ASSIGNMENT - 2

SHIVANI. T
126018047
N-46

COMPUTER NETWORKING

1) Original Block Address : 12.44.184.0 /21

Number of host bits : 32 - 21 = 11

Block size : 2^{11} = 2048 IP addresses.

Address range : 12.44.184.0 - 12.44.~~190~~ 191.255

Medium - size company Block Address : 12.44.184.0 / 22

Number of host bits : 32 - 22 = 10

Block size : 2^{10} = 1024 IP addresses

Address range : 12.44.184.0 - 12.44.187.255

First small company Block Address : 12.44.188.0 / 23

Number of host bits : 32 - 23 = 9

Block size : 2^{9} = 512 IP addresses.

Address range : 12.44.188.0 - 12.44.189.255

Remaining Block Address for second small company } = ?

Remaining Address Range : 12.44.190.0 - 12.44.191.255

512 IP addresses ⇒ 23 network bits, 9 host bits.

Block address : 12.44.190.0 / 23

When a datagram arrives, the router looks at all matching prefixes, even if all three companies' blocks are within a single /21, router can forward datagrams correctly by using longest prefix match.

2) ISP block : 16.12.64.0 /20

Divide among 8 organization, each needing 256 addresses.

a)

Number of host bits = 32 - 20 = 12

Number of addresses = 2^{12} = 4096 IP addresses.

IP address range : 16.12.64.0 - 16.12.79.255

b)

Each require 256 addresses → 2^8 ⇒ /24

Organization	CIDR Block	Range
1	16.12.64.0 /24	16.12.64.0 - 16.12.64.255
2	16.12.65.0/24	16.12.65.0 - 16.12.65.255
3	16.12.66.0/24	16.12.66.0 - 16.12.66.255
4	16.12.67.0/24	16.12.67.0 - 16.12.67.255
5	16.12.68.0 /24	16.12.68.0 - 16.12.68.255
6	16.12.69.0 /24	16.12.69.0 - 16.12.69.255
7	16.12.70.0 /24	16.12.70.0 - 16.12.70.255
8	16.12.71.0 /24	16.12.71.0 - 16.12.71.255

Remaining unallocated Blocks:

16.12.72.0/24 - 16.12.79.255 /24

c) Address Distribution Table:

Use	CIDR Block	Range
ISP total	16.12.64.0/20	16.12.64.0 - 16.12.79.255
Org 1	16.12.64.0 /24	16.12.64.0 - 16.12.64.255
Org 2	16.12.65.0/24	16.12.65.0 - 16.12.65.255
Org 3	16.12.66.0/24	16.12.66.0 - 16.12.66.255

Use	CIDR Block	Range
Org 4	16.12.67.0/24	16.12.67.0 - 16.12.67.255
Org 5	16.12.68.0/24	16.12.68.0 - 16.12.68.255
Org 6	16.12.69.0/24	16.12.69.0 - 16.12.69.255
Org 7	16.12.70.0/24	16.12.70.0 - 16.12.70.255
Org 8	16.12.71.0/24	16.12.71.0 - 16.12.71.255
Unallocated	16.12.72.0/24	16.12.72.0 - 16.12.79.255

Forwarding Table:

Destination Network	Next Hop
16.12.64.0/24	Organization 1
16.12.65.0/24	Organization 2
16.12.66.0/24	Organization 3
16.12.67.0/24	Organization 4
16.12.68.0/24	Organization 5
16.12.69.0/24	Organization 6
16.12.70.0/24	Organization 7
16.12.71.0/24	Organization 8.

3)a) Block address : 130.56.0.0/16

Number of host addresses : $2^{32-16} = 65,536$ addresses.

To divide into 1024 subnets,

$65536 \div 1024 = \underline{\underline{64}}$ addresses per subnet.

b) To create 1024 subnets $\Rightarrow 2^{10} = 1024$

$\therefore$ we borrow 10 bits

New subnet prefix $= 16 + 10$

$= \underline{\underline{/26}}$

Scanned with OKEN Scanner

c) **First Subnet:**

Start IP : 130.56.0.0

End IP : 130.56.0.63

d) **Last Subnet:**

Start IP : 130.56.0.0 + (1023 * 64 addresses)

= 130.56.0.0 + 65472 addresses

(65472 ÷ 256 = 255)

= 130.56.255.0

Last IP : 130.56.255.63

4) a) M bit = 1 , offset field = 0

⇒ **First Fragment**

b) M bit = 1 , offset field ≠ 0

⇒ **Middle Fragment** (Since MF = 1 , not the last or only fragment)

5) **Bellman - Ford Equation:**

$D_{ay} = 5$ $\quad C_{xa} = 2$

$D_{by} = 6$ $\quad C_{xb} = 1$

$D_{cy} = 4$ $\quad C_{xc} = 3$

$D_{dy} = 3$ $\quad C_{xd} = 1$

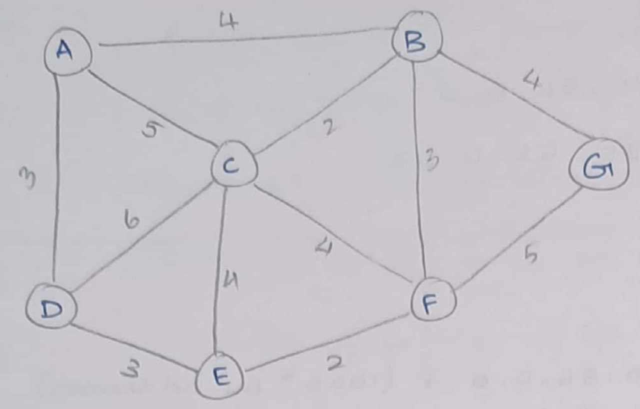
$$D_{xy} = \min(C_{xa} + D_{ay}, C_{xb} + D_{by}, C_{xc} + D_{cy}, C_{xd} + D_{dy})$$

$$= \min(7, 7, 7, 4)$$

$D_{xy} = 4$

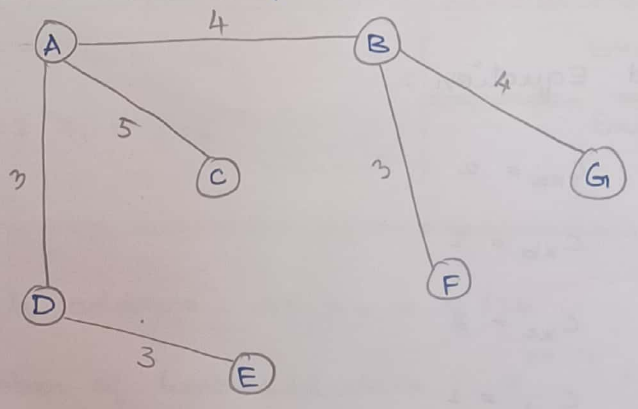
∴ The shortest distance from x to y is 4 through the node 'd'.

6)



Iteration	Tree	B	C	D	E	F	G
Initial	{A}	4	~~∞~~ 5	3	∞	∞	∞
1	{A, D}	4	5	–	6	∞	∞
2	{A, D, B}	–	5	–	6	7	8
3	{A, D, B, C}	–	–	–	6	7	8
4	{A, D, B, C, E}	–	–	–	–	7	8
5	{A, D, B, C, E, F}	–	–	–	–	–	8
6	{A, D, B, C, E, F, G}	–	–	–	–	–	–

<u>Minimum</u> <u>Spanning</u> <u>Tree</u> :



7)a) First 64 zeroes : 0 repeated 64 times

32 x (01) → 01 repeated 32 times = 64 bits.

64 bits = 4 blocks of 16 bits = 0000 : 0000 : 0000 : 0000

0101 = 5 (hex) ⇒ every 4 bits becomes a 5 .

32 repetitions of 01 = 16 groups of 4 bits → 16 hex digits

Split into 2 blocks of 16 bits : 5555 : 5555

0000 : 0000 : 0000 : 0000 : 5555 : 5555 . 5555 . 5555

b) First 64 zeroes : 0 repeated 64 times

0000 : 0000 : 0000 : 0000

1010 = A (hex) ⇒ every 4 bits becomes a 'A'

0000 : 0000 : 0000 : 0000 : AAAA : AAAA : AAAA : AAAA

c) 64 two-bit (01)s

0101 = 5 (hex) → every 4 bits becomes a 5

5555 : 5555 : 5555 : 5555 : 5555 : 5555 : 5555 : 5555

d) 32 four-bit (0111)s

0111 = 7 (hex) ⇒ every 4 bits becomes a 7

7777 : 7777 : 7777 : 7777 : 7777 : 7777 : 7777 :
7777.

8) 5 bit sequence number can represent from $\}$ $= 0$ to $2^5 - 1$

$= 0$ to 31

Sequence numbers go : 0, 1, 2, . . . , 31 and then wrap around to 0 again.

Sequence number of the 100^{th} packet $\}$ = 100 mod 32

$= \underline{\underline{4}}$

---

9) Bandwidth Delay Product $\}$ = Bandwidth (bits/sec) × RTT (seconds)

Number of Packets $\} = \dfrac{BDP}{\text{Packet size}}$

a)

BDP = 1 Mbps × 20 ms

$= 1 \times 10^6 \times 20 \times 10^{-3}$

$= \underline{\underline{20{,}000 \text{ bits}}}$

Number of packets $\} = \dfrac{20{,}000}{1000}$

$= \underline{\underline{20 \text{ packets}}}$.

---

b) BDP = 10 Mbps × 20ms

$= 10 \times 10^6 \times 20 \times 10^{-3}$

$= \underline{\underline{200{,}000 \text{ bits}}}$

Number of packets $\} = \dfrac{2{,}00{,}000}{2000} = \underline{\underline{100 \text{ packets}}}$

c) $BDP = 1 \text{ Gbps} \times 4 \text{ ms}$

$= 1 \times 10^{9} \times 4 \times 10^{-3}$

$= \underline{\underline{4\,000{,}000 \text{ bits}}}$

$$\left.\begin{array}{l}\text{Number of} \\ \text{packets}\end{array}\right\} = \frac{40{,}00{,}000}{10{,}000}$$

$= \underline{\underline{400 \text{ packets}}}$

---

10)

Bandwidth = 100 Mbps

Distance = 10,000 Km $= 10^{7}$ m

Packet size = 100,000 bits

Propagation speed $= 2 \times 10^{8}$ m/s

$$\left.\begin{array}{l}\text{Transmission} \\ \text{delay } (T_t)\end{array}\right\} = \frac{\text{Packet Size}}{\text{Bandwidth}}$$

$$= \frac{100{,}000}{100 \times 10^{6}}$$

$= 0.001$ sec (1 ms)

$$\left.\begin{array}{l}\text{Propagation} \\ \text{delay } (T_p)\end{array}\right\} = \frac{\text{Distance}}{\text{Propagation Speed}}$$

$$= \frac{10^{7}}{2 \times 10^{8}}$$

$= 0.05$ sec (50 ms)

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

$$= \frac{0.05}{0.001} = 50$$

$$\text{Window size} = 1 + 2a$$

$$= 1 + 2(50)$$

$$= \underline{\underline{101 \text{ packets}}}$$

$$\left.\begin{matrix}\text{Number of} \\ \text{bits } (m)\end{matrix}\right\} \Rightarrow W \leq 2^m - 1$$

$$\Rightarrow 101 \leq 2^m - 1$$

$$2^m \geq 102$$

$$m = \lceil \log_2(102) \rceil$$

$$m = \underline{\underline{7 \text{ bits}}}$$

$$\left.\begin{matrix}\text{Round Trip} \\ \text{Time (RTT)}\end{matrix}\right\} = 2 \times T_p$$

$$= 2 \times 50$$

$$= 100 \text{ ms}$$

Timeout should be a bit more than RTT,

$$\text{Time-out} \approx \underline{\underline{120 \text{ ms}}}.$$

---

11)

Bandwidth = 1 Gbps

Distance = 5,000 Km

Packet Size = 50,000 bits

Propagation Speed = 2×10^8 m/s

$$T_p = \frac{5 \times 10^6}{2 \times 10^8}$$

$$= 0.025 \text{ sec}$$

$$= 25 \text{ ms}$$

$$T_t = \frac{50,000}{1 \times 10^9}$$

$$= 0.00005 \text{ sec}$$

$$= 0.05 \text{ ms}$$

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

$$= \frac{25}{0.05}$$

$$= 500$$

$$\text{Window size} = 1 + 2a$$

$$= 1 + 2(500)$$

$$= \underline{\underline{1001 \text{ packets}}}$$

$$W \le 2^m - 1$$

$$2^m \ge 1001 + 1$$

$$2^m \ge 1002$$

$$m = \lceil \log_2(1002) \rceil$$

$$m = \underline{\underline{11 \text{ bits}}}$$

$$\left.\begin{array}{l}\text{Round Trip}\\ \text{Time (RTT)}\end{array}\right\} = 2 \times T_p$$

$$= 2 \times 25$$

$$= 50 \text{ ms}$$

Time-out should be a bit more than RTT

Time-out $\simeq \underline{\underline{70 \text{ ms}}}$.

---

12) The IP header has a protocol field. This indicates whether the packet uses TCP (6) or UDP (17). This is used to identify and forward to the correct transport layer protocol.

13) Client IP = 122.45.12.7

Client empheral port = 51000

Server IP = 200.112.45.90

Server well-known port = 161 (SNMP)

Client Socket Address : 122.45.12.7 : 51000

Server Socket Address : 200.112.45.90 : 161

---

14) No.

If the private internet uses a protocol suite that is completely different from the TCP/IP protocol suite, it cannot directly use UDP or TCP for message communication. This is because, both UDP and TCP are integral parts of the TCP/IP protocol suite, and they rely on specific protocols to handle the addressing and routing of data across networks.

---

15) a)

If the first segment doesn't carry any data, the next segment remains the sequence number same as the first segment.

Sequence number of the next segment } = 101

---

b) If the first segment consumes 10 sequence numbers, the next segment will be the sequence number of the current segment plus 10

Sequence number of the next segment } = 111

16) SYN, SYN + ACK, FIN segments each consume a sequence number because they are a part of the connection setup, which requires sequence tracking.

ACK segments with no data do not consume a sequence number because they don't carry any data, and thus no byte count is incremented.

17) If a server receives a SYN segment on a well-known port but no process is listening on that port, the server is supposed to respond with a TCP RST (Reset segment).

This informs the client that the server is not accepting connections on the specified port, effectively rejecting the ~~connecting~~ on attempt.

18) The TCP client accepts the 400 byte segment, starting from byte 3001 through 3400.

Then the client sends an ACK for byte 3401, indicating that the next byte it expects to receive is byte 3401

19)

⇒ The server will accept the data it was expecting, starting at byte 6001, which will cover bytes 6001 to 8000.

⇒ The server has data ready to send (4001 to 5000 bytes)

⇒ The server sends both the ACK for byte 8001 and the data from bytes 4001 to 5000 in the same segment.

This is known as piggybacking and it is used by TCP to minimize unnecessary packet overhead.

Scanned with OKEN Scanner

20) In SCTP (Stream Control Transmission Protocol), a packet can carry multiple DATA chunks which has user data and chunk header (12 bytes)

Size of each DATA chunk :

22 bytes (user data) + 12 bytes (chunk header) = 34 bytes.

Total size of packet :

= 2 chunks x 34 bytes per chunk

= 68 bytes.

---

21) The server must run continuously to be available for handling incoming requests from clients at any time. It serves multiple clients simultaneously and ensures uninterrupted service.

The client runs only when needed to make requests to the server. It initiates communication, receives the response, and can shut down once the task is completed.

---

22) Food Ordering :

⇒ Control Connection - Talking to the waiter to place an order and manage the session.

⇒ Data Connection - Like the food delivery to your table

Both connections serve different purpose but work together to complete the process.

---

23) HTTP is better because :

→ Simple

→ Compatible

→ Web integration

→ Efficient.

24) TELNET is useful for remote access to servers and devices, network troubleshooting, accessing legacy systems, and managing devices via a command-line interface.

It allows system administrators to control and configure systems remotely, but ~~no~~ does not support file transfer or web browsing like FTP or HTTP.

25) No, because

→ TCP is reliable

→ ensures error checking & flow control

→ TCP is connection-oriented communication.

These features are lacked by UDP and these are essential for accurate and complete file transfers.