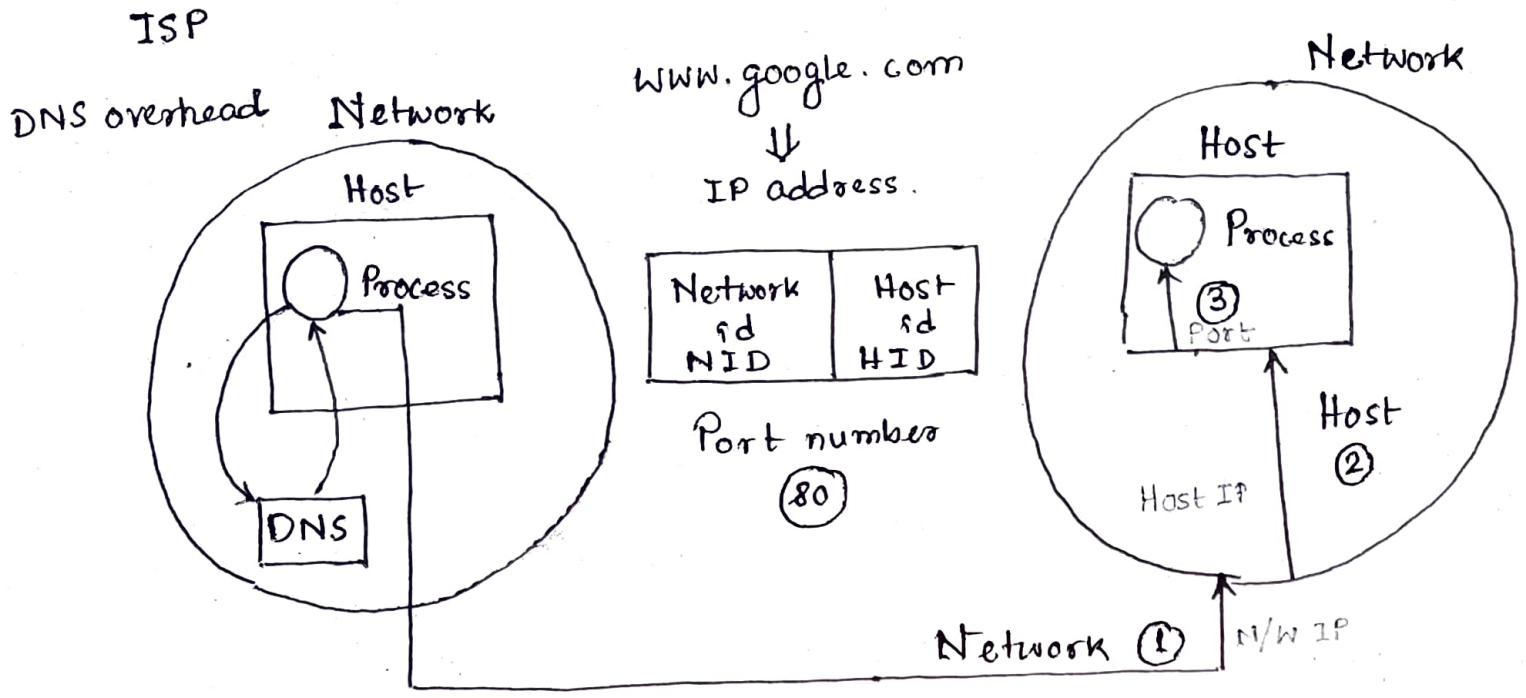
GATE CSE NOTES

by

Joyoshish Saha

IP address subnetting supernetting



→ Domain name service (DNS) provided by ISP to convert domain name to IP address.

→ Port number used to identify a particular process in the host. For well known services the port numbers are already predefined -

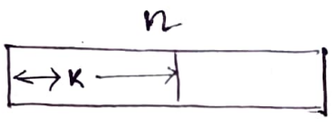
http : 80 smtp : 25 ssh : 22

ftp : 21, 20 https : 443 DHCP : 67, 68

→ ISP provides with DNS. Overhead for this conversion of domain name to IP address is called DNS overhead. To prevent this overhead, we have DNS cache which is a temporary storage of information about previous DNS lookups on a machine's OS or web browser. Keeping a local copy of a DNS lookup allows the OS or browser to quickly retrieve it and thus a website's URL can be resolved to its corresponding IP much more efficiently.

• Binary Number System.

If we choose K bits, the address space/number space will be divided into 2^K parts.



2^n combinations

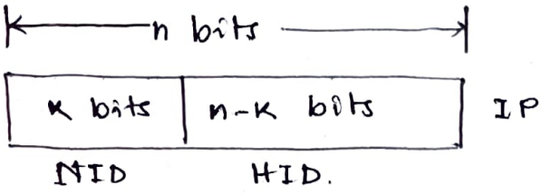
2^K parts

3 bits — Choosing 2 bits = 2^2 parts

000	Part 1 (00)
001	
010	Part 2 (01)
011	
100	Part 3 (10)
101	
110	Part 4 (11)
111	

2^K parts = 2^n numbers.

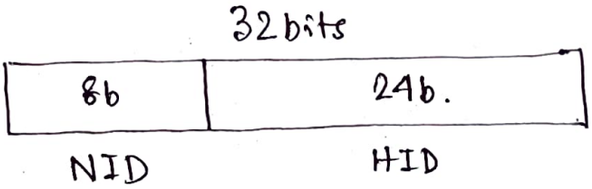
1 part = 2^{n-K} numbers. ⇒



⇒ Size of each N/W = 2^{n-K}

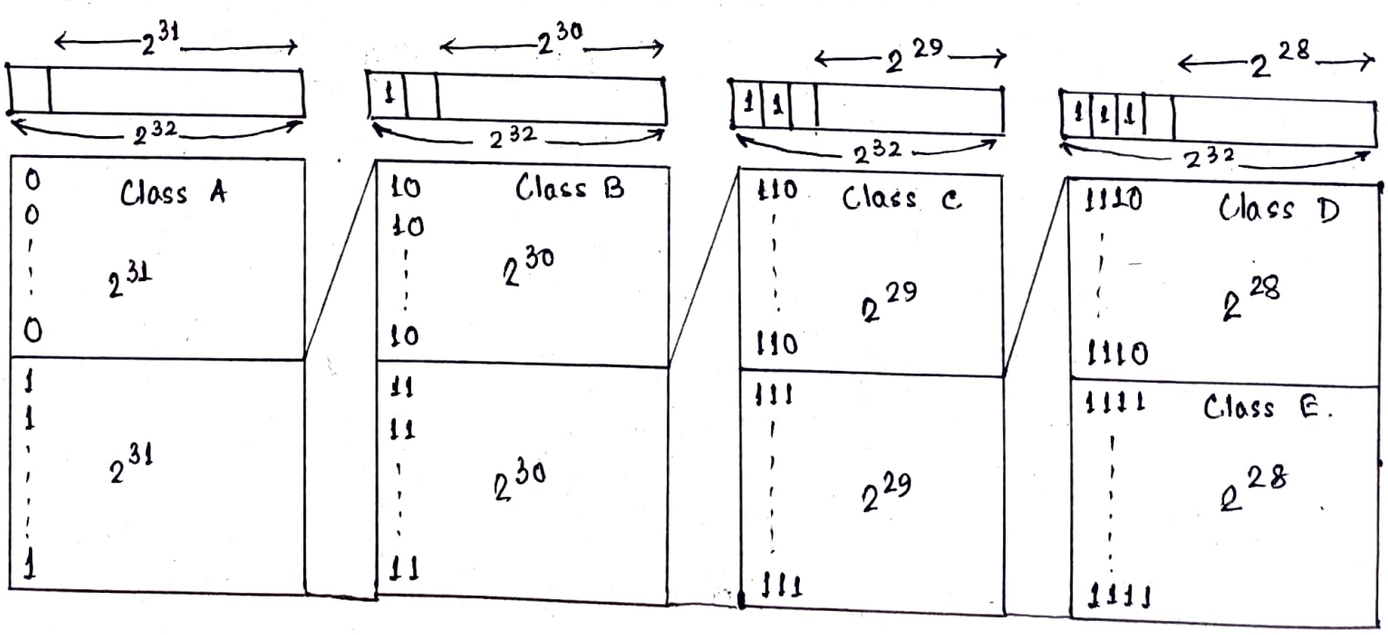
IP address size = 32 b

⇒ 2^{32} IP addresses possible.



2^8 = 256 Networks, each of size 2^{24} = 16 million hosts

* Classful IP address classification.



Dynamic host config protocol
Point-to-point protocol over Ethernet

* IP address. Internet Protocol Address.

Unique address assigned to each computing device on an network.

→ IP addresses types –

a) Static IP address : Once assigned to a N/W element always remains the same. Configured manually. Some ISPs don't provide static IP addresses. More costly than dynamic IP addresses.

b) Dynamic IP address : Temporarily assigned IP address to a N/W element. Can be assigned to a different device if it is not in use. DHCP or PPPoE assigns dynamic IP addresses.

→ IP address format.: 32bit binary address written as 4 numbers (octets) separated by dots. Octets divided into Net ID and Host ID. Net ID represents the IP address of the N/W & is used to identify the N/W. Host ID represents the IP address of the host & used to identify the host within the N/W.

→ Classes of IP addressing –

i) Classful

ii) Classless (CIDR)

→ Classful addressing

1. Class A : If the 32 bit address starts with bit 0.
NID 8 bits ; HID 24 bits.

IP addresses = 2^{31}.

Networks = $2^7 - 2 = 126$

hosts = $2^{24} - 2$ •

Range [1, 126] | 0, 127 unused.

Min	0000 0000	0
Max	0111 1111	127

Class A is used by organisations requiring very large N/Ws.

2. Class B : 32 bit address starting with '10' bit

NID 16 bits ; HID 16 bits

IP address = 2^{30}

Networks = 2^{14}

hosts = $2^{16} - 2$

Range [128, 191]

Min	10 00 0000	128
Max	10 11 1111	191

Used by organizations requiring medium N/Ws like IRCTC, banks etc.

3. Class C : 32 bit address starting with 110.

NID 24 bits; HID 8 bits

IP addresses = 2^{29}

Networks = 2^{21}

hosts = $2^{8} - 2$

Range [192, 223]

Min	110 00000	192
Max	110 11111	223

Used by organisations requiring small to medium size N/Ws. (like colleges, small offices etc.)

4. Class D : 32 bit address starting with 1110.

IP addresses = 2^{28}

Range [224, 239]

Min	1110 0000	224
Max	1110 1111	239

Class D is reserved for multicasting.

In multicasting, there's no need to extract host address from IP address.

5. Class E : 32 bit address starting with 1111.

Not divided into NID, HID

IP addresses = 2^{28}

Range [240, 255]

Min	1111 0000	240
Max	1111 1111	255

Reserved for experimental purposes.

NB

1. All the hosts in a single N/W always have the same NID, but different HID.

2. 2 hosts in 2 different N/Ws can have same HID.

3. A single N/W interface can be associated with more than one IP addresses.

4. No relation between IP and MAC address of a host.

5. IP address of a N/W is obtained by setting all bits for HID to zero.

6. Class A N/Ws accounts for half of the total available IP addresses.

✓ 7. In class A, 0.0.0.0 is reserved for broadcasting requirements and 127.0.0.1 is reserved for loopback address used for S/W testing.

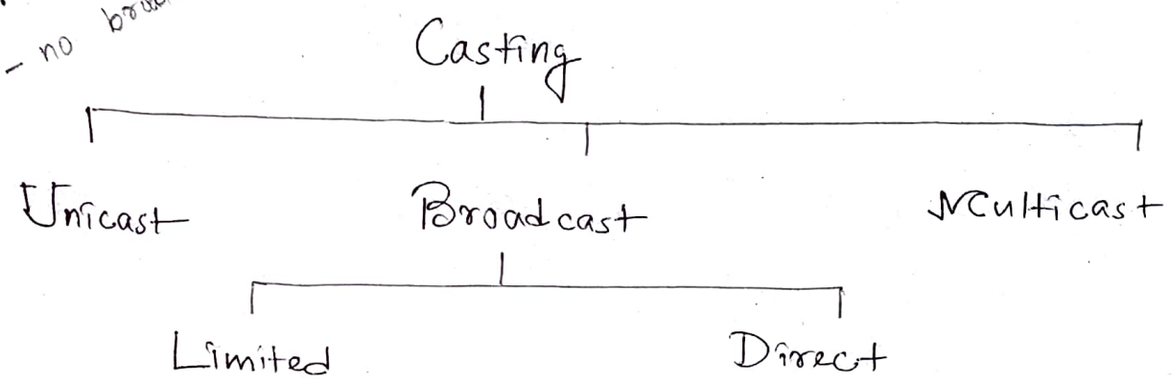
✓ 8. In all the classes, total number of hosts that can be configured are 2 less. When all HID bits are 0, it represents NID of the N/W, when all 1, it represents the DBA. (Direct Broadcast address)

9. Only those devices which have the network layer will have IP address.

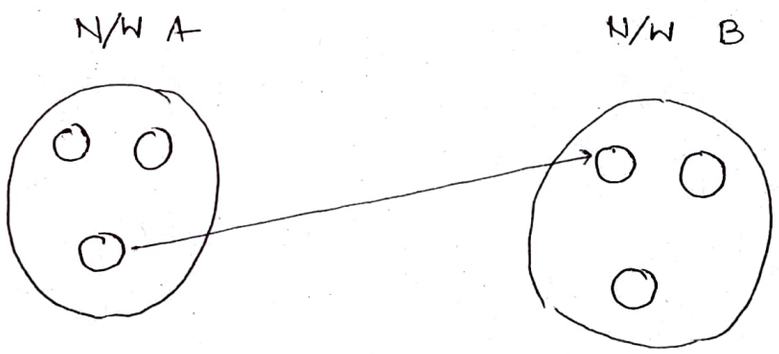
So, switches, hubs and repea-lers does not have any IP address.

* Casting Transmitting data (stream of packets) over the N/W.

(for IPv4 [IPv6 - no broadcasting])



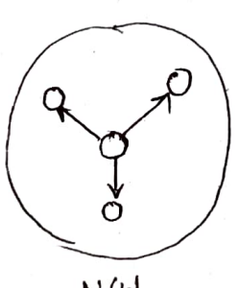
1. Unicast : Transmitting data from one source host to one destination host.
One-to-one transmission.



2. Broadcast : Transmitting data from one source host to all other hosts residing in the same or other N/W. One-to-all transmission.

Based on recipient's N/W,

a) Limited broadcast. Residing in same N/W.

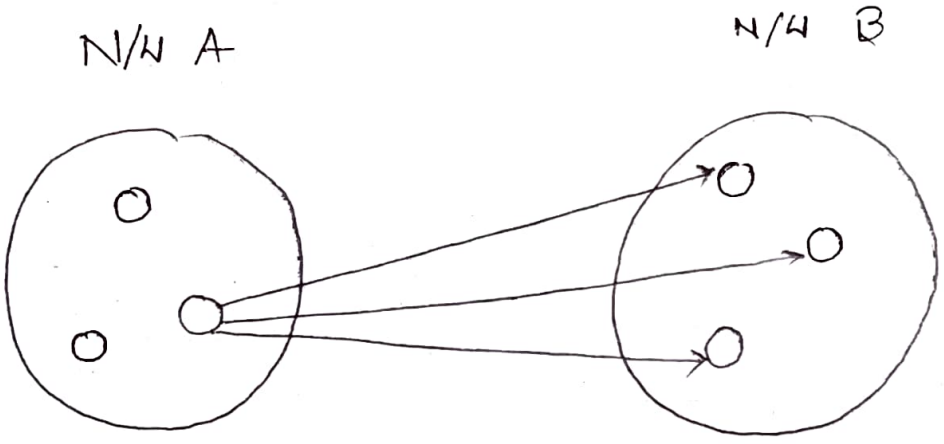


✓ LBA for any N/W = All 32 bits set to 1

Source address = IP address of host A

✓ Destn " = 255.255.255.255

b) Direct broadcast Residing in some other N/W.



DBA for any N/W is the IP address where N/W ID is the IP address of the N/W where all destination hosts are present, and Host ID bits are all set to 1.

eg. Host A (IP 11.1.2.3) sending data to all other hosts residing on the N/W having IP address 20.0.0.0.

Source address = 11.1.2.3

Dest^{n} address = 20.255.255.255.

3. Multicast : Transmitting data from one source host to a particular group of hosts having interest in receiving the data. One-to-many transmission.

eg. Sending a message to a particular group on WhatsApp, teleconference etc.

Multicast makes use of IGMP (Internet Group Management Protocol) to identify its group. Each group is assigned with an IP address from class D of IPv4.

* IP address vs. MAC Address.

12 digit HEX
6 byte Binary

Desc.	IP address 32b	MAC Address 48b
1. Acronym	Internet Protocol	Media Access Control.
2. Address type	Logical address. in N/W layer.	Physical address. in DLL.
3. Provided by	Assigned by user/admin, DHCP or ISP.	Hardware manufacturer (of NIC card)
4. Length	IPv4 uses 32 bit address (dotted notation), IPv6 uses 128 bit in Hex notation.	48 bit address that contains 6 group of 2 Hex digits, separated by hyphens, or colons.
5. Use of classes	IPv4 uses A, B, C, D, E classes for addressing.	No classes used.
6. Spoofing	IP address spoofing possible.	MAC address spoofing possible.
7. Type.	Software address.	H/W address
8. Work on	N/W layer of OSI model.	Data link layer of OSI model.
9. Used for	Numeric representation of a device that uses TCP/IP.	Numeric rep^{n} of a device that uses ethernet.
10. Subnetting	Used.	Not used.
11. Resolution	Address resolution protocol (ARP) used for resolving IP address into physical (MAC) address.	Reverse ARP (RARP) used for resolving physical (MAC) address into IP address.

4 segments
decimal.
192.14.32.10

6 segments
Hex 2 digits each segment
AA:BB:CC:DD:EE:11

$A \times 16^1 + A \times 16^0$
$= 10 \times 16 + 10$
$= 170_{10}$

NB

1.

Range of first octet	Class.
[1, 126]	A
[128, 191]	B.
[192, 223]	C
[224, 239]	D
[240, 254]	E.

NID. #bits

8	0000 0000 0
	1 — 0 — 128
16	11 — 0 — 192
	111 — 0 — 224
24	1111 0 240
	11111 0 248
	111111 00 252
	11111110 254
	11111111 255

2. For any given IP address, IP address of its N/W is obtained by setting all its host ID part bits to 0.

3. For any given IP address, DBA is obtained by setting all its host ID part bits to 1.

4. For any given IP address, LBA is obtained by setting all its bit to 1.

LBA = 255.255.255.255

5. Class D, E IP address are not divided into Net ID and host ID parts.

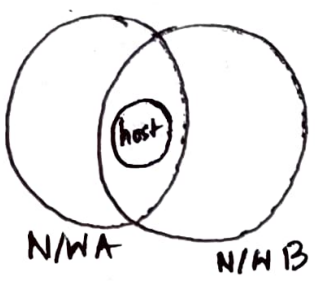
Q A device has to be having 2 or more IP addresses, the device is called -

1. Workstation
2. Router
3. Gateway
4. All.

→ All devices have a N/W layer. So, they will have at least one IP address. In TCP/IP suite, workstation and gateway have all the 5 layers. Router has only 3 layers last layer being network layer.

Workstation : A user may configure more than one IP addresses in his workstation. With more than one IP addresses, it remains present in more than one N/Ws. So, if one N/W goes down, it is always reachable from other N/Ws.

Note IP addresses are assigned to interfaces.



host present in 2 networks

When we buy a new laptop, we usually get 2-3 interfaces. Thus, a workstation can have more than one IP addresses.

Router : A router may be connected to various interfaces. Each interface has a unique IP address. Thus, a router may also have more than IP addresses.

Similare in the case with gateways because gateways are extension of routers.

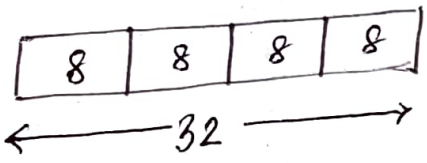
→ All 0 bits in HID → NID.

All 1 bits in HID → DBA
(Directed broadcast)

Number of IP addresses possible in a N/W of class A = 2^{31}, of class B = 2^{30}, of class C = 2^{29}, of class D, E = 2^{28}.

→ Dotted decimal representation of IP address.

32 bits in 4 parts of 8 bits.



→ Class A starts with 0

" B " " 10

" C " " 110

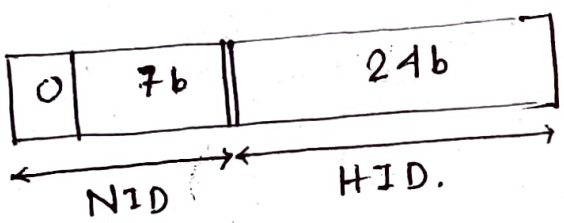
" D " " 1110

" E " " 1111.

Any prefix of a class can't be a prefix of any of the other,

eg. 0 is not a valid prefix of any other class except class A.

→ Class A. (1-126)



N/W = 2^7 = 128 (Theoretical)

Size of each N/W = 2^{24}

= 16 M (# host)

Big corporations will use class A.

1st octet

0 _ _ _ _ _ _ _

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 1 1

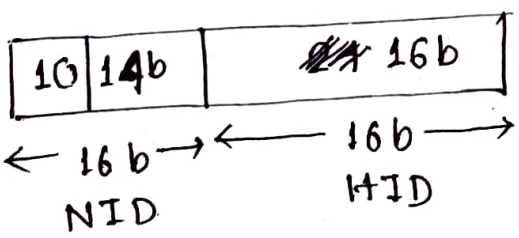
1 1 1 1 1 1 1 127

0, 127 are not used.

So, practically # N/W = 126

Range in class A 1 – 126.

→ Class B (128-191)



N/W = 2^{14} = 16K Networks.

ip addresses / N/W = 2^{16}

Range in class B 128 – 191

10 _ _ _ _ _ _

0 0 0 0 0 0 128

0 0 0 0 0 1 129

1 1 1 1 1 1 191

8b

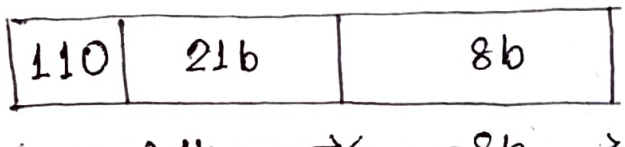
64 numbers × 2^8 = 2^{14} N/Ws

for 2nd octet of NID.

128.0
128.1
128.255
129.0
129.1
129.255
191.255

2^{14} N/Ws

→ Class C (192 - 223)



N/Ws = 2^{21}.

IP addresses / N/W = 2^8
= 256

Range of class C = 192 – 223

110 – – – – –

0 0 0 0 0 → 192
0 0 0 0 1 → 193
⋮
1 1 1 1 1 → 223

32×2^{16}
= 2^{21} N/Ws

192.0.0
192.0.1
192.0.1
⋮
192.0.255
192.1.0
⋮
192.1.255
⋮
192.255.255.

→ For class D and E, entire is left as IP address.

Class D (224 - 239)

1110 – – – –

0 0 0 0 → 224
⋮
1 1 1 1 → 239

Class E (240 - 255)

1111 – – – –

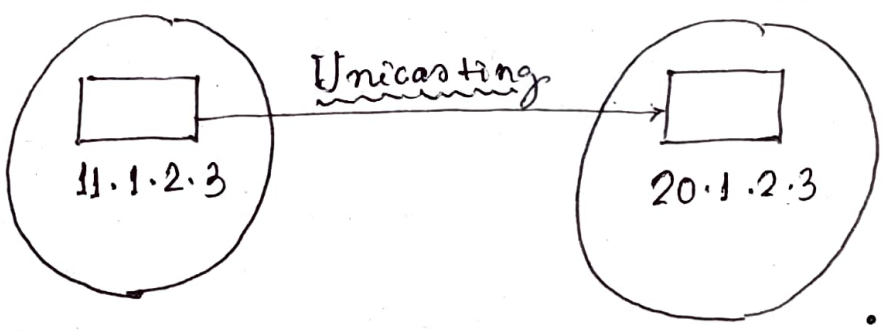
0 0 0 0 → 240
⋮
1 1 1 1 → 255

2^{28} IP addresses for class D, E.

→ #hosts will always be less by 2.

We don't use 2 IP addresses in every class (all 0s, all 1s).

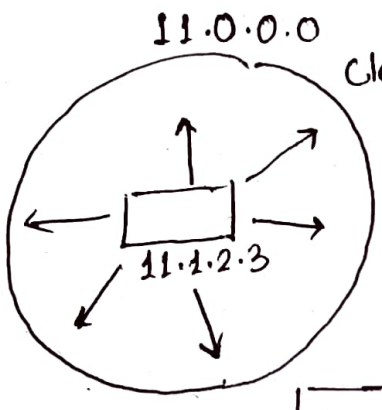
* Casting
  - → Unicast (One host to one host)
  - → Broadcast (One host to many hosts)
    - → Limited (same network broadcasting)
    - → Directed (different network broadcasting)



Data packet

Data	11.1.2.3	20.1.2.3
	Source address	Dest^n address

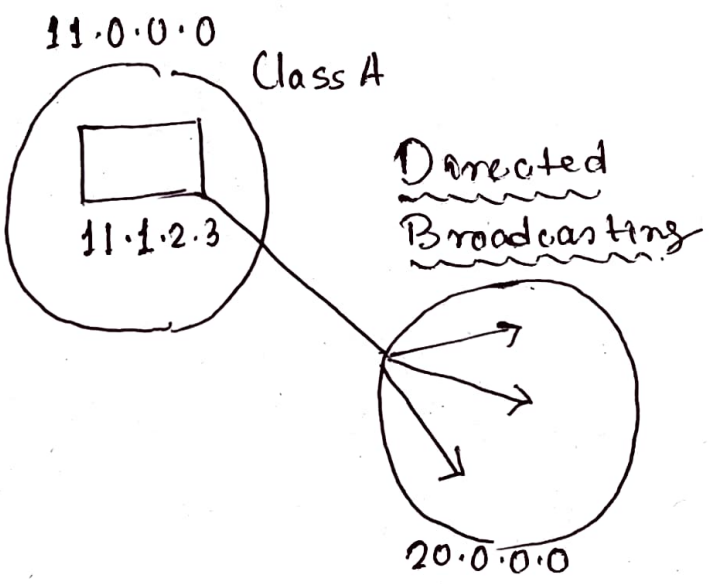
- In the host id, if there is all 0's, it designates the network. This is why, we don't use the 1st IP addr.



Limited Broadcasting.

Data packet.

Data	11.1.2.3	255.255.255.255



Data	11.1.2.3	20.255.255.255

Data packet.

- When dest^n addr. is 255.255.255.255 then the packet is sent to all hosts in the N/W.

Limited broadcasting addr.

LBA = 255.255.255.255

- In the host id, if there is all 1's, it means it is used for directed broadcasting for different N/W. This is why, we don't use this IP address.

Direct broadcast address : NID, HID(all 1's)

• Example. C_A : 1 – 126 8.24 C_c – 192 – 223 24,8

C_B : 128 – 191 16.16

IP	NID	DBA	LBA.
1.2.3.4.	1.0.0.0	1.255.255.255	255.255.255.255
10.15.20.60	10.0.0.0	10.255.255.255	n
130.1.2.3	130.1.0.0	130.1.255.255	n
150.0.150.150	150.0.0.0	150.0.255.255	n
200.1.10.100	200.1.10.0	200.1.10.255	n
220.15.1.10	200.15.1.0	200.15.1.255	n
250.0.1.2	—	—	—
300.1.2.3	—	—	—

* Subnetting

Dividing a single N/N into multiple subnetworks. It improves security and also the maintenance, administration of subnets are easy.

Each subnet has its unique network address known as its subnet ID. The subnet ID is created by borrowing some bits from the host ID part of the IP address. No. of bits borrowed depends on the no. of subnets created.

eg. 2 subnets. class c

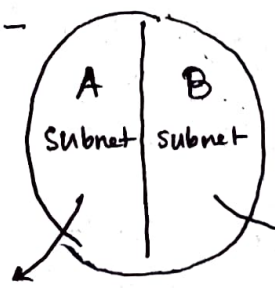
200.1.2.0

200.1.2.0 _____

Range 0 – 127 2^{7-1}

NID = 200.1.2.0

DBA = 200.1.2.127



200.1.2.1 _____

Range 128 – 255 2^{8-1}

NID = 200.1.2.128

Directed broadcast address (DBA)

= 200.1.2.255

200.1.2.x

SID	New HID

Ambiguity ~ NID of the main network and NID of the first subnet same. Also, DBA of the network and DBA of the last subnet are same [Which network or subnet to send packet, depends on situation. If packet is coming from outside, packet should be transmitted to all hosts of the network. If packet is coming from inside, packet should be transmitted to the subnet B's all hosts.

→ 200.1.2.69 S_A eg

Disadvantages of subnetting

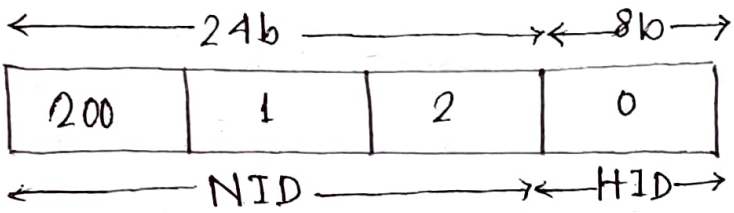
1. Subnetting leads to loss of IP addresses. 2 IP addresses are wasted for each subnet. One IP address is wasted for its network address (SID), another IP address is wasted for its direct broadcasting address (DBA).

2. Subnetting leads to complicated communication process. After subnetting, the communication process becomes complex involving the 4 steps —

1. Identifying the network
2. Identifying the subnet
3. Identifying the host
4. Identifying the process.

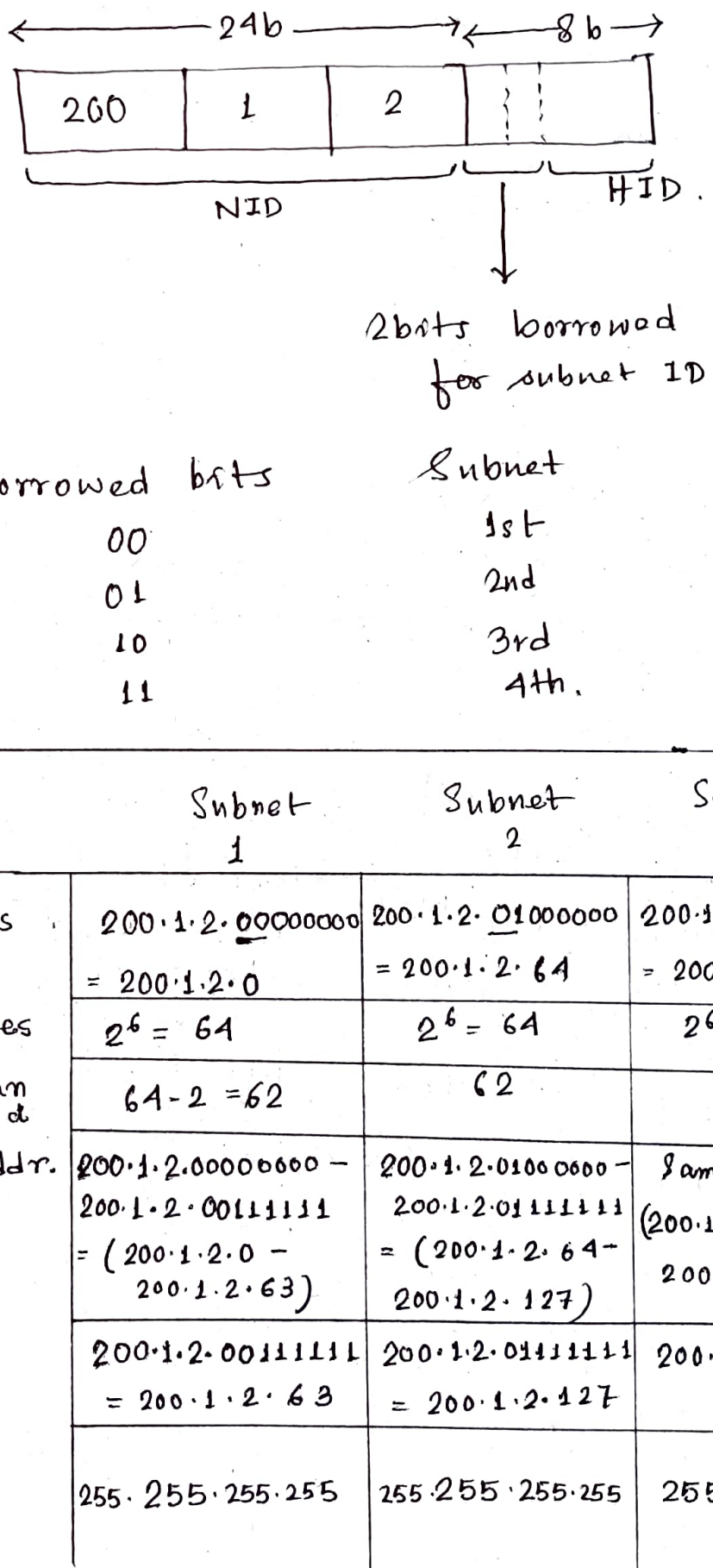
eg We have a big single network having IP address 200.1.2.0. We want to divide this N/W into 4 subnets.

N/W belongs to class C



For creating 4 subnets & to represent their subnet IDs, we require 2 bits.

So, we borrow 2 bits from the HID part.



Now,

Borrowed bits	Subnet	Subnet ID
00	1st	SNID /26
01	2nd	
10	3rd	
11	4th.	

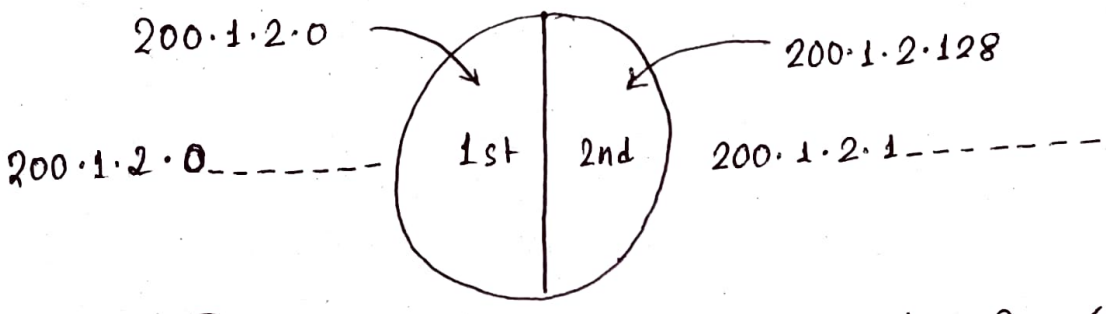
	Subnet 1	Subnet 2	Subnet 3	Subnet 4.
IP address	200.1.2.00000000 = 200.1.2.0	200.1.2.01000000 = 200.1.2.64	200.1.2.10000000 = 200.1.2.128	200.1.2.11000000 = 200.1.2.192
# of IP addresses	$2^6 = 64$	$2^6 = 64$	$2^6 = 64$	$2^6 = 64$
#hosts that can be configured	64-2 = 62	62	62	62
Range of IP addr.	200.1.2.00000000 – 200.1.2.00111111 = (200.1.2.0 – 200.1.2.63)	200.1.2.01000000 – 200.1.2.01111111 = (200.1.2.64 – 200.1.2.127)	Same way (200.1.2.128 – 200.1.2.191)	(200.1.2.192 200.1.2.255)
DBA	200.1.2.00111111 = 200.1.2.63	200.1.2.01111111 = 200.1.2.127	200.1.2.191	200.1.2.255
LBA	255.255.255.255	255.255.255.255	255.255.255.255	255.255.255.

eg. Big single N/W having IP 200.1.2.0. Subnetting into 3 subnets.

Subnetting will be performed in 2 steps:

1. Dividing the given network into 2 subnets
2. Dividing one of the subnets (1 or 2) into 2 subnets.

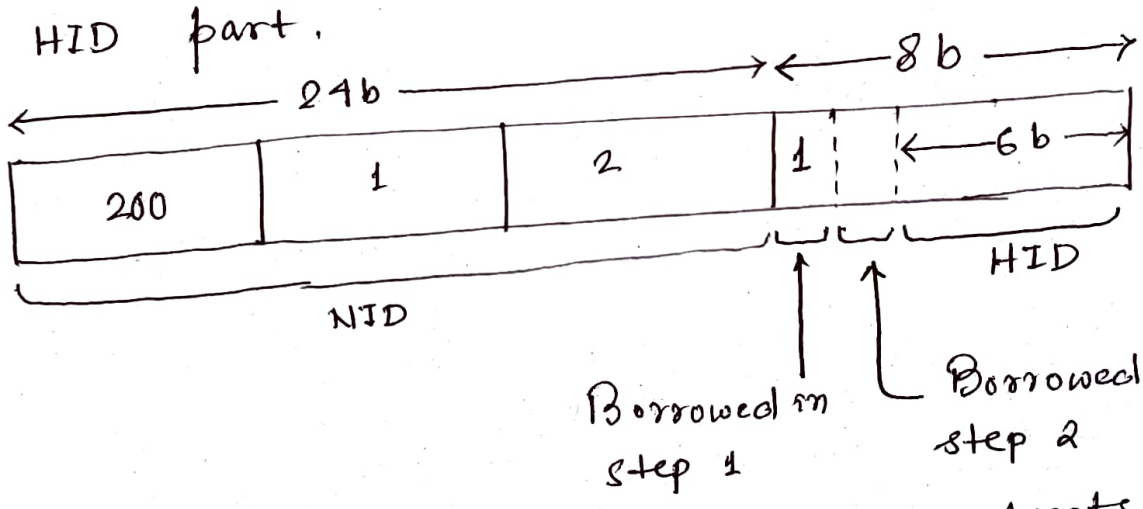
Step 1: Dividing N/W into 2 subnets :-



$2^x = 3$

Step 2: Dividing one subnet into 2 subnets :-

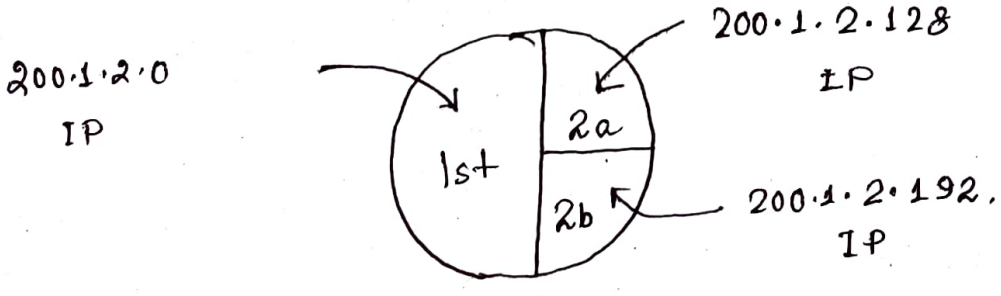
We do subnetting of the 2nd subnet (200.1.2.128) for creating 2 subnets & to represent their subnet IDs, we require 1 bit. So, we borrow 1 bit from HID part.



IP addresses of further divided subnets in the subnet 200.1.2.128 are

200.1.2.10 000000 = 200.1.2.128

200.1.2.11 000000 = 200.1.2.192.



	1	2a	2b
IP address	200.1.2.0	200.1.2.128	200.1.2.192
#IPaddress	$2^7 = 128$	$2^6 = 64$	$2^6 = 64$
#hosts that can be configured	128 - 2 = 126	64 - 2 = 62	64 - 2 = 62
Range of IP addr.	200.1.2.00000000 - 200.1.2.01111111 = (200.1.2.0 - 200.1.2.127)	200.1.2.10000000 - 200.1.2.10111111 = (200.1.2.128 - 200.1.2.191)	200.1.2.11000000 - 200.1.2.11111111 = (200.1.2.192 - 200.1.2.255)
DBA	200.1.2.127	200.1.2.191	200.1.2.255

- Subnet address = First IP address

  Broadcast address = Last IP address.

- **Subnet Mask** It is a 32 bit number which is a sequence of 1's followed by a sequence of 0's where -

  - 1's represent the global network ID part & the subnet ID part
  - 0's represent the host ID part.

Used to determine subnet

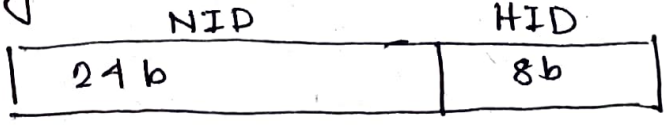
Calculating subnet mask -

For any given IP address, the subnet mask is calculated -

- by setting all the bits reserved for NID part & subnet ID part to 1.
- by setting all the bits reserved for HID part to 0.

eg. N/N having IP address 200.1.2.0

Class c IP.



Subnet mask -

11111111. 11111111. 11111111. 00000000

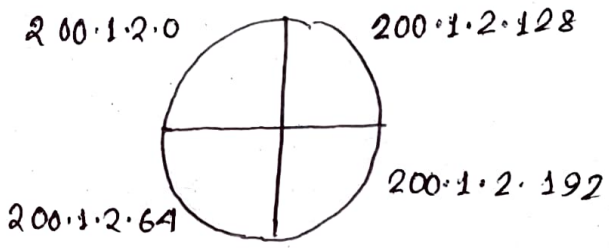
= 255. 255. 255. 0.

Use of subnet mask — ANDing with dest. IP

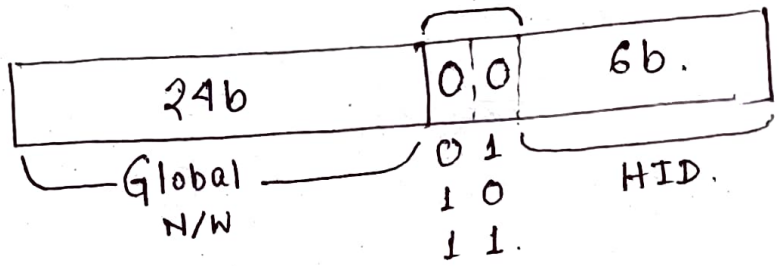
Subnet mask is used to determine to which subnet the given IP address belongs to.

eg. Single network 200.1.2.0 divided into 4 subnets.

Class C.



For each subnet



Subnet mask for each subnet in the global network 200.1.2.0 =

11111111 . 11111111 . 11111111 . 11 000000

= 255 . 255 . 255 . 192 .

Fixed length subnetting — all subnets have same subnet mask since the size of each subnet is same.

- Types of subnetting

1. Fixed length or classful subnetting

Divides the N/W into subnets where all the subnets are of same size, have equal no. of hosts, have same subnet mask.

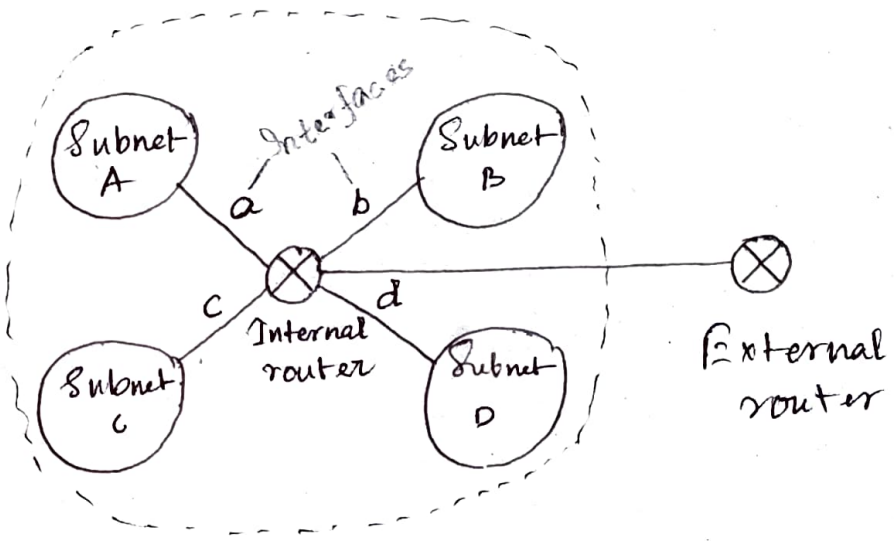
2. Variable length or classless subnetting.

Divides the N/W into subnets where all the subnets are not of same size, don't have equal no. of hosts, don't have same subnet mask.

0	A	1 - 126
10	B	128 - 191
110	C	192 - 223
1110	D	224 - 239
1111	E	240 - 254

Arrangement of subnets.

All the subnets are connected to an internal router. Internal router is connected to an external router. The link connecting the internal router with a subnet is an interface.



Physical view.

Working:

When a data packet arrives,

- External router forwards the data packet to the internal router.
- Internal router identifies the interface on which it should forward the incoming data packet.
- Internal router forwards the data packet on that interface.

* Routing Table.

A table is maintained by the internal router called routing table. It helps the internal router to decide on which interface the data packet should be forwarded.

Routing table consists of the following 3 fields —

1. IP address of the destination subnet
2. Subnet mask of the subnet
3. Interface.

eg. N/N subnetted into 4 subnets.

IP addresses –

1. 200.1.2.0 (S_A)
2. 200.1.2.64 (S_B)
3. 200.1.2.128 (S_C)
4. 200.1.2.192 (S_D).

Routing table –

Destination IP	Subnet mask (1100000)	Interface.
200.1.2.0	255.255.255.192	a
200.1.2.64	255.255.255.192	b
200.1.2.128	255.255.255.192	c
200.1.2.192	255.255.255.192	d
Default	0.0.0.0	e (default interface)

Working of Routing Table

When a data packet arrives to the internal router. –

Step 1 – Router performs bitwise ANDing of destⁿ IP address (mentioned in the data packet) and all the subnet masks one by one.

Step 2 – Router compares each result with their corresponding IP address of the destination subnet in the routing table. Now 3 cases may arise –

a) If there occurs only one match, router forwards the data packet the the corresponding interface.

b) If there occurs more than one match, router forwards the data packet on the interface corresponding to the longest subnet mask (longest run of 1's).

c) If there occurs no match, router forwards the data packet on the interface corresponding to the default entry.

→ NB

(i) In fixed length subnetting, all the subnets have the same subnet mask. So, bitwise ANDing is performed only once. If the result matches to any of the destn subnet IP address, router forwards the data packet on its corresponding interface. Otherwise, it is forwarded on the default interface.

(ii) In variable length subnetting, all the subnets don't have same subnet mask. So, bitwise ANDing is performed once with each subnet mask.

(iii) A host may also be directly connected to the router. In that case, there exists a host specific route from the router to the host. Router saves the IP address of that host in the Destination network column. Router saves 255.255.255.255 in the subnet mask column. ANDing of its destn address & subnet mask yields the IP address of the host. When a data packet arrives for that specific host, bitwise ANDing is performed. When the result of ANDing is the IP address of the host, packet is forwarded to its host specific route.

- (iv) Subnet mask for default route = 0.0.0.0
- Subnet mask for Host specific route = 255.255.255.255

eg.

Dest^n IP in packet: 11001000 . 00000001 . 00000010 . 10000010
200 . 1 . 2 . 130

24-2-6

AND

Subnet mask: 11111111 . 11111111 . 11111111 . 11000000
255 . 255 . 255 . 192.

11001000 . 00000001 . 00000010 . 10000000
200 . 1 . 2 . 128.

Dest^n subnet IP address.

Q. In a class B N/W on the internet has a subnet mask of 255.255.240.0. What is the max. no. of hosts per subnet ?

→ Binary rep^n of subnet mask.

11111111 . 11111111 . 1111 0000 . 00000000
NID. 20 b | HID 12b.

In class B, upper 16 bits form the N/W address & lower 16 bits form the host.

No. of hosts per subnet = $2^{12} - 2 = 4094$. (Ans)

[Also, # bits reserved for subnet ID = 20 - 16 b = 4 b

No. of subnets possible = $2^4 = 16$]

→ NB

1. Defaut subnet masks.

Class	SM.
A	255.0.0.0
B	255.255.0.0
C	255.255.255.0

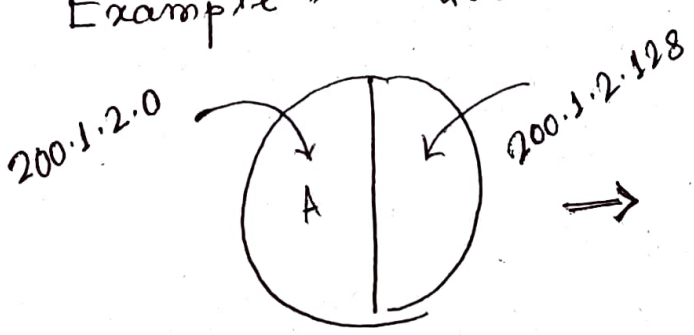
2. Network size is the total no. of hosts present in it. Networks of same size always have the same subnet mask. N/Ws of different size always have the different subnet mask.

3. For a N/W having larger size, its subnet mask will be smaller (no. of 1's less). For a N/W having smaller size, its subnet mask will be larger (no. of 1's more).

4. We should not use 255.255.255.15 like subnet masks. 15 ≡ 00001111. In this case, we can't divide the ranges in subnets properly.

• Variable length subnet masking (VLSM)

Example - 200.1.2.0. Class C.



For subnet A,

24+1 7
1's 0's

SM = 255.255.255.128

For subnet B and C,

24 + 2 bits 6 bits

1's 0's

SM = 255.255.255.192.

0000 0000	0
1000 0000	128
1100 0000	192
1110 0000	224
1111 0000	240
1111 1000	248
1111 1100	252
1111 1110	254
1111 1111	255 .

eg

	Subnet mask	# hosts	Subnets in class A	Subnets in class B	Subnets in class C
A	255.0.0.0	$2^{24}-2$	1	-	-
	255.128.0.0	$2^{23}-2$	2	-	-
	255.192.0.0	$2^{22}-2$	2^2	-	-
	255.240.0.0	$2^{20}-2$	2^4	-	-
B	255.255.0.0	$2^{16}-2$	2^8	1	-
	255.255.254.0	2^9-2	2^{15}	2^7	-
C	255.255.255.0	2^8-2	2^{16}	2^8	1
	255.255.255.224	2^5-2	2^{19}	2^{11}	2^3
	255.255.255.240	2^4-2	2^{20}	2^{12}	2^4
			2^{27-8}	2^{27-16}	2^{27-24}

as 8b NID ⇒ rem. 19b for subnets

8. Routing table.

Dest^n IP	SM	Interface.
144.16.00	255.255.0.0	eth0
144.16.64.0	255.255.224.0	eth1
144.16.68.0	255.255.255.0	eth2 ✓
144.16.68.64	255.255.255.224	eth3.

Packet bearing destn addr. 144.16.68.117 will go to which interface ?

→ . 144.16.68.117 AND 255.255.0.0 = 144.16.0.0. (match)

144.16.68.117 AND 255.255.224.0 = 144.16.64.0 (match)

144.16.68.117 AND 255.255.255.0 = 144.16.68.0 (match)

144.16.68.117 AND 255.255.255.224 = 144.16.68.96

More than one match. 255.255.255.0 has longest run of 1's among the matches.

Router forwards the packet to the interface eth2.

Q. Default route can be described as —

i) Dest^n values of 0.0.0.0 in the routing table

ii) It can be used if N/W has only one next hop router.

iii) It's useful in keeping routing table small.

iv) All.

- When any host connects to the internet, ISP provides following 4 things to the host:

1. IP address – ISP assigns an IP address to the host so that it can be uniquely identified on the internet. (DHCP)

2. Default Gateway – Default router connected to the N/W in which the host is present, is the default gateway for the host.

3. Subnet mask – Used to determine to which N/W the given IP address belongs to.

4. Domain Name Service (DNS) – Used to translate the domain name into an IP address.

- Subnet mask use.

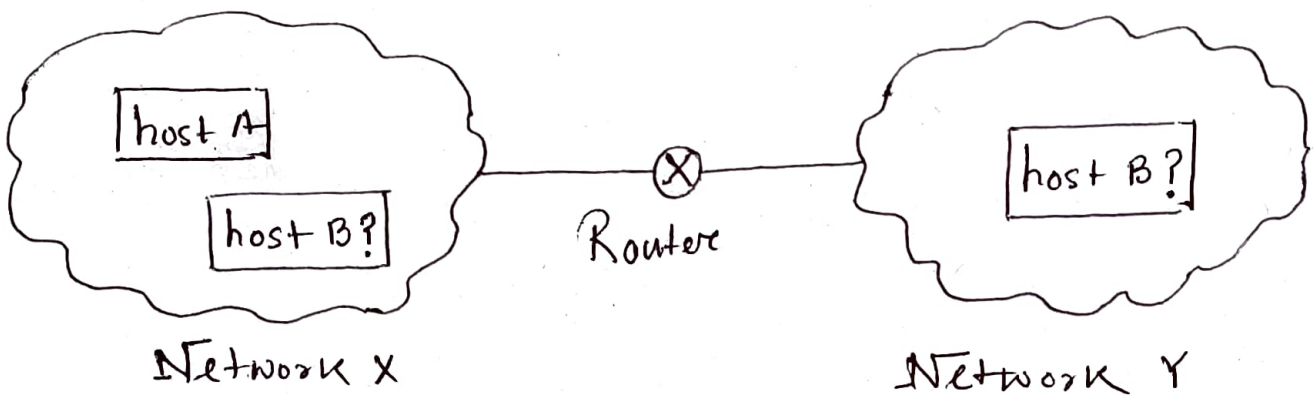
Host uses its subnet mask to determine whether the other host it wants to communicate with is present within the same N/W or not.

If the dest^n host is present within the same N/W, then source host sends the packet directly to the dest^n host.

If the destination host is present in some other N/W, then source host routes the packet to the default gateway. Router then sends the packet to the dest^{n} host.

Example –

There's a host A present in some N/W X. There's a host B. Host A wants to send a packet to host B. Before transmitting the packet, host A determines whether host B is present within the same N/W or not



To determine whether dest^{n} host is present within the same N/W or not, source host follows the following steps –

Source host A
Dest^{n} host B.

Step 1 Source host computes its own N/W address using its own IP address & subnet mask. After this, source host obtains its N/W address wrt itself. (AND opn)

Step 2 Source host computes the N/W address of dest^{n} host using dest^{n} IP address & its own subnet mask. After this, source host obtains the N/W address of dest^{n} host wrt itself. (AND opn)

Step 3 Source host compares the 2 results obtained –

cases : 1. If same, source host assumes that the dest^{n} host is present in the same N/W. Source host sends the packet directly to the dest^{n} host.

2. If different, source host assumes that the dest^{n} host is present in some other N/H. Source host sends the packet via router to the dest^{n} host.

NB

✓ i) Each host knows only its own subnet mask. It does not know the subnet mask of any other host.

✓ ii) The conclusion drawn by a host about the presence of other host within the same or other N/W might be wrong. (Subnetting can be different (CN-8 abr 20:00)

Considering, host A draws some conclusion about host B. Then, same conclusion might not be drawn by host B about host A. Both the hosts have to perform the above procedure separately at their ends to conclude anything.

Q. 2 computers C1 and C2 are configured as follows –

	IP	net mask
C1	203.197.2.53	255.255.128.0
C2	203.197.75.201	255.255.192.0

Which is true?

1. C_1 & C_2 both assume they are on same N/W.
2. ✓ C_1 assumes C_2 is on same N/W, but C_2 assumes C_1 is on a different N/W.

→ At C1,

C1 computes its N/W address

I_1 S_1

203.197.2.53 AND 255.255.128.0

= 203.197.0.0. NID_{11}

C1 computes N/W address of C2

I_2 S_1

203.197.75.201 AND 255.255.128.0

I_{21}

= 203.197.0.0 = C1 N/W address

So, C1 assumes that C2 is on the same N/W.

At C2,

C2 computes its network address

203.197.75.201 AND 255.255.192.0

= 203.197.64.0

C2 computes N/W address of C1

203.197.2.53 AND 255.255.192.0

= 203.197.0.0

Since, results are different, C2 assumes C1 is one a different N/W.

- Unforeseen limitations to classful addressing.

1. During the early days of the internet, the seemingly unlimited address space allowed IP addresses to be allocated to an organisation based on its request rather than its actual need. As a result, addresses were freely assigned to those who asked for them without concerns about the eventual depletion of the IP address space.

2. The decision to standardize on a 32 bit address space meant that there were only ~~232~~ 2^{32} (4,294,967,296) IPv4 addresses available. A decision to support a slightly larger address space would have exponentially increased the number of addresses thus eliminating the current address shortage problem.

3. The classful A, B and C octet boundaries were easy to understand & implement, but they did not foster the efficient allocation of a finite address space. Problems resulted from the lack of a network class that was designed to support medium-sized organisations.

→ NAT used to deal with these problems.

Class C

For example, a /24, which supports 254 hosts, is too small while a /16, which supports 65534 hosts, is too large. In the past, sites with several hundred hosts were assigned a single /16 address instead of 2 /24 addresses. This resulted in a premature depletion of the /16 network address space. Now, the only readily available addresses for medium-sized organisations are /24s, which have the potentially negative impact of increasing the size of the global internet's routing table.

Class B

* Classless Addressing / Classless Inter-Domain Routing

Improved IP addressing system. (CIDR).
Makes the allocation of IP addresses more efficient.
It replaces the older classful addressing.

CIDR block - When a user asks for specific number of IP addresses, CIDR dynamically assigns a block of IP addresses based on certain rules. This block contains the certain required number of IP addresses as demanded by the user. This block of IP addresses is called CIDR block.

Rules for creating CIDR block

1. All the IP addresses in the CIDR block must be ~~contiguous~~ contiguous.
2. The size of the block must be power of 2. Size of the block is total no. of IP addresses contained in the block.
3. First IP address of the block must be divisible by the size of the block.

→ If any binary pattern consisting of (m+n) bits is divided by 2^n, then

- remainder is least significant n bits
- quotient is most significant m bits.

So, any binary pattern is divisible by 2^n, iff its least significant n bits are 0.

eg. 100.1.2.64.

⇓

01100100.00000001.00000010.01000000

6 zeros

Divisible by 2^5 or 2^6

Not divisible by 2^7.

CIDR Notation.

CIDR IP address looks like a.b.c.d/n

Ends with a number, called IP N/W prefix.

IP N/W prefix tells the number of bits used for the identification of N/W. Remaining bits are used for the identification of hosts in the N/W.

eg. 182.0.1.2/28 — 28 bits for NID, 4 bits for HID

→ for writing the CIDR representation, we can choose to mention any IP address from the CIDR block. The chosen IP address is followed by a slash & IP N/W prefix. We generally choose to mention the first IP address.

Q. CIDR representation 20.10.30.35/27. Find range of IP addresses in the block.

→ 27 bits Identification of N/W or block

(32-27) = 5 bits for HID.

Range = (00010100.00001010.00011110.001 00000 to
00010100.00001010.00011110.001 11111)

= (20.10.30.32 to 20.10.30.63)

Q. Consider a block of IP addresses ranging from 100.1.2.32 to 100.1.2.47.

1. Is it a CIDR block?
2. If yes, give CIDR representation.

→ i) All IP addresses are contiguous.

ii) No. of IP addresses = 47 - 32 + 1 = 16 = 2^4

iii) 1st IP address 100.1.2.32

32 ≡ 0010 0000

reduce only this much bits from right in the IP

100.1.2.32 is divisible by 2^4 since its least significant 4 bits are zero.

All rules satisfied.

So, given block is a CIDR block.

Now, size of block = 2^4 = # IP addresses

To have 2^4 IP addresses, total 4 bits are required in the HID part.

No. of bits in the NID part = 32 - 4
= 28

CIDR representation -

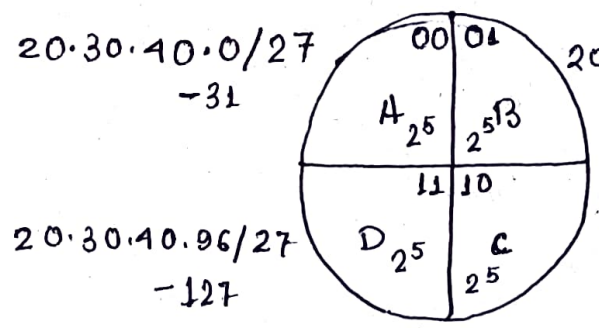
100.1.2.32/28.

* Subnetting in CIDR.

20.30.40.10/25

20.30.40.0 | 0001010

NID or BID | HID. → for 4 subnets



25 + 2 = 27 bits for NID in the subnets

(32 - 27) = 5 bits for HID. ⇒ # hosts = $2^5 - 2$ configurable

IP addresses = 2^5

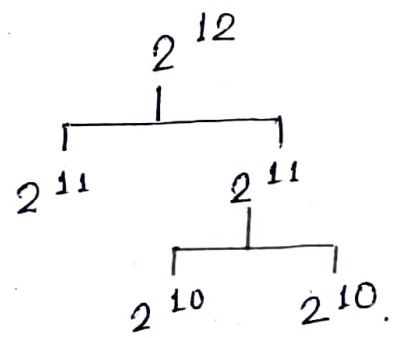
* VLSM or CIDR.

Variable Length SM

40.30.10.10/20

NID 20 bits

HID 12 bits.



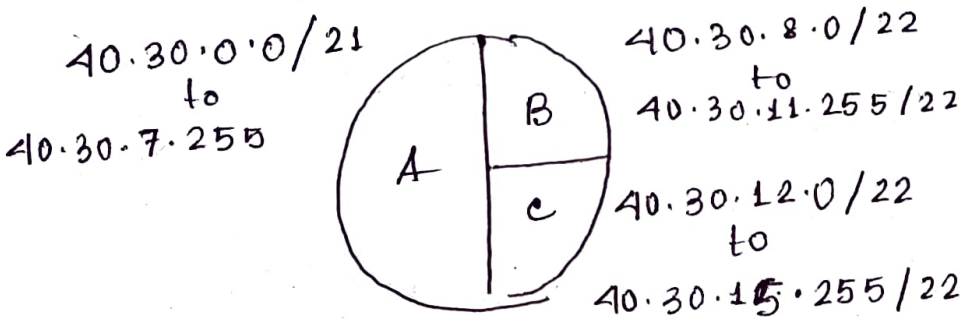
40.30.10.10.

⇓

$2^3 2^2$

40.30.0000 | 1010.00001010

BID HID.



1st subnetting:

A 40.30.0000 | 0 | 000.0000 0000 = 40.30.0.0/21

21 bits

40.30.0000 | 1 | 000.0000 0000 = 40.30.8.0/21

2nd subnetting (on 40.30.8.0/21):

B 40.30.0000 | 10 | 00.00000000 = 40.30.8.0/22

22 bits

C. 40.30.0000 | 11 | 00.00000000 = 40.30.12.0/22.

A range –

40.30.00000 | 111.11111111 = 40.30.7.255/~~22~~ 21

last IP.

B range –

40.30.00001 | 0 | 11.11111111 = 40.30.11.255/22

last IP

C range –

40.30.00011 | 11.11111111 = 40.30.15.255/22

last IP.

NB

1. Subnet mask 255.255.255.255

An address with subnet mask 255.255.255.255 means that the NID and IP address are same.

[As SM AND IP = NID]. So, every device connected to the device will also get NID as its own IP address. If A wants to send packet to B, then it will send it to the gateway first & afterwards to B. [End to end connection].

It's not a network, but a standalone host. So, this N/W with SM as 255.255.255.255 puts each device inside its own subnet, forcing them to communicate with the router before communicating with any other device.

A

I_A 10.12.0.1

SM_A 255.255.255.255

AND

10.12.0.1 NID_{AA}

B

I_B 23.12.1.0

SM_A 255.255.255.255

23.12.1.0 NID_{BA}

Different

⇒ A sends packet via router to the host B

2. Given DBA, # subnets possible

For a DBA, host bits part is all 1's.

DBA is represented as valid N/W id + all hosts bits as 1's.

eg. 201.15.16.31 DBA

201.15.0001 0000.000|11111

5 bits → HID

000|1 1111
000 1 1 111
000 1 1 1 1 1
000 1 1 1 1

If it is a DBA, then at most last 5 bits can be host bits.

Possibilities.:

	N/W / Subnet ID	N/W / Subnet Mask	Hosts configured.
1.	201.15.16.0 (000 | 00000)	255.255.255.224 (NID part 27, HID part 5)	$2^5 - 2 = 30$
2.	201.15.16.16 (0001 | 0000)	255.255.255.240 (NID 28 | HID 4)	$2^4 - 2 = 14$
3.	201.15.16.24 (00011 | 000)	255.255.255.248 (NID 29 | HID 3)	$2^3 - 2 = 6$
4.	201.15.16.28 (000111 | 00)	255.255.255.252 (NID 30 | HID 2)	$2^2 - 2 = 2$
5.	201.15.16.30 (0001111 | 0)	255.255.255.254 (NID 31 | HID 1)	$2^1 - 2 = 0$

* Supernetting

Opposite of subnetting. In supernetting, multiple networks are combined into a bigger network termed as a supernetwork or supernet.

Supernetting is mainly used in Route Summarization, where routes to multiple networks with similar network prefixes are combined into a single routing entry, with the routing entry pointing to a super network, encompassing all the networks. This in turn significantly reduces the size of routing tables & also the size of routing updates exchanged by routing protocols.

So, supernetting is used in route aggregation to reduce the size of routing tables and routing table updates.

→ While performing supernetting on networks, we have to keep in mind –

1. All the IP addresses should be contiguous.
2. Size of all the small networks should be equal & must be in form of 2^n.
3. First IP address should be exactly divisible by whole size of supernet.

Eg 4 small networks of class C: /24

200.1.0.0
200.1.1.0
200.1.2.0
200.1.3.0

Build a bigger network that have a single network ID.

Before supernetting, routing table

NID	Subnet mask	Interface.
200.1.0.0	255.255.255.0	A
200.1.1.0	255.255.255.0	B
200.1.2.0	255.255.255.0	C
200.1.3.0	255.255.255.0	D.

Condition checking:

1. Contiguous: Range of 1st Network is from 200.1.0.0 to 200.1.0.255. If we add 1 in last IP address of first N/W that is 200.1.0.255 + 0.0.0.1, we get the next N/W ID that is 200.1.1.0.

All N/Ws are contagious having size 256 hosts.

2. Equal size of all N/W: As all N/Ws are of class C, so all of them have a size of 256 which in turn equals to 2^8.

3. First IP address exactly divisible by total size:

First IP address 200.1.0.0.

Size of supernet $4 \times 2^8 = 2^{10}$

10 bits 0 ⇒ / by 2^{10}

200.1. 0000 0000 . 0000 0000

First IP address is divisible by total supernet size.

- add $\log_2 4$ bits to HID part.

So, we can join all of them and make a supernet.

NID	HID
22	10

New supernet ID 200.1.0.0 / 22.

- Finding supernet ID using supernet mask:

4 networks' IP addresses

- Changing / variable part 0's
- Constant part 1's

← constant	variable →
200.1 . 0000 0000 .	0000 0000
200.1 . 0000 0001 .	0000 0000
200.1 . 0000 0010 .	0000 0000
200.1 . 0000 0011 .	0000 0000

Supernet mask 255. 255. 252 . 0

Supernet ID = (Any IP address among the 4) AND (Supernet mask)

= 200.1.0.0

→ Advantages of supernetting :

1. Control & reduce N/W traffic.

2. Helpful to solve the problem of lacking IP addresses.

3. Minimises the routing table

→ Disadvantages of supernetting :

1. It cannot cover different area of N/W when combined.

2. All the N/Ws should be in same class & all IP should be contiguous.

eg. 100.1.2.0/25 (1)

100.1.2.128/26 (2)

100.1.2.192/26 (3)

We can apply supernetting on (2) and (3) first that yields supernet 100.1.2.128/25

Now 100.1.2.128/25 and 100.1.2.0/25 are supernetted.

We get supernet 100.1.2.0/24.

* If we do supernetting on 2^n subnets, the Supernet ID part decreases by n bits.

Supernetting four /24 subnets

⇓

Supernet is /22 (24 - 2)

* Private IP addresses.

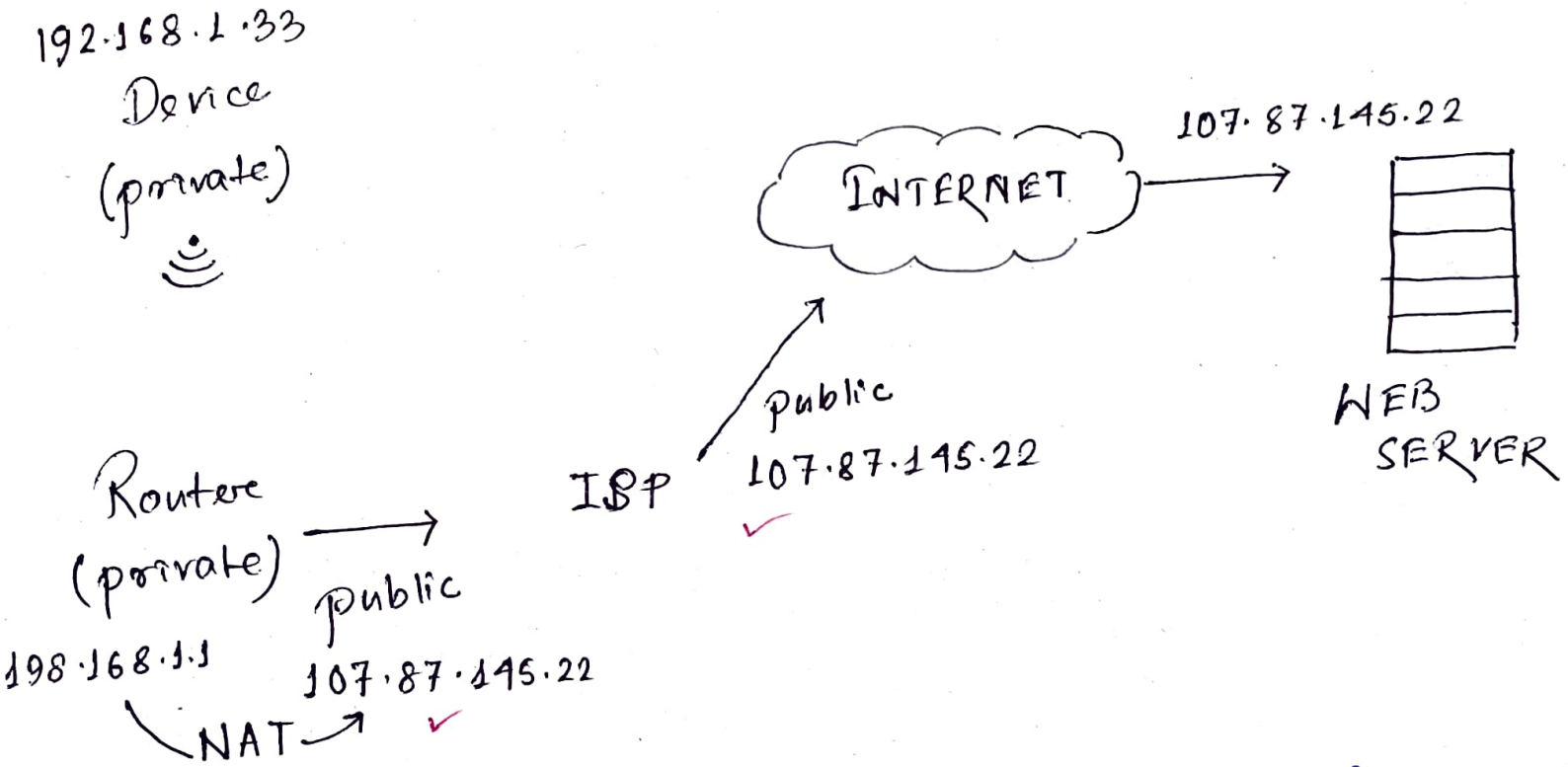
IANA - Internet Assigned Numbers Authority

- ✓ 192.168.0.0 – 192.168.255.255 (class C 256×2^8, 65536 IP addresses)
- ✓ 172.16.0.0 - 172.31.255.255 (Class B, $16 \times 2^{16} = 1,048,576$)
- ✓ 10.0.0.0 – 10.255.255.255 (class A, 2^{24} (16,777,216))

IANA reserves these IP address blocks for use as private IP addresses.

→ Another range of private IP addresses is 169.254.0.0 to 169.254.255.255, but those addresses are for automatic private IP addressing (APIPA) use only.

→ Instead of having devices inside a home or business network each use a public IP address, of which there's a limited supply, private IP addresses provide an entirely separate set of addresses that allow access on a N/W but without taking up a public IP address space. All the devices that are contained within private networks around the world can use a private IP address with virtually no limitation, which can't be said for public IP addresses.



Q. G'12 An ISP has the following chunk of CIDR-based IP addresses available with it: 245.248.128.0 /20. The ISP wants to give half of this chunk of addresses to organisation A, and a quarter to organisation B, while retaining the remaining with itself. Which of the following is a valid allocation of addresses to A and B.

a) 245.248.136.0/21 & 245.248.128.0/22

b) 245.248.128.0/21 & 245.248.128.0/22

c) 245.248.132.0/22 & 245.248.132.0/21

d) 245.248.136.0/24 & 245.248.132.0/21

→ 245.248.128.0

⇓

245.248.1000|0000.0000 0000

Different subnet splittings

245.248.1000 [1] [000.0]

↓

245.248.136.0/21

or

245.248.1000 [0] [000.0]

↓

245.248.128.0/21

245.248.1000 [00] [00.0]

245.248.128.0/22

245.248.1000 [01] [00.0]

245.248.132.0/22

245.248.1000 [10] [00.0]

245.248.136.0/22

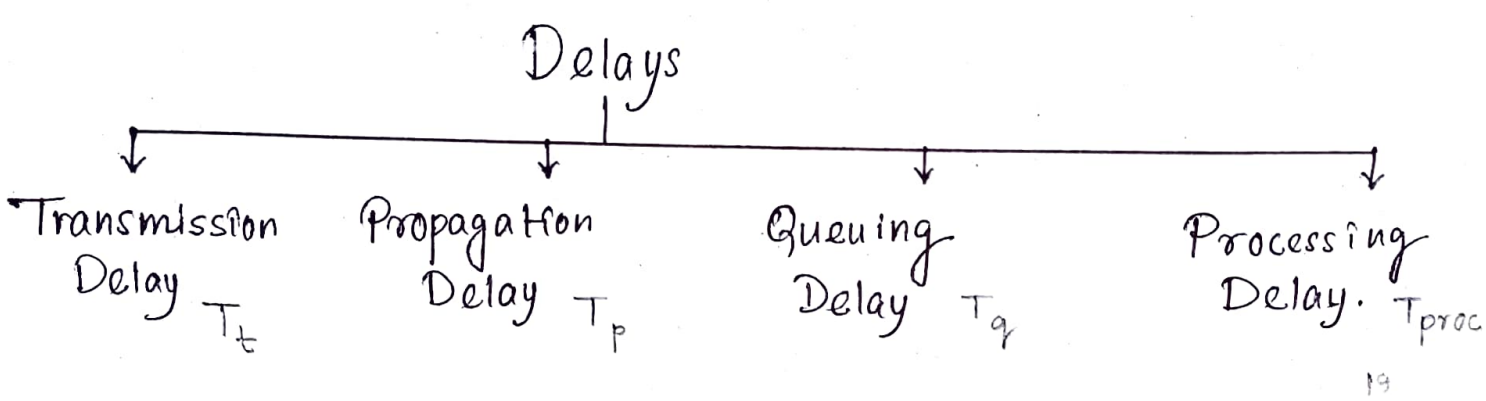
245.248.1000 [11] [00.0]

245.248.140.0/22

Flow Control Methods.

* Delays in CN.

Consider two hosts A and B are connected over a transmission link / transmission media. A data packet is sent by the host A to host B.



1. Transmission Delay. T_t

Time taken to put the data packet on the transmission link.

$T_t \propto$ Length / or Size of data packet

$T_t \propto \frac{1}{\text{Bandwidth}}$ / or

$T_t = \frac{\text{Length / Size of data packet}}{\text{Bandwidth of N/W}}$

$T_t = \frac{L}{B}$

BW → transmission rate

2. Propagation Delay T_p

Time taken for one bit to travel from sender to receiver end of the link.

$T_p \propto$ Distance between sender & receiver

$T_p \propto \frac{1}{\text{transmission speed}}$

$T_p = \frac{\text{Distance b/w sender \& receiver}}{\text{Transmission speed.}}$

$T_p = \frac{d}{v}$

3. Queuing delay T_q

Time spent by the data packet waiting in the queue before it is taken for exec^n.

It depends on the congestion in the N/W.

4. Processing delay T_{proc}

Time taken by the processor to process the data packet.

It depends on the speed of the processor. Processing of the data packet helps in detecting bit level errors that occur during transmissi

NB.

1. Total delay in sending one data packet

or

✓ End to End time =

$$T_t + T_p + T_q + T_{proc}$$

2. In optical fibre, transmission speed of data packet = 2.1×10^8 m/sec.

(70% of speed of light)

3. Both queuing delay & processing delay are dependent on the state of the system.

This is because -

- If dest^n host is busy doing some heavy processing, then these delays increase.
- If dest^n host is free, then data packets will be processed immediately & these delays will decrease.

4. For any particular transmission link, bandwidth and transmission speed are always constant. This is because they are properties of the transmission medium.

*5. Bandwidth is always expressed in powers of 10 and data is always expressed in powers of 2.

1 kilo bytes = 2^{10} bytes

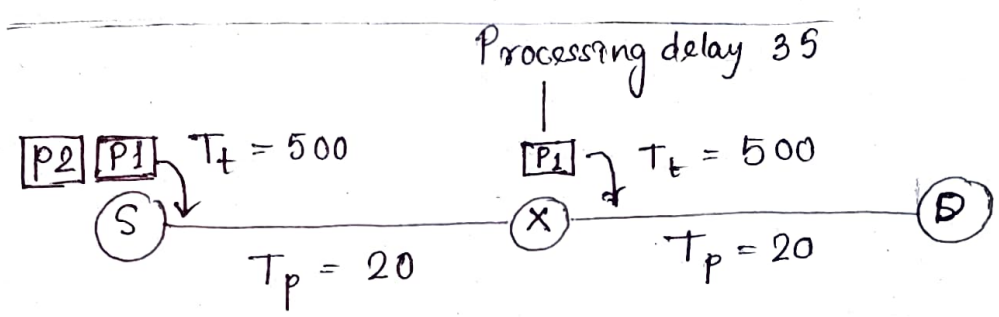
1 kilo bytes per second = 10^3 bytes per second

* Q. G'15. Since it is a N/W that uses switch, every packet goes through two links, one from source to switch and other from switch to destination. Since there are 10000 bits and packet size is 5000, two packets are sent. Transmission time for each packet is $5000/10^7$ sec. Each link has a propagation delay of 20 μs. The switch begins forwarding a packet 35 μs after it receives the same. If 10000 bits of data are to be transmitted between the 2 hosts using a packet size of 5000 bits, the time elapsed between transmission of 1st bit of data and the reception of last bit of data in microseconds is —

Sender host transmits first packet to switch, the transmission time is $5000/10^7$ that is 500 μs. After 500 μs, the 2nd packet is

transmitted. The first packet reaches destination in 500 + 35 + 20 + 20 + 500 = 1075 μs. While the 1st packet is traveling to dest^n, the 2nd packet starts its journey after 500 μs & rest of the time taken by second packet overlaps with 1st packet.

So, overall time = 1075 + 500
= 1575.



Time taken by first packet P_1 to reach switch = transmission time + propagation delay
= 500 + 20 = 520 μs.

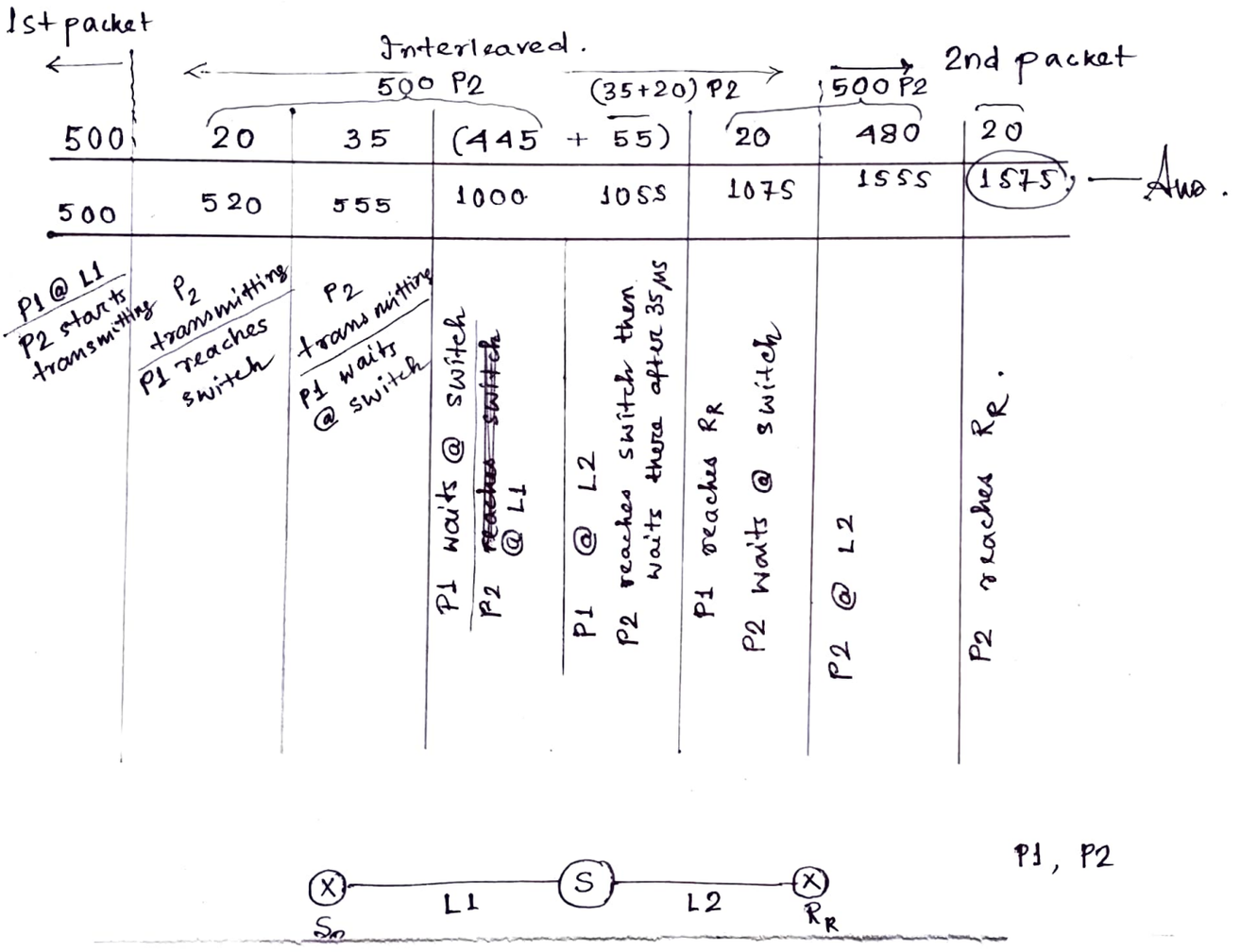
Time taken by P_1 to reach receiver from switch = 35 + 500 + 20 = 555 μs.

Time taken by P_1 to reach receiver from sender = 520 + 555 = 1075 μs.

After 520 μs switch starts receiving second packet P_2, i.e. at 520 + 500 = 1020 μs P_2 is completely received by switch.

Now, time taken by P_2 to reach from switch to receiver = 35 + 500 + 20 = 555 μs.

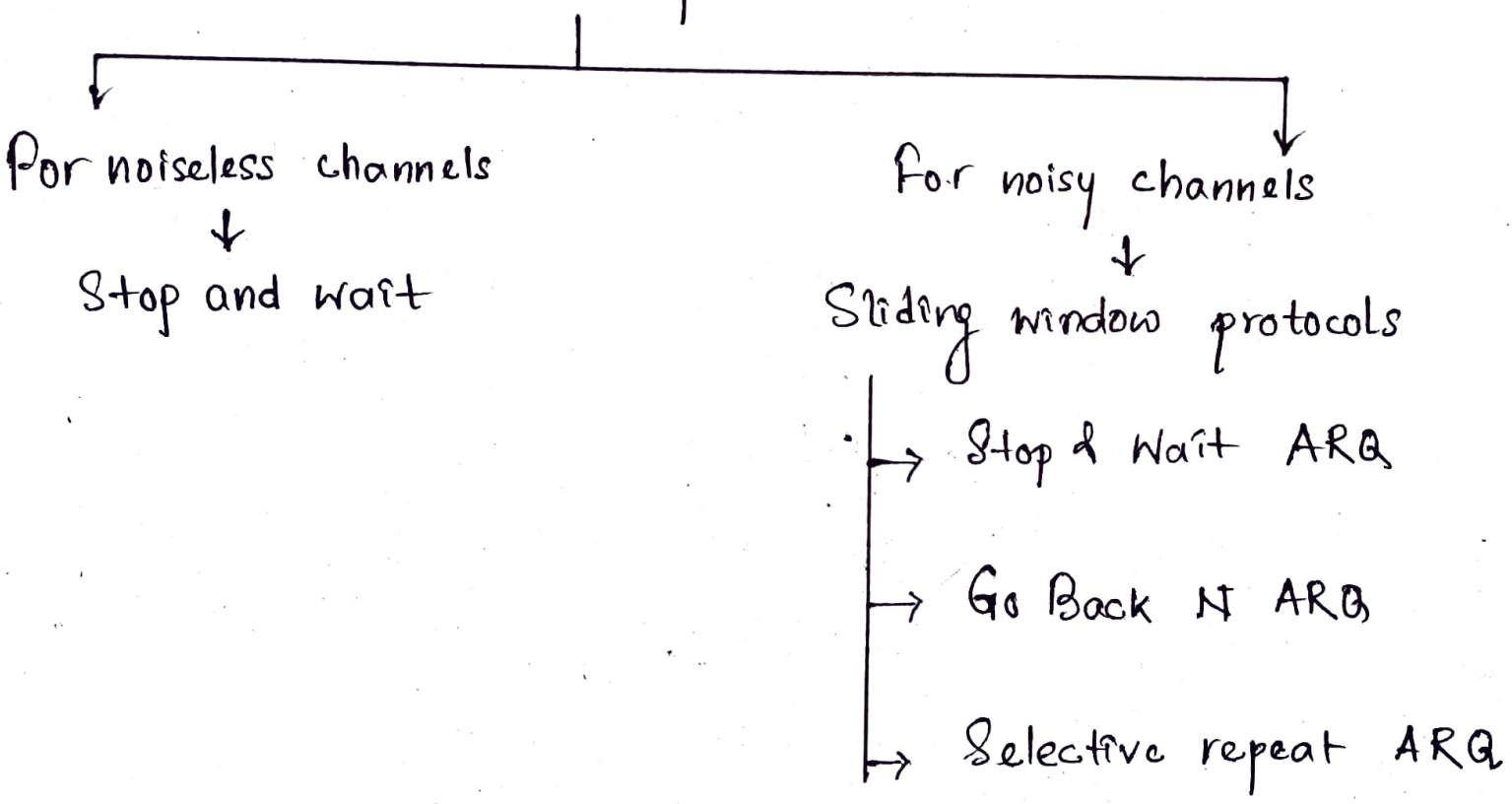
It means at 1020 + 555 = 1575 μs P_2 reaches the destination.



* Flow control. (Data Link Layer)

A set of procedures which are used for restricting the amount of data that a sender can send to the receiver.

- Flow control protocols.



- Stop and Wait Protocol. Simplest flow control protocol.

Works under assumptions :

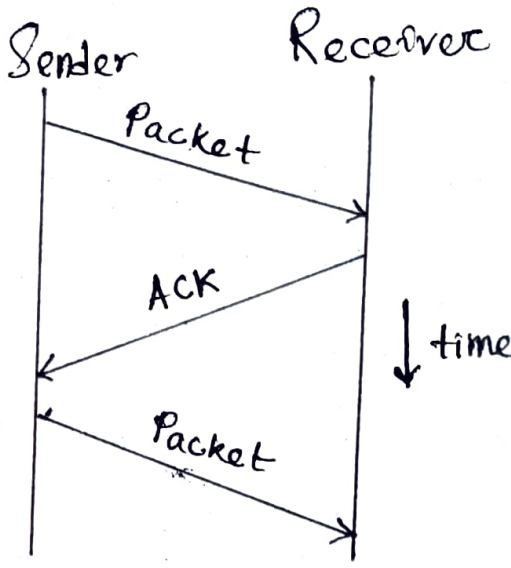
a) Communication channel is perfect

b) No error occurs during transmission.

→ Working : 1. Sender sends a data packet to the receiver. 2. Sender stops and waits for the acknowledgement for the sent packet from the receiver. 3. Receiver

receives and processes the data packet.
4) Receiver sends an acknowledgement to the sender. 5) After receiving the acknowledgement, sender sends the next data packet to the receiver

→ Detailed steps.



1. Sender puts the data packet on the transmission link. 2. Data packet propagates towards the receiver's end. 3. Data packet reaches the receiver & waits in its buffer. 4. Receiver processes the data. 5. Receiver puts the ACK on the transmission link. 6. ACK propagates towards the sender's end. 7. ACK reaches the sender & waits in its buffer. 8. Sender processes the ACK.

→ Total time taken in sending one data packet =

$$(T_t + T_p + T_q + T_{pr})_{packet} + (T_t + T_p + T_q + T_{pr})_{ACK}.$$

p – propagation

pr – processing

Assuming -

Queuing delay and processing delay to be zero at both sender & receiver side,

Transmission time for the acknowledgement to be zero since its size is very small.

So, total time in sending one packet,

$$T_{total} = (T_t + T_p)_{packet} + (T_p)_{ACK}.$$

Now, we know, propagation delay depends on the distance(d) and speed (v). So, it would be same for both data packet and ACK.

$$T_{total} = (T_t)_{packet} + 2 \times T_p$$

→ Efficiency, η of any flow control protocol is given by

$$\text{Efficiency, } \eta = \frac{\text{Useful time}}{\text{Total time}}$$

where,

useful time = Transmission delay of data packet

$= (T_t)_{packet}$

useless time = Time for which sender is forced to wait & do nothing

$= 2 \times$ Propagation delay $= 2T_p$

$$\eta = \frac{(T_t)_{packet}}{(T_t)_{packet} + 2T_p} = \frac{1}{1 + 2\left(\frac{T_p}{T_{t,packet}}\right)}$$

$$\eta = \frac{1}{1+2a}, \quad a = \frac{T_p}{T_{t,packet}}$$

Efficiency, also referred as - Link utilisation, line utilisation, sender utilisation, utilisation of sender.

- **Factors affecting efficiency**

$$\eta = \frac{T_{t,packet}}{T_{t,packet} + 2T_p} = \frac{1}{1 + 2\frac{T_p}{T_{t,packet}}}$$

$$= \frac{1}{1 + 2 \times \left(\frac{\text{Distance}}{\text{Speed}}\right)\left(\frac{\text{Bandwidth}}{\text{Packet length}}\right)}$$

$$\eta = \frac{1}{1 + 2 \cdot \frac{D}{v} \cdot \frac{B}{L}}$$

So,

$\eta \propto \frac{1}{\text{Distance between sender and receiver}}$

$\eta \propto \frac{1}{\text{Bandwidth}}$

$\eta \propto$ Transmission speed

$\eta \propto$ Length of data packet.

→ Throughput, is defined by number of bits that can be sent through the channel per second.

Throughput is also called as –

Bandwidth utilisation,

Effective bandwidth,

Maximum data rate possible,

Maximum achievable throughput.

Throughput = Efficiency (η) × Bandwidth (B)

→ **Round trip time = 2 × Propagation delay**

$RTT = 2T_p$

See last pg of flow control

→ S&W good for thin pipes (less capacity) ↑ Efficiency

bad for thick pipes (more capacity)

$$\eta = \frac{1}{1+2a}$$

$$= \frac{1}{1+2.\frac{T_p}{T_t}}$$

$$= \frac{1}{1+\frac{2 \times T_p \times B}{L}}$$

- More η for thin pipes. Why?

Capacity for full duplex

$$= 2 T_p B.$$

So, capacity ↑ $\Rightarrow$ η ↓

BW vs Throughput.

BW is the max amount of data that can be passed through a channel. (potential capacity)

Throughput is how much data actually does travel through the channel successfully. This can be limited by a ton of different things including latency, & what protocol you're using. (effective payload) (practical measure)

→ Advantages of stop & wait protocol :

a) Very simple to implement.

b) Incoming packet from receiver is always an acknowledgement.

→ Limitations :

a) Extremely inefficient because it makes the transmission process extremely slow and it does not use the bandwidth entirely as each single packet and acknowledgement uses the entire time to traverse the link.

b) If the data packet sent by the sender gets lost, then sender will keep waiting for the ACK for eternity and receiver will keep waiting for the data packet for eternity.

c) If ACK sent by the receiver gets lost, then sender will keep waiting for the ACK for eternity & receiver will keep waiting for another data packet for eternity.

→ Stop and wait protocol performs better for LANs than WANs ; ~~Because~~ because,

- Efficiency of the protocol is inversely proportional to the distance between sender & receiver.
- The protocol works better where the distance between sender & receiver is less.
- Distance is less in LANs as compared to WANs.

• Stop and Wait ARQ Protocol

In stop and wait protocol, sender sends one packet and then waits for its acknowledgement and sender sends the next packet only after it receives the ACK for the previous packet. Problem of S&W is the occurrence of deadlock due to loss of data packet or acknowledgement

Stop and wait ARQ is an improved & modified version of stop & wait protocol.

S&W ARQ assumes —

- Communication channel is noisy.
- Errors may get introduced in the data during the transmission.

→ Working :

S&W ARQ (Automatic repeat request) provides a solution to all the limitations of S&W protocol. S&W ARQ includes following 3 extra elements :

- Timeout timer
- Sequence numbers for data packets
- Sequence n for acknowledgements

Thus,

S&W ARQ = (S&W) + TO timer + Sequence numbers for data packets & acknowledgements.

→ # Sequence numbers required.

* For any sliding window protocol to work without any problem, condition that must be satisfied —

Available sequence numbers ≥ Sender window size + Receiver window size.

Stop & wait ARQ is a one bit sliding window protocol where sender & receiver window size both is 1.

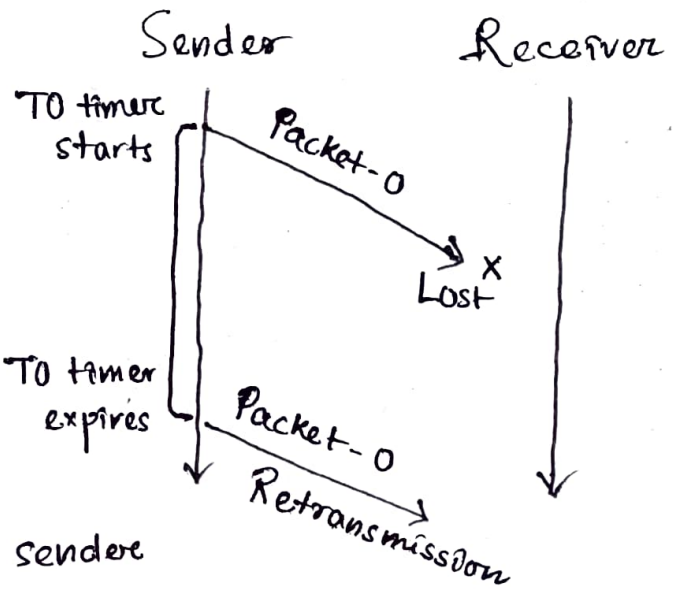
Thus, min. # of sequence numbers required = 1+1 = 2

Two sequence numbers used are 0 & 1.

→ S&W ARQ as a solution to problems in S&W

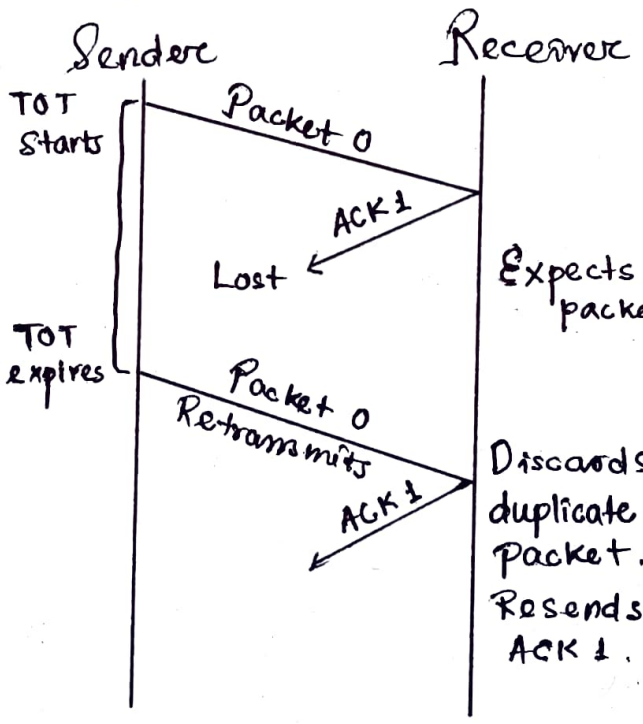
1. Problem of lost data packet.

TO timer helps to solve this problem. After sending a data packet to the receiver, sender starts the TO timer. If the data packet gets acknowledged before the timer expires, sender stops the TOT. If the timer goes off before receiving the acknowledgement, sender transmits the same data packet. After retransmission, sender resets the timer. This prevents occurrence of deadlock.



2. Problem of lost acknowledgement

Sequence numbers on data packets helps to solve the problem. When the ACK sent by the receiver gets lost, sender retransmits the same data packet after its timer goes off (if didn't get ACK). This prevents deadlock. The sequence number on the data packet helps the receiver to identify the duplicate data packet. Receiver discards the duplicate packet & resends the same acknowledgement.



- **Role of sequence numbers on data packets**

1. Sender sends a data packet with sequence number-0 to the receiver.

2. Receiver receives the data packet correctly. Receiver now expects data packet with seq. number 1. Receiver sends Ack 1.

3. ACK-1 gets lost on the way.

4. Sender receives no ACK & timeout occurs. Sender retransmits the same data packet with seq. no-0. This will be a duplicate packet for the receiver.

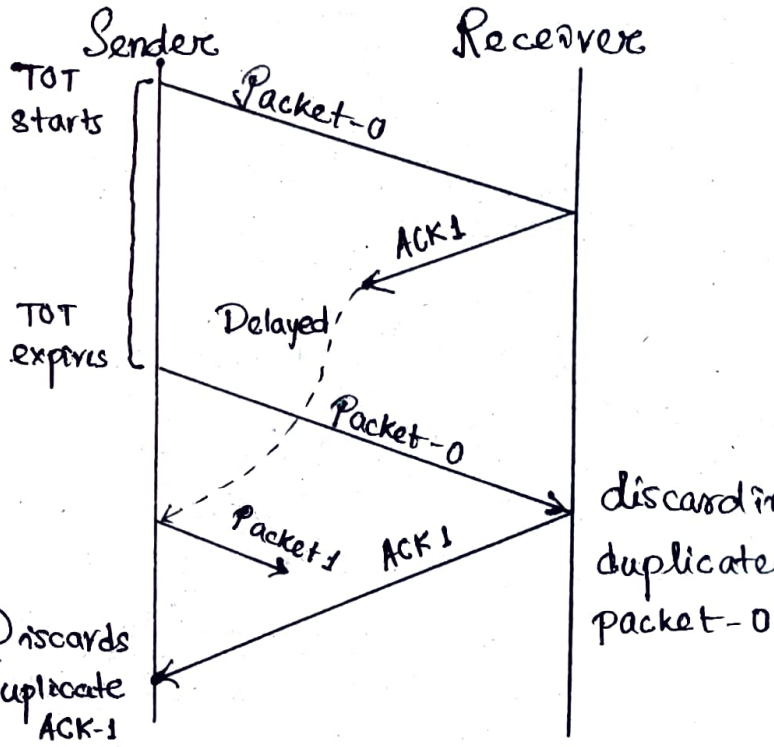
5. Receiver receives the data packet & discovers it is the duplicate packet. It expects the data packet with seq. no-1 but receiving the packet with seq. no.-0. It discards the duplicate packet & resends ACK 1. ACK 1 requests the sender to send a data packet with seq. no. 1.

This avoids inconsistency of data.

Had the sequence numbers not been allotted to the data packets, receiver would have accepted the duplicate data packet thinking of it as the new data packet.

3. Problem of delayed acknowledgement.

Seq. no. on acknowledgement helps to solve problem of delayed acknowledgement.



- Role of seq. no. on ACK.

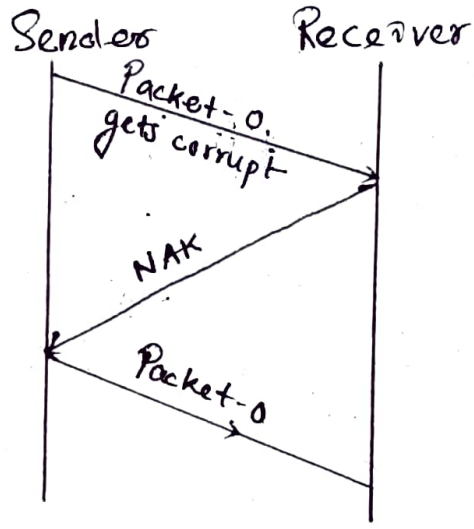
1. Sender sends a data packet with seq. no.-0 to the receiver.
2. Receiver receives the data packet correctly. Receiver expects packet with seq. no. 1 and sends the ACK-1.
3. ACK-1 sent by the receiver gets delayed in reaching the sender.
4. Sender receives no ACK & TO occurs. Sender retransmits the same data packet with seq. no. 0. This will be a duplicate packet for the receiver.
5. Receiver receives the data packet & discovers it is the duplicate packet. It expects the data packet with seq. no. 1 but receiving the data packet with seq. no. 0. It discards the duplicate data packet & resends ACK-1. ACK-1 requests the sender to send a data packet with seq. no. 1.

6. Two ACK-1 reaches the sender. When first ACK-1 reaches the sender, sender sends the next data packet with seq no. 1. When 2nd ACK1 reaches, sender rejects the duplicates ACK, as it has already sent the data packet with seq no. 1 & now sender expects the ACK with seq. no. 0 from the receiver.

Had the seq no. not been allotted to the ACKs, sender would have accepted the duplicate ACK thinking of it as the new ACK for the latest data packet sent by it.

4. Problem of damaged packet:

If receiver receives a corrupt data packet from the sender, it sends a negative acknowledgement (NAK) to the sender. NAK requests the sender to send the data packet again.



→ S&W vs S&W ARQ.

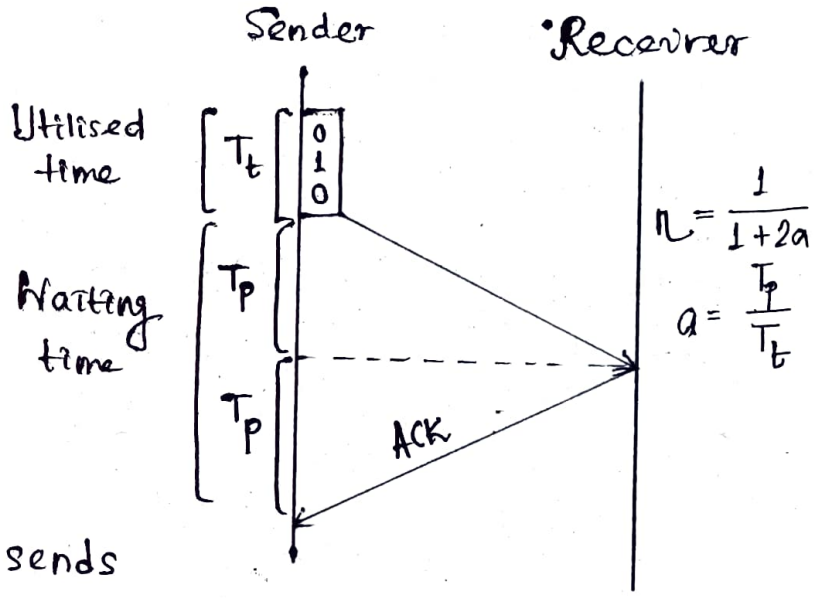
S&W	S&W ARQ
i) Assumes that commun^n channel is noisefree.	i) Imperfect & noisy commun^n channel is assumed.
ii) Data packet sent by the sender can never get corrupt. (read 'should')	ii) May get corrupt.
iii) No concept of -ve ACK.	iii) A -ve ACK is sent by the receiver if the packet is found to be corrupt.
iv) No concept of TO timer.	iv) Sender starts the TOT after sending the data packet.
v) No concept of seq. no.	v) Data packets & ACKs are numbered using seq. no.

→ Limitation of S&W ARQ

The major limitation of S&W ARQ is its very less efficiency. To increase the efficiency, protocols like Go back N & Selective Repeat are used.

Explanation: In S&W ARQ, sender window size is 1. This allows the sender to keep only one frame unacknowledged. So, sender sends one frame & then waits until the sent frame gets acknowledged. After receiving the ACK from the receiver, sender sends the next frame.

Sender uses T_t time for transmitting the packet over the link. Then, sender waits for $2 \times T_p$ time. After $2T_p$ time, sender receives the ACK for the sent frame from the receiver. Then, sender sends the next frame. This $2T_p$ waiting time is the actual cause of less efficiency.



→ Efficiency Improvement.

The efficiency of S&W ARQ can be improved by increasing the window size. This allows the sender to keep more than one unacknowledged frame in its window. Thus, sender can send frames in the waiting time too.

This gives rise to the concept of sliding window protocols.

- Increasing efficiency using larger window size.

unit time assumed - 1 sec.

More than one ~~bits~~ packets - Window

In T_t sec sending 1 packet in SdW

In 1 sec " $1/T_t$ packet in SdW

In total time, $T_t + 2 \times T_p$ sec -

$$\text{sending } \frac{T_t + 2T_p}{T_t} \text{ packets}$$

$$= 1 + 2a \text{ packets}$$

Sending 1 packet in SdW.

Max. data that could be sent $1 + 2a$

That's why $\eta = \frac{1}{1+2a}$

To increase η, when $\eta \longrightarrow 1$,

we have to make numerator also $1 + 2a$

To send more no. of packets we can send multiple bits at a time, thus using larger window.

Q. If the BW (B) of the line is 1.5 Mbps, RTT is 45 msec and packet size is 1 KB, then find link utilisation in stop and wait.

→ B = 1.5 Mbps

RTT = 45 msec.

Packet size = 1 KB.

Transmission delay (T_t) = $\frac{\text{Packet size}}{\text{Bandwidth}}$

= $\frac{1\text{ KB}}{1.5\text{ Mbps}} = \left[\frac{2^{10} \times 8\text{ bits}}{1.5 \times 10^6\text{ bits per s}}\right]$

= 5.461 msec.

→ Propagation delay (T_p) = RTT / 2

= $\frac{45\text{ msec}}{2}$ = 22.5 msec.

Now, η / link utilization = $\frac{1}{1 + 2 \cdot \frac{T_p}{T_t}}$

= $\frac{1}{1 + 2 \cdot \frac{22.5\text{ msec}}{5.461\text{ msec}}}$

= 0.108 = 10.8 %.

Q A channel has a bit rate of 4 Kbps & one way propagation delay of 20 msec. The channel uses S&W. The transmission time of the ACK frame is negligible. To get a channel efficiency of at least 50%, what's the minimum frame size ?

→ B = 4 Kbps ; T_p = 20 msec ; $\eta \geq$ 50 %.

Let, required frame size = L bits.

$T_t = \frac{L\text{ bits}}{4\text{ Kbps}}$ | $a = \frac{T_p}{T_t} = \frac{20\text{ msec} \times 4\text{ Kbps}}{L\text{ bits}}$.

Now, $\eta \geq 0.5 \Rightarrow \frac{1}{1+2a} \geq 0.5 \Rightarrow$ L bits ≥ 2 (20 msec x 4 Kbps)

Frame size at least 160 bits. $\Rightarrow$ L bits $\geq 20 \times 4 \times 2$

= 160 bits.

Q. What is the throughput achievable in S&W protocol by a maximum packet size of 1000 byt and N/W span of 10 km.

→ Throughput = Efficiency × Bandwidth

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

$$= \frac{L/B}{T_t + 2\frac{d}{v}} \times B = \frac{L}{2 \times \frac{d}{v}}$$

As not provided with the N/W's B, ignored the term T_t (although incorrect).

$L = 1000$ Bytes | $d = 10^4$ m | $v = 2.1 \times 10^8$ m/s

So, throughput = $\frac{1000 \text{ Bytes}}{2 \times 10^4 \text{ m} / (2.1 \times 10^8 \text{ m/s})}$

$= 10.5$ MBps

Q If the packet size is 1 KB and propagation time is 15 msec, the channel capacity is 10^9 b/sec, then find the transmission time & utilization of sender in S&W protocol.

→ Packet size = 1KB

$T_p = 15$ msec

Channel capacity, B = 10^9 b/sec

* Generally channel capacity is the total number of bits which a channel can hold. So, its unit is bits. But here, channel capacity is actually given as bandwidth, because its unit is b/sec.

$T_t = \frac{1 \text{ KB}}{10^9 \text{ b/sec}} = 1.024$ μsec

$$a = \frac{T_p}{T_t} = \frac{15 \text{ msec}}{1.024 \ \mu\text{sec}} = 14648.46$$

Sender utilization, $\eta = \frac{1}{1+2a} = 0.00341\ \%$

Q Consider a N/W with average source & dest^n 20 Km apart and one way delay of 100 μsec. At what data rate does the round trip delay equals the transmission delay for a 1KB packet?

→ $d = 20$ Km

$T_p = 100 \ \mu\text{sec}$

Packet size = 1KB

Condition - $RTT = T_t$

$\Rightarrow 2T_p = T_t$

$\Rightarrow 2 \times 100 \ \mu\text{sec} = \frac{1 \text{ KB}}{B}$

$\Rightarrow B = \frac{1 \text{ KB}}{200 \ \mu\text{sec}} = \frac{2^{10}}{200 \times 10^{-6}}$ bytes per second

$\Rightarrow B = 5.12$ MBps

Q Consider two hosts X and Y connected by a single direct link of rate 10^6 bits/sec. The distance b/w the two hosts is 10000 km and the propagation speed along the link is 2×10^8 m/sec. Host X sends a file of 50000 bytes as one large message to host Y continuously. What are T_t and T_p?

→ $B = 10^6$ bits/sec $\quad v = 2 \times 10^8$ m/s

$d = 10000$ km $\quad$ packet size = 50000 bytes.

$$T_t = \frac{50000 \text{ bytes}}{10^6 \text{ bits/sec}} = 400 \text{ msec.}$$

$$T_p = \frac{d}{v} = \frac{10000 \text{ km}}{2 \times 10^8 \text{ m/s}} = 50 \text{ msec.}$$

Q. The values of parameters for the S&W ARQ protocol are as given –

Bit rate of the transmission channel = 1 Mbps

Propagation delay from sender to receiver = 0.75 msec

Time to process a frame = 0.25 msec

No. of bytes in the information frame = 1980

No. of bytes in the ACK frame = 20

No. of overhead bytes in the information frame = 20

Assume that there are no transmission errors. Then the transmission efficiency (in %) of the S&W ARQ protocol for the above params is –

→ B = 1 Mbps Overhead in data frame = 20 bytes

T_p = 0.75 msec

T_{pr} = 0.25 msec

Data frame size = 1980 bytes

Ack frame size = 20 bytes

$$\text{Useful data sending time} = \frac{\text{Useful data bytes sent}}{\text{Bandwidth}}$$

$$= \frac{1980 - 20 \text{ bytes}}{1 \text{ Mbps}}$$

$$= 15.680 \text{ msec}$$

$$\text{Total time} = \underset{\text{data frame}}{T_t \text{ of}} + \underset{\text{data frame}}{T_p \text{ of}} + \underset{\text{data frame}}{T_{pr} \text{ of}} + (T_t + T_p)_{Ac}$$

$$= \frac{1980 \text{ Bytes}}{1 \text{ Mbps}} + 0.75 \text{ msec} + 0.25 \text{ msec} + \frac{20 \text{ Bytes}}{1 \text{ Mbps}} + 0.75 \text{ msec}$$

$$= 17.75 \text{ msec.}$$

$$\eta = \frac{15.680}{17.75} = 88.33\ \%$$

Q A sender uses the S&W ARQ for transmission. Frames are of size 1000 Bytes & the transmission rate at the sender is 80 Kbps. Size of an ACK is 100 Bytes & the transmission rate at the receiver is 8 Kbps. One way propagation delay is 100 msec. Assuming no frame is lost, sender throughput is —

$$T_t\Big|_{\text{data frame}} = \frac{\text{Frame size}}{\text{Sender BW}}$$

$$= \frac{1000 \text{ Bytes}}{80 \text{ Kbps}}$$

$$= 100 \text{ msec}.$$

Frame size = 1000 bytes
Sender BW = 80 Kbps
ACK size = 100 bytes
Receiver BW = 8 Kbps
T_p = 100 msec

$$T_t\Big|_{ACK} = \frac{100 \text{ bytes}}{8 \text{ Kbps}}$$

$$= 100 \text{ msec}$$

Useful time = 100 msec $(T_t \text{ | data frame})$.

Total time = $(T_t + T_p)_{\text{data frame}} + (T_t + T_p)_{ACK}$

$$= 100 \text{ msec} + 100 \text{ msec} + 100 \text{ msec} + 100 \text{ msec}.$$

$$= 400 \text{ msec}$$

$$\eta = \frac{100}{400} = 25\%.$$

Sender throughput = $\eta \times$ Sender BW

$$= 0.25 \times 80 \text{ Kbps}$$

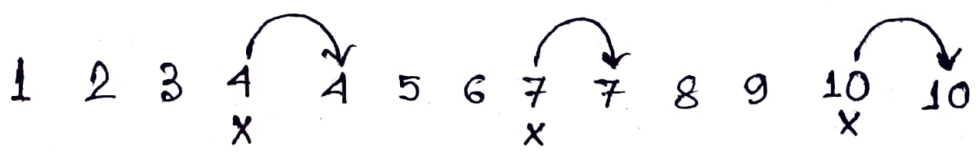
$$= 20 \text{ Kbps}$$

$$= 20 \times \frac{1000}{8} \text{ bytes per sec}$$

$$= 2500 \text{ bytes/sec}.$$

Q. Using SAW, sender wants to transmit 10 data packets to the receiver. Out of these 10 data packets, every 4th packet is lost. How many and which packets sender will have to send extra ?

→ Packets will be sent as



Lost packets 4, 7, 10.

Sender have to send 13 packets in total, 3 being extra as retransmissions.

G'15 Suppose that the SAW protocol is used on a link with a bit rate of 64 kilobits per second and 20 msec propagation delay. Assume that the transmission time for the acknowledgement & the processing time at nodes are negligible. Then the minimum frame size in bytes to achieve a link utilization of at least 50% is —

→ Link utilisation, $\eta = \dfrac{\text{Amount of data sent}}{\text{Max. amount of data that could be sent}}$

Let x be the frame size in bits.

In SAW, once a frame is sent, next frame won't be sent until ACK is received.

$$RTT = (T_t + T_p)_{\substack{data \\ frame}} + (T_t + T_p)_{ACK.} = T_t + 2T_p$$

$$= \frac{x \text{ bits}}{64 \text{ ms}} + 20 \text{ ms} + 0 + 20 \text{ ms} = \left(40 + \frac{x}{64}\right) \text{ms}$$

Method 1 may come handy for some other problems.

Amount of data sent during RTT = x.

Max. amount of data that could be sent =

$$\left(40 + \frac{x}{64}\right) \times 64 \text{ bits}$$

$$= 2560 + x \text{ bits}$$

So, link utilisation,

$$0.5 = \frac{x}{2560 + x} \Rightarrow x = \underline{320 \text{ bytes}}$$

Alternative

$$\eta_{SLW} = \frac{T_x}{T_x + 2T_p} = \frac{1}{1 + 2\frac{T_p}{T_x}} = \frac{1}{1+2a} \quad \Big| \quad a = \frac{T_p}{T_x}.$$

$$T_x = \frac{L}{B} \qquad T_p = \frac{d}{v}.$$

$$\eta = 50\% = \frac{1}{1+2a} \Rightarrow \frac{1}{2} = \frac{1}{1+2a} \Rightarrow 2a = 1$$

$$\Rightarrow 2T_p = T_x \Rightarrow 2 \times 20 \text{ ms} = \frac{L}{B}$$

$$\Rightarrow L = 2 \times 20 \text{ ms} \times (64K \text{ bits}).$$

$$\Rightarrow L = 2 \times 20 \times 10^{-3} \times 64 \times 10^{3} \text{ bits}$$

$$\Rightarrow L = 40 \times 64 \text{ bits} = \underline{320 \text{ bytes}}.$$

$$\eta = \frac{T_t}{T_t + 2 \times T_p}$$

$$\Rightarrow \frac{50}{100} = \frac{f/64K}{f/64K + 2 \times 20 \text{ ms}}$$

$$\Rightarrow \frac{f}{64} = 40 \Rightarrow f = 2560 \text{ bits} = \underline{320 \text{ bytes}}.$$

* Types of communication channels.

1. **Simplex channel.** : Can send the signal only in one direction. Thus, entire bandwidth of the channel can be used during the transmission.

eg. Radio station : Always sends signals to its audience. Never receives signals from the audience.

2. **Half duplex** : Can send signals in both the directions but in only one direction at a time. It may be considered as a simplex communication channel whose transmission direction can be switched.

eg. Walkie-talkie : Has a push-to-talk button. Used to turn on the transmitter but turn off the receiver. When the button is pressed, T_x cannot heare the R_x but R_x can hear the T_x.

3. **Full duplex channel.** : Can send signals in both the directions at the same-time. It increases the efficiency of communication.

eg. Telephone : Both persons can speak as well as hear each other at the same time.

→ **Channel capacity** : Total number of bits a channel can hold.

- Capacity of half duplex channel =

Bandwidth x Propagation delay

$= B \times T_p$

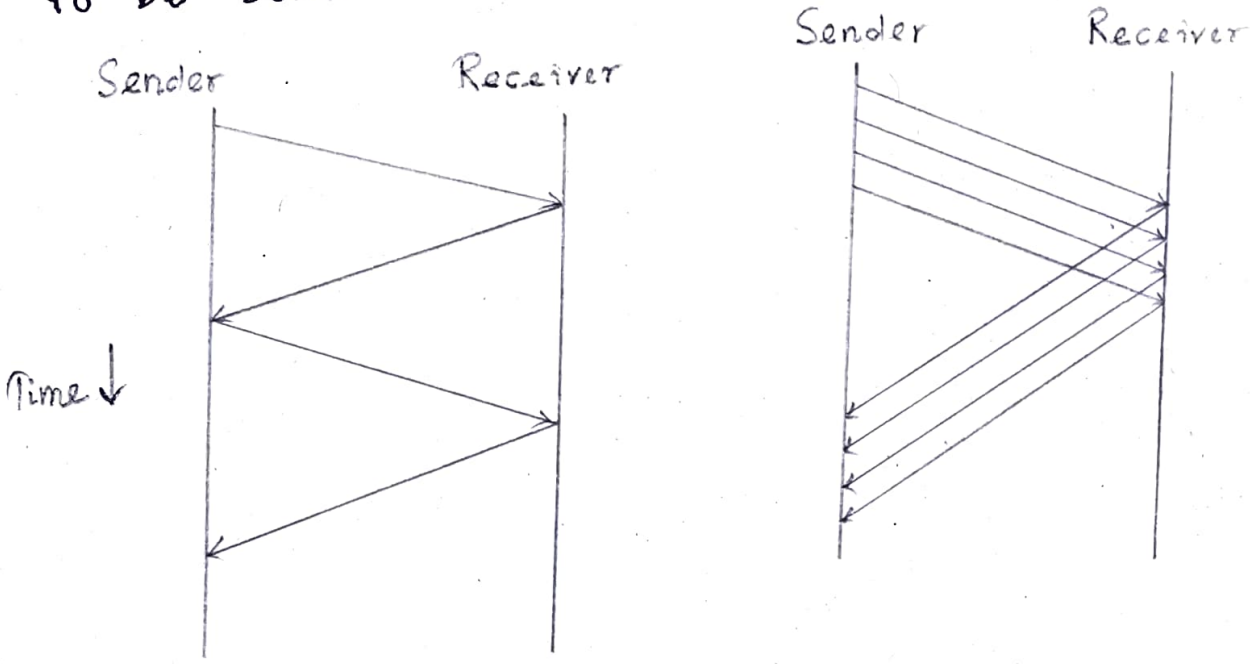
- Capacity of full duplex channel =

$$2 \times \text{Capacity of half duplex channel}$$

$$= 2 \times B \times T_p$$

- Pipelining in CN.

Method of sending multiple data units without waiting for an acknowledgement for the first frame sent. Pipelining ensures better utilization of N/W resources & also increases the speed of delivery, particularly in situations where a large number of data units make up a message to be sent.



Flow control protocols that uses pipelining -

1. Go Back N : Pipelining of frames i.e. sending multiple frames before receiving the acknowledgement for the first frame.

2. Selective repeat : Receiver window size is greater than 1. Only the erroneous or lost frames are retransmitted, while the good frames are received & buffered.

• Sliding Window Protocol.

Allows the sender to send multiple frames before needing the acknowledgements. Sender slides its window on receiving the acknowledgements for the sent frames. This allows the sender to send more frames.

→ Max no. of frames that sender can send without acknowledgement = Sender window size (W_s)

→ **Optimal window size of sender :**

We know $\eta = \frac{T_t}{T_t + 2T_p}$

To get 100% efficiency, $\frac{T_t}{T_t + 2T_p} = 1$

$\Rightarrow T_t = T_t + 2T_p$

Thus, to get 100% η, T_t must be $T_t + 2T_p$. This means sender must send the frames in waiting time ($2T_p$) too.

Now, finding out max. no. of frames that can be sent in time $T_t + 2T_p$.

In T_t time, sender sends one frame.

Thus, in $T_t + 2T_p$ time, sender can send $\frac{T_t + 2T_p}{T_t}$ frames i.e. $1 + 2a$ frames.

Thus, optimal sender window size = $1 + 2a$. $(1/\eta_{-s}$

→ **Required sequence numbers :**

Each sending frame has to be given a unique sequence number. Max. no. of frames that can be sent in a window = $1 + 2a$. So, minimum no. of sequence numbers reqd. = $1 + 2a$

Thus,

Minimum number of bits reqd. in sequence number field $= \lceil \log_2 (1+2a) \rceil$

When minimum number of bits is asked, take ceil.

When maximum " " " " " , take floor.

→ Choosing a window size : Size of the sender's window ~~size~~ is bounded by

1. Receivers ability = If receiver cannot process the data fast, sender has to slow down & not transmit the frames too fast.

2. Sequence number field = If seq. no. field contains n bits, then 2^n sequence numbers are possible. Thus, maximum number of frames that can be sent in one window. $= 2^n$.

So, for n bits in seq. no. field,

✓ Sender window size $= \min (1+2a, 2^n)$

→ 2 well known implementations of sliding window protocol are –

1. Go Back N
2. Selective Repeat.

→ Efficiency, $\eta = \frac{\#\text{frames sent in one window}}{\text{Total } \#\text{frames that can be sent in 1 window}}$

$$\eta = \frac{\text{Sender window size in the protocol}}{\text{Optimal sender window size.}}$$

$$\eta = \frac{\text{Sender window size}}{1+2a}$$

- In S&W ARQ, $w_s = 1 \Rightarrow \eta = \frac{1}{1+2a}$

Q T_t = 1 msec, T_p = 49.5 msec in a sliding window protocol, then

1. W_s for max η
2. Min # bits in seq. no. field
3. If only 6 bits are reserved for seq. no. field what is η?

→ 1. $W_s = 1 + 2a = 1 + 2\frac{T_p}{T_t} = 100$.

2. Min # bits = $\lceil \log_2 (1+2a) \rceil$

$= \lceil \log_2 100 \rceil = 7$.

3. If only 6 bits are reserved in the seq. no. field, then

Max sequence numbers = $2^6 = 64$.

$$\eta = \frac{\text{Sender window size}}{\text{Optimal sender window size}}$$

$$= \frac{\text{Max. seq. numbers}}{100} = \frac{64}{100} = 0.64.$$

Q T_t = 1 msec, T_p = 99.5 msec, then

1. W_s for max η
2. Min no. of bits reqd. for seq. no. field
3. If only 7 bits are reserved for seq. no.s, What will be the η?

→ 1. $W_s = 1 + 2a = 200$

2. $\lceil \log_2 (1+2a) \rceil = 8$.

3. Max. seq. no.s possible = 2^7

$$\eta = \frac{2^7}{200} = 0.64.$$

Q A 3000 km long trunk operates at 1.536 Mbps & is used to transmit 64 byte frames & uses sliding window protocol. If propagation speed is 6 μsec/km, how many bits should the sequence number field be?

→ d = 3000 km

6 μsec/km = Propagation speed — careful! watch units.

B = 1.536 Mbps

packet size, L = 64 bytes.

$$T_t = \frac{64 \text{ bytes}}{1.536 \text{ Mbps}} = \frac{64 \times 8 \text{ b}}{1.536 \times 10^6 \text{ bps}} = 333.33 \text{ μsec}$$

$$T_p = 3000 \times 6 = 18000 \text{ μsec.}$$

$$a = \frac{T_p}{T_t} = 54.$$

Bits reqd. in seq. no. field = $\lceil \log_2 (1+2a) \rceil$ = 7 bits.

[We use only $1+2a$ = 109 seq. numbers out of 2^7.]

Q Compute approximate optimal window size when packet size is 53 bytes, RTT is 60 msec & bottleneck bandwidth is 155 Mbps.

→ $$T_t = \frac{53 \text{ bytes}}{155 \text{ Mbps}} = \frac{53 \times 8 \text{ bits}}{155 \times 10^6 \text{ bps}} = 2.735 \text{ μsec.}$$

$$T_p = \frac{RTT}{2} = \frac{60}{2} = 30 \text{ msec.} \quad | \quad a = \frac{T_p}{T_t} = 10968.921$$

Optimal window size for sender =

$$1 + 2a$$
$$= 1 + 2 \times 10968.921$$
$$= 21938.84.$$

Approximate optimal window size = 21938.

Q A sliding window protocol is designed for a 1 Mbps P2P link to the moon which has a one way latency (delay) of 1.25 sec. Assuming that each frame carries 1 KB of data, what is the minimum number of bits required for sequence number?

→ $B = 1$ Mbps
$T_p = 1.25$ sec
$L = 1$ KB.

$T_t = \frac{1\text{KB}}{1\text{ Mbps}} = 8.192$ msec.

$a = \frac{T_p}{T_t} = 152.59$

Bits reqd $= \lceil \log_2(1+2a) \rceil = 9$ bits.

Q Host A is sending data to host B over a full duplex link. A and B are using the sliding window protocol for flow control. The sender & receiver window sizes are 5 packets each. Data packets (sent only from A to B) are all 1000 Bytes long and the transmission time for such a packet is 50 μs. Acknowledgement packets (sent only from B to A) are very small & require negligible transmission time. The propagation delay over the link is 200 μs. What is the max achievable throughput in this communication?

→ $W_S = W_R = 5$ $\quad T_t = 50$ μs
$L = 1000$ bytes $\quad T_p = 200$ μs

We know $T_t = \frac{L}{B} \Rightarrow B = \frac{L}{T_t} = \frac{1000 \text{ bytes}}{50\,\mu s}$

$\Rightarrow B = 160$ Mbps

$a = \frac{T_p}{T_t} = 4$

Optimal window size $= 1 + 2a = 1 + 2 \times 4 = 9$

$$\eta = \frac{\text{Sender window size}}{\text{Optimal window size}} = \frac{5}{9} = 55.55\%$$

Max. achievable throughput $= \eta \times B$

$= 0.5555 \times 160$ Mbps

$= 88.88$ Mbps

$= 11.11 \times 10^6$ bps

- Go back N protocol.

1. In GBN,

   { Sender window size is N and
   receiver window size is always 1.

   eg. In Go back 10, $W_S = 10$, $W_R = 1$.

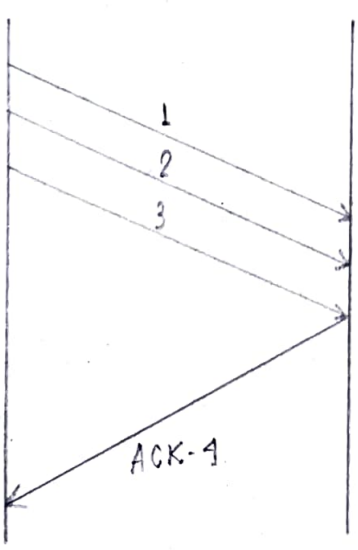
2. There are 2 kinds of acknowledgements -

   a) Cumulative acknowledgement

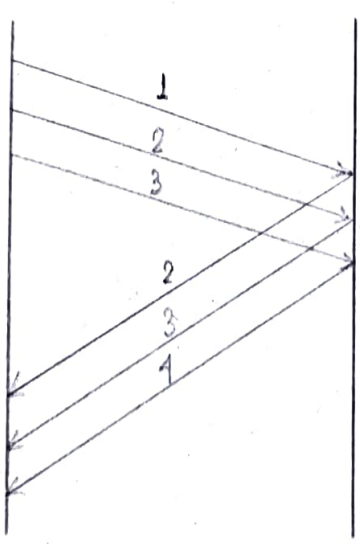
One ACK is used for many packets. The main advantage is traffic is less. A disadvantage is less reliability as, of one ACK is lost that would mean that all the packets sent are lost.

   b) Independent acknowledgement.

If every packet is going to get acknowledgement independently. Reliability is high here but a disadvantage is that traffic is also high since for every packet we are receiving independent ACK.



Cumulative



Independent

3. Go back N uses cumulative acknowledgements.

In GBN, receiver maintains an acknowledgement timer. Each time the receiver receives a new frame, it starts a new ACK timer. After the timer expires, receiver sends the cumulative ACK for all the frames that are unacknowledged at that moment.

A new acknowledgement timer does not start after the expiry of old acknowledgement timer. It starts after a new frame is received.

4. Go back N may use independent acknowledgements too. The kind of acknowledgement used depends on the expiry of ACK timer.

Consider after the expiry of ACK timer, there is only one frame left to be acknowledged. Then, GBN sends the independent acknowledgement for that frame.

5. GBN does not accept the corrupted frames and silently discards them.

In GBN, if receiver receives a frame that is corrupted, then it silently discards that frame. The correct frame is retransmitted by the sender after the time out timer expires. Silently discarding a frame means - simply rejecting the frame & not taking any action. (Like not sending a -ve ACK, NACK to the sender to send the correct frame.)

✓ 6. GBN does not accept out of order frames and silently discards them.

In GBN, if receiver receives a frame whose sequence number is not what the receiver expects, then it silently discards the frame. All the following frames are also discarded. This is because receiver window size is 1 and ✓ therefore receiver cannot accept out of order frames.

⇓

1 2 3 4 5 → ✗ → 3 4 5

7. GBN leads to retransmission of entire window if for any frame, no ACK is received by the sender.

If for any particular frame, sender does not receive any ACK, then it understands that along with that frame, all the following frames must also have been discarded by the receiver. So, sender has to retransmit all the following frames too along with that particular frame. Thus, it leads to the retransmission of entire window.

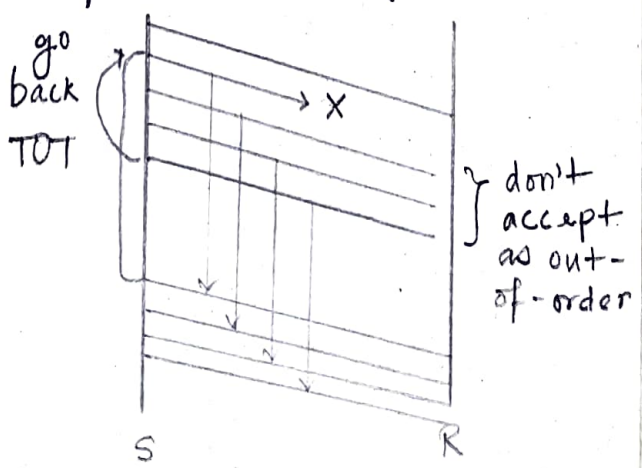
gfg

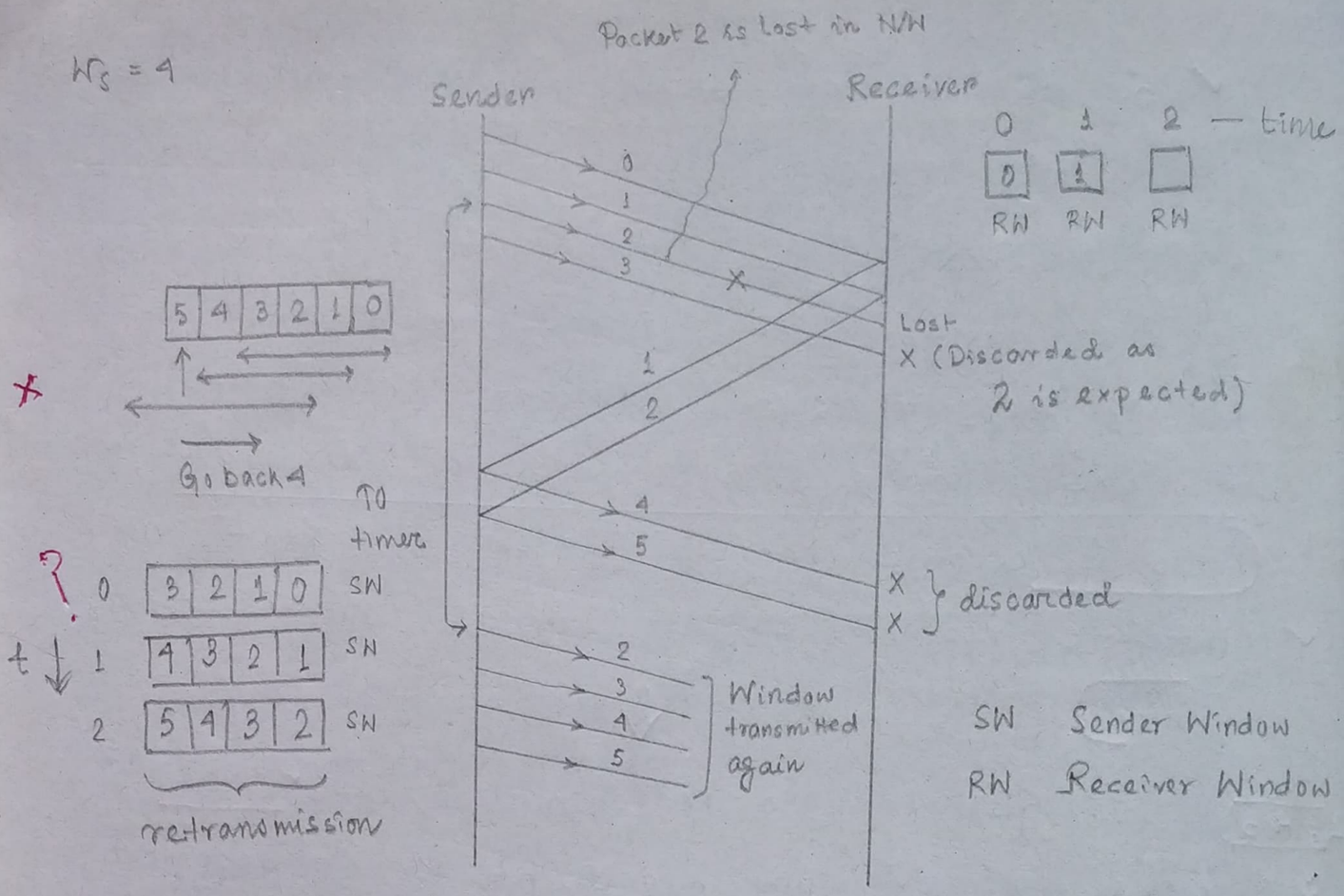
This is why, the protocol is named as GBN.

SWP

N - size of sender window

Go back means sender has to go back N places from the last transmitted packet in the unacknowledged window & not from the point where the packet is lost.





8. GBN leads to retransmission of lost frames after expiry of TOT.

9. Efficiency, $\eta_{GBN} = \dfrac{N}{1+2a}$ $\quad N \rightarrow W_S$

10.* As GBN uses cumulative acknowledgement, at the receiver side, it starts an acknowledgement timer whenever receiver receives any packet which is fixed and when it expires, it's going to send a cum. ACK for the no. of packets received in that interval of timer. If receiver has received N packets, then the ACK no. will be N+1. Important point is ACK timer will not start after the expiry of first timer but after receiver has received a packet. see nxt pg example

Timeout timer at the sender side should be greater than acknowledgement timer (otherwise TOT timer will expire more often & lead to retransmissions)

11. Relationship b/w window sizes and sequence no.

W_S W_R

Minimum sequence numbers required in GBN = $N + 1$

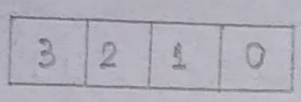
Bits required in seq. no. field = $\lceil \log_2 (N+1) \rceil$

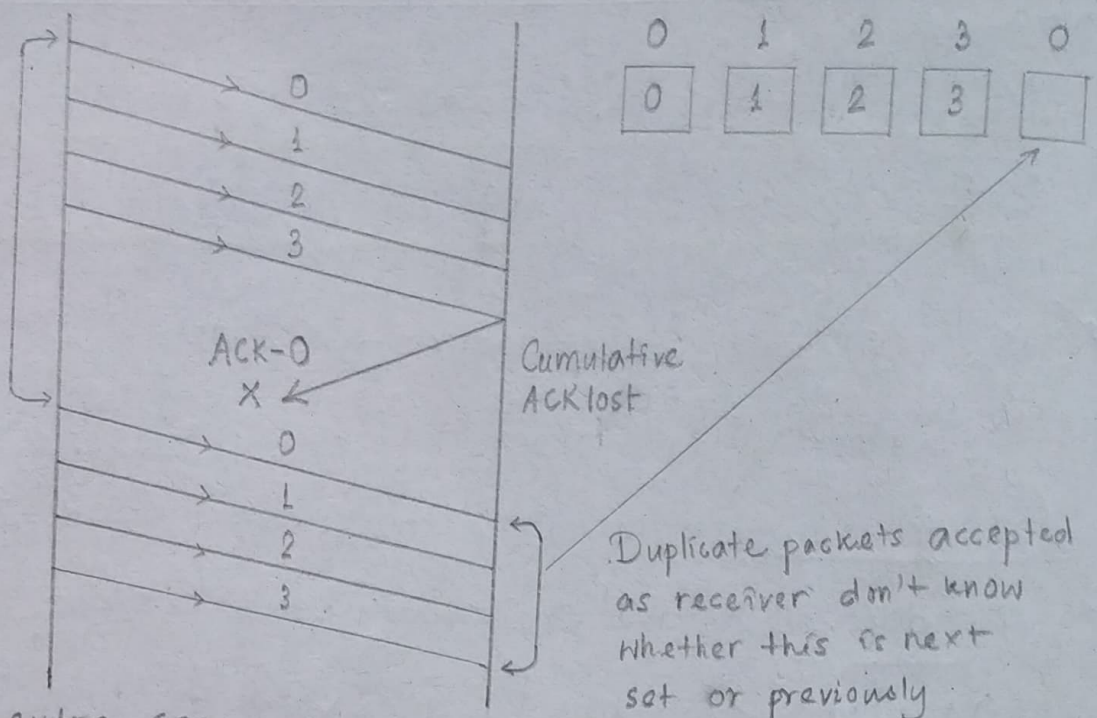
eg. GB4. Cumulative ACK lost

Sender window size is 4, therefore we require a minimum of 4 sequence numbers to label each packet in the window. Now, if receiver has received all the packets (0,1,2 and 3 sent by sender) & has been waiting for packet number 0 again, then it cannot use 4, as we have only 4 sequence numbers. Now, suppose the cumulative ACK for the above 4 packets is lost in the network. On sender side, there will be timeout for packet 0 & hence all the 4 packets will be transmitted again. Problem now is receiver is waiting for new set of packets which should have started from 0 but now it will receive the duplicate copies of the previously accepted packets. In order to avoid this, we need one extra sequence number. Now, the receiver could easily reject all the duplicate packets which were starting from 0 because now it will be waiting for packet number 4 (added an extra sequence number now).

Sequence numbers 4 N = 4

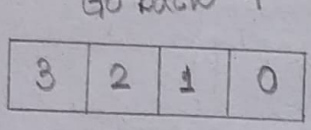
Go Back 4

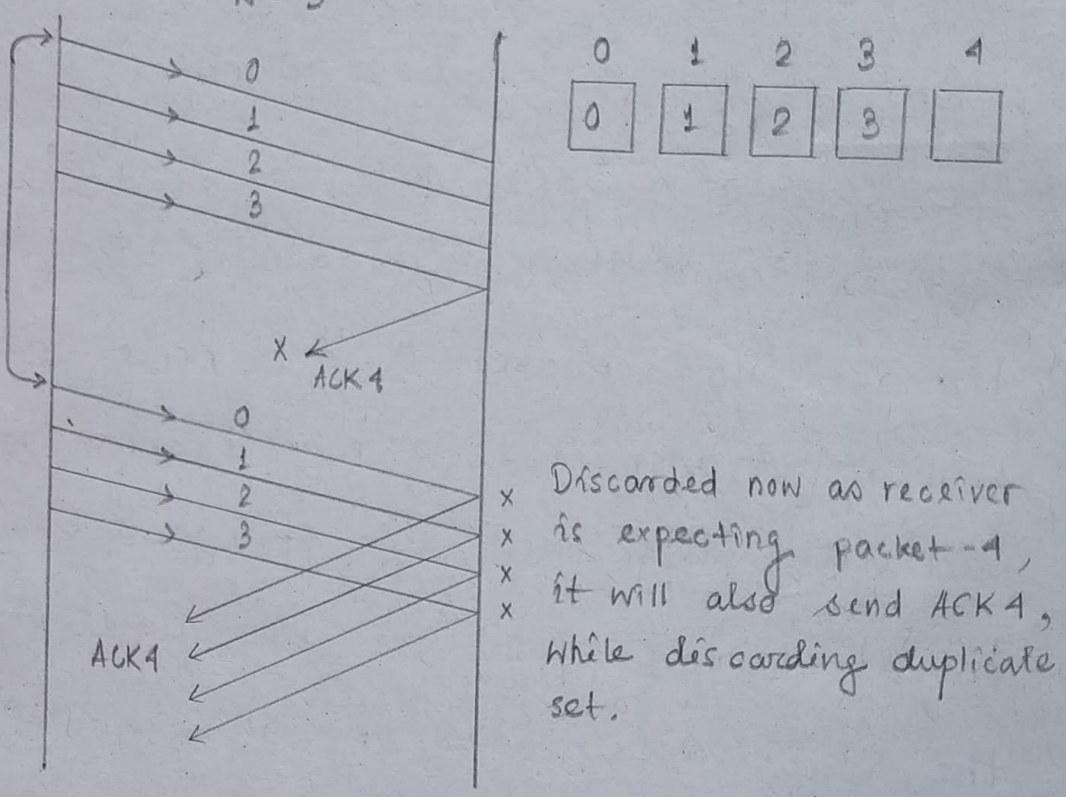




Trying with one extra seq. no.

Go back 4 N = 5





Minimum seq. numbers required thus is N+1. Extra 1 is required in order to avoid the problem of duplicate packets.

Q A 20 kbps satellite link has propagation delay of 400 ms. The transmitter employs the Go back N ARQ with N set to 10. Assuming each frame is 100 bytes long, what is the maximum data rate possible?

→ $B = 20$ kbps $\quad L = 100$ bytes

$T_p = 400$ ms $\quad N = 10$.

$$T_t = \frac{100 \text{ bytes}}{20 \text{ kbps}} = 40 \text{ msec} \quad | \quad a = 10.$$

$$\eta = \frac{N}{1+2a} = \frac{10}{1+2\times10} = 47.6\%$$

Maximum data rate possible / throughput =

$$\eta \times B = 9.52 \text{ kbps}$$

8 Consider the GBN protocol with a sender's window size of n. Suppose that at time t, the next inorder packet the receiver is expecting has a seq. no. of k. Assume that the medium does not reorder messages.

a) What are possible sets of seq. no. inside the sender's window at time t. Assume the sender has already received the ACKs. [k, k+n-1]

→ Receiver window size = 1.

Receiver has processed all the packets ranging from 0 to k-1.

Outstanding packets in sender's window waiting for the acknowledgement starts from k.

Sender window size = n.

Therefore, last packet in sender's window will have sequence number k+n-1.

b) If ACKs are still on their way to sender, what are all possible values of the ACK field in the messages currently propagating back to the sender at a time t? [k, k-n+1]

Receiver expects the packet having sequence number K at time t. It means it has received the packets ranging from 0 to K-1 whose acknowledgements are on the way. For the $K-1^{th}$ packet, ACK no. would be K. For the $K-2^{th}$ packet, ACK no. K-1 & so on.

At any time, max no. of outstanding packets can be n. This is because sender's window size is n. Therefore, the possible values of ACK no. ranges from [K-n+1, ..., K-3, K-2, K-1, K] (total n values). Here, we have assumed that the acknowledgement for all the packets are sent independently. ~~[K, K-n]~~ [K, K-n+1]

Q. Station A needs to send a message consisting of 9 packets to station B using a sliding window (window size 3) and go back n error control strategy. All packets are ready and immediately available for transmission.

If every 5th packet that A transmits gets lost (but no ACKs from B ever get lost), then what is the number of packets that A will transmit for sending the message to B?

→ No. of packets to be sent = 9

Go back N is used where N = 3

Every 5th packet gets lost.

GBN - SWP
Khurram Tanvir
YouTube

1. Since sender window size is 3, so sender sends 3 packets (1,2,3)

3	2	1

Total packets sent = 3

2. After receiving the ACK for packet-1, sender slides its window & sends packet 4.

4	3	2

Total packets sent = 4

3.

5	4	3

Total packets sent = 5

4.

6	5	4

Total packets sent = 6

5.

7	6	5

Total packets sent = 7

Packet 5 gets lost & when time out occurs, sender retransmits packet 5. In GBN, all the following packets are also discarded by the receiver. So, packet-6 and packet-7 are discarded by the receiver and they are also retransmitted. Thus, the entire window is retransmitted.

7	6	5

Total packets sent = 10

Now, the next 5th packet that will be lost is packet 7 (6,7,5,6,7)

7. After receiving the acknowledgement for packet-5, sender slides its window & sends packet-8.

8	7	6

Total packets sent = 11

8.

9	8	7

Total packets sent = 12

9. Packet-7 gets lost & when time out occurs, sender retransmits packet-7 and the following packets.

9	8	7

Total packets sent = 15

Next 5th packet that'll be lost is packet 9 (8,9,7,8,9)

10. After receiving ACK for packet 7, Sender slides its window.

Total packets sent = 13

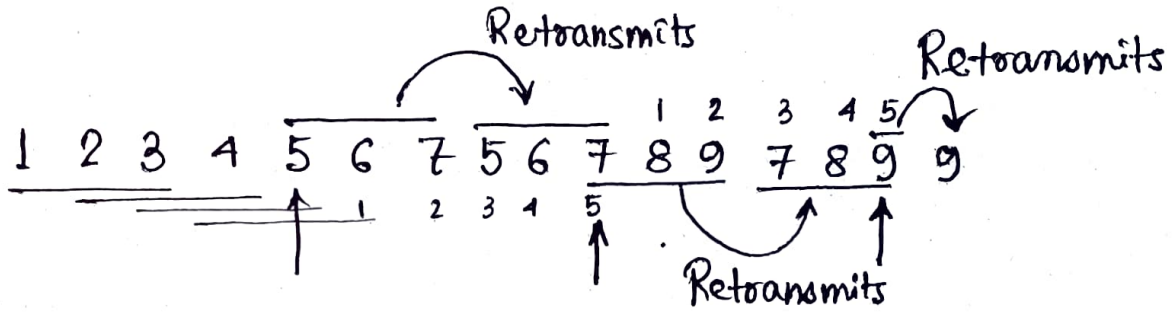
11. 

n = 15

12. Packet 9 gets lost & when time out occurs, sender retransmits packet 9.

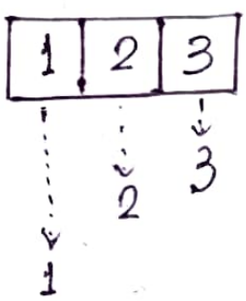
Total packets sent = 16

CN L-13 YT
14.40
20.33

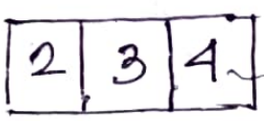


In GBN, $W_S = N$, $W_R = 1$.

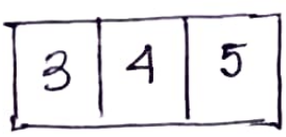
Sender can send N packets without getting acks. for these N packets. When sinder gets the ack for the packets one by one (from first sent to last sent packet), the window is shifted (slided by 1 place); so a new packet is added to the windo
d sent by sender. (ack timer < TOT)

1. 

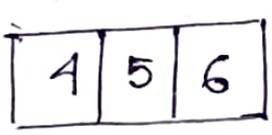
(3) packets sent as $W_S = 3$.

2. ack for 1 received, window shifted by 1 place
(4) 

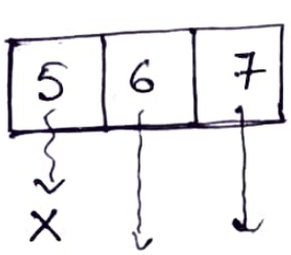
waiting for other acks (2,3,..)
4 packet sent.

3. Similarly,
(5) 

5 sent, but lost.

4. (6) 

shifted for getting ack for 3

5. (7) 

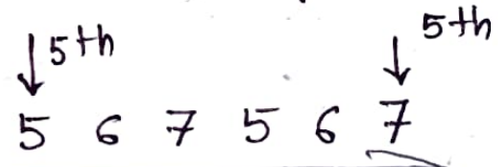
5,6,7 sent but not acknowledged as 5 was not received & GBN does not receive out-of-order packets

6. Again (5,6,7) sent when TOT for 5 in the sender goes off (remember sendoe knows it should have received

the ack for 5th packet before TOT for it goes off as TOT > ack. timer in S in R.
So, ack is always sent before ack timer goes off in the receiver, and obviously before TOT expires)

(10) | 5 | 6 | 7 | retransmission of lost packets

7. Ack for 5 is received => window shifted



(11) | 6 | 7 | 8 |

8. Ack for 6 is received => window shifted

(12) | 7 | 8 | 9 | Now, this 7th packet is lost

X

9. Retransmission of (7,8,9) after TOT for 7 expires at sender

(15) | 7 | 8 | 9 |

10. Ack for 7 received

| 8 | 9 | |

11. Ack for 8 received

| 9 | | | this 9th packet lost

12. Retransmission of (9)

(16) | 9 | | |

Ans = 16

5. A 1 Mbps satellite link connects two ground stations. The altitude of the satellite is 36504 k & speed of the signal is 3×10^8 m/s. What should be the packet size for a channel utilization of 25% for a satellite link using Go back 127 protocol ?

→ BW = 1 Mbps

$d = 2 \times 36504 = 73008$ km

$v = 3 \times 10^8$ m/s

$\eta = 25\% = 0.25$

$N = 127$

packet size, say = L bits

$T_t = \frac{L \text{ bits}}{1 \text{ Mbps}}$

$= L \ \mu\text{sec}$

$T_p = \frac{d}{v} = 243360 \ \mu s$

$a = 243360/L$

$$\eta = \frac{N}{1+2a} \Rightarrow \frac{1}{4} = \frac{127}{1 + 2\,\frac{243360}{L}}$$

$$\Rightarrow \quad L = 960 \text{ bits} \quad \text{or} \quad \underline{120 \text{ bytes}}.$$

Q. Consider a N/W connecting 2 systems located 8000 km apart. The BW of the N/W is 500×10^6 bps. Propagation speed of the media is 4×10^6 m/s. It is needed to design a Go back N protocol. Average packet size is 10^7 bits. (The N/W is to be used to its full capacity.) Assuming processing delays at nodes to be negligible, the minimum size in bits of the sequence number field –

→ $d = 8000$ km

$BW = 500 \times 10^6$ bps

$v = 4 \times 10^6$ m/s

packet size $= 10^7$ bits. $= L$

$\eta = 1$, when used to full capacity.

or when sender window size $= 1 + 2a$

$$T_t = \frac{10^7 \text{ b}}{4 \times 10^6 \text{ m/s}} \quad \frac{L}{B} = 0.02 \text{ sec}$$

$$T_p = \frac{d}{v} = 2 \text{ sec.} \quad | \quad a = 100.$$

$1 + 2a = 201.$

Min. no. of bits in the seq. no. field =

$$\lceil \log_2 (1+2a) \rceil$$

$$= \lceil \log_2 (201) \rceil$$

$$= \lceil 7.65 \rceil = 8.$$

Selective Repeat Protocol.

1. Sender window size is always same as receiver window size.

$$W_S = W_R.$$

Size is of course greater than 1, otherwise protocol will become SAW ARQ.

If n bits are available for sequence numbers, then

$$W_S = W_R = 2^n/2 = 2^{n-1}$$

2. SR protocol uses independent acknowledgements only.

Receiver acknowledges each frame independently As receiver receives a new frame from the sender, it sends its acknowledgement.

3. SR protocol does not accept the corrupted frames but does not silently discard them.

Receiver handles the situation efficiently by sending a negative acknowledgement. (NACK). Negative ACK allows early retransmission of the corrupted frame. It also avoids waiting for the TO timer to expire at the sender side to retransmit the frame.

4. Selective repeat attempts to retransmit only those packets that are actually lost. (due to errors).

Receiver must be able to accept packets out of order. Since receiver must release

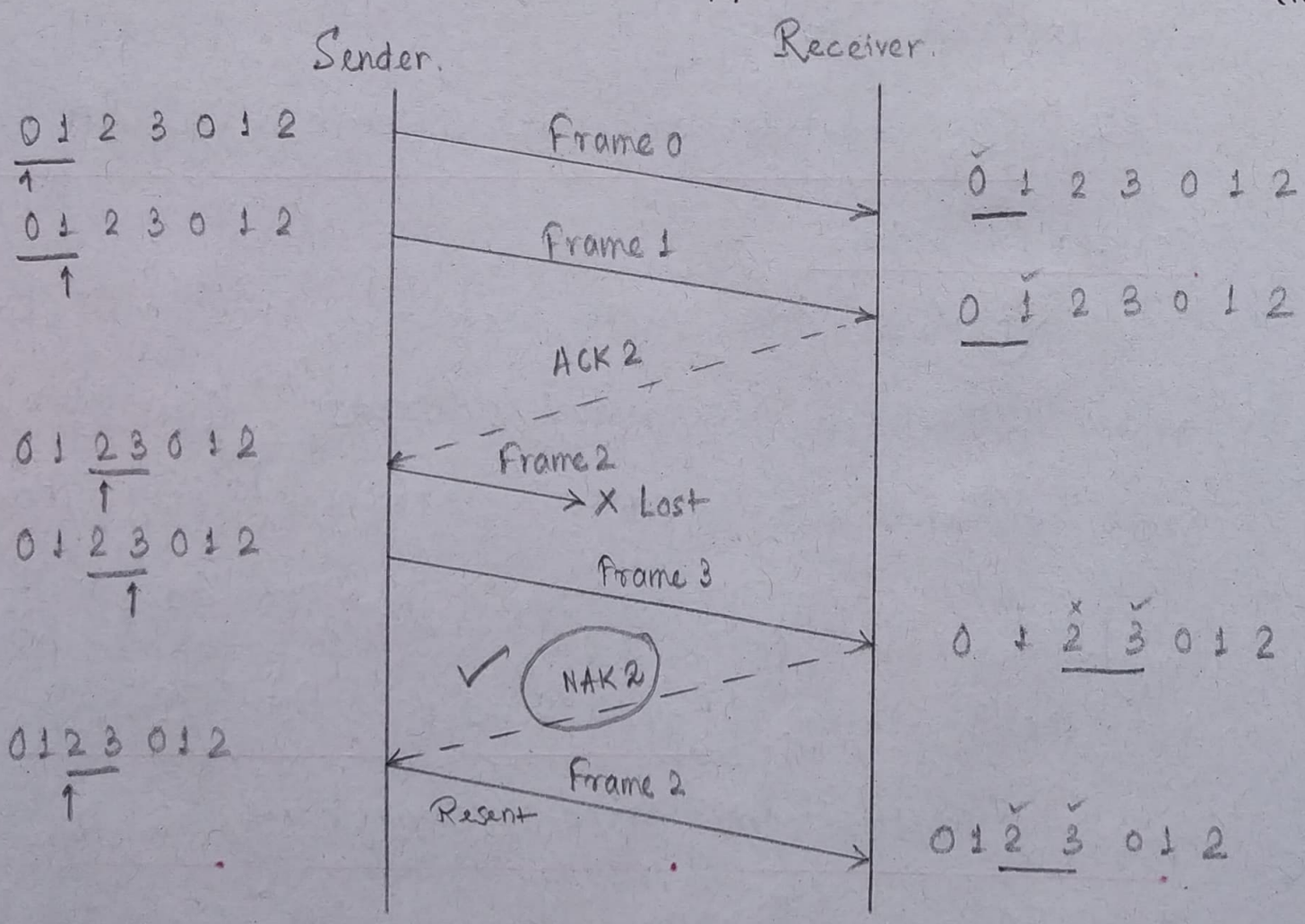
packets to higher layer inorder, the receiver must be able to buffer some packets.

Retransmission requests:

a) Implicit : Receiver acknowledges every good packet; packets that are not ACKed before a time-out are assumed lost or erroneous. This approach must be used to be sure that every packet is eventually received.

b) Explicit : An explicit NAK (selective reject) can request retransmission of just one packet. This approach can expedite the retransmission but is not strictly needed.

- One or both approaches are used in practice.



5. SR protocol accepts the out of order frames.

If receiver receives a frame whose seq. no. is not what the receiver expects, then it does not discard that frame rather accepts it & keeps it in its window.

6. SR protocol requires sorting at the receiver's side.

Receiver's window is implemented as a linked list. When receiver receives a new frame, it places the new frame at the end of the linked list. When the received frames are out of order, receiver performs the sorting. Sorting sorts the frames in the correct order.

7. SR protocol requires searching at the sender's side.

Receiver does not reject the out of order frames. Receiver accepts the out of order frames & sort them later. Thus, only the missing frame has to be sent by the sender. For sending the missing frame, sender performs searching & finds the missing frame. Then, sender selectively repeats that frame. Thus, only selective frame is repeated, not the entire window.

8. SR protocol leads to retransmission of lost frames after expiry of time out timer.

9. Sender must buffer all packets until they are acked. Upto $W_S = W_R$ unacked packets are possible.

10. Receiver must buffer packets until they can be delivered in order, i.e. until all lower numbered packets have been received. Needed for orderly delivery of packets to the higher layer. Upto $W_S = W_R$ packets may have to be buffered.

11. Efficiency of SR protocol.

$$\eta = \frac{N}{1+2a} \quad \Big| \; W_S = W_R = N$$

Q. Maximum window size for data transmission using the selective repeat protocol with n bit frame sequence numbers is —

→ With n bits, total number of seq. no.s possible $= 2^n$.

Min. available sequence numbers $= W_S + W_R$

$$2^n = W_S + W_R = 2W \quad \Big| \; W_S = W_R = W$$

$$\Rightarrow W = 2^{n-1}.$$

Max. window size possible of S & R $= 2^{n-1}$

Q In SR protocol, suppose frames through 0 to 4 have been transmitted. Now, imagine that 0 times out, 5 (a new frame) is transmitted, 1 times out, 2 times out and 6 (another new frame) is transmitted. At this point, what will be the outstanding packets in sender's window ?

→ Only required frame is retransmitted & not the entire window.

1. Frames through 0 to 4 have been transmitted -

   4, 3, 2, 1, 0

2. 0 times out. So, sender retransmits it -

   0, 4, 3, 2, 1

3. 5 (a new frame) is transmitted.

   5, 0, 4, 3, 2, 1.

4. 1 times out. So, sender retransmits it

   1, 5, 0, 4, 3, 2

5. 2 times out. So, sender retransmit it.

   2, 1, 5, 0, 4, 3

6. 6 (new frame) is transmitted.

   6, 2, 1, 5, 0, 4, 3.

Q. The SR protocol is similar to GBN protocol except in the following way -

1. Frame formats are similar in both the protocols.
2. The sender has a window defining maximum number of outstanding frames on both the protocols.

3. Both uses piggybacked ACKs where possible and does not acknowledge every frame explicitly.

4. Both uses piggyback approach that acknowledges the most recently received frame.

→ 1. Both protocols use the same frame formats because both are sliding window protocols. Variation is in the coding & implementation.

2. In both the protocols, sender has a window which defines the maximum number of outstanding frames.

3. Both protocols use piggybacked ACKs wherever possible. Sending ACKs along with the data are called as piggybacked ACKs. But GBN protocol uses cumulative ACKs & does not ACK every frame explicitly. SR uses independent ACK.

4. Both protocols use piggyback approach. GBN acknowledges the most recently received frame by sending a cumulative ACK which includes the ACK for previous packets too if any. SR protocol acknowledges all the frames independently & not only recently received frame.

Q Consider a 128×10^3 bps satellite communication link with one way propagation delay of 150 msec. SR protocol is used on this link to send data with a frame size of 1 KB. Neglect the transmission time of acknowledgement. The minimum number of bits required for the sequence number field to achieve 100% utilization is —

→ $B = 128 \times 10^3$ bps
$T_p = 150$ msec
$L = 1$ KB.

100% utilization ⇒
η 100% ⇒ $W_s = 1 + 2a$

$$T_t = \frac{1 \text{ KB}}{128 \times 10^3 \text{ bps}} = 64 \text{ msec}$$

$$a = T_p / T_t = 2.34.$$

Optimal sender window size = $1 + 2a = \lceil 5.68 \rceil$
$= 6.$

Min. no. of sequence numbers reqd. = $6 \times 2 = 12$

Min. no. of bits reqd. in seq. no. field =
$\lceil \log_2(12) \rceil = 4.$

(Use only 12 sequence numbers, rest 4 remains unused.).

• Comparison of flow control protocols.

	S&W ARQ	GBN	SR	Remarks
Efficiency	$\frac{1}{1+2a}$	$\frac{N}{1+2a}$	$\frac{N}{1+2a}$	GBN and SR gives better efficiency than S&W ARQ.
Window size	$W_S = 1$ $W_R = 1$	$W_S = N$ $W_R = 1$	$W_S = N$ $W_R = N$	Buffer requirement in SR is very large. If does not have lots of memory then choose GBN
Min. no. of seq. no.s reqd.	2	N+1	2N.	SR requires large no. of bits in seq. no. field.
Retransmissions reqd. if a packet lost	Only lost packet retransmitted.	Entire window retransmitted.	Only lost packet retransmitted.	SR far better than GBN in terms of retransmissions reqd.
BW requirement *	Low.	High because even if a single packet is lost, entire window has to be retransmitted.	Moderate. → If error rate is high, it wastes a lot of BW.	SR is better than GBN in terms of BW requirement.
CPU usage *	Low	Moderate	High due to sorting and searching reqd. at S & R side.	GBN is better than SR in terms of CPU.
Level of difficulty in implementation	Low	Moderate	Complex as it requires extra logic & sorting & searching.	GBN better than SR.
Acknowledgements	Independent ACK.	Uses cumul. ACK (may use ind. ack. as well)	Independent ACK.	C.ACK reduces traffic in the N/W but if lost, ACKs for all corresponding packets are lost.
Type of transmission	Half duplex.	Full duplex.	Full duplex.	GBN and SR are better.

NB

1. GBN is more often used than other protocols.
2. SR protocol is less used because of its complexity.
3. SAW ARQ is less used because of its low efficiency.
4. Depending on the context & resources' availability, GBN or SR is employed.
5. SR & SWARQ are similar in terms of retransmissions.
6. GBN & SR are similar in terms of efficiency if sender window sizes are same.

✓ *7. Protocols at data link layer like HDLC (low level protocols) use GBN.

high level data link control

Because -
- BW is high.
- CPU is very busy doing routing job.
- Error rate is low since out of order packets are not possible in wired medium.

so low no. of retransmission

✓ *8. Protocols at transport layer like TCP (high level protocols) use SR.

*Q If the BW between the S & R is sufficient, CPU & buffers are moderate, which flow control method is suggested ?

→ Go back N

* Q If the BW between the S & R is moderate, CPU & buffers are sufficient, then which flow control method is suggested?

→ Selective Repeat Protocol.

Q G'10 Which is not a client-server application?

a) Internet chat
b) Web browsing
c) E-mail
✓d) Ping → reachability test utility. (uses ICMP) If A pings B, B isn't aware of it.

Q 10 packets, every 4th packet lost, using SR protocol. How many retransmissions?

→

1	2	3	4	4	5	6	7	7	8	9	10	10
			↑	1	2	3	4	1	2	3	4	
1	2	3	4				↑				↑	

3 retransmissions.

13 transmissions from sender side

Q Host A is sending data to host B over a full duplex link. A and B are using the sliding window protocol. The sender & receiver window sizes are 5 packets each. Data packets (sent only from A to B) are all 1000 bytes long & the transmission time for such a packet is 50 μs. Acknowledgement packets (sent only from B to A) are very small & require negligible transmission time. The propagation delay over the link is 200 μs. What is the max achievable throughput?

→ $L = 1000$ bytes $= 8000$ bits

$T_t = 50\ \mu s$

$T_p = 200\ \mu s$

$a = \frac{T_p}{T_t} = 4$

$W_s = W_R = 5.$

$$\eta = \frac{N}{1+2a} = \frac{5}{9}.$$

$$T_t = \frac{L}{B} \Rightarrow B = \frac{L}{T_t} = 8 \times \frac{10^9}{5} \text{ bps}$$

Max throughput $= \eta \times B$

$$= \frac{5}{9} \times 8 \times \frac{10^9}{5} \text{ bps}$$

$$= 11.11 \times 10^6 \text{ Bps.}$$

Q Station A uses 32 byte packets to transmit messages to station B using a sliding window protocol. The RTT between A & B is 80 msec. & the bottle-neck bandwidth on the path between A & B is 128 kbps. What is the optimal window size that A should use?

→ $RTT = 80$ ms

$T_p = \frac{RTT}{2} = 40$ ms

$B = 128$ kbps

$L = 32 \times 8 = 256$ bits

$T_t = \frac{L}{B} = 2$ ms

Optimal window size $= 1 + 2a = 1 + 2 \cdot \frac{T_p}{T_t}$

$= 41$

When T_t considered in RTT

2 ms $+ 2T_p = 80$ ms

$T_p = 39$ ms

$W_s = 1 + 2 \cdot \frac{39}{2} = 40$ (Ans)

Q. In what protocols is it possible for the sender to receive an ACK for a packet that falls outside its current window?

1. S&W 2. SR 3. GBN ✓ 4. All. refer GO

→ Delayed acknowledgements fall outside the current window.

Q On a wireless link, the probability of packet error is 0.2. A s&w protocol is used to transfer data across the link. The channel condition is assumed to be independent from transmission to transmission. What is the average number of transmission attempts required to transfer 100 packets?

→ While transferring 100 packets, erroneous packets are $100 \times 0.2 = 20$.

20 packets retransmitted.

While transferring 20 packets, erroneous packets = $20 \times 0.2 = 4$.

While transmitting 4 packets, erroneous packets = $4 \times 0.2 = 0.8 \approx 1$

Average number of transmission attempts = $100 + 20 + 4 + 1 = \underline{125}$.

OR No. of transmission attempts =

$$100 + 100 \times 0.2 + 100 \times 0.2^2 + \dots \infty$$

$$= \frac{100}{1 - 0.2} = 125.$$

- If there are n packets & p is the probability of packet error, then no. of transmission attempts reqd = $n + np + np^2 + np^3 + \dots \infty$

$$= \frac{n}{1-p}.$$

8 Compute the fraction of bandwidth that is wasted on overhead (headers & retransmissions) for a protocol on a heavily loaded 50 Kbps satellite channel with data frames consisting of 40 bits header & 3960 data bits. Assume that the signal propagation time from the earth to the satellite is 270 msec. ACK frames never occur. NAK frames are 40 bits. The error rate for data frames is 1% & the error rate for NAK frames is negligible.

→ Considering 100 frames are being sent.

Useful data sent = 100 x 3960 bits

Useless data sent / overhead :

Due to headers, retransmissions & NAKs.

Error rate for data frames 1%

Out of 100 sent frames, error occurs in one frame. This causes the NAK to follow that causes retransmission.

So,

$$\text{overhead}\Big|_{\text{headers}} = 100 \times 40 = 4000 \text{ bits}$$

$$\text{overhead}\Big|_{\text{NAK}} = 40 \text{ bits.}$$

$$\text{overhead}\Big|_{\text{retrans.}} = \begin{matrix} 40 \text{ b header} \\ + \\ 3960 \text{ b data} \end{matrix} = 4000 \text{ bits}$$

Total overhead = 8040 bits

$$\eta = \frac{396000}{396000 + 8040} = 0.9801$$

BW utilization = $\eta \times B$ = 0.9801 x 50 Kbps

= 49.005 Kbps

BW wasted = (50 − 49.005) = 0.995 Kbps

Fraction of BW wasted = $\frac{0.995}{50}$ = 1.99 %.

Q Consider 1 Mbps error free line. The maximum frame size is 1000 bits. New packets are generated about 1 sec apart. The TO interval is 10 msec. If the ACK timer is eliminated, how many times the average message be transmitted?

→ $T_t = \frac{1000 \text{ bits}}{1 \text{ Mbps}} = 1 \text{ msec.}$

After packet is put on the link, the TO timer is started which is 10 msec long. The next packet is transmitted after 1 sec = 1000 msec.

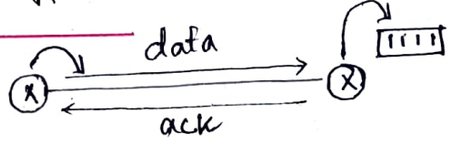
If no ACK is received within 10 msec, the packet will be retransmitted. Retransmissions depends on T_p. If T_p is more, TO will occur & retransmission will take place but if less, no time out. Since, T_p is not given, we can't say anything.

Q. What's the effect on line utilization if we increase the number of frames for a constant message size?

1. Lower line efficiency
2. Higher line efficiency
3. No change in line efficiency
4. No rel^n between line efficiency & frame size.

→ Whether the entire message is sent as a single entity or the entire message is divided into frames & then frames are sent, line utilization remains the same. Because, line contains the same amount of data in both cases.
So, if the number of frames are increased by dividing the message, there is no change in line efficiency.

* General formula for calculating RTT:



RTT = T_r. delay (@ R_1) + 1
P_r. delay (@ $R_1 \xrightarrow[data]{} R_2$) + 2
Que. delay (@ R_2) + 3 — 0
Proc. delay (@ R_2) + 4 — 0
T_r. delay (@ R_2) + 5 — 0
P_r. delay ($R_2 \xrightarrow[ack]{} R_1$) 6

only when the ack is sent individually as ack. packet is small, $T_t \approx 0$. Though, not the case for piggybacked ack.

So, RTT = $T_t + 2T_p$.
Sometimes, for small packet transfer this T_t also ignored, so RTT = $2T_p$.

When receiver only sends acks, T_t for acks is ignored
⇓
RTT = $2T_p$ (used while calculating min frame size in CSMA/CD)

When acks are piggybacked, it's basically sending a complete packet.
⇓
RTT = $T_t + 2T_p$ (While calculating optimal window size $w = 1 + 2a$)

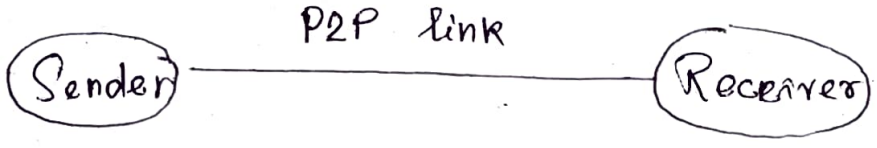
Access Control Methods.

* Communication Links.

Enable the stations to communicate with each other.

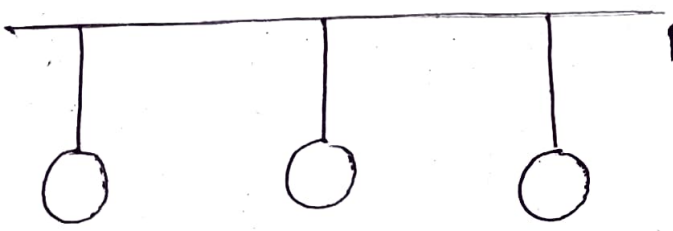
Types:

a) Point to point link: Dedicated link that exists between the two stations. The entire capacity of the link is used for transmission between the two connected stations only.



b) Broadcast Link: Common link to which multiple stations are connected. Capacity of the link is shared among the connected stations for transmission.

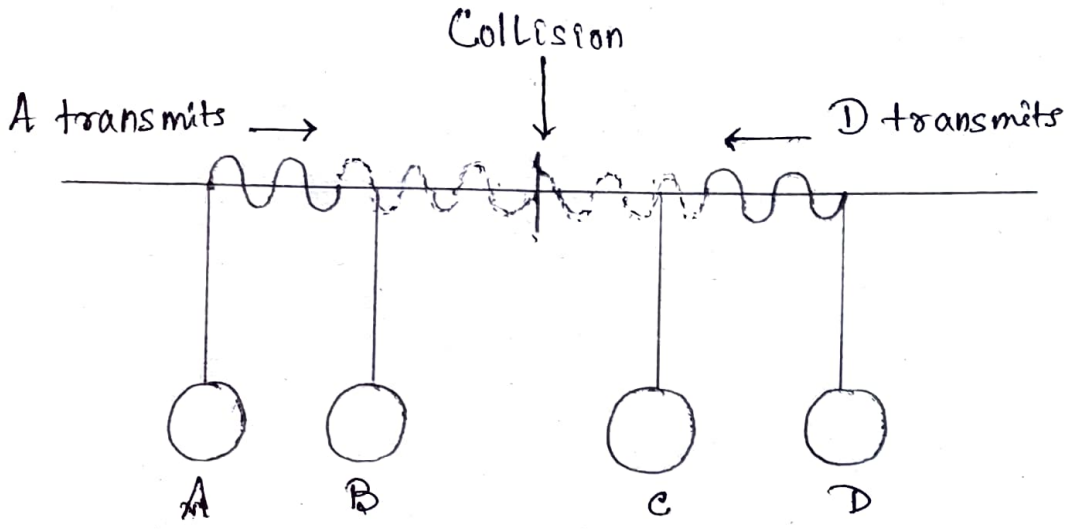
Bus topology.



* Access control: Mechanism that controls the access of stations to the transmission link.

Broadcast link requires the access control.

→ Need of Access control: To prevent the occurrence of collision or if the collision occurs, to deal with it. Collision of data packets causes the data to get corrupt.



* Access control methods.

Implemented at the data link layer of the OSI reference model.

Methods —

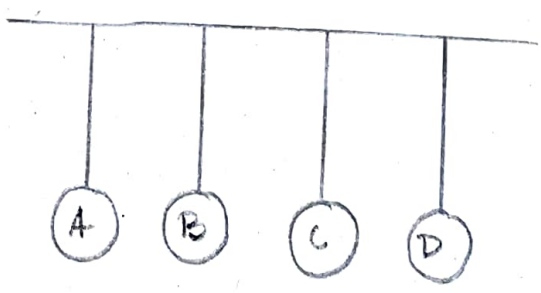
a) Time division multiplexing
b) Polling
c) CSMA/CD
d) Token passing
e) Aloha.

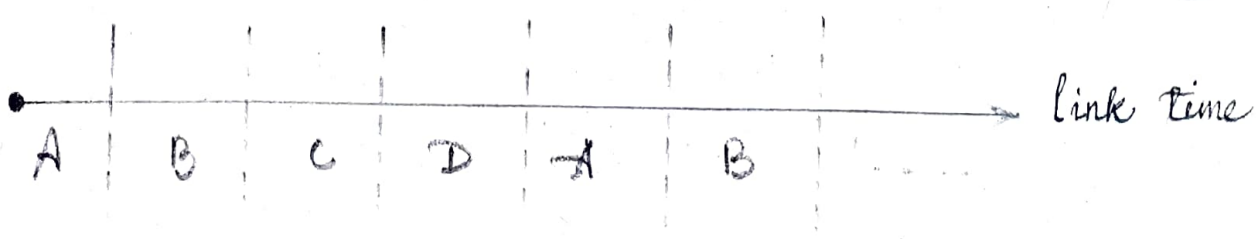
- Time division multiplexing (TDM) Fixed time slots

• Time of the link is divided into fixed size intervals called as time slots or time slices

• Time slots are allocated to the stations in Round Robin manner.

• Each station transmits its data during the time slot allocated to it. In case, station does not have any data to send, its time slot gets wasted.





- Size of time slots.

The size of each time slot is kept such that each station gets sufficient time for following tasks –

a) To put its data packet on to the transmission link

b) Last bit of the packet is able to get out of the transmission link.

Size of each time slot = $T_t + T_p$.

- To keep the size of time slots constant, we have assumed that all the stations want to send the packets of same size. This keeps T_t constant for all the stations. We have considered the worst case when both the stations are present at the two extreme ends. This ensures T_p will be maximum & all the stations will get sufficient time to propagate their data.

- Efficiency. $\eta = \frac{\text{Useful time}}{\text{Cycle time or total time}}$

Useful time = T_t

Useless time = T_p.

$$\eta_{TDM} = \frac{T_t}{T_t + T_p} = \frac{1}{1 + a}$$

Where $a = \frac{T_p}{T_t}$

- Time slot$|_{TDM}$ = $T_t + T_p$.

$$\eta = \frac{1}{1+a}, \quad a = \frac{T_p}{T_t}$$

Throughput = ηB.

- Maximum available effective bandwidth = Throughput

Total no. of stations × BW requirement of 1 station

- Disadvantage:

a) If any station does not have the data to send during its time slot, then its time slot goes wasted.

b) This reduces the efficiency.

c) This time slot could have been allotted to some other station willing to send data.

Q In TDM, T_t = 1 ms, T_p = 1 ms, B = 4 Mbps, then.

1. Find the efficiency.
2. Find the effective bandwidth
3. How many maximum stations can be connected to the network if each station requires 2 Kbps bandwidth ?

→ 1. $a = \frac{T_p}{T_t} = 1$

$$\eta = \frac{1}{1+a} = 50\%$$

2. Effective bandwidth =

$$0.5 \times 4 \text{ Mbps}$$

$$= 2 \text{ Mbps}$$

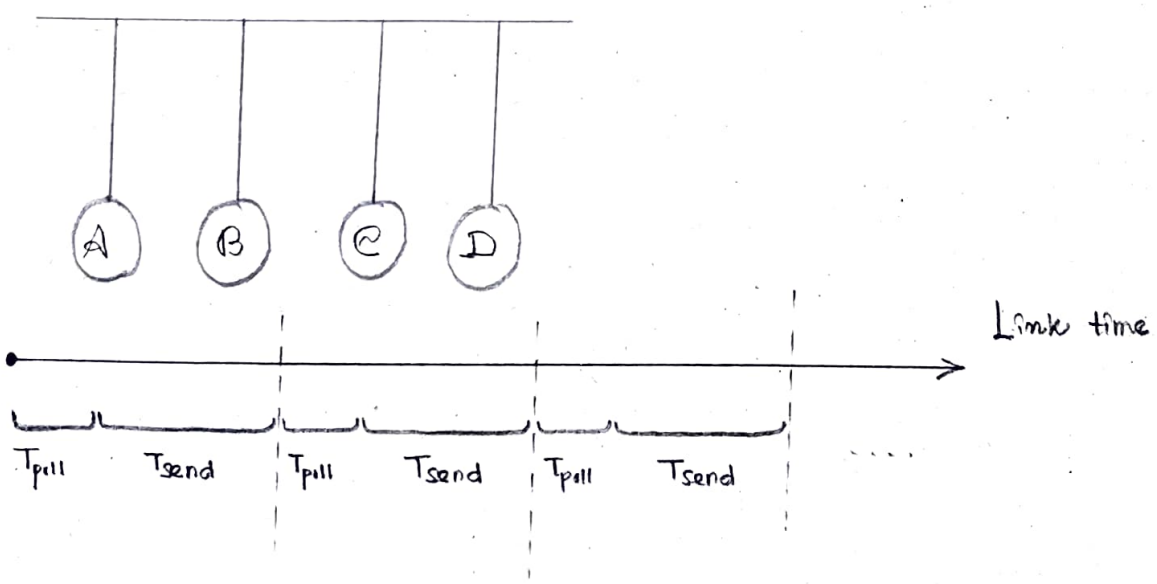
3. Let the total number of stations that can be connected be N.

$$2 \text{ Mbps} = N \times 2 \text{ Kbps}$$

$$\Rightarrow N = 1000.$$

- **Polling** (Time slot given to certain station after polling)

  - A polling is conducted in which all the stations willing to send data participates.
  - The polling algorithm chooses one of the stations to send the data.
  - The chosen station sends the data to the destination. After the chosen station has sent the data, the cycle repeats.



T_{send} = Time taken for sending the data

= Transmission delay + Propagation delay

= $T_t + T_p$

- Efficiency.

Useful time = Transmission delay = T_t

Useless time = $T_{poll} + T_p$

$$\eta = \frac{T_t}{T_{poll} + T_t + T_p}$$

- Advantages.

Unlike TDM, no time slot is wasted.

Leads to maximum efficiency & bandwidth utilization.

- Disadvantages.

Time is wasted during polling.

Link sharing is not fair since each station has the equal probability of winning in each round. Few stations might starve for sending the data.

- Maximum available effective bandwidth =

Total no. of stations × BW requirement of 1 station.

* CSMA/CD used in early Ethernet technology, in LAN.

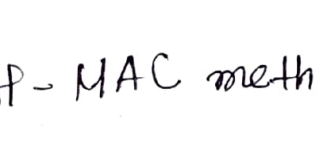
- **CSMA/CD Protocol.** (Media Ac. Control - MAC method)

Carrier sense multiple access / collision detection

→ **Sensing the carrier.** Any station willing to transmit the data senses the carrier. If it finds the carrier free, it starts transmitting its data packet otherwise not.

Each station can sense the carrier only at its point of contact with the carrier. It's not possible for any station to sense the entire carrier. So, there's possibility that a station might sense the carrier free even when it's actually not.

→ **Detecting collision.** It's the responsibility of the transmitting station to detect the collision. For detecting collision, CSMA/CD implements following condition,



$$\text{Transmission delay} \geq 2 \times \text{Propagation delay}$$

This condition is followed by each station.

*Each station must transmit the data packet of size whose transmission delay is at least twice its propagation delay. If the size of data packet is smaller, then collision detection would not be possible.

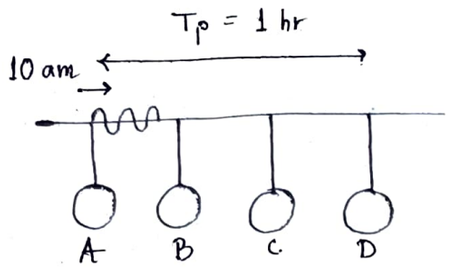
$$T_t \geq 2T_p$$

$$\Rightarrow L/B \geq 2\frac{d}{v}$$

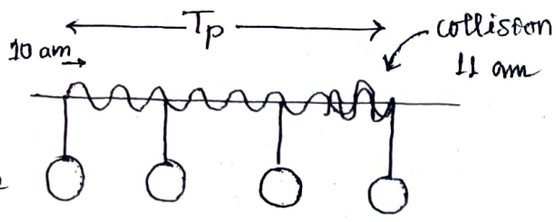
$$\Rightarrow L \geq \frac{2Bd}{v}$$

→ Example ($T_t \geq 2T_p$)

- Consider at 10 am, A senses the carrier to be free & starts transmitting its data packet to station D. Let, T_p = 1 hr.



- At 10:59:59, packet is about to reach station D. At this time, D senses the carrier & finds the carrier free, so starts transmitting data. Now as soon as D starts transmitting, a collision occurs with the data packet of station A. at time ≈ 11 am.



- After collision occurs, the collided signal starts travelling in the backward direction. The collided signal takes 1 hr to reach A after the collision has occurred. For A to detect the collided signal, it must be still transmitting the data. So, transmission delay of station A must be ≥ 1 hr + 1 hr = 2 hr to detect the collision. This is why, for detecting the collision, condition is $T_t \geq 2T_p$.

→ While detecting collision, two cases are possible –

1. If no collided signal comes back during the transmission, it indicates that no collision has occurred. The data packet is transmitted successfully.

2. If the collided signal comes back during the transmission, it indicates that the collision has occurred. The data packet is not transmitted successfully. Then jam signal is released.

→ Releasing jam signal.

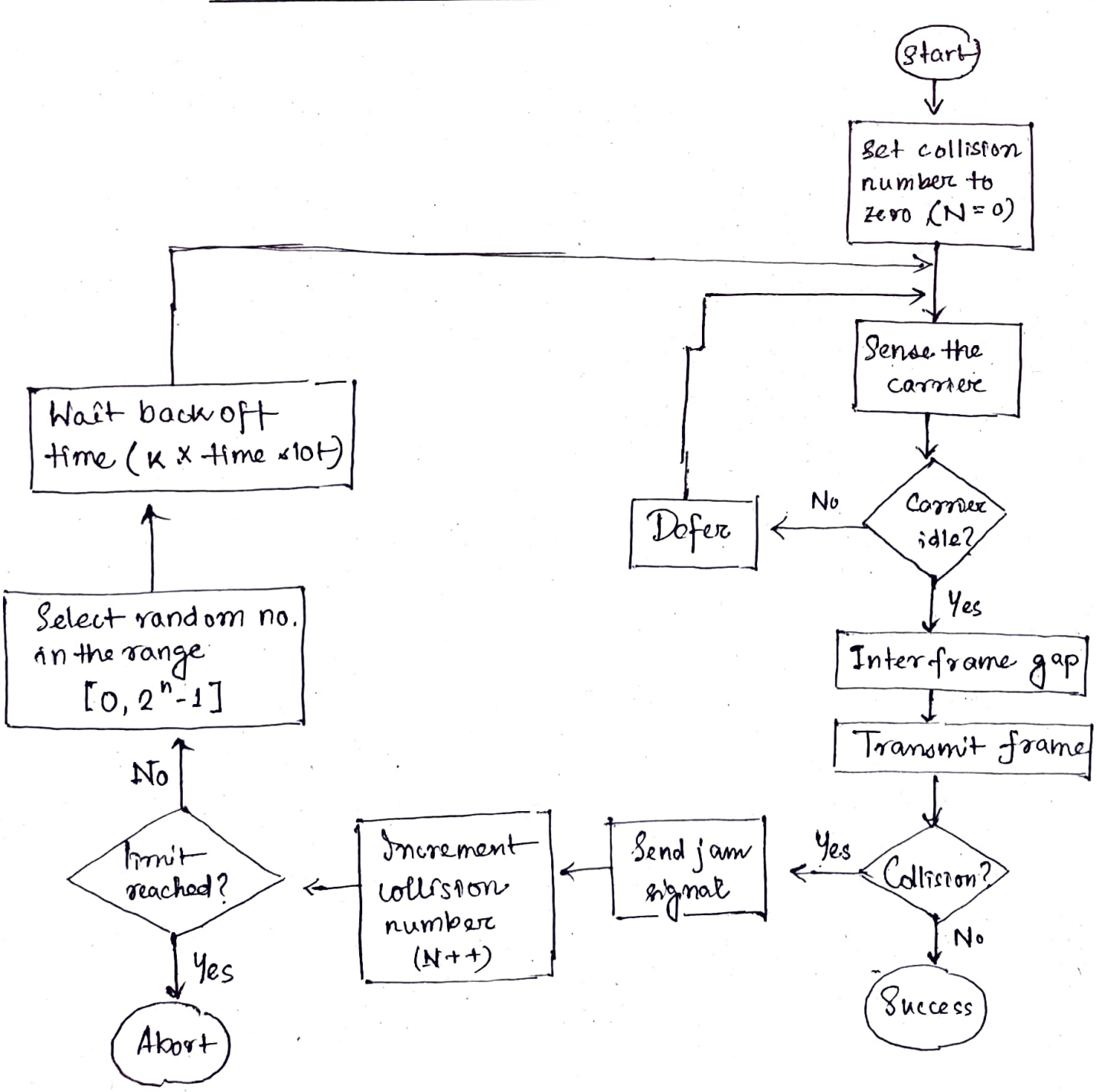
Jam signal is a 48 bit signal. It is released by the transmitting stations as soon as they detect a collision. It alerts the other stations not to transmit their data immediately after the collision. Otherwise, there is a possibility

of collision again with the same data packet. Ethernet sends the jam signal at a frequency other than the frequency of data signals. This ensures that jam signal does not collide with the data signals undergone collision.

→ Waiting for back off time.

After the collision, the transmitting station waits for some random amount of time called as back-off time. After back-off time, it tries transmitting the data packet again. If again the collision occurs, then station again waits for some random back-off time & then tries again. The station keeps trying until the back off time reaches its limit. After the limit is reached, station aborts the transmission. Back off time is calculated using Back off algorithm.

→ CSMA/CD flowchart.



→ Efficiency :

Before a successful transmission, there may occur many number of collisions. $2T_p$ time is wasted during each collision (at max).

Useful time = T_t.

Useless time = Time wasted during collisions + Propagation delay of packet

$= c \times 2T_p + T_p$

c = # of contention slots / collision slots.

$$\eta = \frac{T_t}{c \times 2T_p + T_t + T_p}$$ | c variable.

→ Probabilistic analysis - Average no. of collisions before a successful transmission.

# of stations connected to CSMA/CD N/W = n

Probability of each station to transmit the data = p.

Transmission will be successful when -

one station transmits the data, other n-1 station do not transmit the data.

$$Pr(\text{successful transmission}) = {}^{n}C_1 \times p \times (1-p)^{n-1}$$

For maximum value,

$$\frac{d\,Pr(\text{suc. tra}^n)}{dp} = 0.$$

$\Rightarrow p = 1/n$

$$Pr(\text{successful transmission})_{max} = {}^{n}C_1 \times \frac{1}{n} \times \left(1 - \frac{1}{n}\right)^{n-1}$$

$$Pr\left(\begin{matrix}\text{successful}\\ \text{transmission}\end{matrix}\right)_{max} = \left(1 - \frac{1}{n}\right)^{n-1}$$

When $n \to \infty$, sufficiently large no. of stations,

$$\lim_{n \to \infty} \left(Pr_{suc.\,t^n\,max}\right) = \lim_{n \to \infty} \left(1 - \frac{1}{n}\right)^{n-1} = \frac{1}{e}$$

Number of times a station must try before successfully transmitting the data packet (recurrence interval, MRI, poisson's dist.^{n})

$$= \frac{1}{P_{max}} = \frac{1}{(1/e)} = e.$$

So, average number of collisions that might occur before a successful transmission $= e$.

$$\rightarrow \eta = \frac{T_t}{e \times 2T_p + T_t + T_p}$$

$$\Rightarrow \eta = \frac{T_t}{e \times 2T_p + T_t + T_p} = \frac{T_t}{T_t + 6.44T_p}$$

$$\Rightarrow \eta = \frac{1}{1 + 6.44a}, \quad a = \frac{T_p}{T_t}$$

→ CSMA/CD is used in Wired LANs. It is standardized in IEEE 802.3.

→ CSMA/CD only minimizes the recovery time. It does not take any steps to prevent the collision until it has taken place.

→ Binary exponential back off algorithm.

• Back off time

Waiting time for which the station waits before retransmitting the data. Back off algorithm used for calculating the back off time.

• Algorithm.

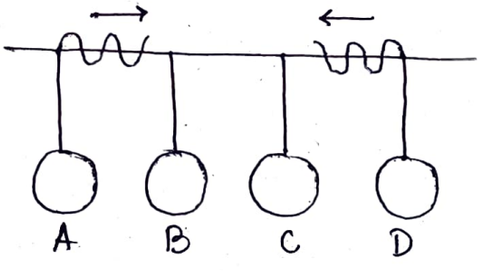
After undergoing the collision, transmitting station chooses a random number in the range $[0, 2^n - 1]$ if the packet is undergoing collision for the n^{th} time. If station chooses a number K then,

Back off time = K x Time slot.

Where value of 1 time slot =

1 round trip time

Example.



A & D start simultaneously.

Consider time slot = 1 unit.

a) Scene 1. (For 1st data packet of both stations).

Both the stations start transmitting their 1st data packet simultaneously. This leads to a collision. Collision number for the 1st data packet of both the stations = 1.

After detecting the collision, A randomly chooses a number in the range $[0, 2^1-1] = [0,1]$. If A chooses K_A number, back off time = K_A units.

Similarly, D randomly chooses in $[0,1]$. If D chooses K_D number, back off time = K_D units.

4 cases (2×2) arise -

K_A	K_D	
0	0	Both stations start retransmitting immediately, leading to collision.
0	1	A starts retransmitting immediately while D waits for 1 unit of time. Successful retransmission of data for A after 1st collision.
1	0	A waits for 1 unit of time while D starts retransmitting its data immediately. Leads to successful retransmission of data for D after 1st collision.
1	1	Both wait for 1 unit & then starts retransmitting leading to collision.

$Pr(\text{A successfully retransmits after 1st collision}) = \frac{1}{4}$

$Pr(\text{D successfully retransmits after 1st collision}) = \frac{1}{4}$

$Pr(\text{Collision after 1st collision}) = \frac{2}{4} = \frac{1}{2}$

Now, considering 2nd case $(0,1)$ - A successfully retransmits its 1st packet after the 1st collision.

Scene-02: For 2nd Data Packet Of Station A And 1st Data Packet Of Station D-

After 2nd case (0,1) of scene 1

Consider after some time,

- Station A starts transmitting its 2nd data packet and station D starts retransmitting its 1st data packet simultaneously.
- This leads to a collision.

At Station A-

- The 2nd data packet of station A undergoes collision for the 1st time.
- So, collision number for the 2nd data packet of station A = 1.
- Now, station A randomly chooses a number in the range $[0, 2^1-1] = [0,1]$. 2 cases
- If station A chooses the number K_A, then back off time = K_A units.

At Station D-

- The 1st data packet of station D undergoes collision for the 2nd time.
- So, collision number for the 1st data packet of station D = 2. 4 cases
- Now, station D randomly chooses a number in the range $[0, 2^2-1] = [0,3]$.
- If station D chooses the number K_D, then back off time = K_D units.

Following 8 cases are possible-

K_A	K_D	Remarks
0	0	• In this case, both the stations start retransmitting their data immediately. • This case leads to a collision again.
0	1	• In this case, station A starts retransmitting its data immediately while station D waits for 1 unit of time. • This case leads to A successfully retransmitting its data after the 2nd collision.
0	2	• In this case, station A starts retransmitting its data immediately while station D waits for 2 unit of time. • This case leads to A successfully retransmitting its data after the 2nd collision.
0	3	• In this case, station A starts retransmitting its data immediately while station D waits for 3 unit of time. • This case leads to A successfully retransmitting its data after the 2nd collision.
1	0	• In this case, station A waits for 1 unit of time while station D starts retransmitting its data immediately. • This case leads to D successfully retransmitting its data after the 2nd collision.
1	1	• In this case, both the stations wait for 1 unit of time and then starts retransmitting their data simultaneously.

		• This case leads to a collision again.
1	2	• In this case, station A waits for 1 unit of time while station D waits for 2 unit of time. • This case leads to A successfully retransmitting its data after the 2nd collision.
1	3	• In this case, station A waits for 1 unit of time while station D waits for 3 unit of time. • This case leads to A successfully retransmitting its data after the 2nd collision.

From here,

- Probability of station A to successfully retransmit its data after the 2nd collision = 5 / 8
- Probability of station D to successfully retransmit its data after the 2nd collision = 1 / 8 (1,0)
- Probability of occurrence of collision again after the 2nd collision = 2 / 8 = 1 / 4 (0,0) (1,1)

Now,

- Consider case-03 occurs. (0,2).
- This causes station A to successfully retransmit its 2nd packet after the 2nd collision.

Scene-03: For 3rd Data Packet Of Station A And 1st Data Packet Of Station D-

Consider after some time,

- Station A starts transmitting its 3rd data packet and station D starts retransmitting its 1st data packet simultaneously.
- This leads to a collision.

At Station A-

- The 3rd data packet of station A undergoes collision for the 1st time.
- So, collision number for the 3rd data packet of station A = 1.
- Now, station A randomly chooses a number in the range $[0, 2^1-1] = [0,1]$.
- If station A chooses the number K_A, then back off time = K_A unit.

At Station D-

- The 1st data packet of station D undergoes collision for the 3rd time.
- So, collision number for the 1st data packet of station D = 3.
- Now, station D randomly chooses a number in the range $[0, 2^3-1] = [0,7]$.
- If station D chooses the number K_D, then back off time = K_D unit.

Following 16 cases are possible-

K_A	K_D	Remarks
0	0	• In this case, both the stations start retransmitting their data immediately.

		• This case leads to a collision again.
0	1	• In this case, station A starts retransmitting its data immediately while station D waits for 1 unit of time. • This case leads to A successfully retransmitting its data after the 3rd collision.
0	2	• In this case, station A starts retransmitting its data immediately while station D waits for 2 unit of time. • This case leads to A successfully retransmitting its data after the 3rd collision.
0	3	• In this case, station A starts retransmitting its data immediately while station D waits for 3 unit of time. • This case leads to A successfully retransmitting its data after the 3rd collision.
0	4	• In this case, station A starts retransmitting its data immediately while station D waits for 4 unit of time. • This case leads to A successfully retransmitting its data after the 3rd collision.
0	5	• In this case, station A starts retransmitting its data immediately while station D waits for 5 unit of time. • This case leads to A successfully retransmitting its data after the 3rd collision.
0	6	• In this case, station A starts retransmitting its data immediately while station D waits for 6 unit of time. • This case leads to A successfully retransmitting its data after the 3rd collision.
0	7	• In this case, station A starts retransmitting its data immediately while station D waits for 7 unit of time. • This case leads to A successfully retransmitting its data after the 3rd collision.
1	0	• In this case, station A waits for 1 unit of time while station D starts retransmitting its data immediately. • This case leads to D successfully retransmitting its data after the 3rd collision.
1	1	• In this case, both the stations wait for 1 unit of time and then starts retransmitting their data simultaneously. • This case leads to a collision again.
1	2	• In this case, station A waits for 1 unit of time while station D waits for 2 unit of time. • This case leads to A successfully retransmitting its data after the 3rd collision.
1	3	• In this case, station A waits for 1 unit of time while station D waits for 3 unit of time. • This case leads to A successfully retransmitting its data after the 3rd collision.
1	4	• In this case, station A waits for 1 unit of time while station D waits for 4 unit of time. • This case leads to A successfully retransmitting its data after the 3rd collision.
1	5	• In this case, station A waits for 1 unit of time while station D waits for 5 unit of time. • This case leads to A successfully retransmitting its data after the 3rd collision.
1	6	• In this case, station A waits for 1 unit of time while station D waits for 6 unit of time. • This case leads to A successfully retransmitting its data after the 3rd collision.

1	7	• In this case, station A waits for 1 unit of time while station D waits for 7 unit of time. • This case leads to A successfully retransmitting its data after the 3rd collision.

From here,

- Probability of station A to successfully retransmit its data after the 3rd collision = 13 / 16
- Probability of station D to successfully retransmit its data after the 3rd collision = 1 / 16
- Probability of occurrence of collision again after the 3rd collision = 1 / 16

In the similar manner, the procedure continues.

Important Notes-

Note-01:

With each successive collision-

- Back off time increases exponentially.
- Collision probability decreases exponentially.

Note-02:

Back Off Algorithm is also known as **Binary Exponential Back Off Algorithm** because-

- It works for only two stations.
- The back off time increases exponentially.
- Collision probability decreases exponentially.

Note-03:

- One disadvantage of Back Off Algorithm is that it shows capture effect.
- It means if a particular station wins the collision one time, then its probability of winning the successive collisions increases exponentially.

Capture effect

If a station(A) gets lucky during some back-off procedure & gets to send its packet before some other station's(B) signal collided, then the probability of sending further packets successfully, increases (exponentially) for station A.

(Because, after each collision the random choice ($2^{n}-1, 0$) is made again & while A successfully transmits its previous packet, (& B couldn't transmit). the # collision for A's packet is 1. & for B's packet its obviously > 1. So, chances are more that A will pick a random no. which is less than the random no. chosen by B in the current round of back-off algorithm.

Q G'15. Consider a CSMA/CD N/W that transmits data at a rate of 100 Mbps (10^8 bps) over a 1 km cable with no repeaters. If the minimum frame size required for this network is 1250 bytes, what is the signal speed (km/s) in the cable?

→ $T_t \geq 2T_p$

$\Rightarrow L/B \geq 2\, d/v$

$\Rightarrow \dfrac{1250 \times 8}{10^8} \geq 2\,\dfrac{1}{v}$

$\Rightarrow v \geq \dfrac{2 \times 10^8}{1250 \times 8} = 20000$ km/s.

Q. In a CSMA/CD N/W running at 1 Gbps over 1 km cable with no repeaters, the signal speed in the cable is 2×10^5 km/s. What is the minimum frame size?

→ $B = 1$ Gbps
$d = 1$ km
$v = 2 \times 10^5$ km/s.

$T_p = \dfrac{d}{v} = 5 \times 10^{-6}$ s

$L_{min} = 2 T_p B = 10000$ bits

Q A 2 km long broadcast LAN has 10^7 bps bandwidth & uses CSMA/CD. The signal travels along the wire at 2×10^8 m/s. What is the minimum packet size that can be used on this network?

→ $T_p = \dfrac{d}{v} = 10^{-5}$ s

$L_{min} = 2 T_p B = 25$ bytes

Q. A & B are the only 2 stations on Ethernet. Each has a steady queue of frames to send. Both A & B attempt to transmit a frame, collide and A wins first back off race. At the end of this successful transmission by A, both A & B attempt to transmit & collide. The probability that A wins the second back off race is —

→ 1st transmission attempt ~

Both attempt to transmit a frame. A collision occurs. Back off algorithm runs. A wins & successfully transmits its 1st data packet.

2nd transmission attempt ~

A attempts to transmit its 2nd data packet & B attempts to retransmit its 1st data packet. A collision occurs.

After 2nd collision -

- at station A : 2nd packet of A undergoes collision for the 1st time. A randomly chooses a number from the range $[0, 2^1-1] = [0,1]$. Then, A waits for back off time & then attempts to retransmit its data packet.
- at station B : 1st packet of B undergoes collision for the 2nd time.

So, B randomly chooses a number from the range $[0, 2^2-1] = [0,3]$. B waits for back off time & then attempts to retransmit its packet.

Following 8 cases are possible —

A	B	Remark
0	0	Collision
0	1	A wins
0	2	n
0	3	n
1	0	B wins
1	1	Collisions
1	2	A wins
1	3	n

$$Pr(A \text{ wins}) = \frac{5}{8} = 0.625$$

Q Suppose nodes A and B are on same 10 Mbps ethernet segment & T_p between 2 nodes is 225 bit times. Suppose A & B send frames at $t = 0$, the frames collide. At what time they finish transmitting a jam signal? Assume a 48 bit jam signal.

→ $T_p = \frac{225 \text{ bit}}{10 \text{ Mbps}} = 22.5 \ \mu s.$

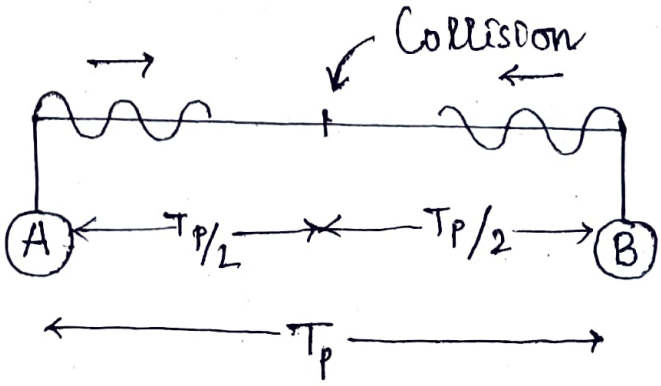
At $t = 0$,

A & B start transmitting their frame.

Since, both start simultaneously, collision occurs at the mid way.

Time after which collision occurs =

half of propagation delay = 11.25 μs.



At $t = 11.25$ μsec,

After collision occurs, collided signals start travelling back. Collided signals reach the respective nodes at $(11.25 + 11.25) = 22.5$ μs.

At $t = 22.5$ μsec,

As soon as nodes discover the collision, they immediately release the jam signal.

Time taken to finish transmitting the jam signal

$= 48$ bit time

$= 48$ bits $/ 10$ Mbps $= 4.8$ μsec.

Time at which the jam signal is completely transmitted $= (22.5 + 4.8) = 27.3$ μsec.

$= 273$ bit times.

Q. Suppose nodes A & B are attached to opposite ends of the cable with $T_p = 12.5$ ms. Both nodes attempt to transmit at $t = 0$. Frames collide & after first collision, A draws $K = 0$ & B draws $K = 1$ in the exponential back off protocol. Ignore the jam signal. At what time (in seconds), is A's packet completely delivered at B if bandwidth of the link is 10 Mbps & packet size is 1000 bits.

→ $T_p = 12.5$ ms

$BW = 10$ Mbps

$L = 1000$ bits $\quad RTT = 2T_p = 25$ ms

Collision occurs at the mid way after time $= 12.5/2 = 6.25$ msec.

Collision is discovered at time $(6.25 + 6.25)$ ms

$= 12.5$ ms

After the collision is discovered,

both nodes wait for some random back off time. A chooses $k = 0$ & then waits for back off time 0×25 ms $= 0$ ms, B chooses $k = 1$ & then waits for back off time $= 1 \times 25$ ms $= 25$ ms. A begins retransmission immediately while B waits for 25 ms.

Waiting time for A -

after winning the back off race, A gets the authority to retransmit immediately. But, A does not retransmit immediately. It waits for the channel to clear from the last bit aborted by it on discovering the collision. Time taken by the last bit ~~aborted~~ to get off the channel $= T_p = 12.5$ ms. So, A waits for time 12.5 ms & then starts the retransmission. Thus,

A starts the retransmission at time $= (12.5 + 12.5)$ ms $= 25$ ms. [A considers it must wait $2T_p$ time before sending another frame, as $T_t \geq 2T_p$]

Time taken to deliver the packet to B =

$$T_t + T_p$$

$$= \frac{1000 \text{ bits}}{10 \text{ Mbps}} + 12.5 \text{ ms}$$

$$= 12.6 \text{ ms}.$$

Thus, at time $(25 + 12.6) = 37.6$ ms the packet is delivered to B node.

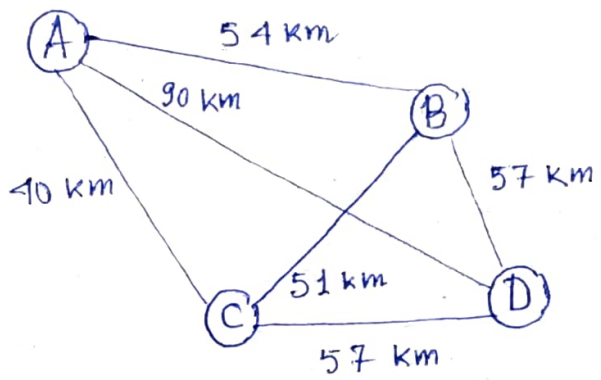
as $T_t \geq 2T_p$

[Or $\overbrace{2T_p} + (T_t + T_p)$

$= 25 + (12.6)$]

As in CSMA, L is set such as $T_t = 2T_p$.

8. The network consists of 4 hosts distributed as -



Assume this N/W uses CSMA/CD & signal travels with a speed of 3×10^5 km/s. If sender sends at 1 Mbps, what could be the minimum size of the packet ?

→ As all the links are P2P, no need to implement CSMA/CD here (CSMA/CD for broadcast link). Stations can transmit whenever they want to transmit.

In CSMA/CD,

$$T_t \geq 2T_p.$$

$$\Rightarrow L \geq 2\frac{d}{v}B$$

We assume that a packet of same length has to be used in the entire network. To get the minimum length of the packet, we should choose the minimum distance. But, then collision would be detected only in the links having distance less than or equal to that minimum distance. For the links having distance greater than the minimum distance, collision would not be detected. So, we choose the maximum distance so that collision can be detected in all the links.

$d = 90$ km

$v = 3 \times 10^5$ km/s

$B = 1$ Mbps

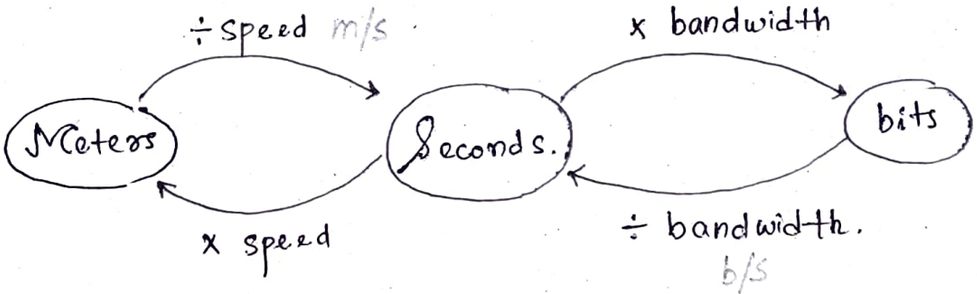
Minimum size of data packet =

$$2 \times \frac{90 \text{ km}}{3 \times 10^5 \text{ km/s}} \times 1 \text{ Mbps}$$

$$= 600 \text{ bits}$$

Access Control Methods (Contd.)

* Time conversion



Token Passing

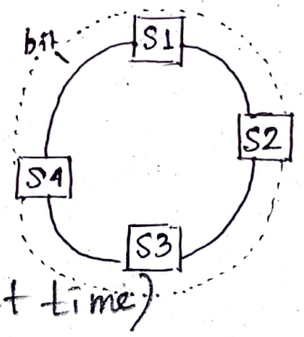
→ **Token**: Small message composed of a special bit pattern that represents the permission to send the data packet. A station is allowed to transmit a data packet if & only if it possesses the token otherwise not.

→ **Ring latency**: Time taken by a bit to complete one revolution of the ring.

Let, length of ring be d

Speed of the bit v

Number of stations N



Bit delay at each station = b (bit time)

(time for which a station holds the bit before transmitting to the other side).

$$\text{Ring latency} = \frac{d}{v} + N \times b$$

$d/v = T_p$ in seconds

$N \times b$ in bits

While calculating ring latency, both terms are brought into the same unit.

After conversion,

$$\text{Ring latency} = \left(\frac{d}{v} + \frac{N \times b}{B}\right) \text{ sec.}$$

$$= \left(\frac{d \times B}{v} + N \times b\right) \text{ bits}$$

→ Cycle time : Time taken by the token to complete one revolution of the ring.

The time for which a station holds the token before transmitting to the other side, is called token holding time (THT).

$$\text{Cycle time} = \frac{d}{v} + N \times THT$$

$$= T_p + N \times THT$$

→ Method :

All the stations are logically connected to each other in the form of a ring. The access of stations to the transmission link is governed by token. A station is allowed to transmit a data packet if it only if it possesses the token. Each station passes the token to its neighbouring station either clockwise or anti-clockwise.

Token passing method assumes -

a) Each station in the ring has the data to send.

b) Each station sends exactly one data packet after acquiring the token.

→ Efficiency :

In one cycle,

✓ Useful time = Sum of transmission delay of N stations since each station sends 1 data packet = $N \times T_t$

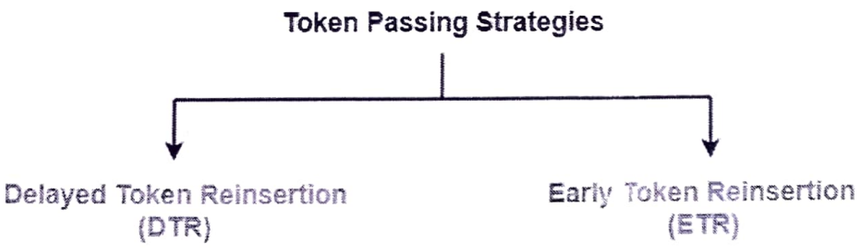
✓ Total time = $T_p + N \times THT$

$\frac{d_{total}}{v}$ (as the station sends data while it's holding token; T_t included in THT)

Thus,

$$\eta = \frac{N * T_t}{T_p + N * THT}$$

THT depends on the strategy implemented.



1. Delayed Token Reinsertion (DTR)
2. Early Token Reinsertion (ETR)

1. Delayed Token Reinsertion-

In this strategy,

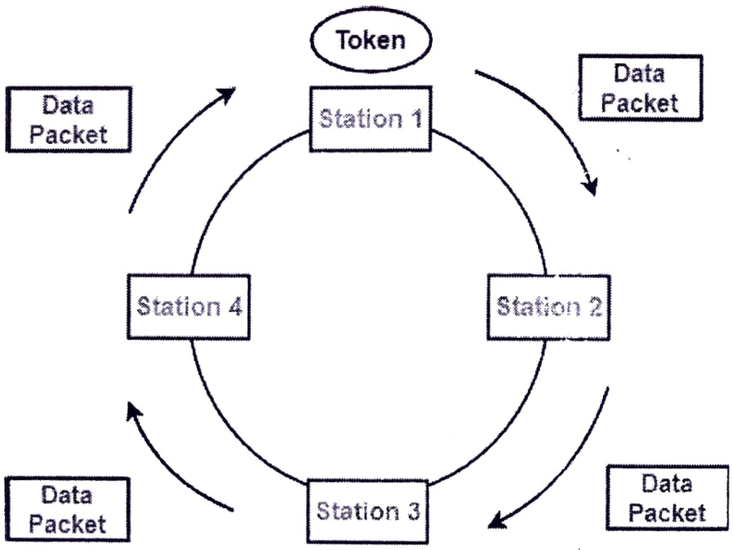
- Station keeps holding the token until the last bit of the data packet transmitted by it takes the complete revolution of the ring and comes back to it.

Working-

After a station acquires the token,

- It transmits its data packet.
- It holds the token until the data packet reaches back to it.
- After data packet reaches to it, it discards its data packet as its journey is completed.
- It releases the token.

The following diagram illustrates these steps for station-1. Same procedure is repeated at every station.



Delayed Token Reinsertion Token Passing

Token Holding Time-

Token Holding Time (THT) = Transmission delay + Ring Latency

We know,

- Ring Latency = T_p + N x bit delay
- Assuming bit delay = 0 (in most cases), we get-

Token Holding Time = $T_t + T_p$

Efficiency-

Substituting THT = $T_t + T_p$ in the efficiency expression, we get-

$$\text{Efficiency } (\eta) = \frac{N \times T_t}{T_p + N \times (T_t + T_p)}$$

(DTR)

OR

$$\text{Efficiency } (\eta) = \frac{1}{\frac{a}{N} + (1 + a)}$$

OR

$$\text{Efficiency } (\eta) = \frac{1}{1 + \left(1 + \frac{1}{N}\right) a}$$

OR

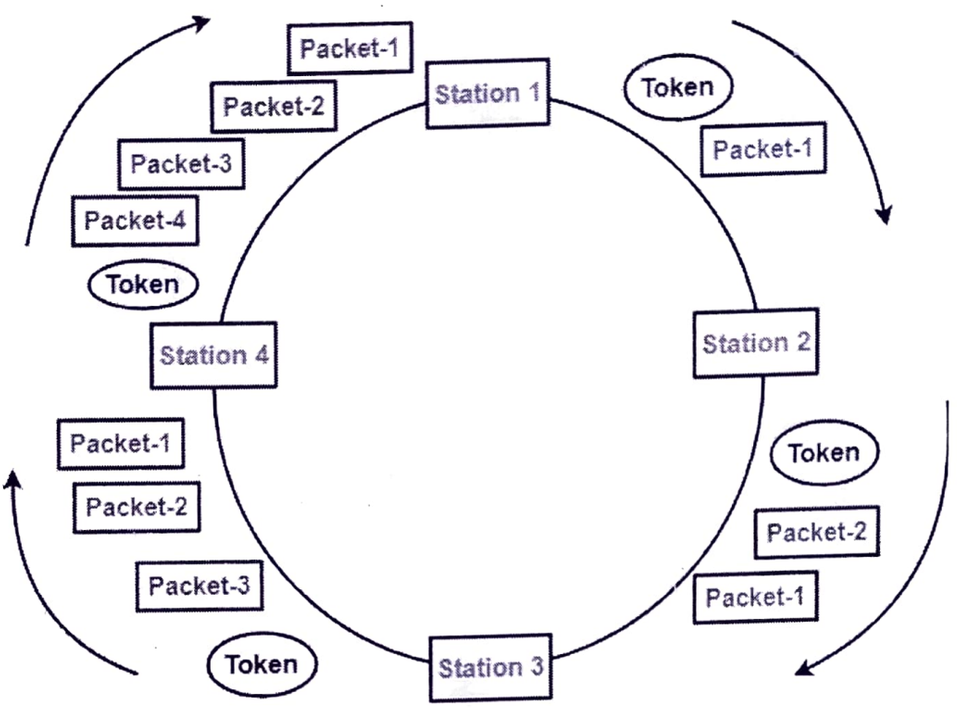
$$\text{Efficiency } (\eta) = \frac{1}{1 + \left(\frac{N+1}{N}\right) a}$$

2. Early Token Reinsertion-

In this strategy,

- Station releases the token immediately after putting its data packet to be transmitted on the ring.

Working:-



Step-01: At Station-1:

Station-1

- Acquires the token
- Transmits packet-1
- Releases the token

Step-02: At Station-2:

Station-2

- Receives packet-1
- Transmits packet-1

} bit delay working

- Acquires the token
- Transmits packet-2
- Releases the token

Step-03: At Station-3:

Station-3

- Receives packet-1
- Transmits packet-1
- Receives packet-2
- Transmits packet-2

} bit delay

- Acquires the token
- Transmits packet-3
- Releases the token

Step-04: At Station-4:

Station-4

- Receives packet-1
- Transmits packet-1
- Receives packet-2
- Transmits packet-2
- Receives packet-3
- Transmits packet-3
- Acquires the token
- Transmits packet-4
- Releases the token

Step-05: At Station-1:

- Receives packet-1
- Discards packet-1 (as its journey is completed)
- Receives packet-2
- Transmits packet-2
- Receives packet-3
- Transmits packet-3
- Receives packet-4
- Transmits packet-4
- Acquires the token
- Transmits packet-1 (new)
- Releases the token

In this manner, the cycle continues.

Token Holding Time-

✓ Token Holding Time (THT) = Transmission delay of data packet = T_t

Efficiency-

Substituting THT = T_t in the efficiency expression, we get-

$$\text{Efficiency } (\eta) = \frac{N \times T_t}{T_p + N \times T_t}$$

(ETR)

OR

$$\text{Efficiency } (\eta) = \frac{1}{1 + \frac{a}{N}}$$

Efficiency

$\frac{Na}{N+a} \times 100\%$

higher than DTR

Differences between DTR and ETR-

Delay Token ~~Retransmission~~ (DTR)	Early Token ~~Retransmission~~ (ETR)
Each station holds the token until its data packet reaches back to it.	Each station releases the token immediately after putting its data packet on the ring.
There exists only one data packet on the ring at any given instance.	There exists more than one data packet on the ring at any given instance.
It is more reliable than ETR.	It is less reliable than DTR.
It has low efficiency as compared to ETR.	It has high efficiency as compared to ETR.

Important Notes-

Note-01:

In token passing,

- It is the responsibility of each transmitting station to remove its own data packet from the ring.

Note-02:

While solving questions,

- If the strategy used is not mentioned, then consider Early Token Retransmission strategy.

Aloha | Pure Aloha | Slotted Aloha

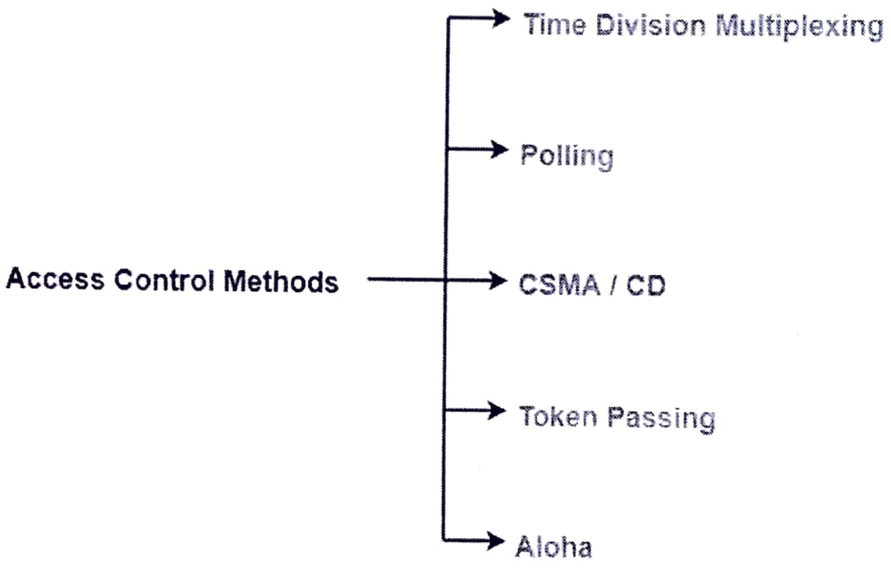
Computer Networks

Access Control in Networking-

Before you go through this article, make sure that you have gone through the previous article on **Access Control**.

We have discussed-

- Access Control is a mechanism that controls the access of stations to the transmission link.
- Broadcast links require the access control mechanism.
- There are various access control methods-

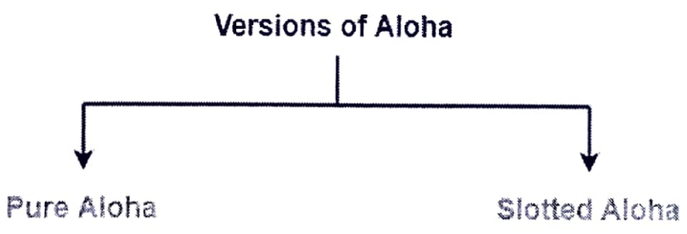


1. **Time Division Multiplexing**
2. **Polling**
3. **CSMA / CD**
4. **Token Passing**
5. **Aloha**

In this article, we will discuss about Aloha and its versions.

Aloha-

There are two different versions of Aloha-



1. Pure Aloha
2. Slotted Aloha

1. Pure Aloha-

- It allows the stations to transmit data at any time whenever they want.
- After transmitting the data packet, station waits for some time.

Then, following 2 cases are possible-

Case-01:

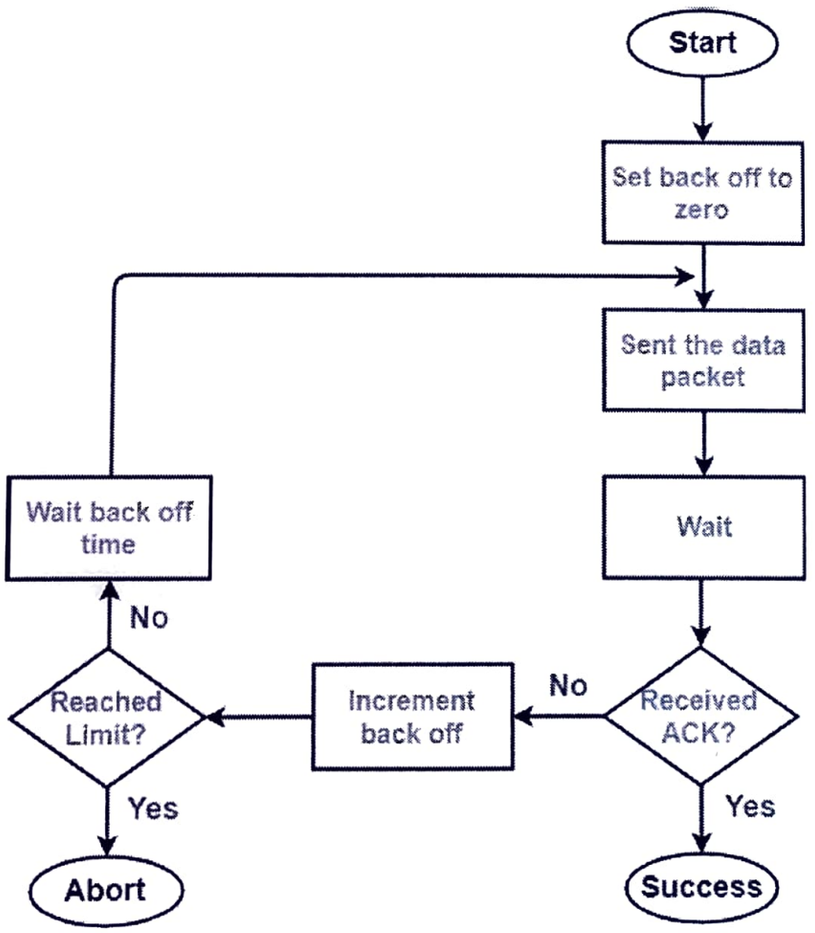
- Transmitting station receives an acknowledgement from the receiving station.
- In this case, transmitting station assumes that the transmission is successful.

Case-02:

- Transmitting station does not receive any acknowledgement within specified time from the receiving station.
- In this case, transmitting station assumes that the transmission is unsuccessful.

Then,

- Transmitting station uses a **Back Off Strategy** and waits for some random amount of time.
- After back off time, it transmits the data packet again.
- It keeps trying until the back off limit is reached after which it aborts the transmission.



Flowchart for Pure Aloha

Efficiency-

Efficiency of Pure Aloha (η) = $G \times e^{-2G}$

where G = Number of stations willing to transmit data

Maximum Efficiency-

For maximum efficiency,

- We put $d\eta / dG = 0$
- Maximum value of η occurs at $G = 1/2$
- Substituting $G = 1/2$ in the above expression, we get-

Maximum efficiency of Pure Aloha

$= 1/2 \times e^{-2 \times 1/2}$

$= 1 / 2e$

$= 0.184$

$= 18.4\%$

Thus,

Maximum Efficiency of Pure Aloha (η) = 18.4%

The maximum efficiency of Pure Aloha is very less due to large number of collisions.

2. Slotted Aloha-

- Slotted Aloha divides the time of shared channel into discrete intervals called as **time slots**.
- Any station can transmit its data in any time slot.
- The only condition is that station must start its transmission from the beginning of the time slot.
- If the beginning of the slot is missed, then station has to wait until the beginning of the next time slot.

- A collision may occur if two or more stations try to transmit data at the beginning of the same time slot.

Efficiency-

Efficiency of Slotted Aloha (η) = $G \times e^{-G}$

where G = Number of stations willing to transmit data at the beginning of the same time slot

Maximum Efficiency-

For maximum efficiency,

- We put $d\eta / dG = 0$
- Maximum value of η occurs at G = 1
- Substituting G = 1 in the above expression, we get-

Maximum efficiency of Slotted Aloha

$= 1 \times e^{-1}$

$= 1 / e$

$= 0.368$

$= 36.8\%$

Thus,

Maximum Efficiency of Slotted Aloha (η) = 36.8%

The maximum efficiency of Slotted Aloha is high due to less number of collisions.

Difference Between Pure Aloha And Slotted Aloha-

Pure Aloha	Slotted Aloha
Any station can transmit the data at any time.	Any station can transmit the data at the beginning of any time slot.
The time is continuous and not globally synchronized.	The time is discrete and globally synchronized.
Vulnerable time in which collision may occur $= 2 \times T_t$	Vulnerable time in which collision may occur $= T_t$
Probability of successful transmission of data packet $= G \times e^{-2G}$	Probability of successful transmission of data packet $= G \times e^{-G}$
Maximum efficiency = 18.4%	Maximum efficiency = 36.8%

(Occurs at G = 1/2)	(Occurs at G = 1)
The main advantage of pure aloha is its simplicity in implementation.	The main advantage of slotted aloha is that it reduces the number of collisions to half and doubles the efficiency of pure aloha.

PRACTICE PROBLEM BASED ON PURE ALOHA AND SLOTTED ALOHA-

Problem-

A group of N stations share 100 Kbps slotted ALOHA channel. Each station output a 500 bits frame on an average of 5000 ms even if previous one has not been sent. What is the required value of N?

Solution-

Throughput Of One Station-

Throughput of each station

= Number of bits sent per second

= 500 bits / 5000 ms

= 500 bits / (5000 x 10^{-3} sec)

= 100 bits/sec

Throughput Of Slotted Aloha-

Throughput of slotted aloha

= Efficiency x Bandwidth

= 0.368 x 100 Kbps

= 36.8 Kbps

Total Number Of Stations-

Throughput of slotted aloha = Total number of stations x Throughput of each station

Substituting the values, we get-

36.8 Kbps = N x 100 bits/sec

$\therefore$ N = 368

Thus, required value of N = 368.

To gain better understanding about Aloha,

refer Forouzan 610

Error Control Methods.

* Error detecting codes are implemented either at data link layer or transport layer of OSI model.

* Basic approach used for error detection is the use of redundancy bits, where additional bits are added to facilitate detection of errors.

Some popular techniques are -

1. Simple parity check
2. Two dimensional parity check
3. Checksum
4. Cyclic redundancy check (CRC).

* Simple Parity Checking (Tanenbaum)

One extra bit called as parity bit is sent along with the original data bits.

→ 2 kinds of parity -

a) Even parity : Total # of 1's in the data unit is made even.

b) Odd parity : Total # of 1's in the data unit is made odd.

Done by adding an extra bit - parity bit.

→ (Original data + Parity bit) is transmitted.

→ At receiver's side,

# of 1's	Parity used	Remark
Even	Even	No error
Even	odd	Error
Odd	Even	Error
Odd	Odd	No error

→ Advantages:

This technique is guaranteed to detect an odd number of bit errors (one, three, five & so on). If odd number of bits flip during transmission, then receiver can detect by counting the # of 1's.

→ Limitation:

Can't detect an even # of bit errors (2,4,6,..). If even number of bits flip during transmission, then receiver can not detect the error.

eg. 1001001 Even parity used

↓

1001001 1 parity bit

↓ Transmission (2 bits flip)

1011011 (# of 1's even)

Receiver assumes no error, although the data is corrupted.

* Two Dimensional Parity checking (Tanenbaum)

Parity check bits are calculated for each row, which is equivalent to a simple parity check bits. Parity check bits are also calculated for all columns; then both are sent along with the data. At the receiving end these are compared with the parity bits calculated on the received data.

Original data.

10011001 11100010 00100100 10000100

	Row parities
10011001	0
11100010	0
00100100	0
10000100	0
11011011	0

{ Even parity.

Column parities

Data Sent

10011001 11100010 00100100 10000100 11011011 0.

* Cyclic Redundancy Check (Tanenbaum)

→ CRC generator: Algebraic polynomial represented as a bit pattern. The power of each term gives the position of the bit and the coefficient gives the value of the bit.

eg. $x^7 + x^6 + x^1 + x^3 + x + 1$

↓

1 1 0 1 1 0 1 1.

→ Properties of CRC generator:

1. Should not be divisible by x. This guarantees that all the burst errors of length equal to the length of polynomial are detected.

2. It should be divisible by $x+1$. This guarantees that all the burst errors affecting an odd number of bits are detected.

→ If the CRC generator is chosen according to the above rules, then

i) CRC can detect all single bit errors.

ii) Can detect all double bit errors provided the divisor contains at least three logic 1's.

iii) Can detect any odd number of errors provided the divisor is a factor of $x+1$.

iv) Can detect all burst error of length less than the degree of the polynomial.

v) Can detect most of the larger burst errors with a high probability.

→ Steps:

1. Calculation of CRC at Sender side:

A string of n 0's is appended to the data unit to be transmitted. n is one less than the number of bits in CRC generator. Binary division is performed of the resultant string with the CRC generator. After division, the remainder so obtained is called as CRC. CRC also consists of n bits.

(We use modulo-2 binary division here — instead of subtraction use XOR operation)

2. Appending CRC to data unit :

The string of n 0's appended to the data unit earlier is replaced by the CRC generator-remainder.

3. Newly formed code word (original data + CRC remainder) is transmitted.

4. Checking at receiver side :

The received code word is divided with the same CRC generator. On division, the remainder so obtained is checked.

2 cases possible -

a) Remainder = 0 : Receiver assumes that no error has occurred & accepts the data.

b) Remainder ≠ 0 : Assumes error & rejects the data asking for retransmission.

→ CRC is based on binary division and a remainder not equal to 0 at the receiver side indicates that the data unit has been damaged in transit & therefore must be rejected.

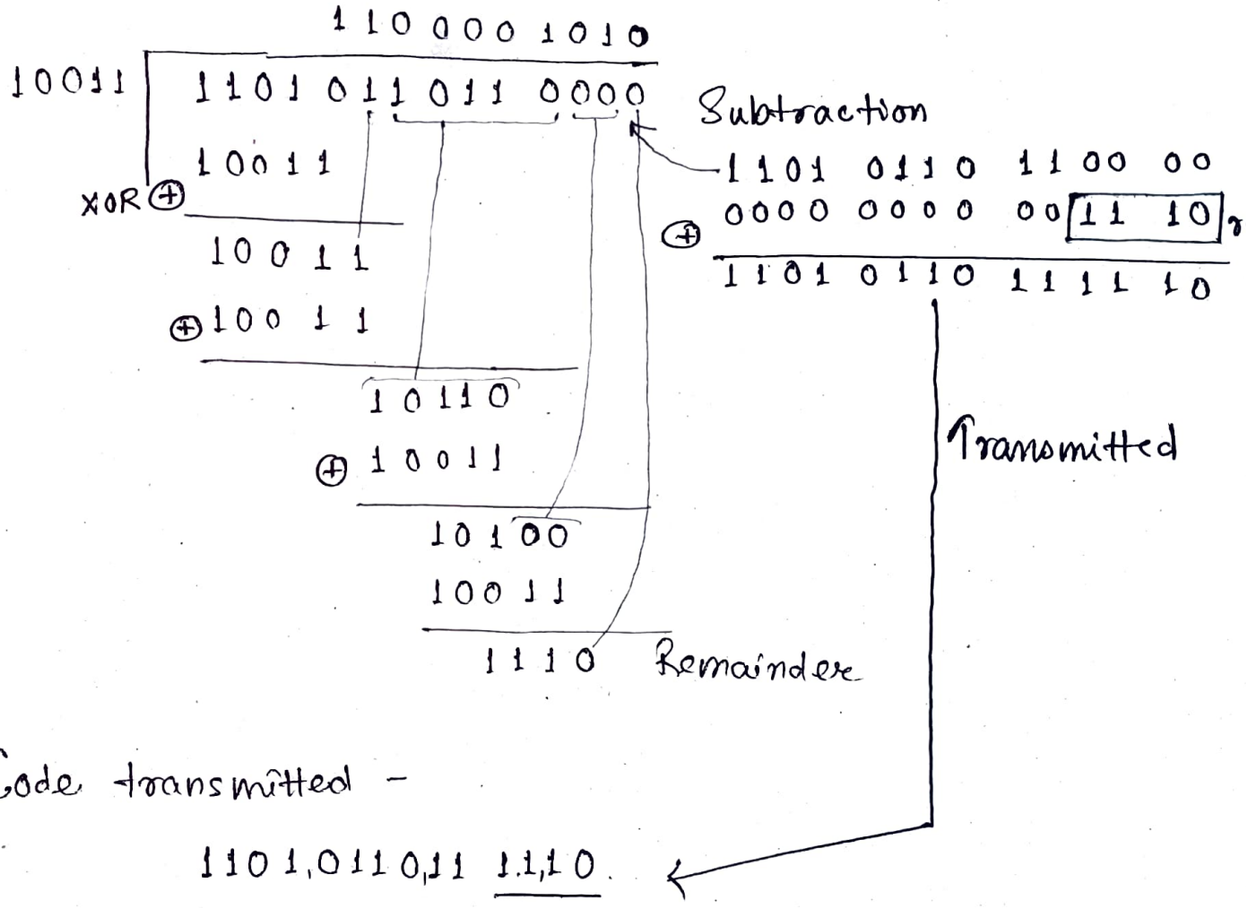
Q. 1101011011 is transmitted using the standard CRC method. Generator polynomial is x^4+x+1. What is actual bit string transmitted to the receiver ?

→ $G(x) = x^4 + x + 1$.

↓ Encoded as

10011 $\quad n = 4.$ | 5-1

Resulting bit stream - 1101011011 0000



Code transmitted -

1101,0110,11 1,1,10.

Q. 10011101 is transmitted using the standard CRC method. $G(x) = x^3 + 1$. Suppose 3rd bit from left is inverted in transit.

→ $G(x) = x^3 + 1 \quad \rightarrow \quad 1001$

Transmission

10011101 000

```
        10001100
    ------------
1001 | 10011101000
       1001
       ----
        .
        .
       ----
       0100   CRC
```

Recovery -

```
        1011 11 | 01100
                |   100
        --------|------
        1011 10 | 10000
```

Transmitted to receiver -

```
10011101100
    ↑
```

Received by receiver

```
        10111101100
        -----------
          10101000
        ------------
1001 | 10111101100
       1001
       ----
        .
        .
       ----
       0100   Remainder ≠ 0
```

Receiver rejects the data & asks for retransmission.

* Checksum. (Tanenbaum)

At sender side,

if m bit checksum is used, the data unit to be transmitted is divided into segments of m bits. The segments are added using 1's complement arithmetic to get the sum. The sum is complemented to get the check-sum.

Data along with the checksum value is trans-mitted to the receiver.

At receiver side,

if m bit checksum is being used, the received data unit is divided into segments of m bits. All the m bit segments are added along with the checksum value. The value so obtained is complemented & the result is checked.

If result = 0, no error.

If result ≠ 0, error. Discards data.

eg. Sender side ~

10011001 11100010 00100100 10000100

K = 4 Segments. | m = 8 bit Segments

4 segments

Sender.

```
  10011001
  11100010
 ----------
  101111011
 ↳        1
 ----------
  01111100
  00100100
 ----------
  10100000
  10000100
 ----------
 100100100
 ↳        1
 ----------
  00100101
```

↓ 1's complement

110 11010

Checksum

Receiver

Received data - correct data

Sum of all segments + checksum = 00000000

⇓

No error assumed.

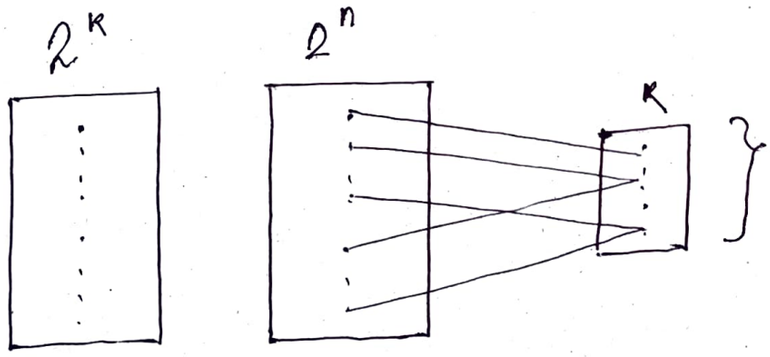
→ While adding the m bit segments, if the result obtained consists of more than m bits, wrap around the extra bits & add to the results so that checksum value consists of m bits only.

→ Checksum is used in the internet by several protocols although not at the data link layer.

* No error detection method is actually reliable, as the error handling bits may be corrupted.

* Meaningful error: Data has been corrupted in such a way that it's unable to detect.

*



duplication of data

$k = n$

Error Detection & Correction.

→ Single bit & burst error.

→ Redundancy bits — central concept of error d^n & c^n.

→ If we need to correct m bit error in an n bit data unit, we need to consider ${}^nC_m$ possibilities.

→ Redundancy achieved through coding schemes (Block coding*, Convolutional coding).

→ Block coding : • Dataword (k bit) + Redundant bits (r bit) = Codeword (n bit)

• 2^k datawords, 2^n codewords.

• $2^n - 2^k$ codewords not used (as block coding is one-to-one)
↓
Invalid or illegal codewords

• Trick in error det^n is these invalid codewords.

→ If the codeword is corrupted during transmission but the received word still matches a valid codeword, the error remains undetected.

→ Parity $k = 2$ $n = 3$ $r = 1$. (Even)

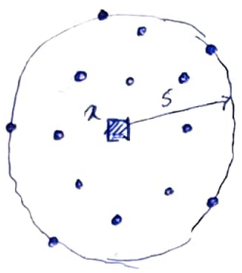
Valid :	k	r	Invalid :
	00	0	001
	01	1	010
	10	1	100
	11	0	111.

Sender 011 → right 2 bits corrupted 000 → Receiver extracts 00 dataword. error goes undetected

→ Hamming distance : # differences b/w corresponding bits. XOR 2 words & count # of 1's.

→ Minimum hamming distance: Smallest h. d. b/w all possible pairs of codewords.

* To guarantee the detection of up to s errors in all cases, the minimum h.d. in a block code must be $d_{min} = s + 1$.



■ valid codeword
• corrupted codeword with 1 to s errors

$d_{min} > s$ or $\geq s+1$

→ Abstract Algebra (Galois Fields)

→ Linear Block Codes.

XOR (addition mod-2) of 2 valid codewords creates another valid codeword.

* Minimum h.d. for linear block codes is the # of 1's in the nonzero valid codeword with the smallest # of 1's.

⊞ Parity check code: $n = k + 1$. $d_{min} = 2$

(XOR bits of dataword) Single bit error detection.

add mod 2

$r_0 = (a_3 + a_2 + a_1 + a_0) \bmod 2$ — sender

$s_0 = (b_3 + b_2 + b_1 + b_0 + q_0) \bmod 2$ — receiver

even parity

syndrome 0 ✓

1 discard.

→ A parity check code can detect an odd # of errors (not even # of errors).

→ Cyclic codes. Also linear block codes.

- In cyclic code, if a codeword is cyclically shifted (rotated), the result is another code-word (wrap-around).

→ CRC subset of cyclic codes.

n K
C(7,4)

Data	Code	Data	Code.
0000	0000 000	1000	1000 101
0001	0001 011	1001	1001 110
0010	0010 110	1010	1010 011
0011	0011 101	1011	1011 000
0100	0100 111	1100	1100 010
0101	0101 100	1101	1101 001
0110	0110 001	1110	1110 100
0111	0111 010	1111	1111 111.

@ Encoder

augmented dataword (add n-k 0 bits to right of it) ← left shift dataword n-k bits

↓

Generator ← Predefined divisor of size n-k+1

↓ mod-2 division

Remainder appended to dataword (codeword)

@ Decoder

Same as generator - checker

↓

remainder Syndrome n-k bits

↓

syndrome = 0 ?

Y → Accept K left most bits.

N → Discard

- mod-2 binary division : add, subtract = XOR
  multiply = AND.

- Generator polynomial:

  Advantage: 1000011 becomes

  $x^6 + x + 1$. (shorter)

⊞ Add-subtract polynomials (XOR)

$x^5 + x^4 + x^2$ 0110 100

$x^6 + x^4 + x^2$ 1010 100

⊕

1100 000 → $x^6 + x^5$.

Combine terms & delete identical pairs.

⊞ Multiply - divide

$x^4 \times x^3 = x^7$

$x^7 \div x^3 = x^4$.

⊞ Multiply 2 polynomials

$(x^5 + x^3 + x^2 + x)(x^2 + x + 1) = x^7 + x^6 + x^5 + x^5 + x^4 + x^3 + x^4 + x^3 + x^2 + x^3 + x^2 + x$

$= x^7 + x^6 + x^3 + x$ (remove 2 identical terms)

⊞ Dividing a polynomial

$x \;)\; x^2 + 1 \;(\; x$

$10 \;)\; 101 \;(\; 1$

10

001

mod-2 division

⊞ Shifting

Left shift m bits ⇒ multiply by x^m (each term)

Right shift m bits ⇒ divide by x^m (each term)

eg.

SL 3 bits 10011 x^4+x+1

↓ $\times x^3$

10011000 $x^7+x^4+x^3$

SR 3 bits 10011 x^4+x+1

↓ $\div x^3$

10 x

⊞ CRC division using polynomials —

dataword x^3+1. $n = 4+3 = 7$

generator x^3+x+1. $K = 4$

x^3+1

↓ augment LS 3 bit X by x^3

x^6+x^3

$$x^3+x+1 \;\big)\; x^6+x^3 \;\big(\; x^3+x$$

$$x^6+x^3+x^4$$

$$x^4$$

$$x^2+x+x^4$$

$$(x^2+x)$$

Codeword — | x^6+x^3 | x^2+x |

- **Cyclic code analysis.**

dataword : $d(x)$ generator : $g(x)$ error : $e(x)$

codeword : $c(x)$ syndrome : $s(x)$

1. $s(x) \neq 0$ one or more bits corrupted.
2. $s(x) = 0$ no bit corrupted or decoder fails to detect.

We need to find the criteria for the generator $g(x)$ so that it can detect specific type of error we want to be detected..

- Received codeword = $c(x) + e(x)$

Receiver divides it with $g(x)$ to find syndrome $\rightarrow \frac{c(x)}{g(x)} + \frac{e(x)}{g(x)}$

$\frac{c(x)}{g(x)}$ has a remainder zero. So, syndrome is the remainder in $\frac{e(x)}{g(x)}$. If it does not have a remainder (syndrome = 0) either $e(x) = 0$ (great!) or $e(x)$ is divisible by $g(x)$ (caution!). Errors divisible by $g(x)$ goes undetected.

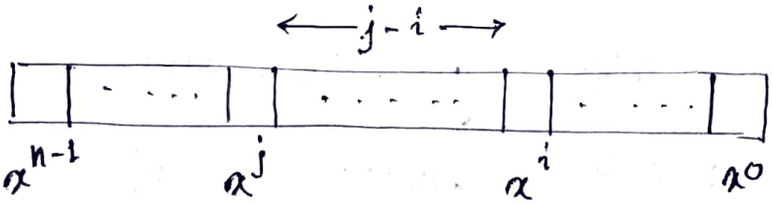
→ Single bit error. $e(x) = x^i$ (i: position of erroneous bit)

To detect this error x^i must not be divisible by $g(x)$. => If $g(x)$ has at least 2 terms & x^0 coefficient is 1, all single bit errors can be caught.

→ 2 isolated single-bit errors

$e(x) = x^j + x^i$

$= x^i(x^{j-i} + 1)$



If $g(x)$ has more than one term & one term is x^0, it can't divide x^i. So, if $g(x)$ is to divide $e(x)$, it must divide $x^{j-i} + 1$. $g(x)$ must not divide $x^t + 1$ ($0 < t \leq n-1$) to detect the error.

So, if $g(x)$ can't divide x^t+1 (t should be b/w 2 & $n-1$), then all isolated double errors can be detected.

eg. $g(x) = x+1$.

Any 2 consecutive error bits can't be detected. $e(x) = x^2+x \Rightarrow e(x) \mid g(x)$

$= x(x+1)$ no remainder

eg. $x^4+1 \Rightarrow g(x)$.

Can't detect 2 errors that are 4 positions apart. $e(x) = x^7+x^3 = x^3(x^4+1)$.

$e(x) \mid g(x) \Rightarrow$ error undetected.

eg. x^7+x^6+1

↳ good choice to detect double errors.

eg. $x^{15}+x^{14}+1$ cannot divide any error of type x^t+1 if $t < 32768$.

→ Odd number of errors.

Generator containing a factor of $x+1$ can detect all odd numbered errors.

eg. $x^4+x^3+x+1 = (x+1) \times (x^3+x^2+1)$.

↑

Modulo-2 ~~as~~ multiplication

→ Burst errors.

$$e(x) = (x^j + \dots + x^i)$$

$$= x^i (x^{j-i} + \dots + 1)$$

min condⁿ for a generator

x^i ← single bit error.

$g(x) = x^r + \dots + 1$.

$\frac{x^{j-i} + \dots + 1}{x^r + \dots + 1}$ should not have a remainder.

Cases –

i) $j - i < r \Rightarrow L - 1 < r \Rightarrow L < r+1$

L → length of burst error

← • All burst errors of length $\leq r$ will be detected.

case ii) $j - i = r \Rightarrow L = r+1$ syndrome 0

← • Prob. of undetected burst error of length $r+1$ is $\left(\frac{1}{2}\right)^{r-1}$

– eg. $g(x) = x^{14} + x^3 + 1$. $r = 14$. (degree of $g(x)$)

Burst error of $L = 15$ can slip by undetected with prob. $\left(\frac{1}{2}\right)^{13}$ almost 1 in 10,000.

rare iii) $j - i > r \Rightarrow L > r+1$. syndrome 0

← • Prob. of undetected burst err. of $L > r+1$ is $\left(\frac{1}{2}\right)^r$.

eg. $x^{14} + x^3 + 1$ $r = 14$.

Burst error of $L > 15$ can slip by undetected with prob. $\left(\frac{1}{2}\right)^{14}$ or almost 1 in 16,000 cases.

Good $g(x)$:

1. At least 2 terms — (single bit error)
2. Coeff. of x^0 is 1. ✓
3. Should not divide x^t+1 → (isolated double) $(2 \le t \le n-1)$
4. Should have factor $x+1$. (odd)

→ Standard polynomials.

CRC-8 : $x^8 + x^2 + x + 1$. (ATM Header)

100000111

CRC-16 : $x^{16} + x^{12} + x^5 + 1$ (HDLC)

CRC-32 : $x^{\wedge}$ (32, 26, 23, 22, 16, 12, 11, 10, 8, 7, 5, 4, 2, 1, 0)

(LAN)

- Error-correcting codes (refer Tanenbaum)

→ Codeword: n bit unit containing m bit data & r bit check bits. $n = m + r$

Code rate: fraction of codeword that carries information that is not redundant, $\frac{m}{n}$.

→ Hamming distance: # bit positions in which 2 codewords differ (XOR 2 codewords & count #1s)

Significance of hamming distance is that if 2 codewords are a hamming distance d apart, it will require min d single-bit errors to convert one into other.

→ In most data transmission all 2^m possible data messages are legal, but due to the way the check bits are computed, not all of the 2^n possible codewords are used. When there are r check bits, only the small fraction of $2^m/2^n$ or $1/2^r$ of possible messages will be legal codewords.

✓→ To reliably detect d errors, we need a distance d+1 code as with such a code there's no way that d single bit errors can change a valid codeword into another valid codeword. To correct d errors we need a distance 2d+1 code because that way the legal codewords are so far apart that even with d changes the original codeword is still closer to the concerned original codeword than any other codeword.

eg. consider 4 valid codewords:

00000 00000, 00000 11111, 11111 00000, 11111 11111. Code has ham. dist. 5.

So, if error-detection is concerned — $d+1 = 5$
$d = 4$
it can detect quadruple errors.

Similarly, if error-correction is concerned - $2d+1 = 5$
$d = 2$
it can correct double bit errors.