COL334/672: Semester 2023-24-1

Major exam: 120 minutes, closed-book.

Name: KUSHAGRA GUPTA

Entry number: 2021CS50592

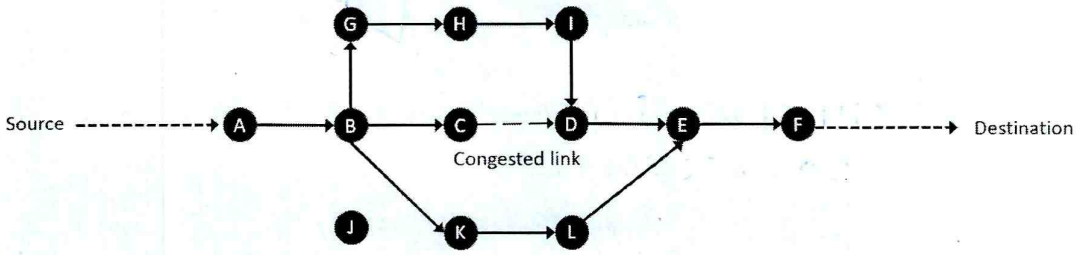
Needless to say, please explain your answers. Zero marks will be awarded if you just state an answer without any explanation. Use the roughwork pages to work out your answers and write them out neatly in the main answer sheets.

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: Kushagra

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]

We would need to add multiple next hop interfaces for every destination IP prefixes to accomodate the possibility of forwarding through multiple paths along with their weights.

So we can use an array / table which contains all possible next hop interfaces & their corresponding weights which could be used for a weighted round robin scheduler.

Eg: B has the table (Dynamic depending on congestion)

C	1
G	10
K	10

★ For the forwarding operations, we can used (weighted) round robin scheduling or priority scheduling to ensure that the distribution of traffic is optimised depending on the congestion state of the bottleneck link on the path to exit gateway router to fill the table

So the lookup table has (Dest IP prefix, Exit gateway, next hop table)

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 the congestion in the network is variable, it could would need to change the weights assigned to each edge dynamically. This way it would be very computation intensive to update the weights fractions of the ~~round~~ weighted round robin scheme.

One of the ~~two~~ problems it could face is — if it selected a path which has a seemingly lower congestion, it could end up routing a lot of traffic to that ~~link~~ path's links and we could have a link even more congested than before. (due to various inter dependencies in all possible paths through common links).

We can resolve this by computing a minimum spanning tree / Steiner tree so that instead of weighting by first / second shortest paths, we weigh using bottleneck links on that tree's path, tree will regulate and not crowd traffic at one location.

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]

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]

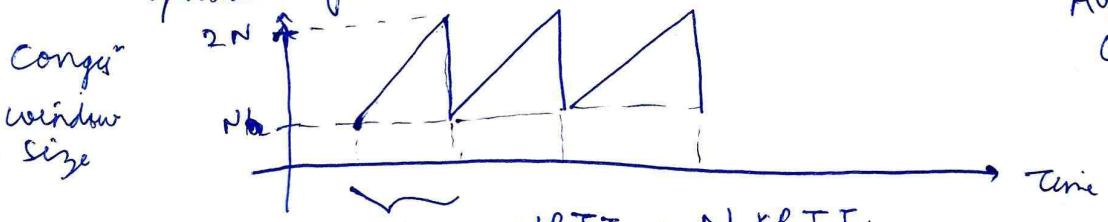
e. Another interesting way in which congestion has been proposed to be managed is by packet trimming. Here, instead of dropping a packet, routers trim it by dropping only the data portion of the packet but not the header. Thus, the destination receives all (or most) of the packets, but some of these 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, because of increased packet reordering, there would be a lot of overhead, can use V

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

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]

The steady state TCP connection in congestion avoidance phase follows the curve



Avg size of Cong-window $\simeq 3N/4$

$\text{Time} = N \cdot RTT = N \times RTT$

$$\frac{1}{L} = N + (N+1) + (N+2) \cdots 2N = (2N+1)N - \frac{N(N-1)}{2}$$

$$= \frac{3N^2}{2} + \frac{5N}{2}$$

$$\therefore \quad 1/L \simeq \frac{3N^2}{2} \Rightarrow N \simeq \sqrt{\frac{2}{3L}}$$

(as the loss rate is ~~number~~ $\frac{1}{\text{Total number of packets sent}}$).

Now the throughput = $\frac{\text{Total bytes sent}}{\text{Total time taken}}$

$$\simeq \frac{(N + (N+1) + (N+2) \cdots 2N) \times MSS}{N \times RTT}$$

$$\simeq \frac{3N^2}{2N} \times \frac{MSS}{RTT}$$

$$\simeq \frac{3N}{2} \times \frac{MSS}{RTT}$$

, now as $N \simeq \sqrt{\frac{2}{3L}}$

$$\frac{3N}{2} \simeq \sqrt{\frac{3}{2L}}$$

$$\simeq \frac{\sqrt{3/2} \times MSS}{RTT \times \sqrt{L}}$$

, as $\sqrt{1.5} \simeq 1.22$

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

Hence Proved.

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]

To ensure the integrity of message delivery, after B has received m', A first computes a hash H of the message to be sent. $H(m)$.

Then, A encrypts the hash using the public key of B, which could have been sent prior by sending B's certificate to A by Helloapp.

A ← K_B^C (certificate) — B

A → $K_B^+(H(m))$

Now B decrypts the hash $K_B^-(K_B^+(H(m))$ to obtain $H(m)$ and it locally computes the hash of received message say $H(m')$ if $H(m') = H(m)$, then by properties of a strong hash function, B knows $m' = m$, so the integrity is validated. If integrity

Note: This could have also been done in reverse, i.e B sends hash $H(m)$ to A.

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]

Instead of sending the message m encrypted using C's public key, B sends the received ~~public~~ encrypted message over A's private key. (where A is the source).

Protocol: The source (here A) first encrypts the message with its own private key, K_A^- & then sends the encrypted message $K_A^-(m)$ by encrypting over B's ~~private~~ public key → $K_B^+(K_A^-(m))$ alongwith A's certificate

Now, when B forwards the encrypted message over it can only apply its private key K_B^- & forward the $K_A^-(m))$ along with A's certificate K_A^c this process would continue & we assure that only A is the source & nothing else.

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]

Similar to above, in this case;

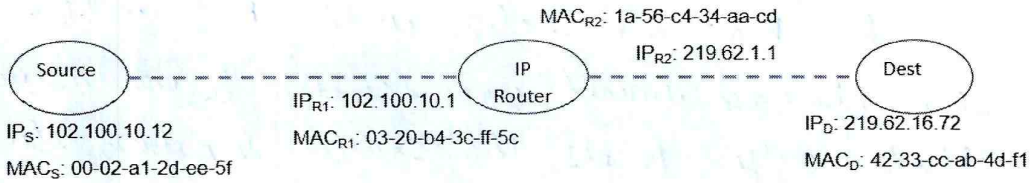
B forwards $K_B^- K_A^-(m))$ alongwith A & B's certificate K_A^c, K_B^c.

This chain would contain -

$K_C^- K_B^- K_A^-(m)$ & hence we would ~~have~~ use the K_A^c, K_B^c, K_C^c.

& we can decrypt at the end, all these certs would conclude full traceability

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	SIP: IP_S	SP: SP	GET http://...
DMAC: MAC_{R_1}	DIP: IP_D	DP: 80	

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_{R_2}	SIP: IP_{R_2}	SP: SP	GET http://...
DMAC: MAC_D	DIP: IP_D	DP: 80	

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]

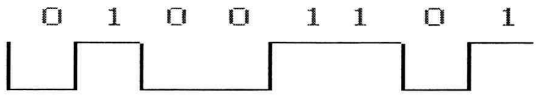
Gateway for source → IP_{R_2} (IP router).
N/w mask for sour → MAC_{R_2}
" for dest^n → IP_{R_1}
" for dest^n → MAC_{R_1}

(as router separates the sour & dest)

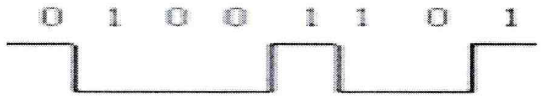
5. Given below is a table for 4B/5B encoding, and an example to help you recall NRZI.

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,

10100011111 [1]

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

0 0 1 0 0 0 0 1 (0 0 0 1 0 0 0 0 1) [1]

the number of cons. 0's with 4B/5B encoding is 3.
max

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

Because due to more than 3 consecutive 0's there ~~is~~ is a lot of error in NRZI encoding as it observes the flips in the data.
So 4B/5B makes certain there would be ~~a lot of~~ at max 3 consecutive 0's.

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]

For the addresses 10001000 to 10001111, they will be forwarded to interface 1 (so a total of 8 addresses). (longest prefix matched)
The addresses 100 00000 to 100 11111 will be forwarded to interface 0 except those from 10001000 to 10001111 (as these have the longest prefix matched for interface 1)
So a total of 32-8 = 24 addresses forwarded to int. 0.
Similarly, address from 101 00000 to 101 11111, a total of 32 addresses forwarded ~~from~~ to interface 2.
The addresses from 1100 0000 to 1100 1111 return ICMP (16 addresses)
The remaining addresses 11 000000 to 11 111111 except above 16 forwarded to interface 3 & all address from 0 0000000 to 0 1111111.

128+48 = 176

Answer :-
Int 1 : (10001000 to 10001111) & add
Int 0 : (100 00000 to 100 00111 and 100 10000 to 100 1111)
Int 3 : (0 0000000 to 0 1111111) and (110 10000 to 110 111111).
ICMP
Int 2 as above

0 → 24, 1 → 8, 2 → 32, 3 → 176, ICMP return → 16 Total = 256.

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]

We can add the prefix 1100 00 in the above table for the customer requesting 4 addresses. In this manner, any requested IP with that prefix would be forwarded to the customer org. (as it is the longest prefix for these 4 IP addresses (6 length) not there are only 4 such IP's as $2^2 = 4$ (2 bits remaining)

7. Short answer questions:

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

We don't have to establish the TCP connection everytime which takes time in the order of RTT. So, we utilize the already established TCP connection

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

There would be no congestion or flow control & this can cause overloading / filled queues or buffers. So UDP will keep on dropping packets due to filled queue & won't receive data.

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

True

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]

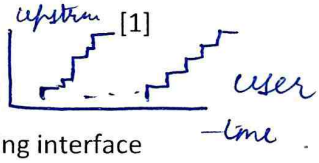
True

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

CUBIC

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

The variable rate of incoming upstream data causes buffering to provide constant rate to user. This is 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]

False

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]

If the packets may get dropped, for eg UDP, it would be better to use inter-packet FEC

In other schemes such as packet corruption or trimming, it would be better to use intra packet FEC in order to ensure that we gain the maximum network utilization as the packets are not completely lost. (inter packet in this case would not accept any packet) whereas for cases like UDP, intra packet would only be able to achieve 2/3rd efficiency.

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]

CRC (Cyclic Redundancy Check) or Checksum (2 dimension)

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

MX type of DNS record.

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]

Parallel TCP connections, 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]

Yes,

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 a very large common buffer.

p. What would you need to do to run a server behind a NAT? ~~IPV6~~ IPV4. [1] for distinguishing

~~(We need to tunnel the packets (IPV6) in IPV4 packets)~~ We provide IP, port inside the network corresponding to every (IP, port) outside the n/w.

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

True.

---------- ROUGH WORK ----------