Computer Networks enable communication between devices through a set of rules called protocols. Among these, **TCP (Transmission Control Protocol)** and **UDP (User Datagram Protocol)** are two of the most widely used transport-layer protocols defined in the TCP/IP model.

TCP is a connection-oriented protocol, meaning a connection must be established before data can be exchanged. It ensures **reliable and ordered delivery** through mechanisms like acknowledgment (ACK), retransmission, flow control, and congestion control. TCP breaks large messages into segments, assigns sequence numbers, and uses ACKs to confirm receipt. If a segment is lost, TCP retransmits it automatically. This reliability makes TCP ideal for applications like web browsing (HTTP/HTTPS), email, and file transfers, where accuracy is more important than speed.

UDP, in contrast, is a connectionless protocol. It sends datagrams without establishing a connection and does not guarantee delivery, ordering, or error correction. Since UDP eliminates overhead, it is extremely fast and suitable for real-time applications like gaming, live video streaming, and VoIP, where speed and low latency are more important than perfect accuracy. For example, in a video call, losing a few packets is acceptable compared to waiting for delayed retransmissions.

Reliable data transfer depends heavily on the characteristics of these protocols. TCP provides reliability using several techniques:

- **Three-Way Handshake:**
  Ensures both sender and receiver are ready for communication before data transfer begins.

- **Flow Control (Sliding Window):**
  Prevents the sender from overwhelming the receiver by adjusting the sending rate.

- **Congestion Control (AIMD, Slow Start):**
  Adjusts sending rate based on network load to avoid congestion collapse.

- **Checksums:**
  Detects errors in transmitted segments.

UDP, though unreliable by design, can still achieve reliability when necessary through application-level mechanisms, such as added sequence numbers or manual acknowledgments. Some modern protocols like QUIC use UDP as a base but add reliability features for improved performance.

Understanding TCP vs UDP helps developers choose the right protocol for different scenarios. TCP suits data-sensitive applications, whereas UDP is the better choice for latency-critical tasks.