Application Layer Protocols - Complete Study Notes

Overview

From competitive exams perspective, the most important application layer protocols are:

- **DNS** (Domain Name System)
- **HTTP** (Hypertext Transfer Protocol)
- **FTP** (File Transfer Protocol)
- **SMTP** (Simple Mail Transfer Protocol)
- **POP** (Post Office Protocol)

Key Point: Each protocol could have a complete book written about it, but exams typically focus on:

1. Port numbers
2. Which applications use which ports
3. Which transport layer protocol is used
4. Basic characteristics and functionality

HTTP Protocol (Hypertext Transfer Protocol)

Basic Information

- **Port Number**: 80
- **Purpose**: Transfer and fetch webpages on WWW (World Wide Web)
- **Transport Layer**: Uses TCP for reliability

Real-life Usage Examples

- Opening any browser and accessing webpages
- Accessing Google, Facebook, or any website
- You can see HTTP protocol usage at the beginning of URLs

Key Characteristics

1. Reliability

- **HTTP itself is NOT reliable**
- It doesn't have built-in functionality for data loss handling
- **Achieves reliability through TCP** in the transport layer
- TCP handles retransmission if data is lost

2. Inband Protocol

- **Definition**: Commands and data travel through the same port
- Only **one port number (80)** is used
- Both control commands and actual data use port 80
- **Contrast with FTP**: FTP uses separate ports for commands and data

3. Stateless Nature

- **HTTP is stateless** by default
- Doesn't store user information (metadata like: which user accessed what, for how long, etc.)
- **Reason**: Millions/billions of requests go through HTTP. Storing all information would cause significant delays

Cookies Concept (Making HTTP Stateful)

Companies like Amazon, Flipkart use cookies to make HTTP appear stateful:

Example Scenario - Amazon Shopping:

1. You check a product on Amazon
2. You close the browser
3. Information gets saved in cookies (client-side storage)
4. Cookies store: username, password, address, previously accessed webpages, browsing history
5. Next time you open Amazon, same information is shown
6. User doesn't need to refill information repeatedly

Result: While HTTP remains stateless, companies create stateful experience through cookies

4. Connection Types

Non-Persistent Connection

Examples: IRCTC, HDFC Banking, ICICI Banking

IRCTC Scenario:

1. You open IRCTC website
2. Access reservation page → New connection created
3. Check payment options → Another new connection created
4. Check train information → Another new connection created
5. **Each new webpage = New connection**

6. Close IRCTC → **All connections immediately lost**
7. Reopen IRCTC → Must fill username and password again

Banking Example:

- Fill information for online transaction
- Stop transaction midway and close browser
- Reopen → Must fill all information from start

Persistent Connection

Example: Gmail

Gmail Scenario:

1. Fill username and password
2. Close browser (not logout)
3. Reopen after one month or one year
4. **Mails open directly** - no need to re-enter credentials
5. Connection remains until you explicitly logout

HTTP Commands

Command	Purpose	Example Usage
HEAD	Access metadata only (not actual content)	Getting information about a webpage without downloading it
GET	Access complete webpage	Most widely used - loading any webpage
POST	Submit information to server	Filling and submitting forms
PUT	Upload information to server	Similar to POST, for uploading data
DELETE	Delete information from server	Removing data from server
CONNECT	Establish connection with server	Used with HTTPS for SSL connections
TRACE	Echo incoming requests	Check/debug incoming requests
OPTION	Inquire about available options	Check what operations are supported

HTTPS vs HTTP

- **HTTPS**: Works on Secured Socket Layer (SSL)
- **Security Feature**: Checks username/password authentication repeatedly with each request
- **Purpose**: Minimize hacking, traffic hijacking, session hijacking attempts
- **Modern Default**: Most websites redirect HTTP to HTTPS automatically
    - Example: Type "http://google.com" → Automatically redirects to "https://google.com"

Practical Example Flow

1. **Client Request**: Browser requests webpage from server
2. **Server Response**: Server provides requested webpage
3. **Protocol Used**: All communication happens through HTTP protocol
4. **Multiple Servers**: If server needs data from other servers, those also use HTTP protocol

---

FTP Protocol (File Transfer Protocol)

Basic Information

- **Port Numbers**:
  - **Port 21**: Control commands (connection establishment)
  - **Port 20**: Data transfer
- **Purpose**: Transfer files to remote locations
- **Transport Layer**: Uses TCP for reliability

Real-life Applications

Software Examples: FileZilla, Techtia

Website Hosting Scenario:

1. You're sitting in Chandigarh
2. Need to host a website
3. Must send website files to server
4. Use FTP protocol through applications like FileZilla
5. **Interface**: Split screen showing your machine and server machine
6. **Process**: Drag files from your system → Files transfer to server system

Key Characteristics

1. Two-Port System (Non-Inband)

- **Port 21**: Control commands
  - Establish connection
  - Send control information
- **Port 20**: Data transfer
  - Actual file transfer
- **Advantage**: Control and data can work in parallel

2. Connection Types

- **Control Connection**: **Persistent**
  - Remains established even during delays
  - Even if you don't send data for some time
  - Connection stays active
- **Data Connection**: **Non-Persistent**
  - Terminates after each file transfer
  - Must re-establish for each new transfer
  - **Reason**: Security purposes

3. Reliability

- **FTP is reliable** - uses TCP protocol
- **Why reliability is crucial**: Files being transferred
- **Critical Requirement**: Even single byte loss can prevent file from opening
- **File integrity**: File opens only when ALL bytes are received correctly

4. Stateful Protocol

- **Stores complete information** about file transfers
- **Tracked Data**:
  - Which file was transferred
  - When it was transferred
  - File size
  - User information
- **Purpose**: Maintain history for future reference
- **Analogy**: Like maintaining a detailed log/history of all activities

FTP vs HTTP Comparison

Feature	HTTP	FTP
Inband/Outband	Inband (single port)	Non-inband (two ports)
Reliability	Uses TCP for reliability	Uses TCP for reliability
State	Stateless	Stateful
Connection Persistence	Varies (persistent/non-persistent)	Control: Persistent, Data: Non-persistent

---

SMTP & POP Protocols

SMTP (Simple Mail Transfer Protocol)

Basic Information

- **Port Number**: 25
- **Purpose**: **Push mail** (sending emails)
- **Key Feature**: **Asynchronous communication**

Asynchronous vs Synchronous

FTP (Synchronous):

- Both users must be active simultaneously for file transfer
- Real-time requirement for both sender and receiver

SMTP (Asynchronous):

- **Advantage**: Recipient doesn't need to be online when mail is sent
- **Process**: Send mail anytime → Recipient reads when they come online
- **Example**: Send email at 2 AM → Recipient reads at 9 AM next day

Mail System Architecture

Components Overview

1. **User → Mail Client → MTA → Another MTA → Mail Client → User**

Detailed Component Explanation

Mail Account

- **Definition**: Space provided by email service to store data
- **Examples**:
  - Google provides 15GB or 25GB space
  - Yahoo provides similar space allocations
- **Usage**: Store emails, attachments, etc.

Mail Client

- **Function**: Interface between user and mail system
- **Examples**: Outlook, web browsers, mobile apps
- **Purpose**:
  - Fetch mail from user (compose interface)
  - Display received mails to user

MTA (Mail Transfer Agent)

- **Function**: Server-side component that handles mail routing
- **Examples**:
  - Gmail.com has its own MTA
  - Yahoo.com has its own MTA
- **Purpose**: Route emails between different email services

Complete Mail Flow Example

Scenario: Gmail user sending email to Yahoo user

Sending Process (SMTP):

1. **User** composes email in Gmail web interface
2. **Mail Client** (web browser) captures the email
3. **Gmail's MTA** receives email from mail client
4. **Gmail's MTA** transfers email to **Yahoo's MTA**
5. Email stored in Yahoo's mail server

Receiving Process (POP):

1. Yahoo user opens email client
2. **POP protocol** retrieves email from Yahoo's MTA
3. **Mail Client** displays email to user
4. User sees the received email

POP (Post Office Protocol)

Basic Information

- **Protocol Version**: POP3 (most commonly used)
- **Purpose**: **Pop/withdraw mail** from mail server
- **Port Numbers**:
  - **Port 110**: Default POP3 (non-encrypted)
  - **Port 995**: Secure POP3 (with encryption)

Security Options

- **Non-encrypted (Port 110)**: Default, basic connection
- **Encrypted (Port 995)**: Secure connection with encryption
- **Usage**: Choose based on security requirements

Modern Email Enhancements

MIME Protocol

- **Full Form**: Multipurpose Internet Mail Extensions
- **Purpose**: Handle multimedia content in emails
- **Supported Content**:
  - Images
  - Videos
  - Audio files
  - Various file attachments
- **Integration**: Used by default with modern email systems

Summary Table - Quick Reference

Protocol	Port(s)	Transport	Reliable	State	Connection Type	Primary Use
HTTP	80	TCP	Via TCP	Stateless	Persistent/Non-persistent	Web browsing
HTTPS	443	TCP	Via TCP	Stateless	Persistent/Non-persistent	Secure web browsing
FTP	21(control), 20(data)	TCP	Yes	Stateful	Control: Persistent, Data: Non-persistent	File transfer
SMTP	25	TCP	Via TCP	-	-	Send emails
POP3	110(normal), 995(secure)	TCP	Via TCP	-	-	Receive emails

Exam-Important Points Summary

Most Frequently Asked Questions

1. **Port numbers** of each protocol
2. **Which applications** use which ports
3. **Transport layer protocol** used (mostly TCP)
4. **Basic characteristics** (reliable/unreliable, stateful/stateless)

Key Differences to Remember

- **HTTP**: Stateless, inband, uses cookies for user experience
- **FTP**: Stateful, non-inband (two ports), file transfer specific

- **SMTP/POP**: Asynchronous email system, multimedia support via MIME

Real-world Applications

- **Banking websites**: Non-persistent HTTP connections
- **Gmail/Social media**: Persistent HTTP connections
- **Website hosting**: FTP for file uploads
- **Email systems**: SMTP for sending, POP for receiving

Note: This covers all essential points for competitive exams like NET, GATE, and other computer science examinations. For deeper understanding, individual protocols can be studied in more detail, but this level is sufficient for most competitive exam requirements.