters that might need re-adjustments or changes in their estimation method. You do not need to provide a solution, just explain the problem and its consequences. [2]

Due to this protocol, the RTT will vary for the different packets since the routes taken by them will be different. This will result in immature timeouts and <u>spurious retransmissions.</u>

Packet reordering will also cause filling of buffers

↑ to app layer

✓	X	✓	✓	✓

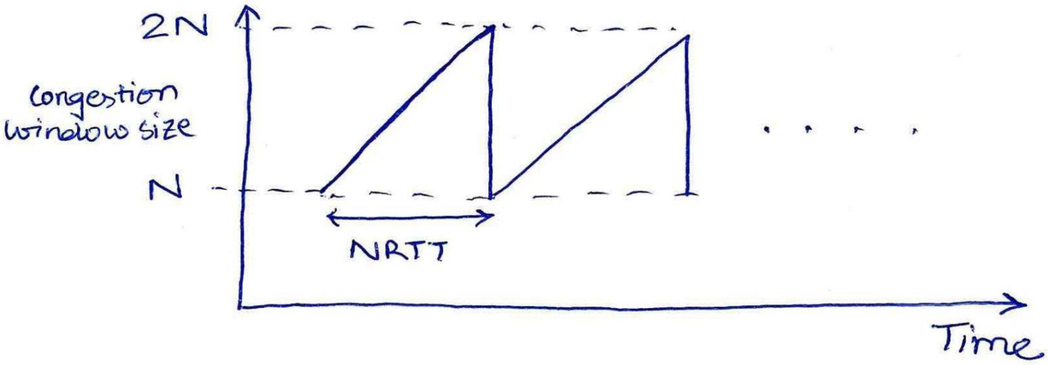
e. Another interesting way in which congestion has been proposed to be managed is by <u>packet trimming</u>. Here, instead of dropping a packet, routers trim it by dropping only the data portion of the packet but not the header. Thus, the destination receives all (or most) of the packets, but some of these packets do not have any data. However, the destination now knows precisely which packets were trimmed and can use negative acknowledgements to request the source to specifically retransmit these packets. Do you think packet trimming can make it easier to handle complications discussed above that would arise for TCP due to increased packet reordering happening with the multi-path routing scheme? [2]

Yes, this can help to improve multipath routing. NAKs can control retransmissions thereby limiting the spurious ones.

2. Show that in a steady state TCP connection working in the congestion avoidance phase, the throughput ~ 1.22 x MSS / (RTT x sqrt (L)) where RTT is the roundtrip time, MSS is the maximum segment size, and L is the loss rate

$$\text{throughput} \sim \frac{1.22 \times MSS}{RTT \times \sqrt{L}}$$

Note that in the congestion avoidance phase, all losses are assumed to be detected through fast retransmits and not timeouts, hence the congestion window rises additively and falls to half its value in a saw-tooth pattern. [5]



Number of packets transmitted before first loss:

$$N + N+1 + N+2 \cdots + 2N$$

$$= \frac{2N(2N+1)}{2} - \frac{N(N-1)}{2} \approx \frac{3N^2}{2}$$

Loss rate $L = \frac{1}{\text{\# packets sent before first loss}} = \frac{1}{3N^2/2}$

$$\Rightarrow N = \sqrt{\frac{2}{3L}}$$

$$\text{Throughput} = \frac{\text{Data sent}}{\text{Time taken}} \approx \frac{\frac{3N^2}{2} \times MSS}{N \times RTT}$$

$$= \frac{3N\ MSS}{2RTT}$$

$$= \frac{\sqrt{1.5}\ MSS}{RTT\sqrt{L}} = \frac{1.22\ MSS}{RTT\sqrt{L}}$$

3. To counter the problem of fake news, the government wants an instant messaging service (let's call it, Helloapp) to provide traceability of messages, but without violating confidentiality, i.e. if a message is forwarded from one user to another and then another and so on, any message recipient should be able to see the entire path over which the message was forwarded, but it can be assured that nobody (including Helloapp itself) other than the recipients along the message chain are able to read the message. Consider two users to start with, A and B. The user devices create their own respective public and private keypairs (K_A^+ and K_A^-, K_B^+ and K_B^-) and Helloapp provides them with certificates for their public keys K_A^C and K_B^C. Assume for simplicity that public keys are directly used to encrypt messages, instead of separately exchanging a session key.

a. In normal course without traceability, A would send a message m to B encrypted on B's public key $K_B^+(m)$, and B would decrypt it by applying its private key $K_B^-(K_B^+(m))$ to get back m. Additionally, B may validate A's identity by checking its certificate. How would the integrity of message delivery be ensured in this setup? A pre-agreed hash function H is available to both A and B. Show all messages that A will send to B. [2]

The hash function H can be used to check the integrity

$A \xrightarrow{K_B^+(m)} B$

m ↓ compute Hash $H(m)$ ↓ encrypt with K_A^- $K_A^-(H(m))$ $\xrightarrow{K_A^-(H(m))}$

$K_B^-(K_B^+(m)) \to m$ ↓ compute hash $H(m)$

$K_A^+(K_A^-(H(m)) \to \cancel{H}(m)$ compare

If calculated hash matches with the one sent by ~~Alice~~, message integrity is ensured.

b. We next want to provide a limited notion of traceability which we call *source traceability*, ie. if B forwards the message further to C using the same protocol as above, C should be able to check that the message originally came from A. We do not want to provide full traceability of the entire transmission chain, just traceability of the original source. What additional information should B send to C? Note that a trivial solution like B just sending the message and information about the source (m, A) does not work because B could easily lie about A being the source. C would need some ways to check that the message really came from A. Please also explain your answer. [2]

* B can send the "signature" $\sigma = K_A^-(H(m))$ along with the message m to C. Since $H(m)$ is encrypted using Alice's Secret key, she alone could have encrypted it.

* To check that the message indeed came from A, C computes $K_A^+(\sigma)$ to recover $H(m)$ and matches it with the hash of the message received from B
* B can also send information about the source so that C gets to know whose public key to use to verify the source.

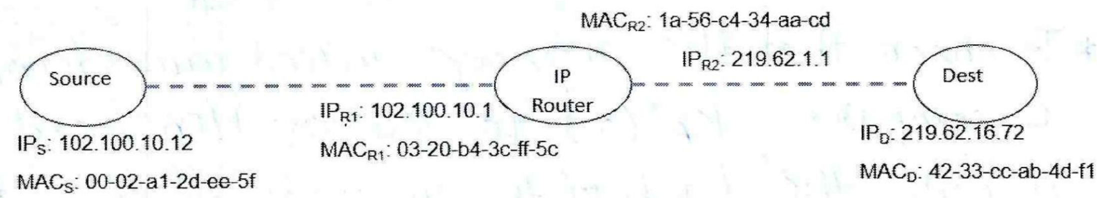
c. We next want to provide full traceability so that C can trace the entire transmission chain, ie. the message was sent by A via B, and make it extensible so that if further C forwards the message then the next recipient can trace the chain from A to B to C, and so on. How can this full traceability be provided? Explain your answer. [2]

All the people in a path can append their Signatures with the message (along with the name of the person he received the message from)

$$A \xrightarrow{\boxed{m \mid K_A^-(H(m))}} B \xrightarrow{\boxed{m \mid K_A^-(H(m)) \mid K_B^-(H(m))}} C \longrightarrow \dots$$

In this way, any person on a path can verify the trace of the message by using the public keys of the people in the trace

4. We have the following topology: a source connected over a LAN to an IP router, which is connected over a different LAN to the destination. The IP and MAC (also called physical or hardware address) addresses of the various nodes are given. The destination has a web server running on port 80.



a. For packets going out from the source, fill in the following information in the TCP/IP/LL headers (shaded regions): Source IP (SIP), Source port (SP), Destination IP (DIP), Destination port (DP), Source MAC address (SMAC), Destination MAC address (DMAC). [3]

You need not write the entire IP or MAC address in the blanks below, just fill in as "SIP: IP_S", "DP: 80", etc.

LL Header	IP header	TCP header	Application data
SMAC: MAC_S DMAC: MAC_{R1}	SIP: IP_S DIP: IP_D	SP: 12345 DP: 80	GET http://...

b. For packets arriving at the destination, fill in the same information in the appropriate places in the shaded regions. [1]

LL Header	IP header	TCP header	Application data
SMAC: MAC_{R2} DMAC: MAC_D	SIP: IP_S DIP: IP_D	SP: 12345 DP: 80	GET http://...

c. What is the gateway for the source? What is a possible network mask for the source? Give the gateway and possible network mask for the destination as well. [2]

Gateway for source: IP Router 102.100.10.1

Network mask (Source): 102.100.10/24

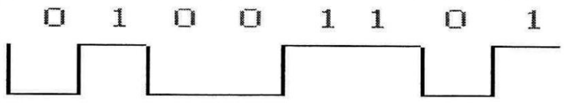
Gateway for destination: IP Router 219.62.1.1

Network mask (Destination: 219.62/16

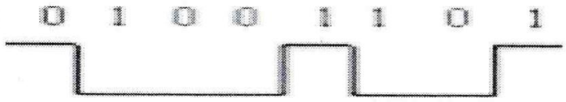
5. Given below is a table for 4B/5B encoding, and an example to help you recall NRZI.

4-Bit Data Symbol	5-Bit Code
0000	11110
0001	01001
0010	10100
0011	10101
0100	01010
0101	01011
0110	01110
0111	01111
1000	10010
1001	10011
1010	10110
1011	10111
1100	11010
1101	11011
1110	11100
1111	11101

Table 2.4 4B/5B encoding.



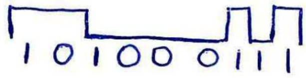
Data



NRZI

a. Encode the following sequence of bits using 4B/5B.
00100111

10100 0111



[1]

b. Give an 8-bit sequence of bits in which the number of consecutive 0s with 4B/5B encoding is maximum.

00100111 (No. of consecutive 0s can be max 3 with 4B/5B) [1]

c. Why is 4B/5B used with NRZI? [1]

Clock recovery: Since the clock at the receiver is reset at every 1, too many consecutive zeros make the clocks of sender and receiver out-of-sync. As NRZI with 4B/5B has at max 3 consecutive zeros, clock can be recovered

6. Suppose a network uses 8-bit addresses. A router is configured with the following forwarding table:

Prefix (in binary)	Interface
100	0
10001	1
101	2
1100	Return ICMP destination unreachable
Otherwise	3

a. For each of the four interfaces 0..3, give the range of matching addresses and the number of addresses. Assume that 00000000 and 11111111 are also valid addresses.[5]

Interface 0

100_ _ _ _ _ $\rightarrow 2^5 = 32$ addresses

But 10001_ _ _ $\rightarrow 2^3 = 8$ addresses will be routed to interface 1. So # addresses = 32 − 8 = 24

Range = 10000000 − 10000111 and 10010000 − 10011111

100 00000 − 10011111 except 10001000 − 10001111

Interface 1

As shown above # addresses = 8. Range = 10001000 − 10001111

Interface 2

101_ _ _ _ _ $\rightarrow 2^5 = 32$ addresses. Range: 10100000 − 10111111

Interface 3

All except the ones mentioned. $\overset{0}{24} + \overset{1}{8} + \overset{2}{32} + 16 = 80$

addresses = $2^8 - 80 = 176$ Range: 0000 0000 − 0111 1111 and 11010000 − 1111 1111

b. Prefix 1100 indicates unallocated address space under control of this ISP. A customer organization requests for 4 addresses. Give an example prefix obtained from the unallocated block that can be added to the forwarding table. [1]

1100 00_ _ will give 4 addresses

So Prefix: 110000

7. Short answer questions:

a. Persistent HTTP improves communication efficiency because... [1]

there is no wastage of time in setting up the connection again

b. UDP does not have congestion control built into it, and this can be a problem because... [1]

UDP may "starve out" TCP traffic, as it may fill up the buffers of the routers, leaving no space for TCP

c. Reverse path forwarding for multicast routing requires all nodes to have computed a shortest path to a centre node. True/False? [1]

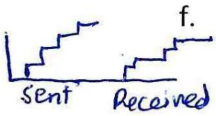
False. The same spanning tree can be used for multiple nodes

d. An Intserv design for QoS is able to provide QoS guarantees because it does not admit new connections if it cannot reserve resources for them. True/False? [1]

~~False~~ True

e. What traffic shaping filter is used to configure an average rate as well as a maximum burst size? [1]

Leaky Bucket Filter



f. Playback delays are introduced in streaming audio and video applications to mask the jitter. What is jitter? [1]

Different packets are received at different times from which they were originally sent. This may lead to distortion of the audio called jitter.

g. Lookup tables in routers for IP multicast addresses have not one outgoing interface corresponding to each prefix but many. True/False? [1]

True

h. FEC (Forward Error Correction) transmits redundant information so that the original information can be recovered even if some parts of the transmitted information are lost or corrupted. 2/3 inter-packet FEC means that that 3 packets are transmitted for every 2 packets, and if any two of the three packets are received then the original two packets can be reconstructed. Similarly, 2/3 intra-packet FEC means that within the same packet, for any 2 blocks of data (of say 512 bytes each), 3 blocks are written in the packet, and the original two blocks can be recovered if any two of the three blocks are received correctly. When would you choose inter-packet FEC and when would you choose intra-packet FEC? Explain your answer. [2]

Inter-packet: It should be used if the channel loses packets often

Intra-packet: It should be used if the channel ~~att~~ causes its data to get corrupted

i. Continuing from the previous question, name one error checking mechanism you would use at the intra-packet level to check whether a data block was received correctly or not. [1]

Checksums or Cyclic Redundancy Check (for each block)

j. What type of DNS record is used to identify the mail server for a domain? [1]

MX

k. Poison reverse is used to solve the count to infinity problem in distance vector routing algorithms. True/False? [1]

True

l. Head of Line blocking in HTTP connections can be solved by opening a new TCP connection for each object to be fetched. True/false? [1]

True

m. Cookies are maintained by the operating system, therefore if you use Chrome and Firefox on the same computer to access the same website, both the browsers will send the same cookie to the website. True/False? [1]

False

n. Can an ISP implement ad blocking if HTTPS is being used? [1]

No. Data is end-to-end encrypted

o. Is Bufferbloat more likely to occur when a router (1) maintains very large per-flow buffers, or (2) when it maintains a very large common buffer? Choose one of the two options. [1]

(2) when it maintains very large common buffer

p. What would you need to do to run a server behind a NAT? [1]

Port forwarding

q. IP multicast uses a special set of IP addresses to identify multicast groups. True/false? [1]

True

--- ROUGH WORK ---