UDP Applications - Complete Study Notes

Introduction to UDP vs TCP

UDP (User Datagram Protocol) is a connectionless protocol that:

- Does not establish any connection before sending data
- Directly sends data without prior setup
- Contrasts with TCP which always transfers data after establishing a connection

Problems with UDP

1. **Unreliable** - Due to no connection establishment
2. **Packets not synchronized** - No synchronization fields in header
3. **Out-of-order delivery** - Packets can reach destination in wrong sequence
4. **Packet loss** - Some packets may get lost during transmission

---

Major Applications of UDP

1. Query Response Protocol

Definition: Applications that work on a query-response model - one request, one reply.

Characteristics:

- No connection establishment required
- Simple request-response pattern
- Minimal overhead

Example 1: DNS (Domain Name Server)

Scenario: When you type a domain name in your browser

- You remember domain names (like google.com), not IP addresses (which are hard to remember)
- Process:
  1. You type "google.com" in browser
  2. Domain name goes to DNS server first
  3. DNS server responds with corresponding IP address
  4. No connection building required

Real-world analogy:

- Like asking a stranger for directions in a market
- You don't establish a relationship first ("Where are you from?", "Have you seen that shop?")
- You simply ask: "What's the address of that shop?"
- They simply reply: "Go to that particular street/area"
- One request → One reply

Example 2: DHCP (Dynamic Host Configuration Protocol)

Process:

- You have MAC address but no IP address
- You request: "My MAC address is this, what's my IP address?"
- Server responds with your IP address
- No prior connection needed

---

2. Speed-Critical Applications

Key Point: Where high speed is required, UDP is preferred due to less overhead.

Why UDP is Faster:

- **Less overhead**: UDP header has only 4 fields (checksum is optional)
- **TCP overhead**: Many fields in TCP header
- **Average UDP overhead**: Only 3 mandatory fields vs many in TCP

Applications Requiring Speed:

Online Games

Problem with TCP:

- Uses Flow Control and Congestion Control
- Uses AIMD (Additive Increase Multiplicative Decrease)
  - Speed increases exponentially (2^n)
  - After threshold, speed drops suddenly
- If single packet lost → retransmission → speed reduction

Gaming Scenario:

- Game pauses for 1-2 seconds due to TCP retransmission
- Other players might kill your character during pause
- Player gets eliminated due to speed issues

- Gaming enjoyment ruined

UDP Solution:

- Maintains constant speed
- No congestion control methods that suddenly reduce speed
- Continuous gameplay without interruptions

Voice Over IP (VoIP)

- Speed matters for real-time communication
- Constant speed required for smooth conversation
- No sudden speed drops acceptable

---

3. Broadcasting and Multicasting

Definition: One sender communicating with multiple receivers simultaneously.

Example: RIP (Routing Information Protocol)

Background:

- Network layer protocol
- Uses distance vector routing
- Every 30 seconds: each node shares its routing table with neighboring nodes

TCP Problems for Broadcasting:

- Would need to establish connection with each receiver
- Multiple connections: connection + connection + connection...
- Each connection requires:
  - Buffer reservation
  - Bandwidth reservation
  - Memory reservation

Resource Problem Example:

- If 1 client needs 1MB buffer
- For 1000 clients = 1000MB space required
- All reserved BEFORE communication starts
- Too much formality just to start communication

UDP Solution:

- Simply broadcast/multicast message to group
- No buffer reservation needed
- No bandwidth/memory reservation required
- Direct data transmission to multiple recipients

Types:

- **Multicasting**: Sending to similar group of people
- **Broadcasting**: Sending to all people on the network

---

4. Continuous Streaming

Requirement: Packets must arrive at continuous speed without interruptions.

Applications:

YouTube Video Streaming

Normal Mode (UDP):

- Continuous streaming maintained
- Speed prioritized over clarity
- Video keeps playing smoothly
- Streaming continues ahead, video plays behind

HD Mode (TCP):

- Every pixel matters for high definition
- If single pixel (packet) drops → TCP stops
- Retransmits that specific pixel
- Video resumes only after retransmission
- **Result**: Slower speed in HD mode due to TCP reliability

Skype Video Chat

UDP Benefits:

- Continuous video chat maintained
- Prevents sudden black screens
- Prevents disappearing of chat partner

- Maintains speed and continuity over clarity

TCP Problems:

- Video chat suddenly shows black screen
- Person disappears from screen
- Creates worry about broken connection
- Interrupts communication flow

UDP Philosophy: "Clarity may be low, but speed and continuity are maintained"

---

5. Stateless Protocol Advantage

Protocol Types:

- **Stateful**: Saves connection values and maintains state
- **Stateless**: Does not store connection information

Stateful Example: Amazon/Flipkart

What they track:

- Pages you clicked
- Items selected
- Cart contents
- Color preferences checked
- Size preferences checked
- Payment page visits
- Complete shopping journey

Overhead: High due to state maintenance

Stateless Example: HTTP Protocol

Characteristics:

- You access thousands of web servers
- HTTP doesn't store all that data
- Each request treated independently
- No connection state maintained

UDP Advantage:

- Completely stateless
- Maintains overhead as low as possible
- No connection state tracking
- Minimal resource usage

Summary of UDP Applications

Application Type	Examples	Why UDP?	Key Benefit
Query-Response	DNS, DHCP	No connection needed	Low overhead
Speed-Critical	Online Games, VoIP	Constant speed required	No sudden speed drops
Broadcasting	RIP Protocol	Multiple receivers	No multiple connections
Streaming	YouTube, Skype	Continuous flow needed	Uninterrupted streaming
Stateless	HTTP-like protocols	No state maintenance	Minimal resource usage

Key Takeaways

1. **UDP trades reliability for speed** - accepts some data loss for better performance
2. **Connection establishment overhead** makes TCP unsuitable for simple request-response scenarios
3. **Constant speed requirement** in real-time applications favors UDP over TCP's variable speed
4. **Resource efficiency** makes UDP ideal for broadcasting to multiple recipients
5. **Stateless nature** reduces overhead and improves scalability

UDP is chosen when **speed, simplicity, and low overhead** are more important than **guaranteed delivery and reliability**.