# TCP vs UDP - Complete Study Notes

## Overview

Both TCP (Transmission Control Protocol) and UDP (User Datagram Protocol) are **transport layer protocols** used for data transmission, but they have fundamental differences in their approach and characteristics.

---

## Key Differences Between TCP and UDP

### 1. Connection Type

#### TCP - Connection Oriented

- Must establish a connection before data transmission
- Data transmission starts only after connection is established
- Similar to a telephone network - connection must be made first

#### UDP - Connectionless

- No connection establishment required
- Starts data transmission immediately upon receiving data from application layer
- Doesn't care about establishing connections

---

### 2. Reliability

#### TCP - Highly Reliable

- Provides guarantee that all data bytes will definitely reach the receiver
- Uses connection-based reliability (like telephone network)
- If data doesn't reach once, uses **re-transmission**
- Has **acknowledgment system**:
  - Receiver sends acknowledgment when packet is received
  - Can use negative acknowledgments
  - If no acknowledgment received, sender retransmits

#### UDP - Less Reliable

- No guarantee of data delivery
- Lost packets are simply lost - no retransmission

- No acknowledgment system
- "Fire and forget" approach

---

## 3. Packet Ordering

**TCP - Maintains Order**

- Packets always reach in the correct sequence
- Example: If sending packets P1, P2, P3 → they arrive as P1, P2, P3
- Uses single network path, ensuring ordered delivery
- Like telephone conversation - first line reaches first, then second, then third

**UDP - No Ordering Guarantee**

- Packets can arrive in any order
- Example: Sending P1, P2, P3 → might arrive as P2, P1, P3 or any sequence
- Uses different network paths for different packets
- Since packets follow different networks, they may reach out of order

---

## 4. Error Control

**TCP - Mandatory Error Control**

- **Checksum is mandatory**
- Must calculate checksum before sending
- Sender includes checksum with data
- Receiver calculates checksum and matches with sender's checksum
- If checksums match → no error detected

**UDP - Optional Error Control**

- **Checksum is optional**
- UDP header has 16-bit checksum field
- Can send data without checksum by putting all zeros (0000000000000000) in checksum field
- If you want error detection, can include checksum

---

## 5. Transmission Speed

**TCP - Relatively Slower**

- Not inherently slow, but slower compared to UDP
- Must establish connection first
- Follows single network path
- If congestion occurs on that path, speed decreases
- Additional overhead due to reliability mechanisms

**UDP - Faster Transmission**

- No connection establishment delay
- Starts transmission immediately upon receiving data from application layer
- Uses multiple network paths (different routes)
- If one route is congested, other routes remain available
- Packets reach faster due to multiple path availability

---

## 6. Overhead

**Overhead Definition:**

- **Payload** = Pure data from application layer (the actual message)
- **Overhead** = Extra information (headers) needed to send the payload
- For M amount of data, need to add X amount of header information

**TCP - Higher Overhead**

- Header size: **20-60 bytes**
- Minimum header: 20 bytes (160 bits)
- Maximum header: 60 bytes
- Contains many fields: flow control, error control, flags, options, etc.

**UDP - Lower Overhead**

- Header size: **Fixed 8 bytes only**
- Regardless of data size, always 8 bytes of header
- Much simpler header structure
- **Comparison**: TCP minimum (20B) > UDP fixed (8B) → TCP has more overhead

---

## 7. Flow Control and Congestion Control

**TCP - Has Both Controls**

*Flow Control:*

- Considers receiver's capacity to accept data
- "How much data can the receiver handle?"

*Congestion Control:*

- Considers network capacity
- "What is the network's current load capacity?"
- Uses algorithms like **AIMD (Additive Increase Multiplicative Decrease)**
- Cannot directly measure network load, so uses algorithms to estimate

*Handling Congestion:*

- If router receives too much data and becomes overloaded
- Your packet may get discarded due to buffer overflow
- TCP detects this and performs **re-transmission**
- Sends the same packet again

**UDP - No Flow/Congestion Control**

- No algorithms for flow or congestion control
- If packet is lost due to congestion → it's permanently lost
- No retransmission mechanism
- Doesn't care about receiver capacity or network load

---

## Real-World Applications and Examples

### TCP Applications (Reliability Required)

**1. HTTP (HyperText Transfer Protocol)**

- Used for webpage transmission
- HTTP itself is not reliable
- Always uses TCP to achieve reliability
- Ensures no data loss during webpage loading

**2. FTP (File Transfer Protocol)**

- Used for file transfers to/from servers
- **Critical requirement**: Every single byte must reach destination
- Example: If downloading a file and even 1 byte is lost → file becomes corrupted and won't open

- File integrity is crucial → TCP's reliability is essential

**3. YouTube (High Definition)**

- HD video streaming uses TCP
- Ensures all video data reaches user
- User gets complete data as requested
- May cause slower buffering due to TCP's reliability mechanisms

### UDP Applications (Speed Required)

**1. DNS (Domain Name Server)**

- When finding IP address from domain name
- Send domain name → receive IP address
- Speed is more important than perfect reliability for quick lookups

**2. BOOTP, DHCP, RIP**

- **BOOTP**: Bootstrap Protocol
- **DHCP**: Dynamic Host Configuration Protocol
- **RIP**: Routing Information Protocol
- All require fast transmission rather than guaranteed delivery

**3. Live Video Conferencing (Skype, etc.)**

- **Requirement**: Screen should not freeze/stop
- **Acceptable**: Little blur or quality reduction
- **Not acceptable**: Complete screen loss
- Speed is prioritized over perfect quality
- Better to have slightly blurred real-time video than perfect but delayed video

**4. YouTube (Standard Definition)**

- Lower quality video streaming can use UDP
- Acceptable to lose some data for faster streaming
- Trade-off between quality and speed

---

## Decision Criteria: When to Use Which?

### Use TCP When:

- **Reliability is crucial**
- Data integrity must be maintained
- Complete data delivery is required
- Examples: File downloads, web browsing, email, database transactions

### Use UDP When:

- **Speed is the priority**
- Real-time communication needed
- Some data loss is acceptable
- Examples: Live streaming, online gaming, DNS queries, live video calls

---

## Summary Table

| Feature | TCP | UDP |
|---|---|---|
| Connection | Connection-oriented | Connectionless |
| Reliability | High (guaranteed delivery) | Low (no guarantee) |
| Ordering | Maintains packet order | No ordering guarantee |
| Error Control | Mandatory checksum | Optional checksum |
| Speed | Slower (relative to UDP) | Faster |
| Overhead | High (20-60 bytes header) | Low (8 bytes header) |
| Flow Control | Yes | No |
| Congestion Control | Yes (AIMD algorithm) | No |
| Retransmission | Yes | No |
| Acknowledgments | Yes | No |

---

## Key Analogies to Remember

1. **TCP = Telephone Network**: Must establish connection before communication, reliable delivery, ordered conversation
2. **UDP = Postal System**: Send letters immediately, no guarantee of delivery order or even delivery, but faster to send
3. **TCP Overhead = Sending a package with tracking, insurance, signature confirmation** (more overhead but guaranteed delivery)
4. **UDP Overhead = Sending a postcard** (minimal overhead, quick delivery, but no guarantees)