Interview questions - TCP and UDP

1. What is the key difference between TCP and UDP?

TCP (Transmission Control Protocol) and UDP (User Datagram Protocol) are both **transport layer protocols**, but they have fundamental differences:

- **TCP** is a **connection-oriented** protocol. It establishes a reliable connection before sending data and ensures that data packets arrive **in the correct order and without loss**.
- **UDP** is a **connectionless** protocol. It **sends data without establishing a connection** and does not guarantee delivery, order, or error correction.

Feature	TCP (Transmission Control Protocol)	UDP (User Datagram Protocol)
Connection	Connection-oriented (handshake required)	Connectionless (no handshake)
Reliability	Reliable (ensures all packets arrive)	Unreliable (some packets may be lost)
Speed	Slower due to error checking & acknowledgments	Faster as it skips reliability checks
Use Cases	Web browsing, file transfer, email	Streaming, gaming, VoIP, DNS lookups

Summary: TCP prioritizes **reliability**, whereas UDP prioritizes **speed**.

---

2. When should you use TCP over UDP?

TCP should be used when **data integrity is critical**, and transmission **reliability is required**.

✅ **Common use cases for TCP:**

- **Web Browsing (HTTP, HTTPS):** Ensures that all webpage data is received without corruption.
- **File Transfers (FTP, SFTP):** Guarantees complete file downloads/uploads without missing chunks.

- **Email Communication (SMTP, IMAP, POP3):** Ensures emails are received in full, without data loss.
- **Database Queries:** Prevents corruption of data when sending/receiving requests and responses.

TCP is essential whenever **missing or corrupted data cannot be tolerated**.

---

3. When should you use UDP over TCP?

UDP is preferred when **speed is more important than reliability**, and **occasional data loss is acceptable**.

✅ **Common use cases for UDP:**

- **Video Streaming (YouTube, Netflix):** A few lost frames don't significantly affect the user experience.
- **Online Gaming (Multiplayer Games):** Low latency is crucial for real-time interactions.
- **VoIP Calls (Skype, Zoom):** Small packet losses are tolerable, but delays should be minimal.
- **DNS Lookups (Domain Name System):** Quick responses are needed, and lost packets can be retried without noticeable delay.

UDP is ideal for **real-time communication** where **low latency matters more than guaranteed delivery**.

---

4. How does TCP ensure reliability?

TCP guarantees reliable communication using the following mechanisms:

- **Three-Way Handshake:** Establishes a connection before data transmission.
- **Acknowledgments (ACKs):** Confirms receipt of packets.
- **Retransmission of Lost Packets:** If a packet is lost, TCP resends it.
- **Error Checking:** Uses checksums to detect corrupted packets.

- **Ordered Data Delivery:** Reassembles packets in the correct order before passing them to the application.

These features ensure that **TCP delivers all data accurately and in sequence**.

---

5. Why is UDP faster than TCP?

UDP is faster because:

🚀 **No connection establishment** – It does not use a handshake like TCP.
🚀 **No acknowledgments** – Data is sent without waiting for confirmation.
🚀 **No retransmission** – Lost packets are **not** resent, reducing delays.
🚀 **Lower overhead** – It has a simpler header (8 bytes vs. TCP's 20 bytes).

Since **UDP doesn't waste time on ensuring reliability**, it delivers data **as quickly as possible**.

---

6. What are the main disadvantages of TCP and UDP?

🚨 **TCP Disadvantages:**

- **Slower performance** due to reliability mechanisms.
- **Higher resource consumption** (CPU and memory usage).
- **Not suitable for real-time applications** (e.g., VoIP, live streaming).

🚨 **UDP Disadvantages:**

- **No guarantee of delivery** – Some packets may be lost.
- **No built-in error correction** – Applications must handle errors manually.
- **Unordered delivery** – Packets may arrive out of sequence.

Each protocol **has trade-offs**, and choosing between TCP and UDP **depends on the application's needs**.

---

7. How do applications handle reliability when using UDP?

Since UDP **doesn't ensure reliable delivery**, applications must implement their own mechanisms:

- **Sequence Numbers:** Ensures packets arrive in the correct order.
- **Custom Acknowledgments:** Applications can request ACKs for important data.
- **Forward Error Correction (FEC):** Adds extra data to recover lost packets.
- **Application-Level Retransmission:** If a packet is lost, the application requests it again.

For example, **online games and video streaming** use **error correction techniques** to reduce the impact of packet loss.

---

8. Can TCP and UDP be used together?

Yes! Some applications **combine TCP and UDP** to optimize performance.

✅ **Examples:**

- **Online Video Streaming (Netflix, YouTube):** Uses UDP for fast video delivery but TCP for control messages (pause, seek, etc.).
- **Online Games:** UDP for real-time actions (player movement) + TCP for non-time-sensitive data (chat, game stats).
- **VoIP (Skype, Zoom):** UDP for low-latency voice data + TCP for connection setup.

By using **both protocols strategically**, applications balance **speed and reliability**.

---

9. Which protocol does DNS use, TCP or UDP?

DNS (Domain Name System) **primarily uses UDP** because:

- **Speed is crucial** – A DNS query must be resolved quickly.
- **Small packet size** – DNS requests/responses fit within UDP's limits.

However, **DNS can use TCP** for **larger responses**, such as when retrieving a full DNS zone transfer.

---

10. How do firewalls handle TCP vs. UDP traffic?

Firewalls manage TCP and UDP traffic differently:

- **TCP traffic** is easier to monitor since it has a clear connection setup (handshake) and termination.

- **UDP traffic** is harder to track because it's connectionless, making it more vulnerable to abuse (e.g., DDoS attacks).

To secure UDP traffic, firewalls use **rate limiting, deep packet inspection, and UDP filtering**.