BYJU'S
EXAM PREP

Computer Science & IT

Computer Networks

SHORT NOTES

Short Notes — COMPUTER NETWORKS

- **Default subnet Mask:**

  Class A: 255.0.0.0

  Class B: 255.255.0.0

  Class C: 255.255.255.0

- **IP Addressing**

  Class A → 0 → (1 - 126), No. of IP Addresses = 231

  Class B → 10 → (128 - 191), No. of IP Addresses = 230

  Class C → 110 → (192 - 223), No. of IP Addresses = 229

  Class D → 1110 → (224 - 239), No. of IP Addresses = 228

  Class E → 1111 → (240 - 255), No. of IP Addresses = 228

- **Private Addresses Range:**

  10.0.0.0 to 10.255.255.255 — 1 class A Network

  172.16.0.0 to 172.31.255.255 → 16 class B Network

  192.168.0.0 to 192.168.255.255 → 256 class C Network

CLASS	**NUMBER OF NETWORKS**	**NUMBER OF HOSTS**
Class A	2^7 -2=126	2^{24} -2 = 1,67,77,214 hosts
Class B	2^{14} =16384	2^{16} -2 = 65,534 hosts
Class C	2^{21} =20,97,125	2^8 -2 = 254 hosts
Class D	No NID and HID, all 28 remaining bits are used to define the multicast address.	
Class E	No NID and HID, it is meant for research and future purpose.	

Layers	**Data Units**	**Functions**
Application Layer	Data	Mail Services, Directory Services, FTAM
Presentation Layer	Data	Encryption/Decryption, Compression
Session Layer	Data	Session Establishment, Synchronization, Dialog Controller
Transport Layer	Segments, Datagram	Segmentation
Network Layer	Packets	Traffic control, Fragmentation, Routing
Data Link Layer	Frames	Flow control, Error control, Access control
Physical Layer	Bits	Bit Synchronization, Bit rate control, Physical Topologies

OSI Model	DoD Model	Protocols	Devices / Apps
Layer 5,6,7	Application	DNS, DHCP, NTP, SNMP, HTTPS, FTP, SSH, TELNET, HTTP, POP3...etc.	Web server, Mail server, Browser, Mail client ...etc.
Layer 4	Host to Host	TCP | IDP	Gateway
Layer 3	Internet	IP, ICMP, IGMP	Router, Firewall layer 3, Switch
Layer 2	Network access	ARP (MAC), RARP'	Bridge, Layer 2 switch
Layer 1		Ethernet, Token ring	Hub

	NID	**HID**		
1.	–	0's	→	Network ID
2.	–	1's	→	DBA
3.	1's	1's	→	LBA
4.	0's	–	→	Host with in the Network
5.	1's	0's	→	Network Mask or Subnet Mask

- **CIDR RULES**

  1. All the 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 the size of the block.

- **NETWORK TOPOLOGIES**

  There are four types of network topologies:-

  1. Bus Topology
  2. Ring Topology
  3. Star Topology
  4. Mesh Topology

- **TRANSMISSION MODES**

  There are three types of transmission modes: -

  1. Simplex Mode
  2. Half-Duplex Mode
  3. Full-Duplex Mode

- **DELAYS IN COMPUTER NETWORKS**

  There are four different kinds of delay that occur during the transmission:

  1. Transmission delay
  2. Propagation delay
  3. Queuing delay
  4. Processing delay

- **TRANSMISSION DELAY (T_d) –**

  $$\text{Transmission delay} = \frac{\text{Length / Size of data packet}}{\text{Bandwidth of Network}}$$

- **PROPAGATION DELAY (P_d) –**

  $$\text{Propagation delay} = \frac{\text{Distance between sender and receiver}}{\text{Transmission speed}}$$

- **QUEUING DELAY –**

  The time spent by the data packet waiting in the queue before it is taken for the execution is called the queuing delay.

- **PROCESSING DELAY –**

  The time taken by the processor to process the data packet is called the processing delay.

- **TOTAL DELAY –**

  The total delay in sending one data packet or End to End time =

  Transmission delay + Propagation delay + Queuing delay + Processing delay

- **THROUGHPUT**

$$\text{Throughput} = \text{Efficiency} \times \text{Bandwidth}$$

$$\text{Throughput} = \frac{T_t}{T_t + 2xTp} \times \text{Bandwidth}$$

$$\text{Throughput} = \frac{L/B}{T_t + 2xd/v} \times B$$

$$\text{Throughput} = \frac{L}{2xd/v}$$

- **EFFICIENCY**

$$\text{Efficiency } (\eta) = \frac{(\text{Transmission delay})_{\text{packet}}}{(\text{Transmission delay})_{\text{packet}} + 2 \text{ x Propagation delay}}$$

OR

$$\text{Efficiency } (\eta) = \frac{T_t}{T_t + 2\,T_p}$$

OR

$$\text{Efficiency } (\eta) = \frac{1}{1 + 2\left(\frac{T_p}{T_t}\right)}$$

OR

$$\text{Efficiency } (\eta) = \frac{1}{1+2a}, \text{ where } a = \left(\frac{T_p}{T_t}\right)$$

- **FLOW CONTROL**

  1. Propagation Delay (P_d) = Distance/Velocity or d/v

  2. Transmission Delay (T_d) = Length of packet / Bandwidth or L/B

  3. $RTT = T_{d(data)} + 2*P_d + T_{d(Ack)} + P_{rd} + Q_d$

  - Optimal window size = 1 + 2a
  - Minimum seq. No required = 1 + 2a
  - Min no. of bits required = $\lceil \log_2(1+2a) \rceil$ in the sequence No. field.

N = Sender's Window Size. (in SR, both sender and receiver window are the same)

a = T_p / T_t

1. Sequence No. >= (Sender's Window Size) + (Receiver's Window Size)
2. Efficiency in TDM (polling) = $T_t / (T_{poll} + T_t)$
3. In CSMA/CD, $T_t >= 2*T_p$
4. Hence, minimum frame length = $2*T_p*B$
5. In CSMA/CD, Efficiency = 1/(1 + 6.44a)

- **MAXIMUM DATA RATE(CHANNEL CAPACITY) FOR NOISELESS AND NOISY CHANNELS**
  - **Noiseless Channel: Nyquist Bit Rate**
  - Bit Rate = 2 * Bandwidth * log2(L)
  - where L is the number of signal levels used to represent data.
  - **Noisy Channel: Shannon Capacity**
  - Capacity = bandwidth * log2(1 + SNR)
  - where SNR is the signal-to-noise ratio

	Stop & wait	**GBN**	**SR**
Efficiency	η = 1/1+2a or η = useful time/ Total time or η = T_d/RTT	η = N/1+2a or η = useful time/ Total time or η = N*T_d/RTT	η = W_S/1+2a or η = useful time/ Total time or η = $W_S * T_d$/RTT
Throughput	Length of data Pkt/RTT or η * B	N*Length of data Pkt/RTT Or η * B	W_S * Length of data Pkt/RTT Or η * B
Buffer	1 + 1	N + 1	N + N
Seq No.	2	N + 1	2N
Sequence No. = K bit		$W_S / 2^K - 1$ $W_R / 1$	$W_S / 2^{K-1}$ $W_R / 2^{K-1}$

- **BACKOFF ALGORITHM FOR CSMA/CD**
  1. Waiting time = back–off time
     Let n = collision number or re-transmission serial number.
     Then, Waiting time = $K * T_{slot}$
     where K = [0, 2^n – 1]
  2. N = No. of stations
     Early Token Reinsertion : Efficiency = 1/(1 + a/N)
     Delayed Token Reinsertion : Efficiency = 1/(1 + (N+1)a/N)
  3. Pure Aloha Efficiency = 18.4 %
     Slotted Aloha Efficiency = 36.8%
- **Steps to Calculate SWP Problem:**
  1. Calculate RTT
  2. Based on the given Bandwidth and RTT calculate No. of bits we are able to transfer within RTT and Equate it as a window in terms of bits (W_{bits})=B*RTT
  3. W_{pkt} or W_p = W_{bits} /Packet Size(bits)
  4. Minimum sequence No. required = W_p
  5. $2^K = W_P$
     Where K= number of bits required in the sequence number

- **ERROR CONTROL**
  1. To detect 'd' bit error minimum Hamming distance required = d +1
  2. To correct 'd' bit error minimum Hamming distance required = 2d + 1
  3. In Hamming code No. of redundant bit or check bits or Parity bits:

$$r=(m+r+1) <= 2^r \text{ (lower limit)}$$

- **CRC**
  1. If the generator has more than one term and the coefficient of x° is 1, all single-bit errors can be detected.
  2. If a generator cannot divide x^t+2 (t between 0 and n-1), then all isolated double errors can be detected.
  3. A generator that contains a factor of x+1 and detects all odd-numbered errors.

  A good polynomial generator needs to have the following characteristics:-
  1. It should have at least two terms.
  2. The coefficient of the term x° should be 1.
  3. It should not divide x^t+1 for t between 2 and n-1.
  4. It should have the factor x+1.

- **ACCESS CONTROL**

1. Minimum size of the frame to detect the collision in Ethernet (CSMA/CD)

$$T_d >= 2 * P_d + T_{d(\text{Jam signal})}$$

2. Backoff Algorithm

Waiting time = K * Slot duration

= K * RTT

= K * 2 * P_d

Range of K = 0 to 2^{n-1}, where n is the collision number

3. Efficiency in Ethernet (CSMA/CD)

$\eta = 1/1+6.44a$ or η = Useful time / Total time

= T_d / (Collision time + T_d + P_d)

4. $P(1-P)^{N-1}$ = Probability of success for single station

$N(1-P)^{N-1}$ = Probability of success for any station among all station [Throughput of the channel]

5. Ethernet [Packet Size]

Min Size	**Max size**
46	1500 [Data]
64	1518 [Frame]

- **NETWORK LAY**

Address Class	**First octet (Decimal)**	**First octet bits (red bits don't change)**	**Network (N) and Host (H) portion**	**Default Subnet Mask**
A	1-127	00000000-01111111	N.H.H.H	255.0.0.0
B	128-191	10000000-10111111	N.N.H.H	255.255.0.0
C	192-223	11000000-11011111	N.N.N.H	255.255.255.0
D	224-239	11100000-11101111	n/a (multicast)	
E	240-255	11111111-11111111	n/a (experimental)	

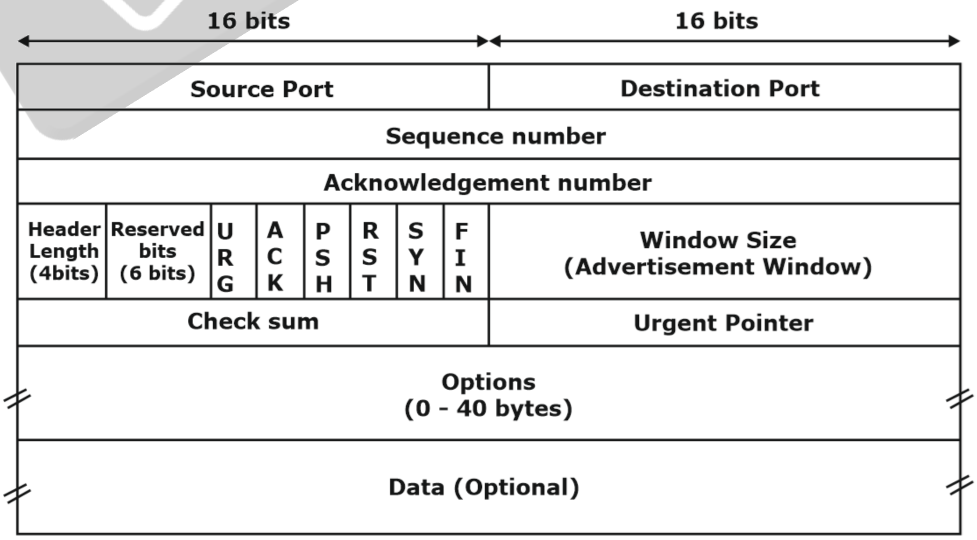
- **DISTANCE VECTOR ROUTING VS LINK STATE ROUTING**

Distance Vector Routing (DVR)	Link State Routing (LSR)
Sends the entire table	Sends only link state information
Slow convergence	Fast convergence
Susceptible to routing loops	Less susceptible to routing loops
Updates are sometimes sent using broadcast	Always uses multicast for the routing updates
Doesn't know the network topology	Knows the entire network topology
Simpler to configure	Can be harder to configure
Examples: RIP, IGRP	Examples: OSPF, IS-IS

- **TCP 3-WAY HANDSHAKE PROCESS**
  - STEP 1: SYN
  - STEP 2: SYN + ACK
  - STEP 3: ACK

- **TCP HEADER**

Part No.	Name
0-1023	Well known port No.
1024-49151	Registered Port No.
49152-65535	Dynamic



TCP Header

SYN = 1 → 1 Seq Number

ACK = 1 → 0 Seq Number

FIN = 1 → 1 Seq Number

1 Data byte → 1 Seq Number

SYN	Ack	Meaning
1	0	request
1	1	reply
0	1	Ack
0	0	Data

- Wrap Around Time (WAT) = Total sequence Number / Bandwidth
  (Bandwidth in byte/sec)
- Min sequence number required to avoid wrap around time with in Life time
  = B* LT
- Min number of bits required to avoid wrap around time with in LT
  $= = \lceil \log_2 B * LT \rceil$

- **TIME OUT TIMER IN TCP**

Basic Algorithm	Jacobson's Algorithm
TO = 2 * RTT NRTT = α(IRTT) + (1 − α) ARTT $0 \le \alpha \le 1$	TO = 4 * ID + RTT NRTT = α (IRTT) + (1 − α) ARTT $0 \le \alpha \le 1$ AD = |IRTT - ARTT| ND = α (ID) + (1 − α)AD

- **CONGESTION CONTROL**

Slow start	Congestion Avoidance	Congestion Detection
1. If Ack Arrives $W_c = W_c + 1$	1. IF Ack Arrives $W_c = W_c + 1/W_c$	1. Time out
2. After one RTT $W_c = 2 * W_c$	2. After one RTT $W_c = W_c + 1$	2. 3 duplicate Ack

- **TOKEN BUCKET**

  Maximum Avg rate for Token Bucket (m) = (c+rt)/t

  m/1 = (c+rt)/t

  mt = c+rt,

  mt – rt = c

  (m-r)t = c

  t = c/(m-r)

  c → Token Bucket Capacity

  r → Token Arrival Rate

- **SWITCHING**

  Switching are basically classified into two categories: -

  1. Circuit Switching
  2. Packet Switching

  - **CIRCUIT SWITCHING:**

    Total time = Setup Time(S) + T_d + P_d + Tear Down Time (T)

    TT = S + L/B + X.d/v + T

  - **PACKET SWITCHING:**

    X → Hop

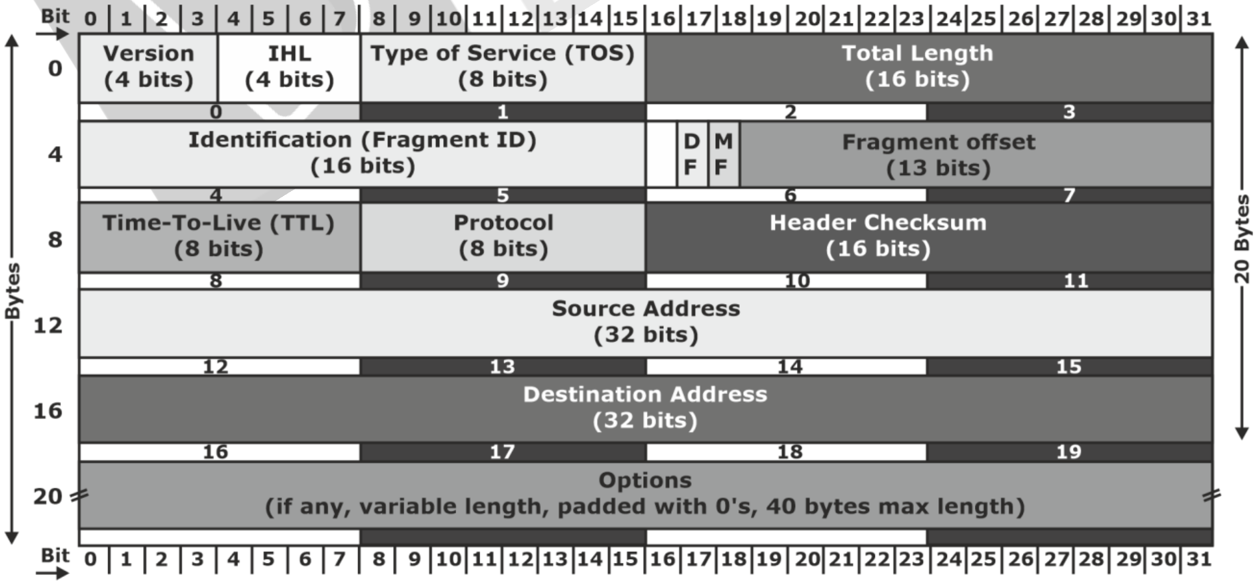
    N → Packet

    Total time = X[T_d + P_d] + X-1 [P_d + Q_d] + N-1[T_d]

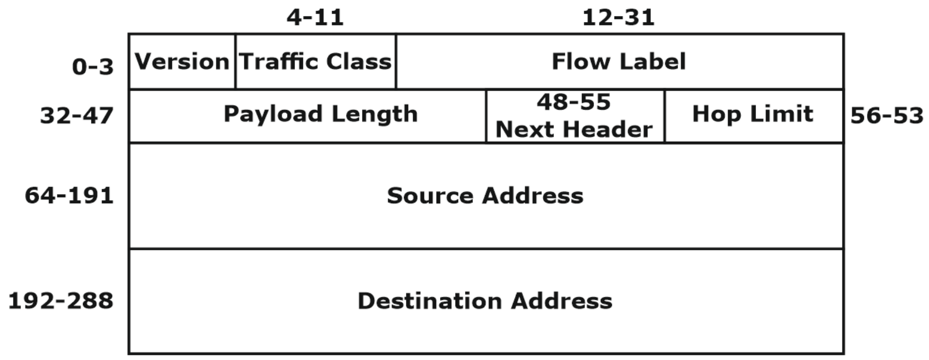
Application	**Port No**	**Transport Protocol**
DNS	53	UDP
HTTP	80	TCP
FTP	20 (Data connection) 21 (Control connection)	TCP
SMTP	25	TCP
POP	110	TCP
SNMP	161, 162	UDP
TFTP	69	UDP
IMAP	143	TCP
Telnet	23	TCP

SHORT TRICK	DNS	HTTP	SMTP	POP	IMAP	FTP
Stateful/ Stateless	Stateless	Stateless	Stateless	Stateful	Stateful	Stateful
Transport Protocol Used	UDP	TCP	TCP	TCP	TCP	TCP
Connectionless/ Connection Oriented	Connection Less	Connection less	Connection oriented	Connection oriented	Connection oriented	Connection oriented
Persistent/Non-persistent	Non-persistent	HTTP 1.0 is non persistent HTTP 1.1 is persistent.	Persistent	Persistent	Persistent	The control connection is persistent. Data connection is non-persistent.
Push/Pull	-	-	Push	Pull	Pull	Can't
Port Number Used	53	80	25	110	143	20 for data connection. 21 for the control connection.
In-band/ Out-of-band	In		-band In-band In-band	In -band	In-band	Out-of-band

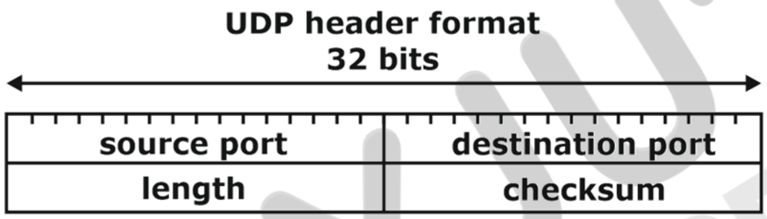
- **IPv4**



- **IPv6**



- **UDP HEADER**



- **DIFFERENCE BETWEEN TCP AND UDP**

TCP	UDP
Keeps track of lost packets. Makes sure that lost packets are re-sent	Doesn't keep track of lost packets
Adds sequence numbers to packets and reorders any packets that arrive in the wrong order	Doesn't care about packet arrival order
Slower, because of all added additional functionality	Faster, because it lacks any extra features
Requires more computer resources, because the OS needs to keep track of ongoing communication sessions and manage them on a much deeper level	Requires less computer resources
Examples of programs and services that use TCP: - HTTP - HTTPS - FTP - Many computer games	Examples of programs and services that use - UDP: - DNS - IP telephony

- **APPLICATION LAYER**

Service	Type	Direction.
DNS	UDP	Out
HTTP/HTTPS	TCP	Out
FTP	TCP/UDP	Out
TELNET	TCP/UDP	Out
POP3	TCP	Out
SMTP	TCP	Out
IRCU	TCP/UDP	Out
IDENT	TCP	In
Private File Service	TCP/UDP	In/Out
NNTP	TCP/UDP	Out
NTP	TCP/UDP	Out
Remote Desktop	TCP/UDP	In/Out

Application Type	Application-layer protocol	Transport Protocol
Electronic mail	Send: Simple Mail Transfer Protocol SMTP [RFC 821]	TCP 25
	Receive: Post Office Protocol v3 POP3 [RCF 1939]	TCP 110
Remote terminal access	Telnet [RFC 854]	TCP 23
World Wide Web (WWW)	Hypertext Transfer Protocol 1.1 HTTP 1.1 [RFC 2068]	TCP 80
File Transfer	File Transfer Protocol FTP [RFC 959]	TCP 21
	Trivial File Transfer Protocol TFTP [RFC 1350]	UDP 69
Remote file server	NFS [McKusik 1996]	UDP or TCP
Streaming multimedia	Proprietary (e.g., Real Networks)	UDP or TCP
Internet telephony	Proprietary (e.g., Vocaltec)	Usually, UDP

****