8. Protocols

Protocols are likened to rulebooks that define how data is transmitted, and receives accross network. They are essential for maintaining order and standardization in network communication.

TCP and UDP:- Data delivery methods

TCP (Transmission control protocol):- Ensure reliable data delivery by confirming data arrival and order, like sending a registered letter.

UDP (User Datagram Protocol):- Focuses on speed, sending data without confirmation, akin to droped a post card in the email.

Most common protocols

OSI Layer	Protocols
Layer 7: Application	HTTP, HTTPS, FTP, SMTP, DNS
Layer 6: Presentation	SSL/TLS, JPEG, MPEG, GIF
Layer 5: Session	NetBIOS, RPC, PPTP
Layer 4: Transport	TCP, UDP, SCTP
Layer 3: Network	IP, ICMP, IGMP, ARP
Layer 2: Data Link	Ethernet, PPP, Frame Relay
Layer 1: Physical	Ethernet (cabling), DSL, USB

TCP/IP Layer	Protocols
Layer 4: Application	HTTP, HTTPS, FTP, SMTP, DNS
Layer 3: Transport	TCP, UDP, SCTP
Layer 2: Internet	IP (IPv4, IPv6), ICMP, ARP
Layer 1: Network Interface	Ethernet, Wi-Fi (802.11), PPP

Introduction to protocol:- Protocols are like the secret rules for procedures for data transmission between electronic devices, ensuring smooth communication the traffic flaws of internet highways.

HTTP and HTTPS:- Web browsing essentials

HTTP(Hyper text transfer protocol:- Foundation for web data exchange, governing how files (text, images, videos are transmitted online.)

HTTPS(https secure):- A secure version of HTTP, encrypting communication between browsers and websites, crucial for secure data transmission like credit card into.

FTP:- Efficient file transfer

FTP (File Transfer Protocol:- A workhorse protocol for moving large files or group of files between.

SMTP (simple mail transfer protocol):- Sends emails from send to recipent's email server.

IMAP (Internet Message access protocol):- Retreives email from the email server, allowing access from multiple devices.

POP(Post office protocol):- Retrieves email from the email server but typically download them to a single device, removing them from the server.

TCP protocol (transmission control protocal)

TCP Connection Establishment (Three-Way Handshake)

1. **SYN**: The client sends a SYN (synchronize) packet to the server to initiate a connection.
2. **SYN-ACK**: The server responds with a SYN-ACK (synchronize-acknowledge) packet to acknowledge the receipt of the SYN.
3. **ACK**: The client sends an ACK (acknowledge) packet back to the server, completing the handshake and establishing the connection.

Error detection is done by `ARQ`

> ARQ (Automatic repeat request) : When a packet is lost receiver don't send a ACK to sender and sender get to know something want wrong and send the packet again.

Termination of Connection

TCP connections are terminated gracefully using a four-way handshake:

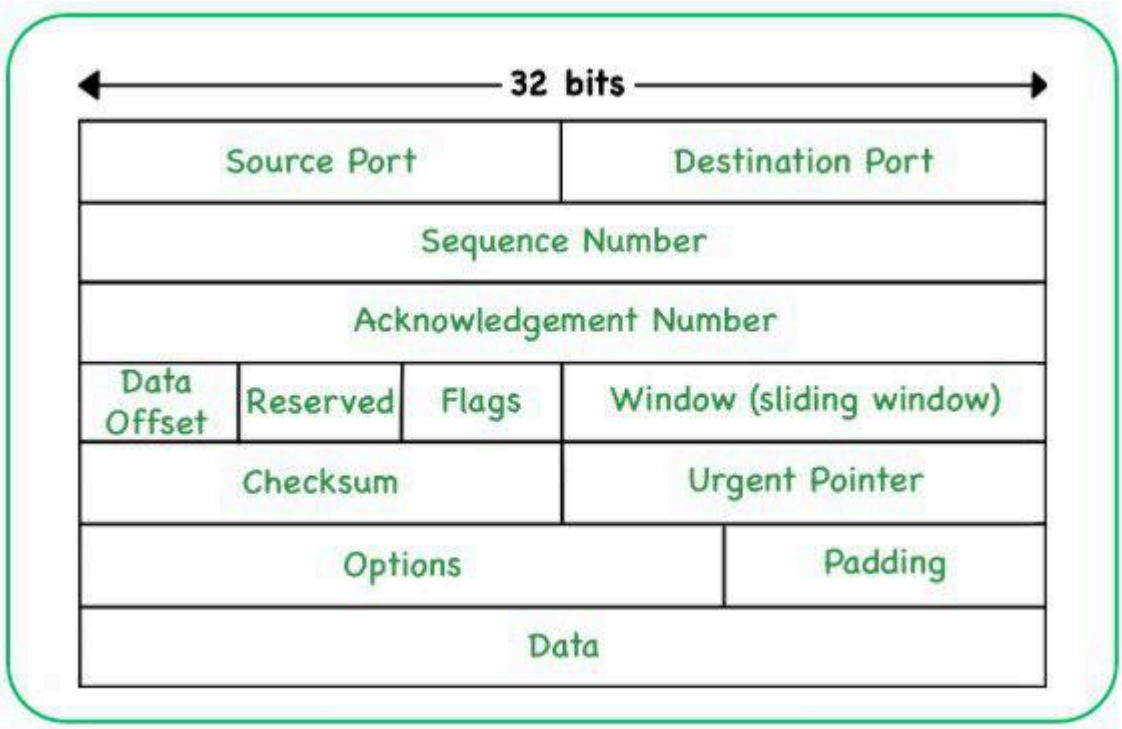
1. **FIN**: One side (e.g., client) sends a FIN (finish) packet to terminate the connection.
2. **ACK**: The other side acknowledges the FIN with an ACK.

3. **FIN**: The second side then sends its own FIN to terminate the connection.
4. **ACK**: The first side acknowledges this FIN, and the connection is fully closed.

TCP Header Structure

The TCP header is typically between 20 byte to 60 bytes

Field	Size (Bits)	Description
Source **Port**	16	The port number of the sender
Destination **Port**	16	The port number of the receiver
Sequence Number	32	The sequence number of the first byte of data
Acknowledgment Number	32	The next sequence number that the sender expects
Data Offset	4	The size of the TCP header in 32-bit words
Reserved	3	Reserved for future use
Flags	9	Control flags (e.g., URG, ACK, PSH, RST, SYN, FIN)
Window Size	16	Size of the sender's receive window (flow control)
Checksum	16	Used for error-checking of the header and data
Urgent Pointer	16	Indicates if there is urgent data
Options (if any)	Variable	Additional options (e.g., Maximum Segment Size)
Padding	Variable	Padding to ensure the header is a multiple of 32 bits



IP protocol

IPv4 Header Format

Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Version				IHL				DSCP						ECN		Total Length															
4	32	Identification																Flags			Fragment Offset											
8	64	Time To Live								Protocol								Header Checksum															
12	96	Source IP Address																															
16	128	Destination IP Address																															
20	160	Options (if IHL > 5)																															
24	192																																
28	224																																
32	256																																

IP Header Structure

The IP header (for IPv4) is typically 20 bytes but can also be longer with options. The structure is as follows:

Field	Size (Bits)	Description
Version	4	IP version (e.g., 4 for IPv4)
IHL (Internet Header Length)	4	Length of the IP header in 32-bit words
Type of Service (ToS)	8	Defines the quality of service parameters
Total Length	16	Total length of the IP packet (header + data)
Identification	16	Unique identifier for the fragment
Flags	3	Control flags for fragmentation
Fragment Offset	13	Position of the fragment in the original packet
Time to Live (TTL)	8	Limits the packet's lifetime
Protocol	8	Indicates the transport layer protocol (e.g., TCP=6, UDP=17)
Header Checksum	16	Used for error-checking of the header
Source IP Address	32	IP address of the sender
Destination IP Address	32	IP address of the receiver
Options (if any)	Variable	Additional options (if any)
Padding	Variable	Padding to ensure the header is a multiple of 32 bits

UTP

UDP is suitable for applications such as:
UDP is ideal for real-time and low-latency applications where some packet loss is acceptable, such as
✓ video streaming
✓ online gaming
✓ VoIP (Voice over IP)
✓ DNS queries
✓ IoT device communication.

DHCP SERVER

> **DHCP Server** (Dynamic Host Configuration Protocol) : Routers can assign IP addresses to devices on the network automatically, which simplifies network management.