GATE CSE NOTES

by

Joyoshish Saha

CN 1

- **IP & subnetting - supernetting.**

→ Classful IP addressing (A-E) → Casting → IP vs MAC → Subnetting → Subnet mask → Var. - fixed length subnetting → Routing table (working) → VLSM → SMask usage in N/W resolution → Limitations of classful addressing → CIDR / Classless addressing (CIDR block rules) → Subnetting in CIDR (Fixed-Variable) → SM 255 → Given DBA ⇒ # subnets possible → Supernetting (conditions) → Private IP addresses.

- **Flow Control Methods.** DLL - GBN | TL - SR (TCP)

→ Delays (T_t, T_p, T_q, T_{proc}) → Stop & Wait protocol (η, Throughput, RTT, Total time) → S&W ARQ (TOT, SNo. for pkt & ack, SN 0&1, Solutions to lost data pkt, lost ack, delayed ack, damaged pkt - S&W ARQ discards duplicate packets $\eta = \frac{1}{1+2a}$ S&W ARQ uses NAK) → S&W vs S&N ARQ (NAK, TOT, SN) → Communication channels (capacity) → Sliding window protocol (optimal window size of sender $1+2a$, req. seq. no., choosing window size) → GBN Protocol ($W_S = N$, $W_R = 1$, Cumulative ack, ack timer, GBN doesn't accept corrupted or out-of-order frames & silently discards them, no NACK, retransmission of entire window, TOT > AT, $\eta = \frac{N}{1+2a}$ - problem of duplicate packets ⇒ #SN = N+1) → SR Protocol ($W_S = W_R = N$, indep. ack, does not accept corrupted frame & uses NAK for retransmission, implicit & explicit retransmission request, accepts out-of-order packets, sorting at R., searching at S, retransmission after TOT expiry, $\eta = \frac{N}{1+2a}$) → Comparison of flow control protocols.*

- **Access Control Methods.** (at DLL)

→ Communication links (P2P, broadcast) → Topologies of CN (SRMBH) → TDM (fixed size intervals allocated in round robin manner, time slot = $T_t + T_p$, $\eta = \frac{1}{1+a}$, effective bandwidth) → Polling (Polling algo chooses one station to let send data, $\eta = T_t/(T_{poll} + T_t + T_p)$ → CSMA/CD ($T_t \geq 2T_p$ used in early Ethernet, Jam signal - 48b, back-off time, back-off limit, $2T_p$ time wasted at max for each collision, $\eta = \frac{1}{1+6.44a} = \frac{T_t}{c \times 2T_p + T_t + T_p}$, used in wired LANs, CSMA/CD only minimises recovery time, does not prevent collision, Binary exponential backoff algo., backoff time increases exp"y, collision prob. decreases exp"y, Capture effect: Winner always wins) → Token Passing (Ring latency $\frac{d}{v} + \frac{Nb}{B}$, Token holding time, $\eta = \frac{NT_t}{T_p + N(THT)}$, Delayed Token & Early token reinsertion, DTR - hold token until pkt transmitted by it takes complete revolution of the ring & comes back to it - THT = T_t + Ring latency

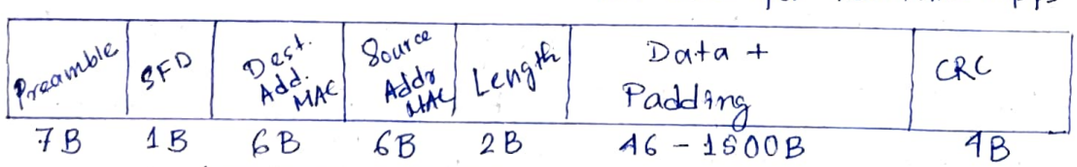
THT = $T_t + T_p$ (bit delay = 0), $\eta = \frac{NT_t}{T_p + N(T_t + T_p)}$, ETR - release token immediately after putting pkt to the ring - THT = $T_t - \eta = \frac{NT_t}{T_p + NT_t}$, ETR has higher efficiency → Aloha (Pure Aloha - $\eta = Ge^{-2G}$ - $\eta_{max} = 18.4\%$, Slotted Aloha - $\eta = Ge^{-G}$ - $\eta_{max} = 36.8\%$)

• ~~Access~~ Error Control Methods. (DLL or TL)

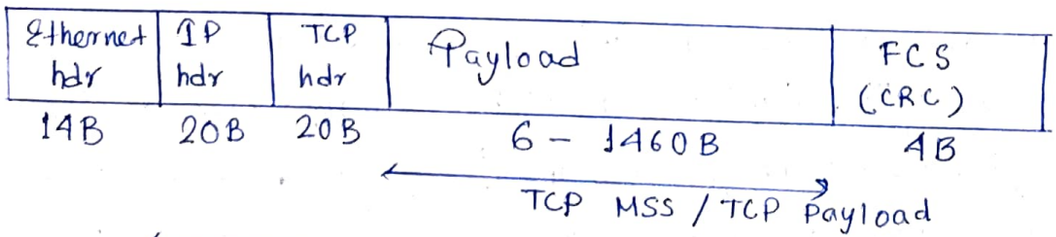
→ Redundancy bits → Simple parity checking (Even-odd-parity bit - can detect odd no. of bit errors) → 2D parity checking → CRC (Properties of CRC generator (! divisible by x, divisible by x+1), detect single bit errors - double bit - odd bit errors - burst errors) → CRC steps (@S - CRC code - add n-1 zeros to data with CRC code - divide data with CRC code - attach remainder to data - @R - divide received data with CRC code - check if rem = 0 ✓ ≠ 0 ✗) → Checksum (for m bit csum, segments of m bits data are added (use wrap around) - then 1's complement to get csum - sent with data - @R sum of all segments + csum ?= 0) - Meaningful error (unable to detect)

• ISO/OSI Model → PDNTSPA → Information exchange → Encapsulation (@ n-1 layer of n layer) → PL (Bit rate control, Bit sync., transmission mode, topology, encoding & signaling) - Encoding (NRZ-L/I, Biphase Manchester, Differential Manchester, 4B/5B → DLL (Node-Node delivery of msg, framing, physical addressing, flow - error - access control LLC & MAC sublayer, Bit - Byte stuffing) → NL (Host 2 host connectivity, switching, routing, logical addressing, congestion control, fragmentation) → TL (Process 2 process delivery, service point addressing, segmentation & reassembly, connection control, connection control, flow control, error control, multiplexing - demux'ng) → SL (Dialogue control, checkpoints - sync) → PL (Syntax - semantics of information, translation, encryption, compression) → AL (User interface)

• LAN Technologies. → Ethernet (DLL) (Bus topology, CSMA/CD, Manchester encoding, not used for real time apps as high no. of collisions



64 - 1518 B size. Ethernet frame.



CN 2

- **Switching** → Circuit, Msg, Packet (Virtual circuit, Datagram)
  PL — Store & forward — DLL — NL
  → Optimal packet size in packet switching → In CS: $T_p > T_t$
  → In PS: $T_t > T_p$. → In packet switching, after (#hops × T_t) time (when 1 packet is delivered to R), every T_t time the R will receive a packet because of pipelining.

- **IPv4**

IPv4 header

Ver (4)	HLen (4)	TOS (8)	TL (16)
Ident. (16)	0	DF	MF	F. offset (13)
TTL (8)	Protocol (8b)	HCsum (16b)
SIP (32)	DIP (32b)
Options (0–40B)

20B (4B × 5)

→ HLen (Scaling factor 4)
→ Frag. offset: # of data bytes ahead of it in the original unfragmented datagram
→ Scaling factor $2^{16}/2^{13} = 8$
→ TTL: Max #hops allowed to reach dest^n @ dest^n TTL should be ≥ 0.
→ Protocol: ICMP-1, IGMP-2, TCP-6, UDP-17
→ Elimination of datagrams from buffer – ICMP > IGMP > UDP > TCP
→ HCsum: of entire header (only)
→ Record route (9), Source routing (9), Padding (In worst case, 3B of dummy data).

- **Fragmentation** → Check size of datagram, MTU of dest^n N/W, DF bit → Changes made (TL field, MF ← 1 except last pkt, F. offset field) → Amt. of data sent in 1 frag. is chosen such that i) it's as large as possible but ≤ MTU ii) it's multiple of 8 (for F. offset field → Last frag. can have any amt. of data.
→ Reassembly algorithm. → Fragmentation overhead. → Fragm^n done at routers (intermediary devices)

- **Protocols @ NL** → Internet (NL) does not have concept of broadcasting, only LANs have. → NAT (Private ⇄ Public) → ARP (Know other host's MAC given their IP) → Localhost → RARP (Know your own IP given you know your MAC, # IP > active hosts @ RARP table) → BOOTP (RARP + centralised table + relay agent) → DHCP (BOOTP + dynamic table, DORA)
→ ICMP (Error handling – feedback – TTL exceed, parameter problem, Source quench, Source redirect, dest^n unreachable; Req. & Reply – Echo, Time stamp, N/W mask, router solicitation & advertisement) – ICMP for TCP, UDP both – ICMP for 1st fragment only – Application of ICMP: Traceroute, PMTUD, Recordroute (using TTL exceed ICMP) – PMTUD using DHU ICMP DF=1.

- **Routing** → DVR (Distributed Bellman-ford (dest^n, weight, NH): $D(v) = \min\{D(v), D(w) + c(w,v)\}$
  ~~LSR~~ → (Dijkstra's SS shortest path: $d_x(y) = \min_v\{c(x,v) + d_v(y)\}$
  & flooding of LS packets

• Transport layer protocols. → TCP (Options field - Time stamp, window scale extension, param. negotiation, padding)

TCP header -

Source port 16b	Dest^n port 16b
Seq. no. 32b	
Ack. no. 32b	
H.Len 4b | Res. 6b | URG | ACK | PSH | RST | SYN | FIN	Adv. window size 16b
CSum 16b	Urg. ptr 16b
options + Padding (0-40B)	

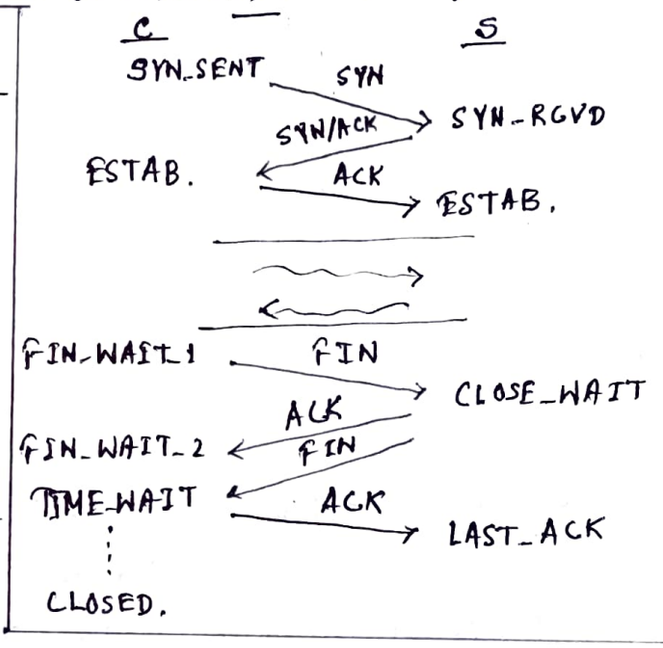
Checksum

Pseudoheader. -

Src. addr. (IP) 32b		
Dst^n addr 32b		
Res. 8b	Protocol from IP 8b	TCP segment len. 16b

Wrap around time, Life time (180s) NAT > LT to avoid same SN for bytes.

→ 3 way handshake (SYN-req, SYN-ACK-reply, ACK - pure ack) → SYN=1 - 1 SN, ACK=1 0 SN, FIN=1 - 1 SN, 1 data byte - 1 SN → TCP Conn^n termination (FIN, ACK, FIN, ACK)



→ TCP flow control (W_S = min (cwnd, rwnd))

→ TCP error control (CSum, Ack, retransmission - after TOT expires, after revng 3 duplicate acks) → TCP Congestion control (Congestion policy - SS, CA, CD - SS: W_c init = 1 MSS, cwnd / W_c = 2 x previous W_c till threshold = W_R/MSS / 2 - CA: linear $W_c = W_c + 1$ ÷ CD:

i) TOT expiry - thresh = $\frac{\text{current } W_c}{2}$ - W_c = 1 MSS - resume SS ii) 3 dup. acks - thr = $\frac{\text{current } W_c}{2}$ - W_c = thr. - resume CA) → TCP timers (TOT, Time wait, ACK timer, persistent (client), keep alive (server)) → Setting TOT (Actual RTT ↑ → TOT ↑) → Basic algo ($IRTT_1$ → $TOT_1 = 2\,IRTT_1$ → $IRTT_{n+1} = \alpha\,IRTT_n + (1-\alpha)\,ARTT_n$ → $TOT_2 = 2 \times IRTT_2$) → Jacobson's algo ($IRTT_1$, ID_1 → $TOT_1 = 4ID_1 + IRTT_1$ → $AD_1 = |IRTT_1 - ARTT_1|$ → $IRTT_{n+1} = \alpha\,IRTT_n + (1-\alpha)\,ARTT_n$ → $ID_{n+1} = \alpha\,ID_n + (1-\alpha)AD_n$ → $IRTT_2 = \alpha\,IRTT_1 + (1-\alpha)\,ARTT_1$ → $ID_2 = \alpha\,ID_1 + (1-\alpha)AD_1$ → $TOT_2 = 4ID_2 + IRTT_2$)

→ Karn's Mod^n (ack arrives after TOT goes off - no initial TOT value as ack is delayed - sol^n: When retransmission double TOT whenever TOT goes off & retransmit). → Traffic shaping (Leaky bucket - same avg rate always - use queue; Token bucket - bucket of tokens - tokens generated at each tick - has max capacity - ready packet acquires a token & goes out to the N/W - max o/p rate = $\frac{c + \rho \cdot t}{t}$) → Silly window syndrome (Clark's, Nagle's algo.)

→ UDP (Length - header + data, Csum on hdr, data, pseudo IP header.

Source port 2B	Dest. Port 2B
Length 2B	CSum 2B

Application - DNS, Trivial FTP (TFTP), broadcast - multicast, real time app, streaming, DHCP, SNMP. - does not guarantee in order delivery.

CN 3

- **Application layer protocols.**

DNS (Stateless, UDP, connectionless, non-persistent, port 53, inband), HTTP (Stateless, TCP, connectionless, HTTP 1.0 non persistent, HTTP 1.1 persistent, port 80, inband), FTP (Stateful, TCP, conn^n oriented, control conn^n persistent, data conn^n is non-persistent) out-of band, port 20 for data, 21 for control conn^n), (SMTP – Stateless, TCP, conn^n oriented, persistent, port 25, inband), POP (Stateful, TCP, conn^n oriented, persistent, port 110, inband).

- **Wifi** (Collision avoidance using RTS, CTS, Exposed & Hidden terminal problem, positive acknowledgement system, MACA).

- **N/W Security.** → Passive attacker, Active attacker (Masquerade, replay, modification, DoS) → Symmetric or secret key cryptography (DES, AES) → $a \equiv b \bmod n \Rightarrow (a-b) \mid n$ → Multiplicative inverse $ab \equiv 1 \bmod p \Rightarrow b \equiv a^{-1} \bmod p$ → a, b coprimes (gcd 1) $\Rightarrow \exists c \in I,\ ac \equiv 1 \pmod b$ → Euler's totient fn $\phi(n)$ = # of +ve integers upto n that are coprime to n $\phi(9) = 6$ (1, 2, 4, 5, 7, 8) → If n is prime $\phi(n) = n-1$ → If m, n coprimes, $\phi(m)\,\phi(n) = \phi(mn)$ → Euler's product formula $\phi(n) = n\left(1-\frac{1}{p_1}\right)\cdots\left(1-\frac{1}{p_K}\right)$ when $n = p_1^{a_1} p_2^{a_2} \cdots p_K^{a_K}$, p_i are prime factors → Euler's theorem for coprimes $a^{\phi(n)} \equiv 1 \bmod n$; a, n being coprimes → Fermat's little theorem $a^{\phi(n)+1} \equiv a \bmod n$; a, n being coprimes. → Asymmetric encryption algo (Diffie Hellman key exchange, RSA algo).

→ **RSA Algo** (pu: (e, n) pr: (d, n) ; @S, $c = p^e \bmod n$; @R, $p = c^d \bmod n = p^{ed} \bmod n$ - $ed \equiv 1 \bmod \phi(n)$; picking n, e, d : $n = p_1 p_2$ - p_1, p_2 large primes, pick $1 \le e \le \phi(n)$ s.t. $\gcd(e, \phi(n)) = 1$ so that $ed \equiv 1 \bmod \phi(n)$.

→ Residue class, primitive root, multiplicative order, discrete logarithm problem (finding x s.t. $a^x \equiv b \bmod p$, np-hard, p - prime, a, b non zero int) → one way function.

→ DHKE. 1. P_u: P, G. Pr_S: a Pr_R: b. 2. S: $x = G^a \bmod P$

R: $y = G^b \bmod P$ 3. Exchange x, y 4. secret key :

S: $y^a \bmod P$ R: $x^b \bmod p$ ⇒ algebraically $x^b \bmod p =$

$y^a \bmod P$. → Firewalls → Digital signatures.

- C - channel capacity, λ - arrival rate of frames (frames/sec), $1/\mu$ - no. of bits/frame ⇒ delay time $T = \frac{1}{\mu C - \lambda}$

If N subchannels $T' = \frac{N}{\mu C - \lambda}$