Major

● Graded

Student

Saharsh Laud

Total Points

38 / 75 pts

Question 1

TCP Reliability

3 / 9 pts

– **0 pts** Correct

– **2 pts** Incorrect/Not Attempted part A

✔ – **3 pts** Incorrect/Not Attempted part B

✔ – **2 pts** Incorrect/Not Attempted part C

– **2 pts** Incorrect/Not Attempted part D

+ **0.5 pts** Correct answer but no explanation/incorrect explanation in part A

– **0.5 pts** Partial correct justification in part A

+ **0.5 pts** Correct answer but no explanation/incorrect explanation in part B

– **1 pt** Partial/Incomplete correct justification in part B

+ **0.5 pts** Correct answer but no explanation/incorrect explanation in part C

– **0.5 pts** Partial correct justification in part C

✔ – **1 pt** Incorrect relation in part D

– **0.5 pts** Partial correct relationship in part D
Correct relation is S-A<=min(C,R)

– **0.5 pts** Partial correct role of receiver-advertised window

– **1 pt** Incorrect/Missing role of receiver-advertised window

Question 2

TCP CCA

5 / 8 pts

+ 3 pts A1 - (6 + 6 + 7 +7 + 8 + 8) / 6 = 7 MSS / RTT
show 6, 7, 8 instead of 6.5, 7.5, 8.5 respectively
directly or indirectly - MSS sized pkt only

+ 1.5 pts A2 - Working with 6,6.5/7,7.5/8,8.5

✔ **+ 0 pts** A3 - Part A Completely wrong / Unattempted

✔ **+ 2 pts** B -
start - 1 MSS

RTT	MSS
1	2
2	4
3	8
4	16

4 RTTs = 40ms

+ 0 pts B2 - Wrong

✔ **+ 3 pts** C1 -
Consider 2 initial bandwidths w1, w2, s.t., w1 < w2
Clearly show difference of growth and stunting of bandwidths based on their initial bandwidth values
i.e., how one can hog more bandwidth

+ 1 pt C2 - Mention not fair - without reasoning / wrong reasoning

+ 0.5 pts C3 - underutilization/oscillation reasoning

+ 0 pts C4 - Part C Completely Wrong / Unattempted

Question 3

TCP Throughput

5 / 6 pts

✔ **– 0 pts** Correct

– 6 pts Incorrect / Not attempted

– 3 pts Part (a) - Incorrect / Not attempted

– 2 pts Part (a) - Incorrect derivation of total no.of packets / Not attempted

– 1 pt Part (a) - Incorrect / missing final loss-rate derivation

– 3 pts Part (b) - Incorrect / Not attempted

– 1 pt Part (b) - Incorrect approximation of p / Not attempted

– 1 pt Part (b) - Incorrect throughput formula / Not attempted

– 1 pt Part (b) - Incorrect final substitution / Not attempted

– 1 pt Total no.of packets = 3W/4*(W/2 + 1)

Question 4

Network-assisted

3 / 4 pts

- ✔ **+ 1 pt** Correct order (B>C>A)
- **+ 1 pt** FIFO gives advantage to loss based flow (TCP Reno)
- ✔ **+ 1 pt** Round robin gives fair share to both flows
- ✔ **+ 1 pt** Packets dropped probablisticly moderating TCP Reno 's aggressive behavior.
- **+ 0 pts** Wrong
- **+ 0 pts** Unattempted

Question 5

DASH

1 / 6 pts

- ✔ **+ 1 pt** **Part A:** Correct
- ✔ **− 0.5 pts** **Part A:** Edge case not handled or wrong(e.g., missing max(0,·))
- **+ 0 pts** **Part A:** Incorrect or unattempted
- ✔ **+ 2 pts** **Part B:** Correct complete expression
- ✔ **− 0.5 pts** **Part B:** Missing min(Bk, Bmax) (uses Bk directly).
- ✔ **− 0.5 pts** **Part B:** Missing max(0,·) for the no-negative-buffer edge case.
- **− 0.5 pts** **Part B:** Missing +L (the chunk duration).
- ✔ **− 0.5 pts** **Part B:** Incorrect or missing download time expression □□
- **+ 0 pts** **Part B:** Incorrect or unattempted
- **+ 1 pt** **Part C:** Correct expression
- **− 0.5 pts** **Part C:** Missing max(0,·)
- **− 0.5 pts** **Part C:** Missing min(Bk, Bmax)
- ✔ **+ 0 pts** **Part C:** Incorrect or unattempted
- **+ 2 pts** **Part D:** Provides all three metrics (x, y, z) with correct, meaningful interpretations.
- **− 0.5 pts** **Part D:** Writing vague term for x
- **− 0.5 pts** **Part D:** Writing vague term for y
- **− 0.5 pts** **Part D:** Writing vague term for z
- ✔ **+ 0 pts** **Part D:** Incorrect or unattempted

Question 6

HTTP

2 / 8 pts

– **0 pts** Regrade Requests Guidelines:
1. Check whether your points had been adjusted Manually.
2. Part C - FCFS != Inorder Delivery, Loss retransmission.
2.1 Segmentation had already been implemented in HTTP/2
3. Absence or non inferable efficiency metric had been penalised.
4. **Frivolous requests might be penalised.**

– **0 pts** Correct

– **8 pts** Incorrect/Not Attempted

✔ – **2 pts** Object Count

✔ – **3 pts** Caching Not Attempted/Incorrect

– **0.5 pts** Efficient for JPEG

– **1 pt** Frequency of updates comparison

– **1.5 pts** Bandwidth/Load/Size

– **3 pts** TCP HOL blocking ~ Not Attempted/Incorrect

✔ – **2 pts** Resoning for not working/Partial reasoning had been adjusted Manually

– **1 pt** Solution To problem

+ **1 pt** C

Question 7

P2P/DNS

4 / 5 pts

✔ + **1 pt** Correct Advantage of in-order download

✔ + **1 pt** Correct Limitation of in-order download

✔ + **1 pt** Alternative chunk selection policy that addresses the limitation

✔ + **1 pt** Problems caused by removing MX records

+ **1 pt** Clear explanation of why these are problems

+ **0 pts** Not attempted/Wrong

Question 8

DHT

3 / 5 pts

+ **1 pt** Use of lookup(x) to find successor

+ **2 pts** Correct identification of predecessor p and setting local pointers

+ **1 pt** Correct neighbor updates using update predecessor and update successor

+ **1 pt** Correct key transfer procedure using get keys, store, del

+ **0 pts** Wrong/Not Attempted

+ **3 pts** Point adjustment

Question 9

Security

2.5 / 6 pts

− **0 pts** Correct

− **6 pts** Incorrect / Not attempted

− **2 pts** Part (a) - Incorrect / Not attempted

− **1 pt** Part (a) - Incorrect symmetric key count / Not attempted

✔ − **1 pt** Part (a) - Incorrect keys count in public key encryption / Not attempted

✔ − **2 pts** Part (b) - Incorrect / Not attempted

− **1 pt** Part (b) - Incorrect / No explanation

− **2 pts** Part (c) - Incorrect / Not attempted

− **1.5 pts** Part (c) - Incorrect / No explanation

− **0.5 pts** Part (c) - Vague/Incomplete explanation

− **0.5 pts** Part (b) - Vague/Incomplete explanation

− **0.5 pts** Part (c) - did not explain how knowing 's' will help attacker

Question 10

Security2

4 / 6 pts

+ 0 pts A. Incorrect Answer

✔ **+ 0.5 pts** A. Correct Answer

+ 1 pt A. Vague explanation

✔ **+ 1.5 pts** A. Correct Explanation

✔ **+ 1 pt** B. Correct definition of Truncation attack

✔ **+ 1 pt** B. Correct explanation of how TLS mitigates the Truncation attack

+ 0.5 pts C. Correctly explained the purpose of the **padding** field

+ 0.5 pts C. Correctly explained the purpose of the **pad length** field

+ 1 pt C. Correctly explained the purpose of the **next-header** field

Question 11

Firewall

2 / 4 pts

+ 0 pts Incorrect Answer / Not Attempted

+ 1 pt Mentioned the installation of default switch rules by the controller:
1. **Outgoing SYN (internal→external):** match `in_port=internal`, TCP SYN; actions: *forward normally* + *send copy to controller*.

2. **Incoming TCP (external→internal):** low-priority match on `in_port=external, ip_proto=TCP`; action: *drop*.
Thus, all inbound TCP traffic is dropped unless a later-installed allow rule overrides this.

✔ **+ 1 pt** Mentioned controller state update on outgoing SYN:
When the controller receives a `PACKET_IN` for an outgoing SYN, it extracts the 4-tuple **(internalIP, internalPort, externalIP, externalPort)** and adds an entry to its connection table (e.g., state = *SYN_SENT*) if it is not already present. This marks the flow as permitted to receive inbound packets.

✔ **+ 1 pt** Mentioned controller actions after updating its state:
After updating its connection table, the controller installs the following **bidirectional flow** rules in the OpenFlow switches:
1. **Internal→External:** match the 5-tuple `(srcIP, dstIP, srcPort, dstPort, ip_proto=TCP)` for the flow; action: forward to external port.
2. **External→Internal:** match the reverse 5-tuple; action: forward to internal port.
These **higher-priority** rules override the default drop rule, allowing packets only for flows whose outgoing SYN was observed.

+ 1 pt Mentioned how clean-up is handled:
When the controller observes FIN/RST packets (via `PACKET_IN`) or when flows become idle, it removes the corresponding state entry and deletes the installed allow rules in the OpenFlow switches. Idle timeouts may also be used, allowing inactive flows to be automatically purged.

Question 12

SBGP **3.5** / 8 pts

– 0 pts Regrade Request Guidelines:

0. Most of your answers were wrong and incoherent so they had been patially awarded.
1. *Please check if anything if carried forward to next page had been checked.*
2. A. General Expectation: Digital Certificate to originating AS signed by Trusted Authority. This can not be solved only by Cryptographic hashing.
3. B. Assymetric Key/Public-Private Key, Hashing/Encryption/Digital Signature of Paths, Message Forwarding/Chains/Recursive Decryption, Message Verification/Decryption.
4. This criterion had been done away for correct but innovative design thoughts, if found.
5. Drawbacks should not be overlapping or same thing written in defferent words.
6. Anyone claiming for **clause 4** might be **harshly penalised** if un satisfactory.

– 0 pts Correct

– 8 pts Not Attempted/Incorrect

✔ **– 2 pts** Digital Certificates signed by trusted certificate Authority

– 1 pt Digital Certificates signed by trusted certificate Authority

– 4 pts Path Verification

✔ **– 2 pts** Path Verification

– 1 pt Path Verification

– 2 pts Limitation Incorrect/Not Atteempted

✔ **– 1 pt** Limitations only one correct/ Similar Limitation

💬 **+ 0.5 pts** Point adjustment

Name: SAHARSH LAUD Entry Number: 2024MCS2002

Major Exam

COL334/672: Computer Networks

Sem I, 2025-26

There are 12 questions and 17 pages in this quiz booklet (including this page). There are **75 total points**, and you have **120 minutes** to answer the questions.

- **Feel free to think outside the box but write inside the box**
- **Write concise answers**
- **Do not start the exam until instructed to do so**

I Transport Layer

1. [9 points]: Answer the following questions:

A. TCP uses Go-Back-N for implementing reliability. True or False. Justify your answer.

B. Consider a connection-termination protocol in which each side must first send a "done" message and also wait to receive a "done" message from the other side before simultaneously closing the connection. Is this practical in real-world networks? Explain why or why not.

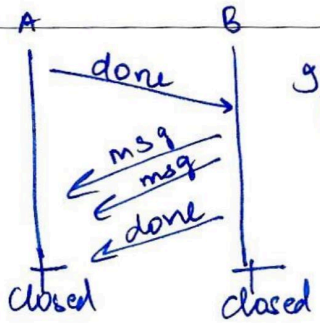
C. In TCP, if the instantaneous round trip time at any given time is t sec, the value of the retransmission timeout is always set to greater than or equal to t sec. True or False. Justify your answer.

D. Consider a TCP connection where the last byte ACKed is A, latest byte sent but not yet ACKed is S, current congestion window is C bytes, and the receiver advertised window is R bytes. What is the relationship among these variables? What role does the receiver-advertised window play?

A. False : TCP uses hybrid of GoBackN and Selective Repeat to implement Reliability.
TCP use Cumulative ACK's like GBN and selective retransmission like Selective Repeat.

B. This seems to be practical only when the sender who sends "done" message first is able to accept any messages the other party is sending before they also send "done" message.

2024MCS2002



If The host A is not able to receive message after sending done then this is not practical but if it happens like in the diagram then this is feasible solution for connection termination.

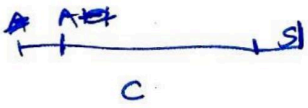
C. True: Since the RoundTrip time includes time to send packet & receive Ack we should wait atleast t sec before time out to prevent Premature Timeouts. ∴ Timeout time ≥ t sec

D. Last Byte Acked = A
Latest Byte sent but not yet Acked = S
Congestion window = C
Receiver window = R

→ As per Congestion Control → Congestion wnd ≤ Receiver wnd

$\therefore \quad C \leq R$

Next Ack will be A+S



→ $S - A = C$

→ Receiver window helps in Flow Control since it assists in regulating sender flow so that sender doesn't overwhelm receiver buffers.

2024 MCS2002

2. [8 points]: Answer the following questions:

A. Suppose TCP uses AIMD congestion control with no slow start and an initial window of 6 MSS. *The receiver sends one ACK for every two packets, and the sender transmits only MSS-sized packets.* Assuming a constant RTT and no loss, what is the average throughput (in terms of MSS and RTT) over the first 6 RTTs?

B. Assuming TCP is using slow start with initial window = 1 MSS on a line with a 10-msec round-trip time and no congestion. The receiver-advertised window is 24 KB and the maximum segment size is 2 KB. How long does it take before the sender can transmit a full receiver window?

C. Suppose TCP starts using Multiplicative Increase, Multiplicative Decrease for congestion control. Would such an algorithm be TCP-fair? Justify your answer.

A → Initial = 6 MSS → 1 ACK for 2 Packets

Final = 3 MSS ∴ Avg = $\frac{6+3}{2} = \frac{9}{2} = 4.5$ MSS

∴ Avg Throughput = $\frac{4.5 \text{ MSS}}{\text{RTT}} = \frac{6 \text{ MSS}}{\text{RTT}}$

B → w_i = 1 MSS ; RTT = 10 ms

rcvnd = 24 KB MSS = 2 KB

rcvnd = $\frac{24 \text{ KB}}{2 \text{ KB}} = 12$ MSS

1 MSS $\xrightarrow{1 \text{ RTT}}$ 2 $\xrightarrow{1 \text{ RTT}}$ 4 $\xrightarrow{1 \text{ RTT}}$ 8 $\xrightarrow{1 \text{ RTT}}$ (6) → Now sender can send full recvr window

~~1 MSS per RTT~~

∴ Sender can Transmit → 4 RTT's

= 4 x 10 = 40 msec

2024MCS2002

c. No such an Algo Multiplicative Increase Multiplicative decrease must be TCP Fair.

$$\text{Inc} \longrightarrow \quad w_{i+1} = a * w_i \quad ; \; a > 1$$

$$\text{Dec} \longrightarrow \quad w_{i+1} = b * w_i \quad ; \; b < 1$$

In such a case the rich get richer so a flow with higher ~~state~~ window will ramp faster than slower one

$F_1 = \quad w_1 = 20 \qquad a = 5$

$F_2 = \quad w_2^0 = 6 \qquad b = 1/2$

$\text{Inc} \longrightarrow \quad F_1 = 20 \times 5 = 100$

$F_2 = 6 \times 5 = 30$

$\text{Dec} \longrightarrow \quad F_1 = 100/2 = 50$

$F_2 = 30/2 = 15$

So we can see Rich get Richer so convergence to common flow not possible Hence its not TCP-Fair.

2024MCS2002

3. [6 points]: Recall the macroscopic description of TCP throughput. In the period of time from when the connection's rate varies from $\frac{W}{2\times RTT}$ to $\frac{W}{RTT}$, only one packet is lost (at the very end of the period).

A. Show that the loss rate (fraction of packets lost) is equal to $\frac{1}{\frac{3}{8}W^2+\frac{3}{4}W}$

B. Use the result above to show that if a connection has a loss rate p, then its average throughput is approximately given by $\frac{1.22\times MSS}{RTT\sqrt{p}}$

A → Rate → $\frac{W}{2RTT} \rightarrow \frac{W}{RTT}$

Avg window = $\frac{\frac{W}{2}+W}{2} = \frac{3W}{4}$ ⇒ Avg Packets $\leq$ 3

$\frac{W}{2} \rightarrow W \rightarrow$ Packets = $W/2$

Total packets = Throughput = $\frac{3W\ MSS}{4\ RTT}$

Cycle = $W/2$ window = $\frac{3W}{4}$ ⇒ # Packets = $\frac{3W}{4}\times\frac{W}{2} = \frac{3W^2}{8}$

∴ Total Packets = $\frac{3W^2}{8} + \frac{3W}{4}$ → lost = 1

∴ Loss Rate ⇒ 1 Packet send = $\frac{1}{\frac{3W^2}{8}+\frac{3W}{4}}$ Lost

B → $P = \frac{1}{\frac{3W^2}{8}}$ ⇒ $W^2 = \frac{8}{3P}$ ⇒ $W = \sqrt{\frac{8}{3P}}$

Avg Throughput = $\frac{\text{Avg window}}{RTT} = \frac{3W\ MSS}{4\ RTT}$

⇒ $W = \sqrt{\frac{8}{3P}}$ ⇒ $\frac{3}{4}\times\sqrt{\frac{8}{3P}}\times\frac{1}{RTT}$

$= \sqrt{\frac{8}{16}\times\frac{9}{8}} = \sqrt{1.5} = 1.22$ ⇒ $\frac{1.22\times MSS}{\sqrt{P}\times RTT}$ = Avg Throughput

2024 MCS2002

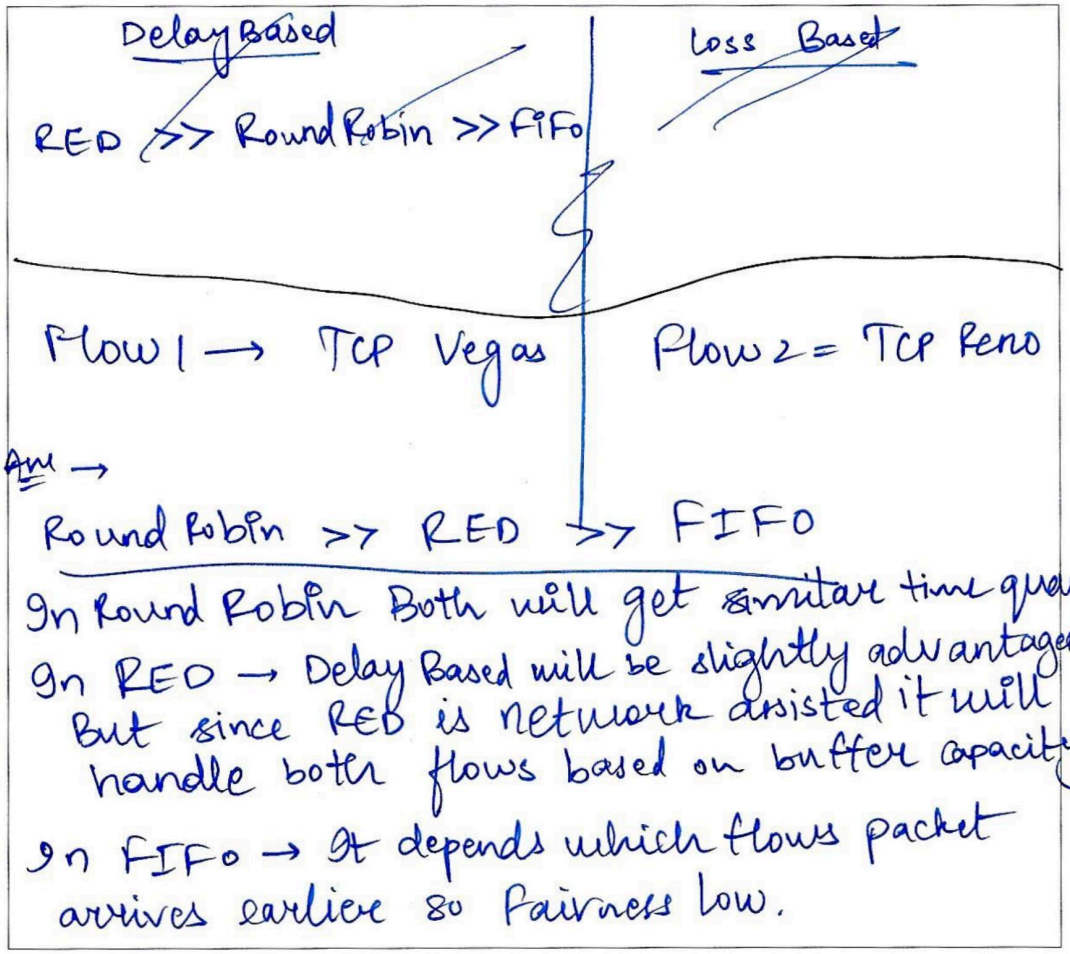
4. [4 points]: Consider a single bottleneck link shared by two long-lived flows. Flow 1 uses TCP Vegas, a delay-based congestion control algorithm, while Flow 2 uses TCP Reno, a loss-based congestion control algorithm. Assume both flows have the same RTT and traverse the same bottleneck router. The router can be configured in three ways:

A. as a simple drop-tail FIFO queue, or

B. using a per-flow round-robin scheduling policy, or

C. using Random Early Detection (RED), which drops packets probabilistically before the buffer is full.

Rank the three cases in decreasing order of the fairness achieved between the delay-based and loss-based flows. Justify your answer.



2024MCS2002

II Application

5. [6 points]: Consider a DASH-based video streaming client. The video is divided into chunks of duration L seconds each. Let:

- B_k = buffer occupancy (in seconds) immediately after chunk k has finished downloading and has been added to the buffer
- B_{max} = buffer threshold (in seconds) beyond which the client pauses new chunk requests
- R_{k+1} = bitrate selected for $k+1^{th}$ chunk (bits/sec)
- C_{k+1} = observed bandwidth during the $k+1$ download (bits/sec)

Assume that playback continues during downloading as long as the buffer is non-empty. A chunk is available only after it has been fully downloaded.

A. How long does the client wait (if at all) before sending the request for chunk k+1?

B. Derive an expression for the buffer occupancy B_{k+1} (in terms of variables above) at the end of downloading chunk $k+1$.

C. Give an expression for the stall (rebuffering) duration, if any, during the download of chunk k+1.

D. Suppose you are designing a bitrate adaptation algorithm that optimizes the following function:

$$QoE = f(x) - \alpha \times g(y) - \beta \times h(z)$$

Here x, y, and z are the **application-level** performance metrics and f, g, and h are non-decreasing functions. Briefly state what these metrics could represent in the video streaming context?

A. If given Rate $\geq R_{K+1}$ → No wait ask immediately & Buffer occupancy ~~B_{K+1}~~ $\leq B_{max}$

If Rate $\geq R_{K+1}$ But Buffer Full ie. $\geq B_{max}$ wait till L seconds so that Buffer has space For $k+1^{th}$ chunk

B. Size For $k+1 = L * R_{K+1}$

Time to download $= L * R_{K+1} * C_{K+1}$
$k+1$ chunk

$\therefore B_{K+1} = B_K - (L * R_{K+1} * C_{K+1}) + L$

2024MCS2002

S → Stall = $L * R_{K+1} * C_{K+1}$

D → QoE ⇒ $f(x) \rightarrow \alpha * g(y) - \beta * h(z)$

These metrics can be
- Buffer occupancy
- Last Bit Rate
- Current BitRate

6. [8 points]: Answer the following questions:

A. A Web page consists of multiple objects fetched over a persistent HTTP connection. When these objects are requested sequentially, the total page load time is 3 seconds, of which 150 ms is spent establishing the connection. When HTTP pipelining is used for the same page, the total load time decreases to 200 ms, assuming all object requests are issued at once and each object has the same response time. How many objects (HTTP requests) are required to load this page?

B. The `If-Modified-Since` header can be used to check whether a cached page is still valid. Requests can be made for pages containing images as well as only HTML text. Do you think the effectiveness of this technique is better or worse for JPEG images as compared to HTML? Think carefully about what "effectiveness" means and explain your answer.

C. Why does HTTP/2 still suffer from head-of-line (HOL) blocking, even though it supports multiplexing over a single TCP connection? Briefly describe a solution that avoids this limitation.

2024MCS2002

A. → x objects Page load = 3 ; Connection = 150ms

→ 150 objects are Required to load Page.

B. Worse for Images as the Image may have been created long back & uploaded to the page so even though If-modified-Since shows latest entry we might not be able correctly identify if image has been changed or not.
While its better for HTML Text since we can do checksum or some hash & check if its actually modified or not.
So its less (worst) effective for JPEG images than HTML.

C) Multiplexing is over single TCP Connection so Packet loss or congestion in earlier request can cause HOL Blocking
Using QUIC (HTTP/3) with UDP can resolve it.

2024 MCS 2002

7. **[5 points]:** Briefly answer the following:

A. In BitTorrent, suppose a peer downloads file chunks strictly in order (i.e., chunk 1 first, then chunk 2, and so on). Describe one advantage of this policy. Describe one limitation of this policy and propose an alternative chunk selection policy that addresses the limitation.

B. Recall that DNS MX records are used for email delivery. Suppose DNS were redesigned to eliminate MX records and rely solely on CNAME or A records to identify the mail server for a domain. What problems would arise with such a design?

A. Adv → File is downloaded sequentially so maybe all closest neighbors have sequential parts so it'll be faster to search.
It's simple since only need to search chunks in order
Disadv → Rarest chunk might get lost if its the last in sequence & the peer having it goes down.
Alternate:— Use Rarest chunk selection first policy.

B: Incoming and outgoing mail servers can't be distinguished!
Cannot differentiate spoof server with actual mail server address.

2024 MCS2002

8. [5 points]: A Distributed Hash Table (DHT) stores key-value pairs across multiple nodes arranged in a circular identifier space $[0, 2^m - 1]$. Each node is responsible for the keys in some portion of this space, and maintains pointers to its successor and predecessor nodes.

You are given the following basic primitives:

- `lookup(x)`: returns the node responsible for key x (returns the successor node if x is not present)
- `store(k, v)`: stores key–value pair (k, v) at the local node
- `get_keys(x)`: returns all key–value pairs currently stored at node x
- `del(k)`: deletes key k and its value at the local node
- `update_successor(x, v)`: node x sets node v as its successor
- `update_predecessor(x, v)`: node x sets node v as its predecessor

A new node u with identifier `u_id` wants to join the DHT. It knows one existing node c in the system and can send point-to-point messages. Describe the join protocol for node u, i.e., how does u update the DHT structure and take responsibility for its portion of the key space using the primitives listed above.

→ 1.) update_successor(u, c)
update_predecessor(c, u)

→ 2.) u needs to copy all keys from c so that it can get peer infos & File infos.
Node c → get_keys(c) then → store(k,v) to copy all keys
For each key k in set of all keys at node u →
If k is not responsible for u → del(k)

→ ~~Finally we can do~~ lookup(u) to find keys responsible for node u

→ 1) update_successor(u, c)
update_predecessor(c, u)
At u: 2) get_keys(c) → store(k,v)
At u: 3) lookup(u)
At u: 4) If key not in lookup(n) → delete → del(k)

2024MCS2002

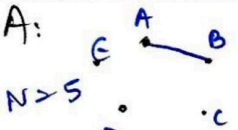
III Network Security

9. [6 points]: Briefly answer the following questions:

A. Suppose N people want to communicate with each of N- 1 other people using symmetric key encryption. All communication between any two people, i and j, is visible to all other people in this group of N, and no other person in this group should be able to decode their communication. How many keys are required in the system as a whole? Now suppose that public key encryption is used. How many keys are required in this case?

B. Suppose a TLS session employs a block cipher with cipher block chaining (CBC). True or false: The server sends to the client the IV in the clear

C. Consider a variation of the MAC algorithm where the sender sends $(m, H(m)+s)$, where $H(m)+s$ is the concatenation of $H(m)$ and s. Is this variation flawed? Why or why not?

A: 

N > 5

A → B, C, D, E 4
B → C, D, E 3
C → D, E 2
D → E 1

→ $\frac{(N-1)(N-1+1)}{2}$ = N(N-1) keys needed

symmetric key ←

→ In Public key we need a pair for everyone so 2 keys per 2 people → $\frac{N(N-1)}{2} \times 2 = N(N-1)$ keys

B: False IV can be used to unravel the entire Block chaining. so it sends encrypted.

C: Yes: If someone has the message: known plaintext attack
Then they can simply hash & remove hash from h(s)|ts & get everything

Instead do H(m+s) so that secret is also Hashed & encrypted.

2024MCS2002

So that even if there is known plain text its not possible to decipher Hash of m+Secretkey

10. [6 points]: Answer the following questions:

A. The secure email protocol we discussed in class did not use *nonces*. Is this a problem? Explain.

B. What is a *truncation* attack, and how does TLS mitigate this attack?

C. Recall that in IPSec ESP mode, the original datagram is followed by padding, a pad length field, and a next-header field. Explain the purpose of each of these fields.

A. Yes its an issue since if attacker sends previous message he can do replay attack so Nonces should be used along with Assymetric encryption in email messages so as to verify the liveliness of the email at Receiver end.

2024MCS2002

B. Truncation Attack is when attacker sends FIN message so that one or more parties close the connection & attacker gets the remaining msg so original msg is truncated.
TLS sends a "done" signal in the Record using a flag to denote closing of connection.

C. Padding ⟶ To prevent any modifications on datagram.

Padlength ⟶ To specify padding length so that we can retrieve original datagram at end.

Next-Header ⟶ Info about next expected packet headers

11. **[4 points]:** Consider the following network policy: drop any incoming TCP packet unless it belongs to a flow for which a valid outgoing SYN packet has already been observed. Explain how this policy can be implemented in an SDN network consisting of OpenFlow switches and a controller.

2024MCS2002

→ Controller will Install Flow table rules in the switches.

~~Flow~~ ~~Table~~ Suppose the flow with valid outgoing SYN Packet has SRC = 10.0.0.1
Dest = 11.0.0.1

→ Flow Table →

SRC IP	DST IP	TYPE	ACTION	PRIORITY
10.0.0.1	11.0.0.1	SYN	PORT 85 (FORWARD)	10
*	*	*	DROP	5

∴ These Flow Rules are advertised to Switches so if its valid SYN TCP Packet its priority is High so Allow & Forward to some port else If it doesn't match this Rule then next priority Rule will drop such Packet.

12. [8 points]: Recall that BGP is a path-vector protocol used for inter-domain routing. Each AS advertises its own prefixes to its neighbors, as well as forwards the announcements it has received from others after appending its own AS information in the AS-PATH. A malicious AS can launch at least two kinds of attacks:

A. It can falsely claim to originate a prefix (e.g., pretend to be the origin of IP prefix P).

B. It can modify the BGP route advertisements it has received from its peers (e.g., by adding or removing ASes in the AS-PATH).

Design an extension to BGP, called Secure BGP (S-BGP), that mitigates both of the above attacks. In your design, describe the cryptographic mechanisms and trust infrastructure used. Also, explain how your design prevents each of the two attacks mentioned above. Finally, mention **two** practical challenges of deploying your design at Internet scale.

2024 MCS2002

1.) Authentication Problem →

Each AS can sign a message using their own Private key and trust only their neighbours. This builds a web of Trust

So if an attacker pretends to be of origin IPx It also needs to advertise its digital signature to all its neighbours and since this won't match with their existing signatures for IPx The attacker would get caught.

2.) Message Integrity Problem →

~~Each AS advertises its public key to neighbors~~

Each neighbor pair does asymmetric key encryption on the AS path + secret msg ~~so~~ then the sender sends m, H(m+s) = h

Receiver decrypts h' if h=h' then message not tampered with.

Issues → 1.) Scalability as each neighbor needs to trust each other

2.) Costs performance overheads due to hashing encryption.

Rough Work

Initial = W

Final = $\frac{W}{2}$

Avg Tput = $\frac{3W}{4RTT}$

$wnd = W$

$Rate = \frac{W}{2 \times RTT} \longrightarrow \frac{W}{RTT}$

$Wi = 1MSS$

$RTT = 10ms$

$rwnd = 24KB$

$MSS = 2KB$

$\frac{12\ MSS}{24KB}$

$2KB$

$\frac{3W}{4} \longrightarrow 1$ lost

$1P \longrightarrow \frac{4}{3W}$

$cycle = \frac{W}{2}$ $\quad wnd = \frac{3W}{4}$

$\Rightarrow$ # Packets = $\frac{3W \times W}{8\ 2} = \frac{3W^2}{8}$

$1 \xrightarrow{1} 2 \xrightarrow{2} 4 \xrightarrow{3} 8 \xrightarrow{4} 16$

$\frac{3W^2}{8}$

$P = \frac{1}{\frac{3W^2}{8}}$

$W^2 = \frac{8}{3}$

$W = \sqrt{\frac{8}{3}}$