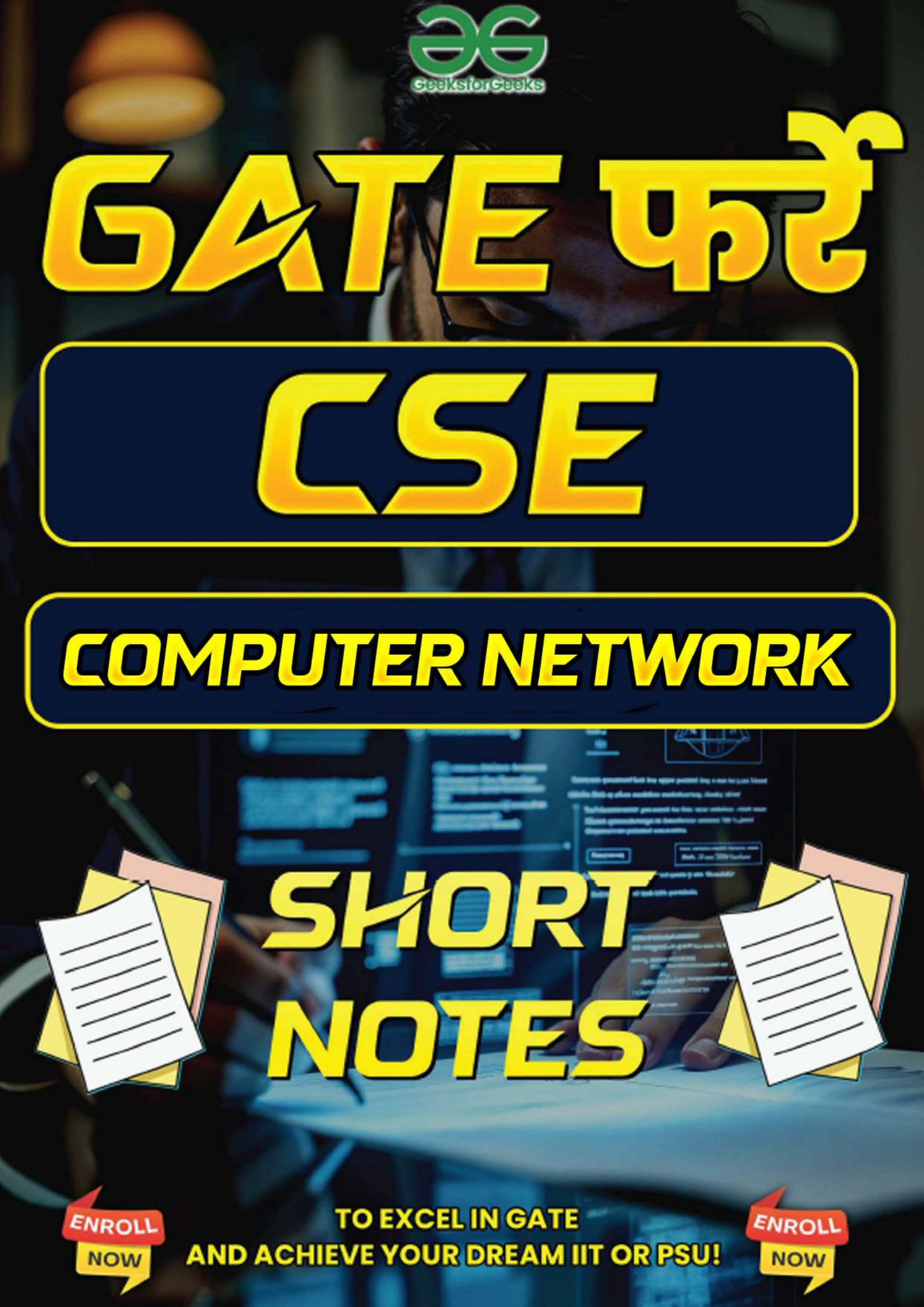

GeeksforGeeks
GATE फर्स्ट
CSE
COMPUTER NETWORK
SHORT NOTES
ENROLL NOW
TO EXCEL IN GATE
AND ACHIEVE YOUR DREAM IIT OR PSU!
ENROLL NOW

1. **IP address: 32 bit** [8 bit | 8 bit | 8 bit | 8 bit]
2. **IP address**: to identify n/w & host
3. **Port no**: to identify process in the device

IP address is divided into n/w ID and host ID

4. **In Classful Addressing**, 32 bits address, starting bits get fixed → to make different classes.

Classful Addressing:

Class	First Octet	Range	No of IP addresses
A	0	[1 – 126]	2^{31}
B	10	[128 – 191]	2^{30}
C	110	[192 – 223]	2^{29}
D	1110	[224 – 239]	2^{28}
E	1111	[240 – 255]	2^{28}

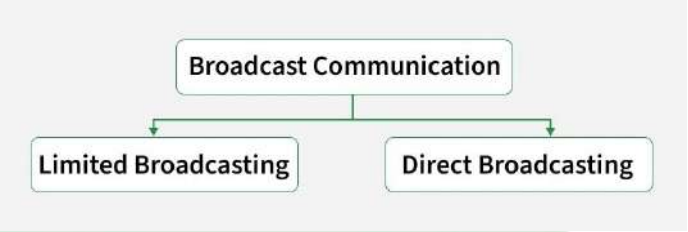
Class	No of N/W	No. of Host/NW
Class A	$2^7 - 2 = 126$	$2^{24} - 2$
Class B	$2^{14} = 16,384$	$2^{16} - 2$
Class C	$2^{21} = 20,97,152$	$2^8 - 2$

Class D	No NID & HID, all 28 bits used for multicast
Class E	No NID & HID, for future purpose

5. Class A has reserved two networks:
   - 0.0.0.0 → Default Route
   - **127.x.y.z → Self loop address**
6. 255.255.255.255 → Limited broadcast address.

Types of Communication:

i) Unicast (1 to 1) → one computer to another computer
ii) Broadcast (1 to all)
iii) Multicast (1 to many) → one computer to many computers



Broadcast Communication:

- **Limited Broadcast** → Transmitting data 1 to all in same n/w
- **Directed Broadcast** → Transmitting data 1 to all in different n/w
- Limited Broadcast address =255.255.255.255
- Directed Broadcast address (DBA) = All HID should be 1
- Can't use as source IP
- Always used as destination address

Special Cases:

NID	HID
valid	all 0's → N/w id of entire n/w
valid	All 1's → Directed broadcast address (DBA)
All 1's	all 1's → Limited broadcast address (LBA)

7. **Subnetting**: Subnetting is borrowing bits from HID.

Subnet mask: It helps to identify which portion of IP is network ID and which portion is host ID.

- No. of 1's = (NID + Subnet ID)
- No. of 0's = HID

Default Subnet Mask:

- Class A: 255.0.0.0
- Class B: 255.255.0.0
- Class C: 255.255.255.0

Subnet mask (AND) IP add = N/w IP

VLSM (Variable Length Subnet Mask):

In VLSM, subnet design uses more than one mask in the same network. → Means more than one mask is used for different subnets of single class (A, B, C).

- In subnets, no of host = no of IP addresses - 2

Classless Addressing:

a.b.c.d /n (where n = NID or subnet mask)

CIDR (Classless Interdomain Routing):

Rules to be followed:

1. All IP addresses in the block must be contiguous
2. Block size must be a power of 2
3. First IP address of the block must be divisible by size of block

Supernetting

- The process of combining two or more networks to get a single network is called supernetting.
- (Subnet mask borrowed from net ID)

Advantages of Supernetting:

- Reduces routing table entries
- Router will take less time for processing packets
- Improves flexibility of IP address allotment

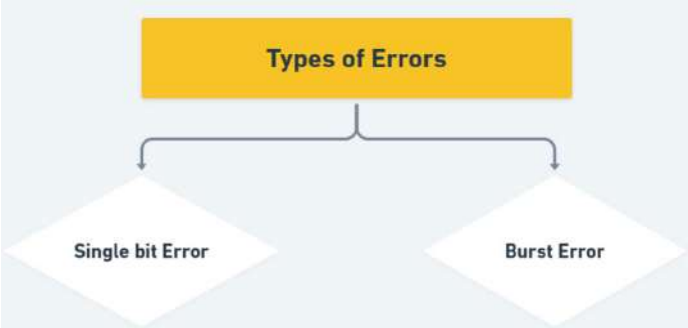
Rules of Supernetting:

1. Networks ID must be contiguous
2. Size of the Network must be same and networks must be multiple of 2
3. First Netwok ID must be divisible by size of supernet
4. Block should be of equal size & belong to same class

Private IP addresses:

- 10.0.0.0 to 10.255.255.255
- 172.16.0.0 to 172.31.255.255
- 192.168.0.0 to 192.168.255.255

Error Control



- Number of corrupted bits = Data rate × Noise duration
- Burst error is more likely to occur than single bit error.
- Error correction is more difficult than error detection.

Error Control:

- **Error Detection:** If error found, discard and ask for retransmission.
- **Error Correction**: If can correct error, retransmission not required.

Methods for Error Detection:

1. Simple parity
2. 2D parity
3. Checksum
4. CRC

Methods for Error Correction:

- **Hamming code**

Block Coding:

- **Message is divided into blocks, each block of size K bits = datawords.**
- **Add r redundant bits → codewords of length n (n = k + r).**
- **Instead of sending datawords, send codewords.**

Valid codeword = 2^k
Invalid codeword = (2^n – 2^k)

Hamming Distance

Hamming Distance:

- Hamming distance between two strings of equal length is the number of positions at which the corresponding symbols (0 or 1) are different.
- Denoted by d(x, y).

Examples:

1. d(000, 101) = 2 hamming distance
2. d(100, 011) = 3 hamming distance
3. d(101, 110) = 2 hamming distance

- Hamming distance can be found by applying XOR operation (⊕) on two codewords and counting the number of 1's in the result.

Minimum Hamming Distance:

- In a set of codewords, the minimum Hamming distance between all possible pairs of codewords is taken.

Example:

Valid codewords:

- a: 010
- b: 101
- c: 110
- d: 001

Distances:

- d(a,b)=3
- d(a,c)=1
- d(a,d)=2
- d(b,c)=2
- d(b,d)=1
- d(c,d)=3

→ min hamming distance = 1

Minimum hamming distances Required for error detections.

- For detecting d bits error → min hamming distance = (d + 1)
- For correcting d bits error → min hamming distance = (2d + 1)

Linear Block Codes

Linear Block Code:

- A code in which XOR(⊕) of two valid codewords is also a valid codeword.
- Minimum HD = minimum number of 1's in non-zero codeword.

Simple Parity Check:

- Data + 1 parity bit.
- Even parity: parity bit added to make number of 1's even.
- Odd parity: parity bit added to make number of 1's odd.
- Limitation: Can't detect even number of errors.

2D Parity Check:

- Information bits organized in a matrix of rows and columns.
- For each row and column → parity bit calculated.
- Detects and corrects all single-bit errors, and detect 2- or 3-bit errors.
- Some patterns with 4 or more errors can be detected.

Cyclic Redundancy Check (CRC)

- Sender adds CRC at end of data.
- Receiver divides received data by divisor. If remainder = 0 → no error. Else → error.

Steps:

- Length of dataword = n
- Length of divisor = k
- Append (k-1) zeros to original message
- Perform modulo-2 division → remainder = CRC
- Codeword = dataword + (k-1) zero's + CRC

Note: CRC must be (k-1) bits

Example:

- Data = 1001001 (n=7)
- Divisor (CRC generator) = 1101 (k=4)

1) Append k−1=3k-1=3k−1=3 zeros to the data
Dividend =1001001000=
1001001\mathbf{000}=1001001000

2) Long division (XOR when the current bit is 1)
Divisor =1101= 1101=1101

Step @pos0: 1001 ⊕ 1101 = 0000 → partial: 0000 001000
Step @pos1: 0010 ⊕ 0000 = 0010 → (no XOR since leading 0, effectively shift)
Step @pos1: 0100 ⊕ 1101 = 1001 → partial
Step @pos2: 1001 ⊕ 1101 = 0000
Step @pos3: 0001 ⊕ 0000 = 0001 → (leading 0s skip)
Step @pos3: 0111 ⊕ 1101 = 0010
Step @pos5: 1000 ⊕ 1101 = 0101
Step @pos6: 1010 ⊕ 1101 = 0111

After division → remainder = CRC

– CRC Polynomial Notation

- Dataword = d(x)
- Codeword = C(x)
- Generator = g(x)
- Syndrome = S(x)
- Error = e(x)

Steps to Apply CRC:

1. Determine degree 'r' of g(x) (highest power).
2. Determine x^r d(x).
3. Divide x^r d(x) by g(x) → remainder.
4. Codeword = x^r d(x) + remainder.

1. CRC (Cyclic Redundancy Check)

- **Dataword:** 1001001
- **Divisor (Generator polynomial):** 1101 = $x^3 + x^2 + 1$
- Dataword polynomial: d(x) = $x^6 + x^3 + 1$
- Append r (degree of divisor = 3) → multiply by x^3
  → $x^3 \cdot d(x) = x^9 + x^6 + x^3$
- Divide by g(x) to get **remainder** (CRC).
- **Remainder = $x^2 + x + 1$**
- **Codeword = $d(x) \cdot x^3$ + remainder = $x^9 + x^6 + x^3 + x^2 + x + 1$**
  → 1001001111

  Final **CRC codeword = 1001001111**

2. Properties of a Good Generator Polynomial

1. Should have **≥ 2 terms**.
2. Coefficient of highest term x^0 should be 1.
3. Should **not divide x^k+1** for k between 2 and n−1.
4. Must have factor (x+1).

3. Checksum

- If 8 bit checksum is used data is divided into 8-8 bits group and added ,its 1's complement is checksum.
- Sender transmits (data + checksum).
- Receiver also does same ,if results come to zero then data is correct ..

4. Hamming Code (Error Correction)

- **Hamming Code** is used for error correction.
- It can **correct** single-bit errors.
- It can **detect** up to two-bit errors.
- m = message bits
- r = redundant (check/parity/extra) bits
- n=m+r total codeword length

According to the Hamming Code condition, the minimum number of redundant bits required is:

$2^r \geq m + r + 1$

Here, r represents the **lower limit** of the redundant bits needed.

-

Key Points to Remember:

- **CRC → remainder after polynomial division**.
- **Checksum → 1's complement sum method**.
- **Hamming Code → error detection + correction (1-bit errors, up to 2-bit detection)**.

Flow Control

Bandwidth:

- Bandwidth refers to **maximum rate of data transfer** across a network or internet connection.

K, M, G are different for Data and Bandwidth:

Unit	Data	Bandwidth
K	2^{10}	10^3
M	2^{20}	10^6
G	2^{30}	10^9

Delay in Computer Networks

1. **Transmission Delay (Td)**
2. **Propagation Delay (Pd)**
3. **Queuing Delay (Qd)** (Considered if not negligible)
4. **Processing Delay (Pr d)**

Transmission Delay: (Td)

- Amount of time taken to **transfer a packet** on to the outgoing link is considered as **transmission delay**.

Td=L/ R
where,

- L = Length of the packet
- R = Transmission rate

Propagation Delay:

- Amount of time taken to **reach back** from one (sender) point to another (receiver) point is called propagation delay.

Pd=Distance / Velocity

Queuing Delay :

- The amount of time packet will wait in the queue of a router before being taken up for processing is called **queuing delay**.

Processing Delay :

- Time required for a router or destination host to receive packet, open its input port, remove the header, perform an error detection, etc.

Total Time / Round Trip Time (RTT):

- It is the additional time between a **request for data** and the **display of that data**.

RTT=Td(frame)+2·Pd+Qd+Prd+Td(ack)

Efficiency / Line Utilization / Link Utilization / Sender Utilization:

Efficiency=Useful Time/Total Time

Throughput / Effective Bandwidth / Bandwidth Utilization:

- Maximum data rate possible.

Throughput=Efficiency×Bandwidth
Throughpu t= L /
Td(frame)+2·Pd+Qd+Prd+Td(ack)

Capacity of Link / Wire / Channel:

- Max number of bits available on a link at any time.

Capacity of Link=B×Pd

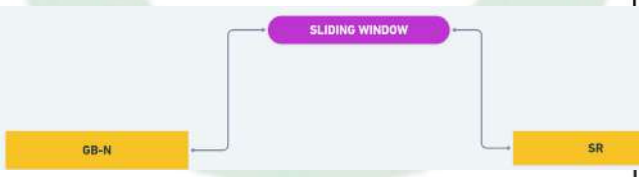
Stop and Wait Protocol:

- Can deal with **frame lost**, **ACK lost**, **ACK delayed**.

- Can work on both full-duplex and half-duplex.

1. Sender can send one data packet at a time.
2. Receiver will receive and consume the data packet.
3. After consuming, the data receiver sends ACK, and ready for the next.
4. Sender can't send next data packet unless ACK of previous packet is received.
5. Sender must maintain a copy of the frame and timeout must be started after sending.

Sliding Window Protocol:

- Instead of sending one packet and waiting for ACK, sender sends **multiple packets** and waits for **acknowledgements**.
- Send window size = WS
- 

Window Size Calculations:

- **Max Window Size**: (1+2a)·Pkt
- where a=Pd/Td
- **Min Seq Number**: (1+2a)
- **Min No. of Bits Req in Sequence No Field**:

ceil(log2(1+2a)

Go-Back-N (GBN):

1. Sender window size = N itself
2. Receiver window size = 1 always
3. Out-of-order packets not received by receiver
4. Timer maintained only for **first frame** in window
5. GBN uses **cumulative ACK**
6. ACK time should be less than timeout time

Selective Repeat (SR):

1. Sender window size = receiver window size = W
2. SR receiver can receive **out-of-order** packets and stores them in buffer
3. Reordering and sorting is done
4. Timer is maintained for **each** frame
5. If a packet is corrupted, NAK is sent, sender resends
6. NAK more than 1 – only packet with NAK is retransmitted
7. Duplicate Packet Problem:
    - Occurs when same data packet is received **more than once**
    - Can be solved by:
        - Increasing sequence number
        - Decreasing sender window size

GeeksforGeeks

GATE CSE BATCH

KEY HIGHLIGHTS:

- 300+ HOURS OF RECORDED CONTENT
- 900+ HOURS OF LIVE CONTENT
- SKILL ASSESSMENT CONTESTS
- 6 MONTHS OF 24/7 ONE-ON-ONE AI DOUBT ASSISTANCE
- SUPPORTING NOTES/DOCUMENTATION AND DPPS FOR EVERY LECTURE

COURSE COVERAGE:

- ENGINEERING MATHEMATICS
- GENERAL APTITUDE
- DISCRETE MATHEMATICS
- DIGITAL LOGIC
- COMPUTER ORGANIZATION AND ARCHITECTURE
- C PROGRAMMING
- DATA STRUCTURES
- ALGORITHMS
- THEORY OF COMPUTATION
- COMPILER DESIGN
- OPERATING SYSTEM
- DATABASE MANAGEMENT SYSTEM
- COMPUTER NETWORKS

LEARNING BENEFIT:

- GUIDANCE FROM EXPERT MENTORS
- COMPREHENSIVE GATE SYLLABUS COVERAGE
- EXCLUSIVE ACCESS TO E-STUDY MATERIALS
- ONLINE DOUBT-SOLVING WITH AI
- QUIZZES, DPPS AND PREVIOUS YEAR QUESTIONS SOLUTIONS

ENROLL NOW

TO EXCEL IN GATE
AND ACHIEVE YOUR DREAM IIT OR PSU!

ENROLL NOW

- Condition to solve:

WS+WR≤ASN
Where:

- WS = Window Size
- WR = Receiver Window
- ASN = Available Sequence Numbers

Final Conditions:

- **Go-Back-N:** Receiver window size is always 1
  WS+1≤ASN
- **SR:** Best condition:
  2WS≤ASN

Comparison Table:

Metric	Stop and Wait	Go-Back-N (GBN)	Selective Repeat (SR)
Efficiency	η = Td / (Total Time)	η = N×Td / (Total Time)	η = WS×Td / (Total Time)
Through put	L / Total Time	N×L / TotalTime	WS×L / TotalTime
Buffer	1 + 1	N + 1	N + N
Seq Formula	2(0 or 1)	N + 1 (0 - N)	2N (0 - 2N–1)
Seq no in K bits .		Ws 2^k-1 Wr 1	Ws 2^(k-1) Wr 2^(k-1)

IPv4 Header

Layers:

- Application Layer → Message
- Transport Layer → Segment
- Network Layer → Datagram

IPv4 Header Format:

VER (4 bits)	HL (4 bits)	Services (8 bits)	Total Length (16 bits)
Identification number (16 bits)		Flags (3 bits)	Fragment offset (13 bits)
Time to Live (8 bits)	Protocol (8 bits)	Header checksum (16 bits)	
Source IP Address (32 bits)			
Destination IP Address (32 bits)			
Option (0–40 bytes)			

- **Min Header Size:** 20B + 0 B = 20 B
- **Max Header Size:** 20B + 40B = 60B

Field Descriptions

Version (4 bits)

- Indicates **IPv4 or IPv6**.

Header Length (4 bits)

- Contains length of the header.
- Min: 20B, Max: 60B
- Formula: Header Length = Field Value × 4(scaling factor)

Services (8 bits)

- First 3 bits = **Priority**
- Next 4 bits = **Type of Service** (TOS)
- Last bit = Not used

Structure:

P	P	P	D	T	R	C	X

- P = Priority
- D = Min delay
- T = Max throughput
- R = High reliability
- C = Min cost

Priority

It is a 3-bit subfield ranging from 0 to 7 (000 to 111 in binary). Priority field is needed if a router is congested and need to discard some datagram, those datagrams which have the lowest priority are discarded first.

Types of Services

It is a 4 bit subfield. Each bit having a special meaning, although a bit can be 0 or 1. One and only one of the bits can have the value 1 in each datagram.

Total Length (16 bits)

- Total Length = Data + Header
- Min Datagram size = 20B
- Max = 2^16 = 65535B

Identification Number (16 bits)

- Each datagram is assigned a sequence number for identification.
- All fragments of a datagram share the same ID.

Flags (3 bits)

- Bit 1: Not used
- Bit 2: DF = Don't Fragment
- Bit 3: MF = More Fragments

Fragment Offset (13 bits)

- Indicates the position of the fragment relative to the original datagram.
- Stored as offset/8 (scaling factor = 8).

Fragment Types:

- FO = 0 → First fragment
- FO ≠ 0→ Middle OR LAST
- FO ≠ 0, MF = 0→ Last fragment
- FO = 0, MF = 0 → No fragmentation

TTL (8 bits) – Time To Live

- Prevents infinite looping of packets.
- Decremented by 1 on each hop.

If the **TTL (Time to Live)** field becomes **zero** before reaching the destination:

- The datagram is **discarded**.
- An **ICMP (Internet Control Message Protocol) message** is sent back to the sender.

Protocol (8 bits)

- Indicates the protocol in data portion:
  - ICMP → 01
  - IGMP → 02
  - UDP → 17
  - TCP → 06

ICMP>IGMP>UDP>TCP

Header Checksum (16 bits)

- Calculated for header only.
- Computed at **each router** as headers may change (e.g., TTL).

Source Address (32 bits)

- IPv4 address of sender.

Destination Address (32 bits)

- IPv4 address of receiver.

Options (0–40 bytes)

The IPv4 datagram header consists of **two parts**:

- A **fixed part**, which is always **20 bytes long**.
- A **variable part**, which can be up to **40 bytes long**.

Types of Options:

1. Strict Source Routing
2. Loose Source Routing
3. Record Routing
4. Timestamp
5. Padding

Not Changed	May be Changed	Definitely Changed
VER	Total Length	TTL
Services	MF (More Fragments)	Header Checksum
Identification Number	Fragment Offset	
DF (Don't Fragment)	HL & Options *(if present, HL may change)*	
Protocol		
Source IP (SIP)		
Destination IP (DIP)		

TCP & UDP

<table>
<tr><td colspan="8">Source Port (16 bits)</td><td>Destination Port (16 bits)</td></tr>
<tr><td colspan="9">Sequence number (32 bits)</td></tr>
<tr><td colspan="9">Acknowledgement number (32 bits)</td></tr>
<tr><td>Header Length (4 bits)</td><td>Reserved bits (6 bits)</td><td>URB</td><td>ACK</td><td>PSH</td><td>RST</td><td>SYN</td><td>FIN</td><td>Window Size (Advertisement Window) (16 bits)</td></tr>
<tr><td colspan="8">Check sum (16 bits)</td><td></td></tr>
<tr><td colspan="9">Options (0-40 bytes)</td></tr>
</table>

Header length: (4 bits)

- Header length is a 4-bit field that contains the length of header.

- Scaling factor = 4
  ➤min^m size = 20B
  ➤max^m size = 60B

Source Port Address: (16 bits)

- This is a 16-bit field that defines the port no of the application/program in the host that is sending the segment.

Destination Port: (16 bits)

- This is a 16-bit field that defines the port no of application program in the host that is receiving the segment.

Sequence Number:

- This is a 32-bit field defines the seq number of the first data byte.

- Every byte is associated with one seq number.

Data size at TL = Total Length (IP) - IP(H) - TCP(H)

- Every packet is associated with one sequence number.

- **TCP Suggests**: Do not start with the sequence number 0.
  ➤Always choose any **random sequence number initially.**

Acknowledgment number (32-bit)

- A 32-bit field that indicates the next expected byte seq number from the other end of the connection.

Flags:

There are 6 different flags and they can have values 0 or 1:

1. **URG**
2. **ACK**
3. **PSH**
4. **RST**
5. **SYN**
6. **FIN**

1. URG (Urgent):
Indicates that the urgent pointer field is valid and contains urgent data that should be processed with priority.

2. ACK (Acknowledgement):
Indicates whether the acknowledgment number field is valid. It's used to confirm the receipt of a data segment.

3. PSH (Push):
Instructs the receiver to deliver data to the application layer **immediately** without buffering.

4. RST (Reset):
Used to abruptly terminate a connection, often in response to some error or invalid segment.

5. SYN (Synchronize):
Used to initiate a connection during the TCP 3-way handshake. It signals the beginning of a TCP conversation.

6. FIN (Finish):
Used to terminate a TCP connection gracefully.
➡When a sender sets FIN, it indicates that it has no more data to send.

SYN / FIN / ACK Consumption Table

Condition	Notes
SYN=1	Consume 1 seq no
FIN=1	Consume 1 seq no
ACK=1	Consume 0 seq no
Data byte	Consume 1 seq no

	SYN	ACK	
1.	1	0	= Request
2.	1	1	= Reply
3.	0	1	= Ack
4.	0	0	= Data Transfer

Wrap Around Time (WAT):

Time taken to wrap around 2^{32} sequence numbers (based on bandwidth)

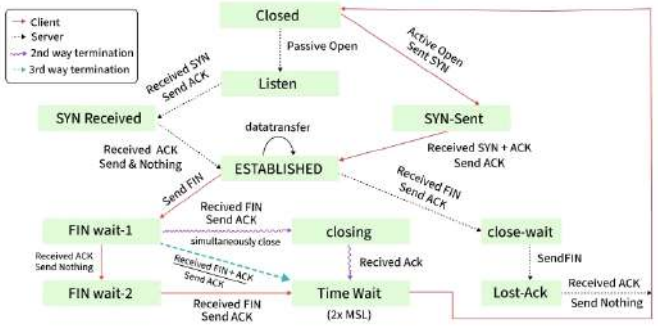
WAT = Total seq no / Bandwidth (B/s)

To avoid wrap around within the lifetime:
Min seq no ≥ LT × B

Transmission Control Protocol (TCP)

- TCP is **reliable**, port-to-port, byte/stream transport layer protocol.
- Supports **full-duplex, connection-oriented** communication.
- TCP connection has **3 phases**:
    1. Connection Establishment
    2. Data Transfer
    3. Connection Termination
- Uses **sliding window protocol** for flow control; the window size is set and controlled by the receiver.
- Each TCP connection is associated with **four windows**.
- Not useful for **broadcasting** and **multicasting**
- Data will be received at destination **in order**
- TCP provides **end-to-end error control and flow control**

TCP State Transition Diagram



Retransmission in TCP:

1. Retransmission after timeout timer
2. Retransmission after 3 duplicate ACKs

Silly Window Syndrome

- When we use capacity of the network inefficiently (overhead >> data transfer)
- Happens when:
  - Receiving capacity becomes 0
  - Produces only one byte at a time
  - Consumes only one byte at a time → leads to **Silly Window Syndrome**

TCP Timer Management

1. Keep-alive timer
2. Persistent timer
3. ACK timer
4. Linear wait timer
5. Timeout timer

1. Persistent Timer

- Deals with **zero-window deadlock**
- Ensures window size info continues even if other end closes receive window

2. Keep-alive Timer

- Server closes the connection if the client does not send any data for a fixed time

3. Time-wait Timer (2MSL)

- Used during connection termination
- Connection stays open for 2MSL to allow TCP to resend final ACK in case ACK is lost

4. Timeout Timer

- Starts after sending TCP segment
- If ACK not received in time → **Retransmission**
- Timeout timer = **Retransmission Timer**
- Should adapt based on traffic:
  - Increase if traffic is high
  - Decrease if low traffic

Algorithms for Computing Timeout Timer

1. Basic Algorithm
2. Jacobson's Algorithm
3. Karn's Algorithm

Timeout Timer = 2 × RTT

Karn's Algo:

- When a segment is retransmitted, do not use Basic/Jacobson's algorithm since RTT is invalid.
- Instead, double the **RTT × TOT** when timeout occurs.

RTT Formulas:

Next RTT (NRTT):

$$NRTT = \alpha(RTT) + (1-\alpha)ARTT$$

$$0 < \alpha < 1$$

Actual Deviation (AD) = $|RTT - ARTT|$

Next Duration (ND) = $\alpha(AD) + (1 - \alpha)(\text{prev AD})$

Congestion in Network / Congestion Control

Congestion:

- State where message traffic is too heavy → slows down response time

Congestion Control:

- Techniques/mechanisms to:
    - Prevent congestion before it happens
    - Handle congestion after it occurs

TCP reacts by reducing sender window size:

Ws = min(Wc, Wr)

Where:

- Wc = congestion window
- Wr = receiver window

Threshold = Wr / 2

TCP Congestion Control Phases:

1. **Slow Start**
2. **Congestion Avoidance**
3. **Congestion Detection**

Slow Start:

- Window size increases exponentially until it hits threshold

After 1 RTT: Wc = 2 × (Wc)
If ACK arrives: Wc = Wc + 1

Congestion Avoidance (Additive Increase):

- Window size increases linearly

After 1 RTT: Wc = Wc + 1
If ACK arrives: Wc = Wc + 1/wc

Congestion Detection:

- If congestion occurs, window size must be decreased

1. Timeout
2. 3 Duplicate ACKs

Threshold new = Wc / 2
Restart from threshold

Traffic Shaping:

Mechanism to **control traffic** sent to the network.

1. Leaky Bucket:

- Fixed output rate even with variable input
- Packets leak at constant rate if there's water (packets) in the bucket

2. Token Bucket:

- Allows bursty traffic at regulated rate
- Max number of packets = C + r × t
  (C = capacity, r = rate, t = time)

UDP - User Datagram Protocol

- **Message-oriented**, **connectionless**, **unreliable** transport protocol
- No flow or error control
- Header is **simple & fixed (8 bytes)**

UDP Header Format:

Source Port (16)	Destination Port (16)
Length (16)	Checksum (16)

- UDP Length = IP Length - IP Header Length

  e.g. 65535 − 20 = 65515

- Max payload = 65515 − 8 = **65507**

Checksum:

- Not mandatory
- No flow/error control
- Depends on IP & ICMP for error reporting

Where We Use UDP:

- Applications requiring:
  - One request, one reply
  - Constant data flow
  - Multimedia streaming
  - Speed over reliability
- Used in:
  - SNMP
  - Route updates
  - Broadcasting/Multicasting
  - Real-time apps
  - TFTP (Trivial File Transfer Protocol)

TCP vs UDP Summary:

Feature	TCP	UDP
Header Size	Dynamic (20–60B)	Fixed (8B)
Flow Control	End-to-End	None
Error Control	Yes (Checksum Mandatory)	Optional (Checksum)
Connection Type	Connection-Oriented	Connectionless
Reliability	Reliable	Not Reliable
Sequence Numbers	Yes	No
Acknowledgment	Yes	No
Overhead	High (20–60B)	Low (8B)
Protocol Examples	HTTP, FTP, SMTP, POP	DNS, SNMP, TFTP, DHCP

State	Description
CLOSED	No connection exists
LISTEN	Waiting for SYN
SYN-SENT	Sent SYN, waiting for ACK
SYN-RCVD	Received SYN, sent ACK
ESTABLISHED	Connection established
FIN-WAIT-1	Sent FIN, waiting for ACK
FIN-WAIT-2	Received ACK for FIN
CLOSE-WAIT	Received FIN, sent ACK
CLOSING	Both sides sent FIN
LAST-ACK	Sent FIN, waiting for final ACK
TIME-WAIT	Final ACK sent, waiting for 2MSL

GeeksforGeeks

GATE CSE BATCH

KEY HIGHLIGHTS:

- 300+ HOURS OF RECORDED CONTENT
- 900+ HOURS OF LIVE CONTENT
- SKILL ASSESSMENT CONTESTS
- 6 MONTHS OF 24/7 ONE-ON-ONE AI DOUBT ASSISTANCE
- SUPPORTING NOTES/DOCUMENTATION AND DPPS FOR EVERY LECTURE

COURSE COVERAGE:

- ENGINEERING MATHEMATICS
- GENERAL APTITUDE
- DISCRETE MATHEMATICS
- DIGITAL LOGIC
- COMPUTER ORGANIZATION AND ARCHITECTURE
- C PROGRAMMING
- DATA STRUCTURES
- ALGORITHMS
- THEORY OF COMPUTATION
- COMPILER DESIGN
- OPERATING SYSTEM
- DATABASE MANAGEMENT SYSTEM
- COMPUTER NETWORKS

LEARNING BENEFIT:

- GUIDANCE FROM EXPERT MENTORS
- COMPREHENSIVE GATE SYLLABUS COVERAGE
- EXCLUSIVE ACCESS TO E-STUDY MATERIALS
- ONLINE DOUBT-SOLVING WITH AI
- QUIZZES, DPPS AND PREVIOUS YEAR QUESTIONS SOLUTIONS

ENROLL NOW

TO EXCEL IN GATE
AND ACHIEVE YOUR DREAM IIT OR PSU!

ENROLL NOW

Multiple Access Control in Networking

Data Link Layer:

- Divided into two sublayers:
  - **Logical Link Control (LLC)** – Error control, flow control.
  - **Media Access Control (MAC)** – Access control.
- Types of communication link:
  - Point-to-Point
  - Broadcast Link

Access Control:

- Ensures fair access to the transmission medium.
- Required in broadcast networks to avoid **collisions**.
- **Collision** causes data corruption → requires control.

Multiple Access Protocols:

1. **Random Access** – ALOHA, CSMA, CSMA/CD, CSMA/CA
2. **Controlled Access** – Reservation, Polling, Token Passing
3. **Channelized Access** – FDMA, TDMA, CDMA

ALOHA:

Pure Aloha	Slotted Aloha
Any station transmits the data at any time.	Any station can transmit the data at the beginning of any time slot.
Vulnerable time in which collision may occur $= 2 * T_f$(T_f – Transmission time for single frame)	Vulnerable time in which collision may occur $= T_f$
Throughput of pure aloha $= G * e^{-2G}$	Throughput of slotted Aloha $= G * e^{-G}$
Maximum throughput $s_{max} = 18.4\%$ (When $G = 1/2$)	Maximum throughput $s_{max} = 36.8\%$ (When $G = 1$)
The main advantage of pure aloha is it simplicity in implementation	The main advantage of slotted aloha is that it reduces the number of collisions to Half and double the throughput of pure aloha

CSMA (Carrier Sense Multiple Access):

- Station **senses** channel before transmitting.
- Still collision possible due to **propagation delay**.
- Types:
  1. **1-persistent:** Continuously senses, sends immediately.
  2. **Non-persistent:** Waits random time if busy.

3. **p-persistent:** It is used if the channel has time slot duration equal to or grater than the propagation time
   If channel has found ideal then:
   (i) with probability p , the station sends its frame.
   (ii) with probability (1-p),the station wait for the beginning of the next time slot and check the line again.

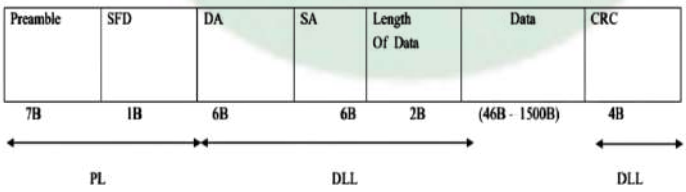
CSMA/CD (with Collision Detection):

- Stops sending when collision is detected.
- Sends jamming signal to inform others.
- **Minimum frame size to detect collision L** $\geqslant 2pd + Td(jam) \times B$

Backoff Algorithm:

- Used in CSMA/CD to wait before retransmitting.
- Backoff time = K×RTT
- K∈[0,2n−1]K \in [0, 2^n - 1]

Ethernet (IEEE 802.3):



- Uses **CSMA/CD** and **bus topology**.
- **Speeds**:
  - 10 Mbps (normal)
  - 100 Mbps (fast)
  - 1 Gbps (gigabit)
- **Frame format:**
  - Preamble: 7B
  - SFD: 1B
  - Destination Address: 6B
  - Source Address: 6B
  - Length: 2B
  - Data: 46–1500B
  - CRC: 4B

Ethernet Frame Fields:

Preamble:

- 7B field contains alternate 0's and 1's.
- It alerts the stations that frame is going to start.

Start Frame Delimiter (SFD):

- It is a 1 byte field which is always set to 10101011.
- Last "11" indicates the end of SFD and marks the beginning of the frame.

Destination Address (DA):

- It is a 6B field that contains the MAC address of the destination for which the data is destined.

Source Address (SA):

- It is a 6B field that contains the MAC address of the source which is sending the data.

Length:

- It is a 2B field which specifies the length of data field.

Data:

- It is a variable-length field which contains the actual data.
- It is also called the **payload field**.
- Range of size = (46B – 1500B).

Frame Check Sequence (CRC):

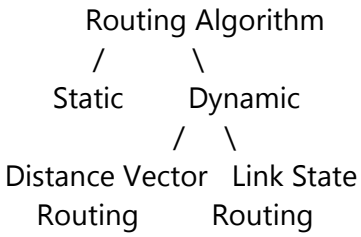
- It is a 4B field that contains the CRC code for error detection.

Types of Packet Switching

1. **Datagram Packet Switching**
2. **Virtual Circuit Packet Switching**

Datagram Switching	Virtual Circuit Switching
No need to make a connection before sending data.	Connection is made first, then data is sent.
Each data packet can go from a different route.	All packets go through the same fixed route.
Packets can reach the destination in any order.	Packets reach in the same order as sent.
Not very reliable – packets can get lost.	More reliable – packets are not usually lost.
Cheaper method.	More expensive – resources are reserved.
No resources are reserved in advance.	First packet books CPU, bandwidth, etc. for others.
Every packet carries full address info.	Only the first packet carries full path info; others carry short info.
If there is no space, packets can be dropped midway.	Packets are never dropped; they are queued or forwarded.
Used in IP networks (like the Internet).	Used in ATM (Asynchronous Transfer Mode).
Works mostly at Network Layer (Layer 3).	Works mostly at Data Link Layer (Layer 2).

Routing Algorithm



Types of Routing Algorithms:

1. Static Routing

- The fixed path is set manually.
- It does not change automatically if the network changes.
- Simple but not flexible.

2. Dynamic Routing

- Routes are updated automatically based on network conditions.
- Divided into two types:

a) Distance Vector Routing

- Each router shares distance info with its neighbors.
- Works slowly and takes time to adjust if the network changes.

b) Link State Routing

- Each router knows the full map of the network.
- Fast and adjusts quickly to network changes.

Distance Vector Routing	Link State Routing
Used in 1980s	Used in 1990s
Uses very less bandwidth (only sends distance info)	Uses more bandwidth (sends full link info)
Router knows only about its neighbors (local info)	Router knows about the entire network (global info)
Uses Bellman-Ford algorithm	Uses Dijkstra's algorithm
Network traffic is low	Network traffic is high
Slow to update (slow convergence)	Fast to update (fast convergence)
Has count to infinity problem	No such problem
Can create permanent loops	May create temporary loops, but gets fixed
Protocol example: RIP (Routing Information Protocol)	Protocol example: OSPF (Open Shortest Path First)

Note:

1.The maximum Hop count allowed For RIP is 15 and Hop count of 16 is considered as Destination unreachable.

2.RIP uses UDP as its transport protocol with the port number – 530

IP Support Protocols

1. ARP (Address Resolution Protocol)

Finds MAC address using an IP address.

Used when a device knows the IP but needs the physical address to send data.

2. ICMP Messages (Internet Control Message Protocol)
ICMP messages are of two types:

1. Error-reporting or Feedback Messages
These are sent when there is a problem in the network.

There are 5 types:

	Error Message Type	Meaning
	Destination Unreachable	The packet cannot reach the final destination.
	Source Quench	Tells the sender to slow down sending packets.
	Time Exceeded	Packet took too long and got dropped.
	Parameter Problems	There's something wrong in the packet's header (info).
	Redirection	Suggests a better route for the packet.

2. Query or Request and Reply Messages
These are used for asking questions or sending requests and getting replies.

There are 4 types:

	Query Message Type	Meaning
	Echo Request and Reply	Used for testing if the other device is active. (Ping uses this)
	Timestamp Request and Reply	Used to check the time between two devices.
	Address-mask Request and Reply	Asks for the subnet mask information.
	Router Solicitation and Advertisement	Used by devices to find nearby routers.

APPLICATION LAYER PROTOCOL

The Application Layer is the topmost layer in the OSI and TCP/IP models. It directly interacts with the end user and provides network services to applications like web browsers, email clients, etc.
There are some Application Layer Protocols

1. HTTP
2. FTP
3. SMTP
4. POP3
5. IMAP
6. DNS

HTTP Protocols-

1. HTTP is used to access data on the World Wide Web (www).
2. It is a client-server protocol, and works on port number 80 using TCP.
3. HTTP is an In-Band protocol – both request and data are sent in the same connection.
4. HTTP is a stateless protocol, which means it does not remember anything about the user or past requests.
5. There are 2 types of HTTP connections:
    - (i) Non-persistent (1.0)
    - (ii) Persistent (1.1)

Non-Persistent (1.0)

In Non-persistent HTTP, a new TCP connection is created for each request-response pair.
Steps:

1. Client opens a TCP connection and sends a request.
2. Server sends the response and closes the connection.
3. If a file contains N images (in separate files on same server), then connection must open and close N+1 times.

➞ That's why it's called non-persistent – the connection doesn't stay open.

Persistent (1.1)

In Persistent HTTP, the TCP connection is kept open for multiple requests.
Key Points:

1. The server keeps the connection open even after sending the response.
2. The connection is closed only when:
    - Client asks to close it, or
    - A timeout occurs.

This makes it faster and more efficient than non-persistent.

FTP-File Transfer Protocol

FTP is a standard internet protocol used to transfer files between computers using a TCP/IP connection.

1. It works on two port numbers:
    - Port 21 for Control Connection
    - Port 20 for Data Connection
2. FTP uses two connections:
    - (i) Control Connection (Port 21): for sending commands
    - (ii) Data Connection (Port 20): for sending actual files
3. The control connection remains open during the entire FTP session.
4. The data connection is opened and closed for each file transfer.
5. When an FTP session starts, control connection opens first.
    - While it's open, the data connection can be opened/closed many times (for multiple file transfers).
6. FTP uses persistent TCP connection for control (i.e., it stays open).
7. FTP uses non-persistent TCP connection for data transfer (i.e., opens and closes per file).
8. FTP is a connection-oriented protocol.
9. FTP is an "Out-of-band" protocol:
    - Control and data do not go through the same connection.
    - They use separate connections.
10. Some protocols (like FTP) send requests and data on different connections, so they are called Out-of-Band.
11. Others (like HTTP & SMTP) use the same connection for both request and data – they are called In-Band protocols.
12. FTP is a stateful protocol – it remembers user information during the session.

Transmission Modes In FTP

FTP can transfer files in three modes:

1. Stream Mode –
   Data is sent as a continuous stream of bytes.
2. Block Mode –
   Data is divided into blocks before sending.
3. Compressed Mode –
   Data is compressed before sending to save bandwidth.

File Types in FTP

FTP supports transfer of 3 types of files:

1. ASCII File –
   Normal text files (readable characters).
2. EBCDIC File –
   File format used by IBM systems.
3. Image File –
   Raw binary data (bit-by-bit transfer), also used for software files.

Data Structure in FTP

While transferring files, FTP understands 3 types of data structures:

1. File Structure –
   File is a simple sequence of bytes (normal file).

2. Record Structure –
   File is divided into records (like a table row).

3. Page Structure –
   File is divided into pages, with each page having its own number.

SMTP (Simple Mail Transfer Protocol)

1. SMTP is used to send emails reliably and efficiently.

2. Works on TCP Port 25.

3. SMTP is a text-based, connection-oriented, and stateless protocol.

4. It uses persistent TCP connections, so multiple emails can be sent in one session.

5. It is an in-band protocol (data and control share the same connection).

6. SMTP is used to push (send) emails from client to server.

Components

- User Agent (UA):
  Creates the email message and envelope.

- Mail Transfer Agent (MTA):
  Transfers the message over the Internet.

Email Flow

- Sender uses SMTP to push email to the receiver's mail server.

- Receiver uses POP3 or IMAP4 to pull/download the email.

Limitations of SMTP

- Can only send 7-bit ASCII text.

- Cannot send images, audio, video, binary files, or non-English text (like Hindi, Japanese, etc.).

Solution-MME

- MIME (Multipurpose Internet Mail Extension) is used with SMTP to send non-text content.

- MIME converts non-ASCII data to ASCII for transmission and back at the receiver side.

- Enables sending of images, audio, video, binary files, and foreign language content via SMTP.

Pull vs Push

- SMTP → Push (Client to Server)

- POP3 / IMAP4 → Pull (Server to Client)

POP3(Post office Protocol version-3)

1. Message access protocol – used for receiving emails from the mail server.
2. It is a pull protocol – pulls mails from server to client.
3. Works on TCP port 110.
4. It is a connection-oriented protocol – connection is required for communication.
5. Uses persistent TCP connection – connection stays open during session.
6. State full protocol – remembers user info during session.
7. In-band protocol – both data and control go through the same connection.

Limitations:

- Cannot preview mail before downloading – full mail must be downloaded first.
- Cannot organize emails on the mail server – no folder structure support.

IMAP4 (Internet Mail Access Protocol v4)

1. IMAP4 is similar to POP3, but it has more features, is more powerful, and a bit complex.
2. It allows users to:
    - Check the email header before downloading.
    - Search content inside emails before downloading.
    - Partially download the email.
    - Create, delete, rename folders on the mail server.
    - Organize mails in a folder hierarchy (like Inbox, Sent, Spam etc.).
3. IMAP4 is a pull protocol (like POP3).
4. Works on TCP Port 143.
5. It is connection-oriented and stateful.
6. Uses persistent TCP connection.
7. IMAP is an in-band protocol (data and control share the same connection).

POP3 VS IMAP4-Key Differences

POP3	IMAP4
Mails can only be accessed from one device.	Mails can be accessed from multiple devices.
Emails are downloaded and deleted from the server.	Emails stay on the server and sync across devices.
Users cannot organize mails on the server.	Users can organize mails in folders on the server.
No support for syncing changes.	Supports syncing changes on all devices.
It is one-way (client to server).	It is two-way (changes reflect both sides).

DNS – Domain Name System

1. DNS is used to translate domain names (like www.google.com) into IP addresses.
2. It is used in both LAN and WAN environments to track computers, services, and resources.
3. DNS is an Application Layer protocol.
4. It uses UDP port 53 by default, and TCP port 53 for large transfers (e.g., zone transfers).
5. It is a pull protocol – client sends query, server replies.
6. DNS follows a client-server architecture.
7. It uses a distributed database to store data in the form of resource records.
8. DNS is a stateless protocol – it does not remember previous queries.

GeeksforGeeks

GATE CSE BATCH

KEY HIGHLIGHTS:

- 300+ HOURS OF RECORDED CONTENT
- 900+ HOURS OF LIVE CONTENT
- SKILL ASSESSMENT CONTESTS
- 6 MONTHS OF 24/7 ONE-ON-ONE AI DOUBT ASSISTANCE
- SUPPORTING NOTES/DOCUMENTATION AND DPPS FOR EVERY LECTURE

COURSE COVERAGE:

- ENGINEERING MATHEMATICS
- GENERAL APTITUDE
- DISCRETE MATHEMATICS
- DIGITAL LOGIC
- COMPUTER ORGANIZATION AND ARCHITECTURE
- C PROGRAMMING
- DATA STRUCTURES
- ALGORITHMS
- THEORY OF COMPUTATION
- COMPILER DESIGN
- OPERATING SYSTEM
- DATABASE MANAGEMENT SYSTEM
- COMPUTER NETWORKS

LEARNING BENEFIT:

- GUIDANCE FROM EXPERT MENTORS
- COMPREHENSIVE GATE SYLLABUS COVERAGE
- EXCLUSIVE ACCESS TO E-STUDY MATERIALS
- ONLINE DOUBT-SOLVING WITH AI
- QUIZZES, DPPS AND PREVIOUS YEAR QUESTIONS SOLUTIONS

ENROLL NOW

TO EXCEL IN GATE
AND ACHIEVE YOUR DREAM IIT OR PSU!

ENROLL NOW

Types of DNS Severs

1. Root Server – First-level server, points to TLD servers.
2. TLD (Top Level Domain) Server – Handles domains like .com, .org, .in, etc.
3. Authoritative DNS Server – Stores the actual IP address of a domain name.
4. Local DNS Resolver – Found in client-side networks, stores cached queries.

Main Services of DNS

- Name Translation – Converts domain names into IP addresses.
- Host Aliasing – Maps multiple domain names to the same IP address.
- Mail Aliasing – Handles email routing using MX records.
- Load Balancing – Distributes incoming traffic across multiple servers.

Comparison Table of Application Layer Protocols.

Comparison Iable of Application Layer Protocols						
Feature	**DNS**	**HTTP**	**SMTP**	**POP**	**IMAP**	**FTP**
Stateful / Stateless	Stateless	Stateless	Stateless	Stateful	Stateful	Stateful
Transport Protocol Used	UDP	TCP	TCP	TCP	TCP	TCP
Connectionless/ Oriented	Connectionless	Connectionless	Connection-oriented	Connection-oriented	Connection-oriented	Connection-oriented
Persistent / Nonpersistent	Non-persistent	HTTP 1.0: Non-persistent HTTP 1.1: Persistent	Persistent	Persistent	Persistent	Control: Persistent Data: Non-persistent
Push / Pull	-	-	Push	Pull	Pull	Can't
Port Number Used	53	80	25	110	143	20 (Data), 21 (Control)
In-band / Out-of-band	In-band	In-band	In-band	In-band	In-band	Out-of-band

Standard Port Numbers and Transport Protocols of Application Layer Services:

Application	Port Number	Transport Protocol
DNS	53	UDP
HTTP	80	TCP
FTP	20 (Data connection) 21 (Control)	TCP
SMTP	25	TCP
POP	110	TCP
SNMP	161, 162	UDP
IFTP	69	UDP
IMAP	143	TCP
Teinet	23	TCP
DHCP	67 (DHCP Server) 63 (DHCP Client)	UDP

Common Commands Used in Application Layer Protocols

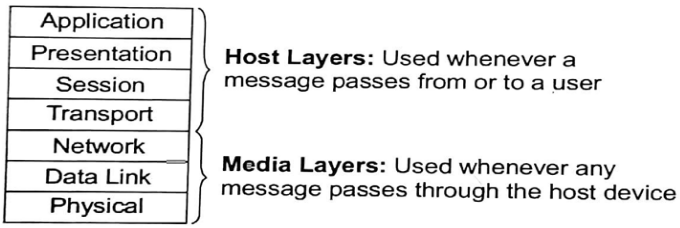
Application	Port Number	Transport Protocol
DNS	53	UDP
HTTP	80	TCP
FTP	20 (Data connection) 21 (Control)	TCP
SMTP	25	TCP
POP	110	TCP
SNMP	161, 162	UDP
TFTP	69	UDP
IMAP	143	TCP
Telnet	23	TCP
DHCP	67 (DHCP Server) 68 (DHCP Client)	UDP

OSI AND TCP/IP PROTOCOLS STACK

OSI Model

1. OSI stands for Open Systems Interconnection.
2. It was developed by ISO in 1984.
3. It is a standard model to divide a communication system into 7 layers.
4. Each layer has similar types of functions and communicates with the layer above and below it.
5. OSI model is mainly used as a reference for designing protocols and devices to ensure compatibility and communication between systems.

7 Layers of OSI Model (Top to Bottom)



Physical Layer

- It is the 1st layer of the OSI model.
- It works at hardware level only.
- It sends bits (0s and 1s) over the network using physical medium.
- Deals with hardware like cables, switches, etc.
- Connects devices with physical media (like fiber, copper wire).
- Controls things like voltage, data speed, distance, and connection type.

Data Link Layer

- Works at Layer 2 of OSI model.
- Transfers frames from one node (hop) to the next.
- Converts:
  - Bits ↔ Packets (Frames)
- Uses Layer 1 (Physical Layer) to send bit stream.

Main Functions:

- Framing – Breaks bit stream into frames.
- Error Control – Detects and handles errors in transmission.
- Flow Control – Manages data speed to prevent overflow.
- Access Control – Controls which device can use the link.
- Physical Addressing – Adds MAC address to identify devices.

Parts of Data Link Layer:

- LLC (Logical Link Control)
  → Flow control, Error control, Synchronization
- MAC (Media Access Control)
  → Framing, Physical addressing, Access control

Network Layer

- Works at Layer 3 of the OSI model.
- Delivers packets from source to destination across networks.
- Hides routing details from upper layers.

Main Functions:

- Host-to-Host Connectivity – Sends data between different devices (hosts).
- Logical Addressing – Assigns IP addresses to identify devices.
- Routing – Chooses the best path to send data.
- Switching – Manages packet switching in the network.
- Fragmentation & Reassembly –
  - Breaks large data into smaller packets.
  - Reassembles packets at the receiver.
- Congestion Control – Avoids overload in the network.

Other Responsibilities:

- Translates logical address → physical address (e.g., IP → MAC).
- Handles packet delivery, error handling, and network traffic issues.

Transport Layer

- Works at Layer 4 of the OSI model.
- Responsible for process-to-process delivery (not just host-to-host).
- Make sure data reaches the right application on the destination device.
- Requests retransmission if packets are missing or corrupted.
- Sends acknowledgement after successful delivery.

Main Functions:

- End-to-End Connectivity – Delivers data between two devices completely.
- Service Point Addressing – Uses port numbers to reach the correct process.
- Flow Control – Controls speed to avoid overwhelming receivers.
- Error Control – Detects and corrects errors, ensures no data loss/duplication.
- Segmentation & Reassembly –
  - Splits data into smaller parts (segments).
  - Rejoins segments at the receiver.
- Congestion Control – Prevents overload in the network.
- Connection Control – Supports both connection-oriented (TCP) and connectionless (UDP) communication.
- Multiplexing & Demultiplexing – Allows multiple applications to use the network at the same time.

Session Layer

- Layer 5 of OSI model.
- Acts as a network dialog controller.
- Starts, manages, and ends communication sessions between devices.
- Only authorized users can join the session.
- Helps in restarting communication smoothly after interruption.

Main Functions:

- Authentication & Authorization – Confirms who can join the session.
- Synchronization (Checkpoints) – Saves session state, so data can resume from the last point if failure occurs.
- Dialog Control – Decides who sends/receives and for how long.

Presentation Layer

- Layer 6 of OSI model.
- Acts as a translator between application and network.
- Handles protocol conversion and graphics/audio format handling.
- Ensures smooth communication between different systems/platforms.

Main Functions:

- Character Translation – Converts data formats from sender to receiver.
- Encryption / Decryption – Secures data during transfer.
- Compression – Reduces data size for fast transmission.

Application Layer

- Layer 7 of OSI model – Topmost layer.
- Directly interacts with the user and applications.

Main Functions:

- Provides network services to users (e.g., email, file transfer, web browsing).
- Supports applications like Mail services, File sharing, etc.
- Gives access to network-based apps (e.g., browser, FTP client).
- Represents what the user sees or uses.

Examples of Services:

- Email (SMTP, POP3, IMAP)
- Web browsing (HTTP, HTTPS)
- File transfer (FTP)
- Remote access (Telnet)

OSI LAYER PROTOCOLS

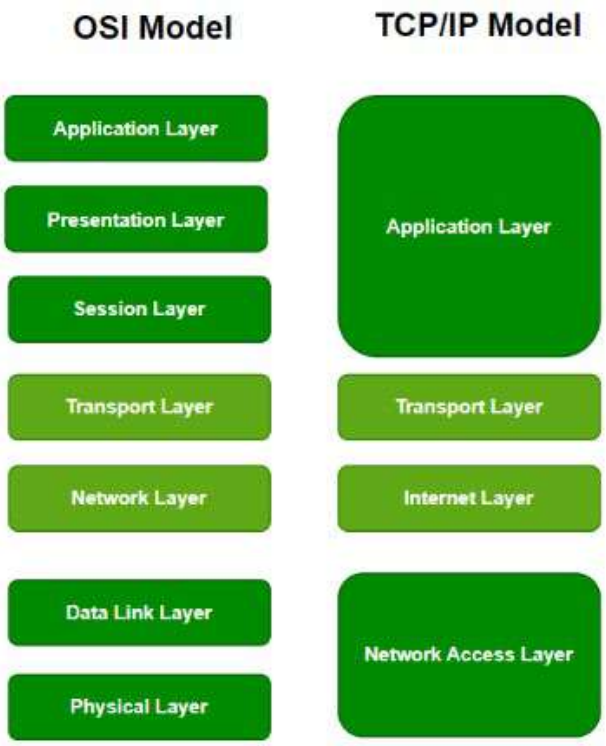
OSI Layer Protocols

Layer	Protocols
Application	NNTP, SIP, SSI, DNS, FTP, Gopher, HTTP, NFS, NTP, SMTP, DHCP, SNMP, Telnet, Netconf
Presentation	MIME, XDR, TLS, SSL
Session	Named Pipes, NetBIOS, SAP, SIP, L2TP, PPTP
Transport	TCP, UDP, SCTP, DCCP
Network	IP (IPv4, IPv6), ICMP, IPsec, IGMP, IPX, AppleTalk
Data Link	ATM, SDLC, HDLC, ARP, CSLIP, SLIP, PLIP, IEEE 802.3, Frame Relay, ITU-T G.hn, PPP, X. 25
Physical	EIATIA-232, EIA/TIA-449, ITU-T V-Series, I.430, I.431, IEEE 802.3, SONET/SDH, PON, OTN, DSL, IEEE 802.1, IEEE 802.15, IEEE 802.16, IEEE 1394, ITU-T G.hn PHY, USB, Bluetooth

TCP/IP Model

There are 5 layers of the TCP/IP model.

1. Physical Layer
2. Data Link Layer
3. Network Layer
4. Transport Layer
5. Application Layer



Physical Layer

- Lowest layer of TCP/IP model.
- Deals with actual transmission of raw bits over physical media (cables, wireless signals).
- Defines electrical and mechanical specifications like voltage, cable type, connectors.
- Examples: Ethernet cables, Fiber optics, Radio waves.

Data Link Layer

- Provides node-to-node data transfer.
- Frames data packets from the Network layer into frames for transmission.
- Handles error detection and correction on the link.
- Uses MAC addresses to identify devices on the same physical network.
- Examples: Ethernet, Wi-Fi (802.11), PPP.

Network Layer (Internet Layer)

- Responsible for logical addressing and routing.
- Delivers packets from source to destination across multiple networks (routing).
- Main protocol: IP (IPv4, IPv6).
- Also uses ARP, RARP for address mapping and ICMP for error reporting.
- Handles fragmentation and reassembly of packets.

Transport Layer

- Provides end-to-end communication between applications on hosts.
- Ensures reliable or unreliable data delivery depending on protocol.
- Main protocols:
    - TCP (Transmission Control Protocol): reliable, connection-oriented, error-checked.

- UDP (User Datagram Protocol): faster, connectionless, no guarantee.

- Performs flow control, error control, segmentation and reassembly.

Application Layer

- Highest layer, directly interacts with user applications.
- Provides services like email, file transfer, remote login, web browsing.
- Supports many protocols:
  - HTTP, FTP, SMTP, DNS, Telnet, POP3, IMAP, SNMP.
- Responsible for data formatting, encryption, and session management as needed.

Comparison between the OSI Model and TCP/IP Model

Parameter	OSI Model	TCP/IP Model
Full Form	Open Systems Interconnection	Transmission Control Protocol/Internet Protocol
Layers	7 layers	4 layers
Usage	Low usage	Mostly used
Approach	Vertical approach	Horizontal approach
Delivery	Guaranteed delivery of data	Delivery not guaranteed
Replacement	Easy to replace tools and change	Harder to replace tools
Reliability	Less reliable	More reliable
Protocol Example	Various protocols for each layer	Common protocols like HTTP, FTP, TCP, UDP
Error Handling	Built into Data Link & Transport layers	Built into protocols like TCP
Connection Orientation	Both connection-oriented and connectionless at Transport	TCP (connection-oriented), UDP (connectionless)

GeeksforGeeks

GATE CSE BATCH

KEY HIGHLIGHTS:

- 300+ HOURS OF RECORDED CONTENT
- 900+ HOURS OF LIVE CONTENT
- SKILL ASSESSMENT CONTESTS
- 6 MONTHS OF 24/7 ONE-ON-ONE AI DOUBT ASSISTANCE
- SUPPORTING NOTES/DOCUMENTATION AND DPPS FOR EVERY LECTURE

COURSE COVERAGE:

- ENGINEERING MATHEMATICS
- GENERAL APTITUDE
- DISCRETE MATHEMATICS
- DIGITAL LOGIC
- COMPUTER ORGANIZATION AND ARCHITECTURE
- C PROGRAMMING
- DATA STRUCTURES
- ALGORITHMS
- THEORY OF COMPUTATION
- COMPILER DESIGN
- OPERATING SYSTEM
- DATABASE MANAGEMENT SYSTEM
- COMPUTER NETWORKS

LEARNING BENEFIT:

- GUIDANCE FROM EXPERT MENTORS
- COMPREHENSIVE GATE SYLLABUS COVERAGE
- EXCLUSIVE ACCESS TO E-STUDY MATERIALS
- ONLINE DOUBT-SOLVING WITH AI
- QUIZZES, DPPS AND PREVIOUS YEAR QUESTIONS SOLUTIONS

ENROLL NOW

TO EXCEL IN GATE
AND ACHIEVE YOUR DREAM IIT OR PSU!

ENROLL NOW

STAR MENTOR CS/DA

KHALEEL SIR
ALGORITHM & OS
29 YEARS OF TEACHING EXPERIENCE

CHANDAN SIR
DIGITAL LOGIC
GATE AIR 23 & 26 / EX-ISRO

SATISH SIR
DISCRETE MATHEMATICS
BE in IT from MUMBAI UNIVERSITY

MALLESHAM SIR
M.TECH FROM IIT BOMBAY
AIR – 114, 119, 210 in GATE
(CRACKED GATE 8 TIMES)
14+ YEARS EXPERIENCE

VIJAY SIR
DBMS & COA
M. TECH FROM NIT
14+ YEARS EXPERIENCE

PARTH SIR
DA
IIIT BANGALORE ALUMNUS
FORMER ASSISTANT PROFESSOR

SAKSHI MA'AM
ENGINEERING MATHEMATICS
IIT ROORKEE ALUMNUS

SHAILENDER SIR
C PROGRAMMING & DATA STRUCTURE
M.TECH in Computer Science
15+ YEARS EXPERIENCE

AVINASH SIR
APTITUDE
10+ YEARS OF TEACHING EXPERIENCE

AJAY SIR
PH.D. IN COMPUTER SCIENCE
12+ YEARS EXPERIENCE