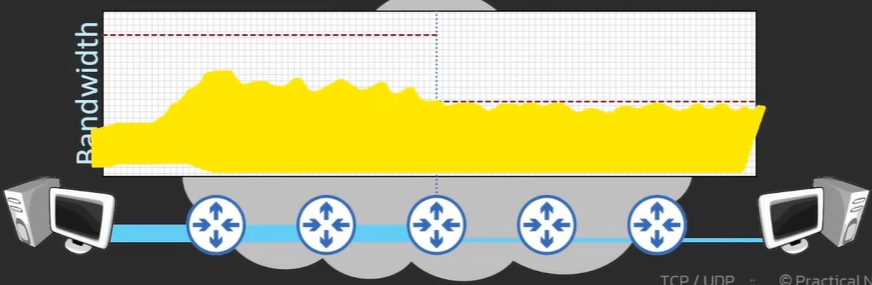
**TCP vs UDP**

**TCP (Transmission Control Protocol)**

* **Reliability**:
  + Ensures reliable data delivery through acknowledgments.
  + Data is delivered in the correct order using sequence numbers and checksums.
  + Detects and retransmits lost packets using timeouts and retransmission queues.
* **Flow Control**:
  + Adjusts the transmission rate to maximize available bandwidth while avoiding congestion.



* **Overhead**:
  + TCP includes significant overhead due to features like error checking, sequence tracking, and acknowledgments.
* **Use Cases**:
  + File transfers (FTP) where reliable delivery is crucial.
  + Email (SMTP).
  + Applications requiring ordered and error-free data transmission.
* **Connection Establishment**:
  + Uses a 3-way handshake to establish a reliable connection:
  + **SYN**: Client sends a synchronization request.
  + **SYN-ACK**: Server acknowledges and synchronizes.
  + **ACK**: Client acknowledges the server.
* **Handling Lost and Out-of-Order Packets**:
  + Lost packets are detected using timers and retransmission queues.
  + Out-of-order packets are reordered using sequence numbers.
* **Data Integrity**:
  + Ensures ordered delivery and correct sequence of data for accurate application processing.

**UDP (User Datagram Protocol)**

* **Unreliability**:
  + No acknowledgments or retransmissions.
  + Packets may be lost or arrive out of order.
* **Flow Control**:
  + Does not include flow control; data is transmitted as fast as possible.
* **Overhead**:
  + Minimal compared to TCP, making it more efficient for lightweight and real-time applications.
* **Use Cases**:
  + Real-time audio and video streaming.
  + Online gaming.
  + Applications with built-in delivery confirmation mechanisms (TFTP).
* **Speed Considerations**:
  + UDP is not inherently faster; its perceived speed comes from reduced overhead and lack of error-checking mechanisms.

A screenshot of a computer

Description automatically generated

A close-up of a number

Description automatically generated

**Packet Order**

* **Why Packet Order Matters**:
  + Many applications rely on ordered data to function correctly (file transfers, database transactions).
* **Consequences of Out-of-Order Packets**:
  + Real-time applications may experience jitter, lag, or degraded performance.
  + TCP retransmits and delays processing until all packets are in order.
* **How TCP Ensures Order**:
  + Sequence numbers assign a unique identifier to each packet.
  + Acknowledgment numbers confirm receipt of data.

A cloud with arrows and arrows

Description automatically generated

**Packet Loss**

* **When Packets Are Lost**:
  + Network congestion.
  + Faulty hardware or unstable connections.
* **How TCP Handles Packet Loss**:
  + Retransmits packets when ACK is not received within a timeout period.
* **Impact of Packet Loss on UDP**:
  + Lost packets are not retransmitted, leading to gaps in data streams.

**Overhead and Handshake in TCP vs. UDP**

* **TCP Overhead**:
  + Higher due to error checking, acknowledgments, and flow control.
  + 3-way handshake adds additional latency during connection setup.
* **UDP Overhead**:
  + Minimal.
* **Handshake Comparison**:
  + TCP requires a 3-way handshake before data transfer.
  + UDP sends data directly without establishing a connection.

**Speed Comparison**

* **TCP**:
  + Slower due to features ensuring reliability and ordered delivery.
  + Suitable for applications where accuracy is more important than speed.
* **UDP**:
  + Faster for lightweight and real-time applications due to reduced overhead.
  + Best suited for time-sensitive tasks like streaming and gaming.

**Why TCP is More Widely Used**

* **Reliability**:
  + Ensures data is delivered without errors.
  + Essential for critical applications like file transfers and email.
* **Ordered Delivery**
* **Error Checking**

**Key Takeaways**

* **TCP**: Reliable, ordered, and error-checked; used where data integrity is critical.
* **UDP**: Lightweight, fast, and suitable for real-time applications where occasional data loss is acceptable.
* **Choosing the Right Protocol**:
  + Use TCP for accuracy and reliability.
  + Use UDP for speed and real-time performance.