

Reasoning Qs of CN MIDI

1. Is it true that HTTP/3 no longer relies on TCP?

Answer:

Yes, this is true.

- HTTP/1.1 and HTTP/2 use **TCP** as the transport protocol.
 - HTTP/3, however, is built on **QUIC**, which runs over **UDP** instead of TCP.
 - QUIC integrates features like **multiplexing, congestion control, and TLS encryption** directly into the transport layer, solving TCP's **head-of-line blocking problem**.
-

2. Is it possible for 2 processes on the same computer to have the same port number if one is using TCP and the other is using UDP?

Answer:

Yes, it is possible.

- The operating system maintains separate namespaces for **TCP** and **UDP** sockets.
 - This means one process can use **TCP port 80** and another can use **UDP port 80** without conflict.
 - However, two different processes cannot bind to the same **protocol + port** combination (e.g., both using TCP port 80).
-

3. A user sends mail out using SMTP. At the receiver's end, can SMTP be used to retrieve that email?

Answer:

No, SMTP cannot be used to retrieve emails.

- **SMTP (Simple Mail Transfer Protocol)** is designed for **sending and forwarding emails** between mail clients and servers.

- To **retrieve** emails from the receiver's mail server, protocols like:
 - **POP3 (Post Office Protocol v3)**: downloads and often deletes emails from the server.
 - **IMAP (Internet Message Access Protocol)**: keeps emails on the server and allows synchronization across multiple devices.
 - Therefore, SMTP handles outgoing mail, while POP3/IMAP handle incoming mail.
-

4. In HTTP/3, if an HTML file is lost and needs to be resent, does it delay the delivery of CSS and JavaScript files?

Answer:

No, it does not delay other files.

- In **HTTP/2 (over TCP)**, packet loss causes **head-of-line blocking**, where one lost packet can delay all streams.
 - In **HTTP/3 (over QUIC/UDP)**, each stream is independent.
 - If the HTML file is lost and retransmitted, it does **not block CSS or JavaScript** delivery — they continue unaffected.
-

5. If HTTP uses TCP and TLS, and a hacker tries to capture the username and password, will it be original or encrypted?

Answer:

They will be **encrypted**.

- With **HTTPS (HTTP over TLS)**, data such as usernames and passwords are encrypted before being transmitted.
- A hacker sniffing the network will only see **ciphertext**, not the plaintext credentials.
- Plaintext exposure happens only if:
 - TLS is **not used** (plain HTTP).
 - TLS is **broken** (e.g., successful MITM with forged certificate).

- The **client or server is compromised** (e.g., malware, keylogger, or breached server).

6. What is the difference between Persistent and Non-Persistent HTTP connections?

Answer:

- **Non-Persistent HTTP:** A new TCP connection is created for each object (HTML, CSS, JS, image). This causes overhead due to multiple TCP handshakes.
- **Persistent HTTP:** A single TCP connection can transfer multiple objects, reducing handshake overhead and improving performance.

7. What is the difference between POP3 and IMAP?

Answer:

- **POP3:** Downloads emails to the client and usually removes them from the server. Suitable for single-device access.
- **IMAP:** Keeps emails on the server and synchronizes across devices. Suitable for multi-device access.

8. How does Traceroute work and what is the role of TTL?

Answer:

- **Traceroute** finds the path packets take to a destination by sending packets with increasing **Time-to-Live (TTL)** values.
- Each router decreases TTL by 1. When TTL reaches 0, the router sends back an **ICMP Time Exceeded** message.
- By analyzing responses, Traceroute reveals each router along the path.

9. Why was packet switching (IP) preferred over circuit switching in the Internet's design?

Answer:

- Packet switching allows **efficient resource sharing**, as network capacity is used only when data is sent.
- It is **resilient to failures** — if one path is unavailable, packets can be rerouted.
- Circuit switching reserves dedicated bandwidth, which is inefficient for bursty data traffic like the Internet.