Two network layer functions:

1). Forwarding: Move packets from router's input to appropriate router output.

2). Routing: Net. layer must determine route or path taken by packets as they flow from sender to receiver. The algos used are routing algorithms.

* Each router has forwarding table. A router forwards packet by examining value of field in arriving packet header, and then using this header value to index into router forwarding table

Services:

1). Guaranteed delievery:

2). Guaranteed delievery with bounded delay.

Services provided to flow of packets b/w source and dest.

1). inorder packet delievery

2). Guaranteed minimal bandwidth.

3). Restrictions on changes in inter-packet spacing.

Internet Net. layer

Transport: TCP, UDP

Network: IP, ICMP, Routing

- Routing Protocols
  - 1). path selection ⟷ Forwarding Table
  - 2). RIP, OSPF, BGP
- IP
- ICMP

Link: Ethernet, Wifi etc

Physical: Copper, Fiber, Wireless

Mon Tue Wed Thu Fri Sat

Date: / /20

* Datagram network provides network layer connectionless services.

(Imp) Question

IP Fragmentation and Reassembly:

MTU: Maximum amount of data that link layer frame can carry is called maximum transmission unit (MTU)

Suppose you receive an IP datagram from one link. Check forwarding table and outgoing links has MTU that is smaller than length of IP datagram. The solution is to fragment data in IP datagram into two or more smaller IP datagrams, then send these smaller datagrams over outgoing link. Each of these smaller datagrams is referred to as fragment. If some datagrams are fragments, it must further determine when it has received last fragment and how fragments it has received should be pieced back together to form original datagram. For this purpose, identification, flag, and fragmentation offset fields in IP datagram.

Mon Tue Wed Thu Fri Sat Date: / /20

Ex#1

MTU 1500bytes → 20bytes for IP payload + 1480 bytes size

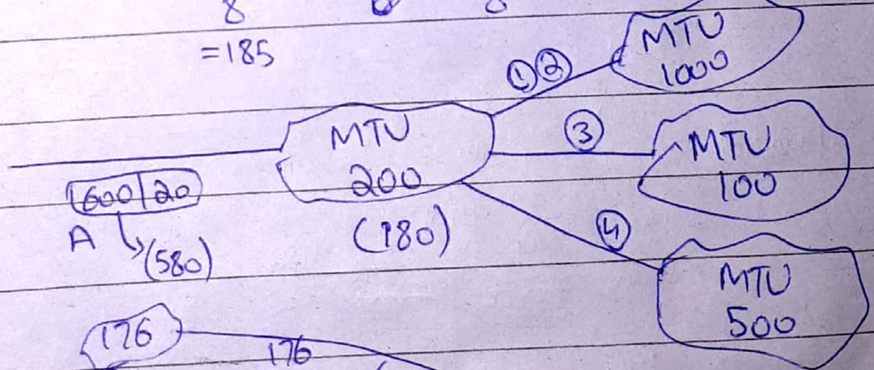
4000 bytes datagram ?
↳ (20 for IP payload)

	Bytes	ID	Offset
1st fragment	1480	777	
2nd "	1480	777	
3rd "	3980 − 1480 − 1480 = 1020	777	

More fragments MF: 1 1 0
↑ is kagy aur fragments nhi hai (3rd frag.)

Offset: 0 $\frac{1480}{8} = 185$ ~~$\frac{1480}{8}$~~ $\frac{1480+1480}{8}$ = 370

Ex#2



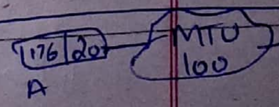
176 ~~180~~ 176 ~~180~~ 176 ~~180~~ 60

→ set To nearest, which is divisible by 8.

MF: 1 1 1 0

Offset 0 $\frac{176}{8} = 22$ $\frac{176+176}{8} = 44$ $\frac{176+176+176}{8} = 66$

Frag#: ① ② ③ ④
↑ refragment because 176>100.



80 80 16

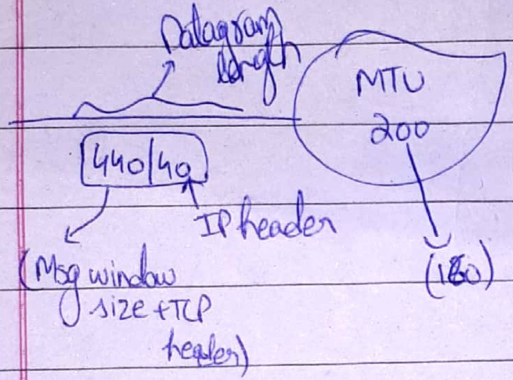
MF: 1 1 0

Offset: 44 54 64

Scanned by CamScanner

Ex#3:

Message size is 400bytes
TCP Header is 40bytes
IP header is 40 bytes
MTU is 200 bytes.

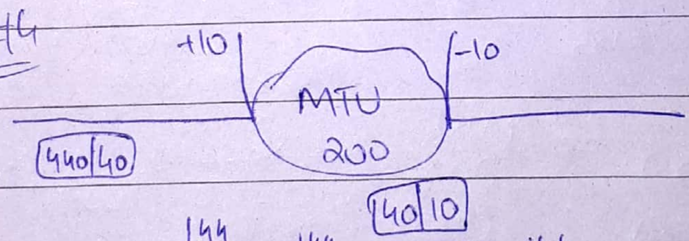


160 160 120

MF: 1 1 0

Offset: 0 20 40

Ex#4



144 144 ~~152~~ 144
~~150~~ ~~150~~ ~~140~~ 8

→ 144 144 144 8

MF: 1 1 1 0

IP addressing:
32 bit identifier for host, router, interface
interface: conn. b/w host/router and physical link.
routers have multiple interfaces.
host typically has one active interface

Mon Tue Wed Thu Fri Sat

Pg#2

15980

Date: __/__/20

880 880 880 880 880 880 880 880 880 880 880
880 880 880 880 880 880 880 880 880 880 160

*. ~~Each~~ One IP address associated with each interface.

Subnet: ~~device interfaces w~~
~~de~~ network interconnecting three host interfaces and ~~called~~ one router interface forms subnet.

Q ~~How ISP get block of addresses?~~
A: ICANN provides addresses from larger block of addresses, which allocates addresses, manages DNS, assign domain names, resolves disputes.

Q How does host get IP address?
A: 1). By Manual Config.
2). By DHCP (Dynamic Host Config Protocol)
It allows host to obtain IP address automatically, when it joins network.
*. Can reuse of addresses.
*. Support for mobile users who want to join network.

(Imp)
NAT: (Network Address Translation)
To convert public address into private and private into public.

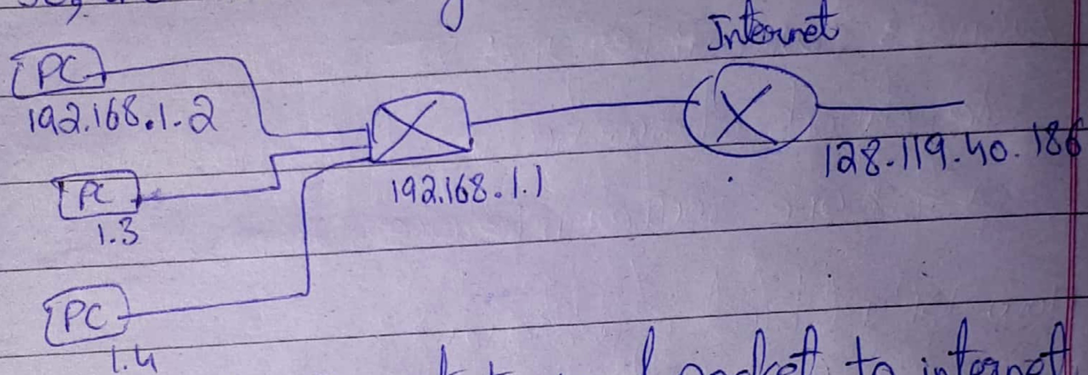
Scanned by CamScanner

Address space: Total no. of addresses used by protocol. $n=32$, $2^n \approx 4.2$ billion

IPv4 → 32bit
↳ 4.2 billion addresses.

Its impossible to assign unique ip address to each device and we have pop. about 7.2 billion.

So, the fix is using NAT.



If 192.168.1.3 wants to send packet to internet then router must contain NAT table.

NAT Table

WAN	LAN
192.168.1.1	192.168.1.3

* NAT router behaves to outside world as single device with single IP address.
* Hiding details of home network from outside world.

Advantages of NAT:

1). NAT conserves legally registered IP addresses.

2). It provides privacy as device IP address, sending and receiving traffic, will be hidden.

3). Eliminates address renumbering when network evolves.

Disad: NAT 1). Translation results in switching path delays.
2). Certain application will not function while NAT is enabled.
3). Ports numbers used for addressing processes and not for addressing hosts.
4). violates end to end argument.
5). routers should only process up to layer 3.

ICMP (Internet Control Message Protocol)
Used by hosts and routers to communicate network level info. to each other.
Used for error reporting.
echo request/reply (used by ping).
ICMP message: type, code plus first 8 bytes of IP datagram causing error.

IPv6: 32 bit address space for IPv4.
Datagram format:
fixed length 40 byte header, no fragmentation allowed

32 bits

Version	Traffic class	Flow Label
Payload len.	Next hdr	Hop limit
Source add. (128 bits)		
Dest. add. (128 bits)		
Data		

Scanned by CamScanner

Mon Tue Wed Thu Fri Sat   Date: __/__/20__

1). Increases size of IP addresses from 32 to 128 bits.

* Header size fixed.
* No option field, no checksum field, no fragmentation. Packet is too big, we cant fragment if packet is big sent msg ICMP to receiver that msg packet is too big.

~~Transitioning from IPv4 to IPv6:~~

IPv4: [Classful]

Classes → A, B, C

	In binary notation	In dotted decimal
	First byte	B
A:	0	0–127
B:	10	128–191
C:	110	192–223

Class A: N $\boxed{H \quad H \quad H}$

B: N N $\boxed{H \quad H}$

C: N N N H

Mask: It helps to find netid and host id.

n bits = 1, (32-n) = 0

A: n = 8	Class A	11111111.00000000.- - -
B: n = 16	" B	11111111.11111111.
C: n = 24	" C	11111111.11111111.11111111.0

Scanned by CamScanner

Pg#3

Mon Tue Wed Thu Fri Sat

Date: __/__/20

IPv4 Classless Addressing

In this, variable length blocks are used that belongs to no class.

Restrictions: i). Addresses in block must be contiguous (one after other)

ii). No. of addresses in block must be 2^n.

iii). The first address must be evenly divisible by no. of addresses.

e.g., Block ← 205.16.37.32 (first address)

next add. is 205.16.37.33

next add. is 205.16.37.34

⋮

Last → 205.16.37.47 (Total 16 addresses)

And $\frac{32}{16} = 2$ (3rd point also cleared).

Mask in classless: can take any value from 0 to 32.

*. In Slash/CIDR notation

$x.y.z.t/\underline{n}$ → mask

1) First address in block can be found by setting rightmost (32-n) bits to 0's.

2). Last address can be found by setting rightmost (32-n) bits to 1's.

3). No. of addresses = 2^{32-n}

e.g., one of addresses is 205.16.37.39/$\underset{n}{\underline{28}}$. Find 1st, last and total

11001101 00010000 00100101 00100111

For 1^{st} add: ___ . ___ . ___ .00100000
(205.16.37.32)

For last add:

___ . ___ . ___ .00101111
(205.16.37.47)

Number of addresses $= 2^{32-n} = 2^{32-28} = 2^4 = 16$

2^{nd} way:

i). First Add: (Address) AND (MASK)
ii). Last Add: (Address) OR (complement of mask)

e.g., 205.16.37.39/28

Mask: 11111111.11111111.11111111.11110000 (Till 28 all 1's)
Add: 11001101.00010000.00100101.00100111

1^{st} add: 11001101.00010000.00100101.00100000 (205.16.37.32)

Comp. of Mask: 00000000.00000000.00000000.00001111

Last Add: 11001101.00010000.00100101.00101111

5 4 3 2 1 0
32+8+4+2+1
(47)

Transition from IPv4 to IPv6:

3 strategies

Dual Stack

Source

Installed on system → IPv4 IPv6

Tunneling

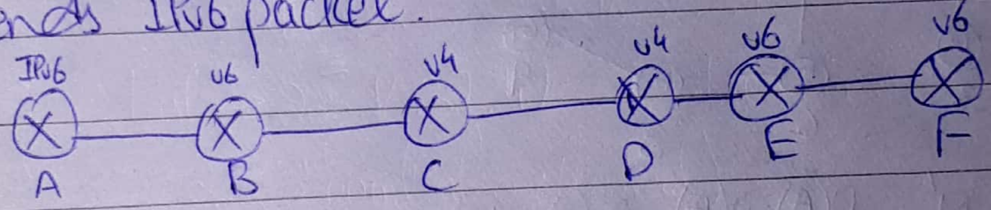
IPv6 Client — IPv4 region — IPv6 Server

1). Dual Stack:

In this technique nodes with IP stack will contain both IPv4 and IPv6.

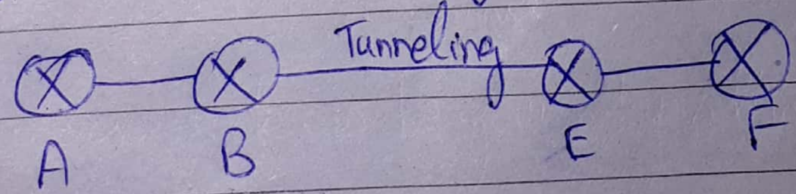
while communicating with IPv6 node, IPv6 part is used and when communicating with IPv4 node it uses IPv4 part.

Source host queries DNS. If DNS server responds with IPv4 address then source responds with IPv4 packet. If DNS returns IPv6 then source sends IPv6 packet.



2). Tunneling:

IPv6 packet is encapsulated in an IPv4 packet when it enters IPv4 region.



Routing Algorithms:

~~1). Link state (LS) / Dijkstra~~

Global: All routers have complete topology, link cost info.
(Link State (LS) algo's)

decentralized: iterative process of computation, exchange of info. with neighbours.

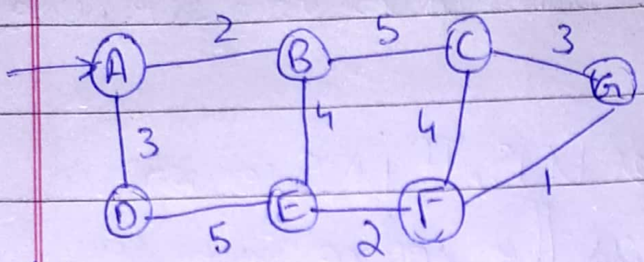
Mon Tue Wed Thu Fri Sat

Date: / /20

No node has complete info. about costs of all network links.

(Distance Vector (DV) algo.)

1). LS Algo. (Dijkstra)

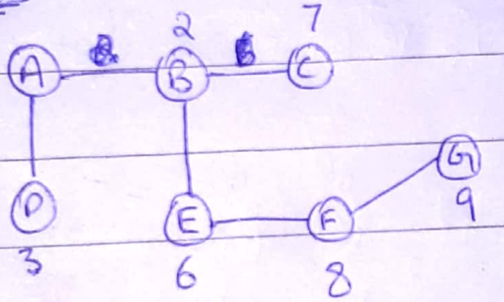


$D(v) = \min(D(v), D(w) + c(w,v))$

where $D(v)$ is dist.

$p(v)$ is previous node

Short Distances



D(G), P(G)	Step	N'	D(A), p(A)	D(B), p(B)	D(C), p(C)	D(D), p(D)	D(E), p(E)
	0	A	∞	2, A	∞	3, A	∞
∞	1	AB	∞		7, B	3, A	6, B
∞	2	ABD	∞				

Scanned by CamScanner

Subnetting:

VLSM + Past Paper VLSM solved

Pg#4

DV Routing Algo: (iterative, distributed, asynchronous)

1). Cost is normally Hop counts.

2). Each router needs to update routing table asynchronously. [when it receives info. from receiver]

3). After update result is sent to all neighbour

4). Each router should keep at least 3 pieces of info. for each route: 1). Destination n/w 2). Cost 3). Next hop.

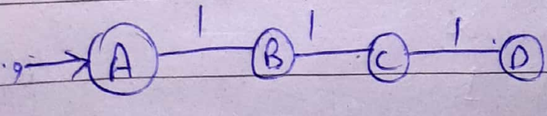
When record arrives, router searches for destination address in routing table

1). If entry is found

↳ a). If record cost plus 1 is smaller, than corresponding cost in table, it means neighbour have found better route.

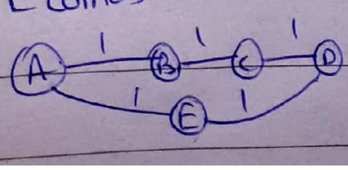
b). If next hop is same, it mean some changes has happened in part of Network.

2). If entry not found, router adds it to table and sorts table according to destination.

e.g.,→ 

$Dest_A$	Cost	via
D	3	B

And E comes

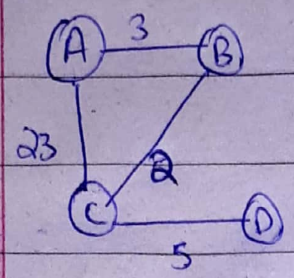


Then

$Dest_A$	Cost	via
~~D~~	~~3~~	~~B~~ ← (discarded)
D	2	E

Mon Tue Wed Thu Fri Sat Date: / /20

e.g.:



Turn #1 Next hop

From A:

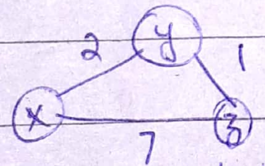
Dest	Cost via A	B	C	D
A				
B		3	25	
C		5	23	
D		10	28	

Turn #2

~~Rec~~ A receives update from C [C→D=5]

A receives update from B {B→C→D=7}

eg#2:



Node X:

	Cost to x	y	z
x	0	2	7
y	∞	∞	∞
z	∞	∞	∞

Node Z

	x	y	z
x	∞	∞	∞
y	∞	∞	∞
z	7	1	0

Node Y:

	x	y	z
x	∞	∞	∞
y	2	0	1
z	∞	∞	∞

Scanned by CamScanner

Turn #2:

Node X

	x	y	z
x	0	2	3
y	2	0	1
z	7	1	0

Node Y

	x	y	z
x	0	2	7
y	2	0	1
z	7	1	0

Node Z

	x	y	z
x	0	2	7
y	2	0	1
z	3	1	0

Turn #3:

Node X:

	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0

Node Y:

	x	y	z
x	0	2	3
y	2	0	1
z	3	0	1

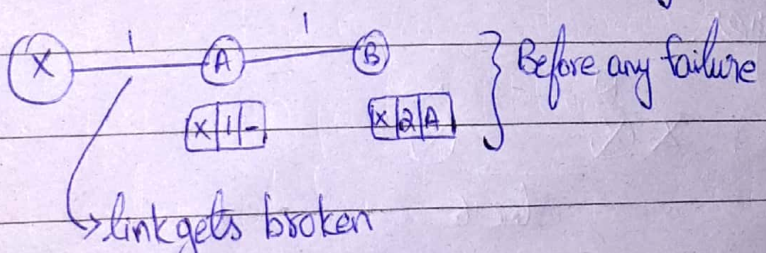
Node Z

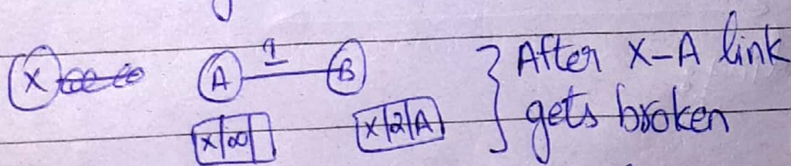
	x	y	z
x	0	2	3
y	2	0	1
z	3	0	1

Count to Infinity Problem:

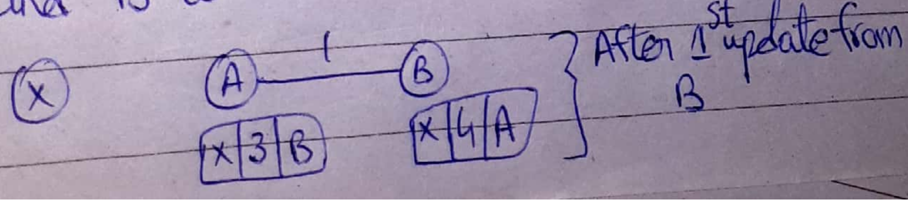
↳ Any decrease in cost, news travels fast

↳ Increase in cost, news travels slowly.

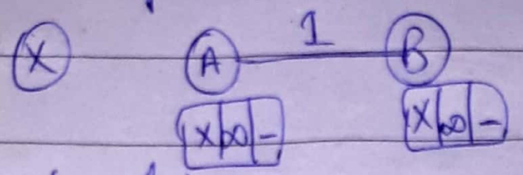




Info. from A to B slow travels and B to A travels fast



After n updates

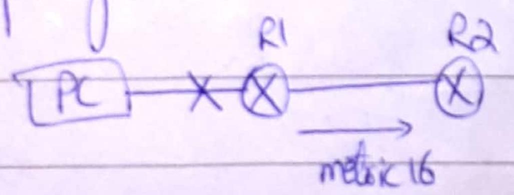


upto infinity.

Solution:

1). Route Poisoning:

When route fails, dv protocols spread bad news about route failure by poisoning route. Routers consider routes advertised with an infinite metric to have failed. Each dv routing protocol uses concept of an actual metric value that represents infinity. Hop limit is 15. And, infinity is 16. Main disad. of poison reverse is that it can significantly increase size of routing announcements in certain fairly common network topologies.



Hierchical Routing:

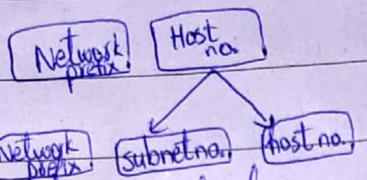
✓ VLSM + Past Paper VLSM Solved

Tue Wed Thu Fri Sat   Date: __/__/20

Subnetting:

Subnetting:

* A practice of dividing network into two or more networks is called subnetting.
* Split host no. portion of an IP address into subnet no. and host no.



* Subnets are treated as seperate network.
* Identified by Subnet mask (32bit)

Class A:

All N = 1, all H = 0

Subnet mask: N | H | H | H

255 0 0 0

Class B: All N=1, H=0

255.255.0.0

N | N | H | H

16bit

Class C:

255.255.255.0

Ranges:

For Class A:

0____.____.____.____ to 01111111.____.____.__

(0 to 127)

127 reserved for local host and loop back is $\boxed{0-126.}$

* Most expensive

For Class B:

~~10 is fixed as LMB. (leftMost Bit)~~

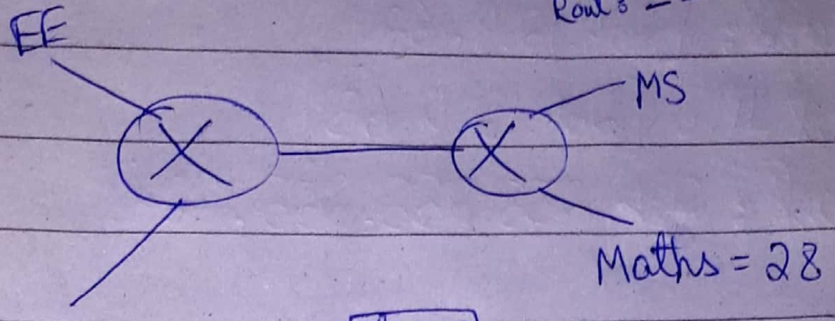
10____.____.____.____

(128 – 191)

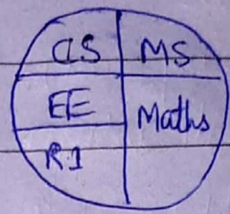
No. of addresses = 2^{30}

No. of networks is $[2^{14}]$

CS: ---
Rout: ---

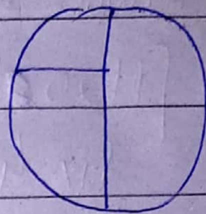


Class C



Maths ---
EE ---
CS ----
MS ----
Rout: ------

NetId: 204.12.4.0/24



255

For Maths:

For 28 hosts
$n = \log_2(28) = 4.8$
we will assign, $n = 5$
Network ID: 204.12.4.0/27
1st host add: 204.12.4.1
[11001100.00001100.00000100.00000000]
[11001100.00001100.00000100.00011111]
31
Last add: [204.12.4.30]
Broadcast ID: 204.12.4.31
Subnet mask: 255.255.255.224

For EE: For 28 hosts
$n = \log_2(28) = 4.8$
$n = 5$
N.Id = 204.12.4.32/27
1st host add. = 204.12.4.33 [00100001]
[11001100.00001100.00000100.$\boxed{001}$11111]

For Router: 1 host Net. IP = 204.12.4.96
[01]100000
n=2 1st host: 204.12.4.97 last host add: 204.12.4.98
Subnet mask is: 255.255.255.252

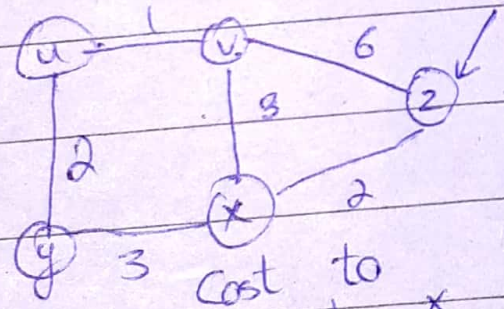
Mon Tue Wed Thu Fri Sat Date: / /20

Last host add: [204.12.4.62]
Broadcast ID: [204.12.4.63]
Subnet mask: ~~255~~. 255.255.255.224.

For CS:
14 hosts
$n = \log_2(14) = 3.8$
We will assign n=4.
Network ID: 204.12.4.64/28
[11001100.00001100.00000100.$\boxed{0100}$0001
Fix 4 leftmost bits in Host Part.
1st host address: 204.12.4.65.
Last host add:
[11001100.00001100.00000100.0100|1111]
79
last add is 204.12.4.78
Broadcast address is 204.12.4.79
Subnet mask is 255.255.255.240

For MS: 7 hosts
$n = \log_2(7) = 2.8$
We will assign, n=3.
Network ID is 204.12.4.80/29
1st host add: [11001100.00001100.00000100.$\boxed{010}$10001]
(204.12.4.81)
Last host add: [11001100.00001100.00000100.$\boxed{010}$11111]
95
last host add is 204.12.4.94.
Broadcast add is 204.12.4.95
Subnet mask is 255.255.255.224

For



	Cost to u	v	x	y	z
v					
x					
z	∞	6	2	∞	0

	Cost to u	v	x	y	z
v	1	0	3	∞	∞
x	∞	3	0	3	2
z	7	6	2	~~∞~~ 5	0

	Cost to u	v	x	y	z
v	1	0	3	3	5
x	4	3	0	3	2
z	6	5	2	5	0

Past Paper Solution

From Node F.

	A	B	C	D	E	F	G
F	∞	∞	∞	14	6	0	17
E	∞	∞	∞	∞	∞	∞	∞
D	∞	∞	∞	∞	∞	∞	∞
G	∞	∞	∞	∞	∞	∞	∞

	A	B	C	D	E	F	G
F	20	∞	∞	14	6	0	17
E	14	∞	∞	18	0	6	23
D	32	∞	∞	0	18	14	12
G	37	∞	∞	12	23	17	0

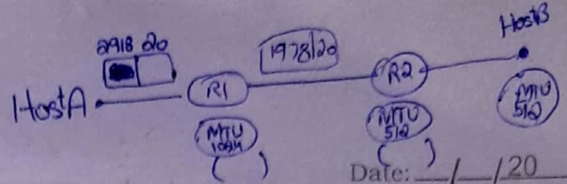
	A	B	C	D	E	F	G
F	20	∞	15	14	6	0	17
E	14	∞	9	18	0	6	23
D	19	∞	16	0	18	14	12
G	31	∞	28	12	23	17	0

	A	B	C	D	E	F	G
F	20	20	15	14	6	0	17
E	14	15	9	18	0	6	23
D	19	21	16	0	18	14	12
G	31	33	28	12	23	17	0

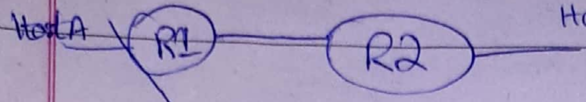
Final

	A	B	C	D	E	F	G
F	20	20	15	14	6	0	17
E	14	15	9	18	0	6	23
D	19	21	16	0	18	14	12
G	31	33	28	12	23	17	0

Past Papers — IP Datagrams



Mon Tue Wed Thu Fri Sat   Date: __/__/20



900 bytes datagram.

20 bytes of TCP header

MTU 900 bytes

A → R1 : Max 1024 bytes framesize (including 14 byte frame header)

R1 → R2 : Max 512 bytes " (including 8 byte frame header)

R2 → B : Max 512 " (including 12 byte frame header)

Link A → R1 :

Length = 900 + 20 + 20 = 940   940/20 MTU 1010

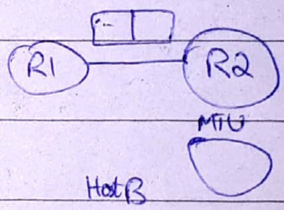
880

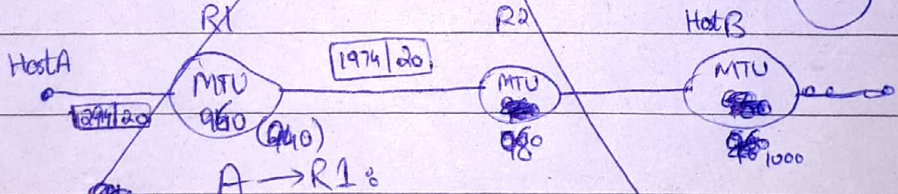
MF = 0

Offset = 0

Link R1 → R2 :

Length =





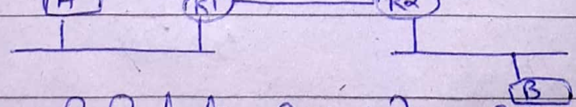
A → R1 :

Length = 940

(2860)

R1 → R2 :

Length = 50



Sol : TCP Packet = 900 + 20 = 920 Bytes

IP Packet = 920 (data) + 20 = 940 Bytes

Link A → R1 : (1010 Bytes IP data packet) supported.

Thus no fragmentation needed.

Link R1 → R2 : (can support only 504 bytes of IP data packet and 504 − 20 (IP header) = 484).

Scanned by CamScanner

So, ~~8x60~~

$8 \times x \le 484$

$x \le 60.5$ Use $x = 60$ bytes

1). IP data 1 = $60 \times 8 = 480$ bytes

Packet size = $480 + 20 = 500$

2). IP data 2 = $920 - 480 = 440$ bytes

Packet size = $440 + 20 = 460$ bytes

i- Length = 500, ID = x, DF = 0, MF = 1, offset = 0

ii- Length = 460, ID = x, DF = 0, MF = 0, offset = $\frac{480}{8}$ = 6

For R2 → B : (500 bytes support)

500 − 20 (IP header) = 480.

1). Length = 480+20 = 500, MF = 1

2). Length = 920 − 480 + 20 = 460, offset = $\frac{480}{8}$ = 60.

MF = 0

40: 198.16.0.0

A = 4000, B = 2000, C = 4000, D = 8000.

A:

s = 20

198.16.0.0 − 198.16.15.255

198.16.$\boxed{0000}$1111.11111111

B:

198.16.16.0 − 198.16.23.255. s = 21

.$\boxed{0001}$0111.11111111

Scanned by CamScanner