Name: ________________ **Entry Number:** ________________

Major Exam – Solution

COL334/672: Computer Networks

Sem I, 2024-25

There are 9 questions and 13 pages in this quiz booklet (including this page). There are **90 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**

1. [6 points]: Multiple choice questions. There can be more than one correct answer. +0.5 for correct, -0.5 for incorrect answers.

A. Which of the following statements are true about TCP? Select all that apply.

○ TCP uses selective acknowledgements for reliability

○ *ssthresh* is not updated when the loss is due to timeout

● A delay-based congestion control algorithm (CCA) reacts faster to congestion than a loss-based CCA

● The ECN bit used by routers to indicate congestion is located in the IP header

B. Application Layer Which of the following statements are true? Select all that apply.

● SMTP is a push-based protocol while HTTP is a pull-based protocol

○ Both HTTP and SMTP use ASCII (We will also consider if you have marked this option)

○ HTTP is stateless protocol while SMTP maintains states across two different sessions.

○ Web relies on DNS while Email does not use DNS.

C. A user requests a Web page `www.iitd.ac.in` and gets the following response:

```
HTTP/1.1 200 OK
Date: Tue, 20 Nov 2024 12:00:00 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 138
Cache-Control:  no-cache,
Connection: keep-alive
<much more document text following here (not shown)>
```

Which of the following statements are true? Select all that apply. .

● Another user downloading the same webpage in the same network has to go to the origin server even if there are web caches in the network

○ The Date: header in the HTTP response message indicates when the object in the response was last modified.

● Any additional objects on that webpage will be downloaded over the same TCP connection.

● The user can also download another webpage `www.iitd.ac.in/students` on the same TCP connection.

2. [12 points]: TCP rate control

A. Explain the difference between flow control and rate control. What is the sliding window, given that the congestion window is W_{cwnd} and the flow window is W_{flow}?

B. Suppose TCP uses AIMD for its congestion control, without slow start. Assuming the congestion window (cwnd) increases by 1 MSS every time a batch of ACKs is received, and that the round-trip time (RTT) remains approximately constant, how long does it take for the cwnd to increase from 6 MSS to 12 MSS (assuming no packet loss)? What is the average throughput (in terms of MSS and RTT) for this connection during this period?

C. Suppose that, instead of using a multiplicative decrease, TCP decreases the window size by a constant amount. Would the resulting AIAD (Additive Increase Additive Decrease) algorithm converge to an equal-share algorithm? Justify your answer.

D. Explain the following statement: "TCP CUBIC has greater RTT-fairness compared to TCP Reno" where RTT-fairness refers to how fairly a congestion control algorithm shares bandwidth among flows with different RTTs.

Solution.

A. Flow control prevents the receiver from being overwhelmed, while congestion control prevents the network from being overwhelmed.
Window = min(W_{cwnd}, W_{flow})

B. Time = 6 RTT
Average Throughput = $\frac{\text{Avg Window}}{\text{RTT}}$
$= \frac{9\,\text{MSS}}{\text{RTT}}$
We will also consider 8.5 $\frac{MSS}{RTT}$.

C. In AIAD (Additive Increase Additive Decrease), the congestion window decreases by a fixed amount regardless of the window, penalizing all flows equally. As a result, two flows – one with a larger window and the other with a smaller window – will not converge to an equal window size.

D. TCP Cubic achieves greater RTT fairness because its congestion control algorithm depends primarily on the time since the last congestion event, rather than the round-trip time (RTT) as in the case of AIMD.

3. [8 points]: Video streaming Answer the following questions:

A. List two advantages of using HTTP/TCP for streaming stored video.

B. You are designing a video streaming service and need to choose a chunk duration for the video stream. The available options are 2 seconds, 4 seconds, and 8 seconds. Explain the trade-offs involved in selecting each of these durations.

C. You are also tasked with selecting the number of bitrate levels for the video stream. You can choose between 4 bitrate levels or 16 bitrate levels. Explain the trade-offs of each option.

Solution:

A. Re-use exsiting CDN infrastructure, HTTP over TCP is middlebox friendly, TCP provides reliability

B. Shorter chunk duration gives greater responsiveness over bitrate adaptation, allowing the system to respond more quickly to changes in network conditions. Longer chunks lead to fewer HTTP requests and less metadata, reducing server and client-side load. Among the options, 4 seconds seems like a good balance (marks have been awarded for other choices, provided proper justification is given)

C. Fewer bitrate levels mean less storage overhead, while more bitrate levels provide a more fine-grained bitrate adaptation by allowing a closer match to network conditions.

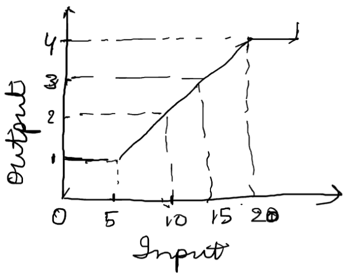

Figure 1: The horizontal and vertical lines discretize the function. Input is first mapped to the ceiling and then the corresponding horizontal line is chosen as the output. For instance f(4) = 1.

4. [8 points]: Bitrate adaptation: Consider an HTTP-based adaptive video streaming service with the following setup: Each video chunk is encoded at bitrates: {1, 2, 3, 4} Mbps and has a duration of 4 seconds. The player last requested a video chunk at 3 Mbps, which took 6 seconds to download. The current buffer occupancy is 10 seconds, with a maximum buffer capacity of 20 seconds. Answer the following:

A. Assuming a rate-based adaptation algorithm, which bitrate will be selected by the player for the next chunk? Explain.

B. Assume a buffer-based adaptation algorithm based on Figure 1. Which bitrate will be selected by the player for the next chunk? Explain.

C. Consider two players streaming the same video and sharing a bottleneck link. Assume the underlying TCP congestion control is fair. One player uses a rate-based adaptation algorithm, while the other uses a buffer-based adaptation algorithm. Will the link bandwidth be shared fairly between the two players? Explain.

Solution:

A. Rate based adaption algorithm would decide based on the last chunk throughput which is 3*4/6 =2 Mbps. The algorithm chooses a rate closest to a value which is less than the achieved throughput. So, the chosen rate = 1Mbps.

B. The buffer-based adaption would choose the rate based on current buffer value. Rate = 2 Mbps.

C. Bandwidth may not be shared evenly. The buffer-based adaptation algorithm quickly fills its buffer with low-quality chunks and eventually start downloading high-quality chunks. At this point, a rate-based algorithm may start observing lower throughput and enter into a negative feedback loop due to cross-layer interactions between TCP congestion control and bitrate adaptation.

Solution:

5. [12 points]: DHT (Distributed Hash Table): In class, we studied the PASTRY protocol for implementing a distributed hash table. Consider the CHORD protocol, which also uses consistent hashing to assign unique identifiers (IDs) to nodes and data items in a circular ID space, ranging from 0 to $2^m - 1$. Like PASTRY, in CHORD, the key k is stored at the *successor(k)* node. However, instead of maintaining random neighbors, each node i maintains a set of neighbors $2^k + i$ or its successor, $\forall k \in \{0, m-1\}\}$. To find a key, a node queries the closest preceding node in its finger table to forward the request.

Consider a system with the nodes: {8, 14, 21, 38, 50, 56} and keys: {10, 15, 25, 40, 60}. Answer the following questions:

A. Which node will store key 60? Who are the neighbors of node 8?

B. Consider when node 50 abruptly exits the system. Can you think of a mechanism to prevent loss of data, i.e., keys in this scenario?

C. When node 50 abruptly exists, how can node 8 update its neighbor list in a decentralized manner once it discovers it can no longer reach node 50?

D. Prove why CHORD can complete the lookups in $O(\log N)$.

Solution:
A. 8
Neighbours of 8 - 14, 21, 38, 50
B. Each node maintains 2 neighbour lists, one of itself and one for its predecessor. So if a node exits abruptly, its successor will add the keys in its own neighbouring list to prevent loss of data
partial marks - move key to successor before exiting
C. Replace 50 with its successor. To find the successor of 50, use key lookup(50).
D. The entries in the neighbor list double the distance between successive nodes (exponential distance). Thus, each lookup can be reduced by a factor of 2 with each hop, leading to at most O(logN) hops to find the correct node responsible for the key.

6. [12 points]: Security

Input	Output	Input	Output
000	110	100	011
001	111	101	010
010	101	110	000
011	100	111	001

A. Consider a 3-bit cipher block chaining scheme with the ciphering process shown in Table . Consider m(1) = 001 and c(0) = 010. What is the encoding for m(1)?

B. Consider RSA with p = 7 and q = 13. Let the public key be (91, 17). What is the private key? Why is it not easy to break the RSA system in practice?

C. Suppose Alice wants to send an e-mail to Bob. Bob has a public-private key pair (K_B^+, K_B^-), and Alice has Bob's certificate. But Alice does not have a public, private key pair. Alice and Bob (and the entire world) share the same hash function H(#).

Is it possible to design a scheme that provides confidentiality for sending the message from Alice to Bob? If so, show how.

Is it possible to design a scheme that allows Bob to verify that Alice created the message? If so, explain how.

Solution: **A.** Consider $m(1) = 001$ and $c(0) = 010$. We perform XOR between $m(1)$ and $c(0)$:

$$m(1) \oplus c(0) = 001 \oplus 010 = 011.$$

From the given table, the output for input 011 is 100. Therefore, the encoding for $m(1)$ is:

$$c(1) = 100.$$

Solution: **B.** The modulus n is calculated as:

$$n = p \cdot q = 7 \cdot 13 = 91.$$

The Euler's totient function $\phi(n)$ is:

$$\phi(n) = (p-1)(q-1) = 6 \cdot 12 = 72.$$

To find the private key d, we need to solve for d such that:

$$e \cdot d \equiv 1 \pmod{\phi(n)}.$$

With $e = 17$ and $\phi(n) = 72$, we solve $17 \cdot d \equiv 1 \pmod{72}$. Using the extended Euclidean algorithm, we find that:

$$d = 17.$$

Therefore, the private key is:

$$\text{Private key} = (91, 17).$$

RSA is not easy to break because it is based on the difficulty of factoring large numbers. While small values of n can be factored easily, larger values (e.g., 2048 bits) make factoring computationally infeasible using current algorithms.

Solution: **C.1** For confidentiality, Alice can encrypt the message using Bob's public key K_B^+:

$$c = \text{Encrypt}(K_B^+, m).$$

Alice sends the encrypted message c to Bob, and Bob decrypts it using his private key

$$m = \text{Decrypt}(K_B^-, c).$$

This ensures that only Bob can read the message, ensuring confidentiality.
C.2 It is not possible to ensure authenticity using a public-private key pair.

7. [12 points]: Secure DNS. The DNS system used on the Internet today is susceptible to various types of network attacks. To address these vulnerabilities, DNS over TLS (DoT) has been proposed, where all DNS queries are sent over a TLS connection operating on port 853. Answer the following questions about DNS over TLS:

A. What specific mechanisms in DNS over TLS protect against eavesdropping?

B. What mechanisms in DNS over TLS protect against impersonation attacks, i.e., an adversary from pretending to be a DNS server?

C. To implement DoT, which entities in the Internet will need to be upgraded?

D. Assuming the RTT to the local DNS resolver is 20 ms, what is the minimum time it will take to get a response from the local DNS resolver using DNS over TLS? Compare this time with that of DNS over UDP. Can you think of another transport protocol that could reduce this time?

E. Assume you are a network operator and decide to block DoT because it reduces your visibility into network traffic, affecting some network management tasks (e.g., blocking illegal websites). Write a firewall rule to block DoT traffic.

Solution:

A. DoT protects against eavesdropping by leveraging the secured tunnel, created during TLS handshake. All the DNS queries and responses in DoT are encrypted via the exchanged session key (or, derived using Master secret key and Key Derivation Function), between DoT Client and DoT Server, thereby providing confidentiality and protection against passive attacks such as eavesdropping, etc.

B. During the TLS handshake, the DoT server is required to exchange the X.509 certificate with the DoT client, thereby authenticating the DoT server and assuring protection against impersonation attacks, such as an adversary pretending to be a DNS server, and sending its ows DNS messages, etc. Refer **Section 4.4.2.** of **RFC 8446**. On the other hand, replay attacks, i.e., an adversary transmitting stale DoT server messages to DoT Client, later in point of time, can be simply avoided using nonces to preserve freshness and MACs to preserve integrity and authentication of data.

C. The DNS clients (or, the stub resolvers) must be upgraded to support DoT at port 853. The DNS servers must also be upgraded to support the DoT queries (e.g., Apart from this, the intermediate routers and middleboxes should also allow DoT communication over port 853. The complete specifications for DoT are provided in RFC 7858.

Solution:

D. **Using DNS over TLSv1.2:** The intial TCP handshake would require 1.5 RTTs [TCP SYN (client) + TCP SYN + ACK (server reply with piggybacked ACK) + TCP ACK (client)]. The client and server hello would require 1 RTT. The certificate exchange would required 1 RTT. The session key exchange would require 1 RTT. And then, the data transmission would require 1 RTT. So, overall DNS over TLS (DoT) would require 1.5+1+1+1+1 = 5.5 RTTs = 5.5 * 20 = 110 ms.

Using DNS over UDP: Since, UDP is connectionless, hence, no handshake is required and data transmission can take place directly, in unencrypted fashion. Hence, overall DNS over UDP would require 1 RTT = 20 ms.

Alternative protocols such as DNS over QUIC (DoQ) can reduce the latency compared to protocols such as DNS over TLS (DoT) and DNS over UDP (DoU).

E.

action	src address	dst address	protocol	src port	dst port
deny	*	*	TCP	853	*
deny	*	*	TCP	*	853

8. [8 points]: 4G Architecture Answer the following questions:

A. How does the base station in 4G network share spectrum with the user devices?

B. What is the role of Mobility Management Entity in a 4G network? Is it a control plane or data plane element?

C. Explain how handovers between Base stations within the same 4G network occur?

Solution.

A. 4G base stations use Orthogonal Frequency Division Multiple Access (OFDMA) which shares the spectrum by allocating users orthogonal resource blocks divided across both frequency and time domain.

B. The MME (Mobility Management Entity) in 4G LTE is responsible for signaling and control functions, including user authentication, mobility management, and session setup.

C. Refer to the slide in Lec 39.

9. [12 points]: System Design: Throughout this course, you have explored various network protocols. Many system design principles, however, repeat across protocols. Below are three such principles:

A. Hierarchy to manage scale.

B. Randomization for distributed algorithms.

C. Soft state to reduce the complexity of maintaining state consistency.

For each principle, provide two examples of network protocols that utilize it. Briefly explain how each protocol applies the principle.

Solution: Note that for each example we are also looking for explanation as asked in the question

A. DNS, IP, Internet topology, Data structure topology

B. Slotted Aloha, Exponential backoff in MAC, using log(N) random neighbors in PASTRY, randomly unchocking a peer in BitTorrent

C. TTLs/expiry values used in HTTP caching, DNS etc.