Q.No : 11) **Score : 4.00 / 4.00**

A. Given the following address 193.45.67.23/24. Extract the following information: i) First address ii) Last address iii) Number of addresses in the block. iv) Network mask.

B. An organization is granted a block of addresses with the beginning address 15.15.15.0/24. The organization needs to have 3 subblocks of addresses to use in its three subnets: one subblock of 12 addresses, one subblock of 62 addresses, and one subblock of 120 addresses. Draw a network topology and Give IP assignments to each subblocks.

Page:1 11

A) 193.45.67.23/24

$\overbrace{\boxed{\text{netid}}}^{24\text{bits}}\overbrace{\boxed{\text{hostid}}}^{8\text{bits}}$

/24 implies

⇒ Network mask = 255.255.255.0

i) First address = (193.45.67.23) AND (255.255.255.0)
= 193.45.67.0

ii) Last address = (193.45.67.23) OR (0.0.0.255)
= 193.45.67.255.

iii) Number of addresses in block = $2^{32-24} = 2^8$
= 256 addresses

iv) Network mask = 255.255.255.0.

B) Starting address : 15.15.15.0/24.

a) Subblock 3 needs 120 addresses ⇒ 128 addresses given
b) Subblock 2 needs 62 addresses ⇒ 64 addresses given
c) Subblock 1 needs 12 addresses ⇒ 16 addresses given

a) 128 addresses needed

⇒ Network prefix = $32 - \log_2 128 = 25$

Subnet address = 15.15.15.0/25

Starting Address = 15.15.15.0

Last Address = 15.15.15.127

Page:2 11

b) 64 addresses needed

⇒ Network Prefix = $32 - \log_2 64 = 26$.

Starting address = 15.15.15.128

last address = 15.15.15.191

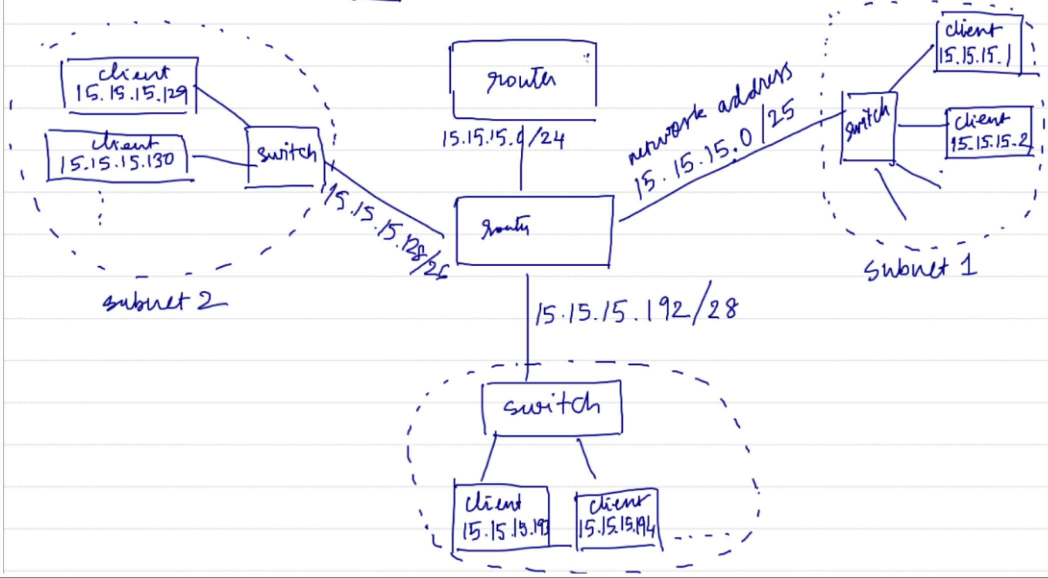
Subnet address = 15.15.15.128/26

c) 16 addresses needed

⇒ Network prefix = $32 - \log_2 64 = 26$.
Starting address = 15.15.15.128
last address = 15.15.15.191
Subnet address = 15.15.15.128/26

c) 16 addresses needed
⇒ Network prefix = $32 - \log_2 16 = 28$
Starting address = 15.15.15.192
Last address = 15.15.15.207
Subnet address = 15.15.15.192/28

Network Topology:



Q.No : 12) **Score : 4.00 / 4.00**

A. Why checksum is used in UDP? Find the checksum for the following:

1110011001100110 and 1101010101010101 .

B. Discuss multiplexing and demultiplexing w.r.t transport layer services with neat diagrams.

Page:1 12

A. UDP is an unreliable data transfer protocol. Packets can be lost or corrupted and UDP does not provide any mechanisms to recover from this.
So, checksum is used to check for any errors in transmission. UDP will retransmit data if the checksum & sum received on the other side does not add up to 0

Checksum calculation :

```
   1 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0
 + 1 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
 ---------------------------------
 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1
```

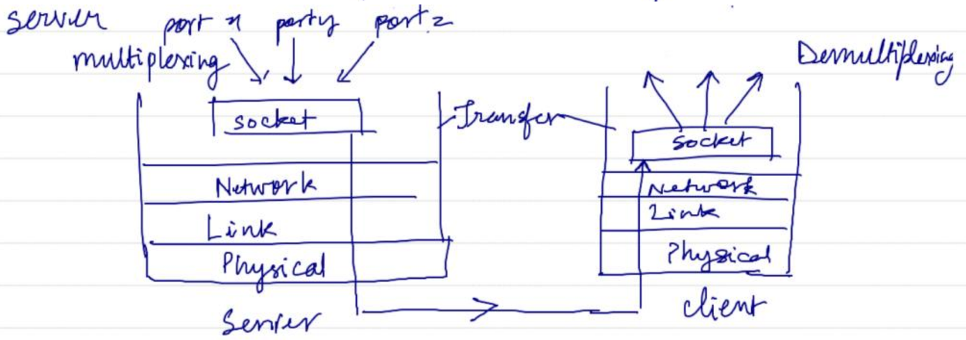
Extra 1 bit is added again

1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1

Sum 1 0 1 1 1 0 1 1 1 0 1 1 1 1 0 0

Checksum = 1s complement of sum
= 0100010001000011

B. Multiplexing : The packets from various hosts on a server are combined and transferred together to the client via the socket.
Demultiplexing: The packets coming from server are separated and send to the respective destination ports.



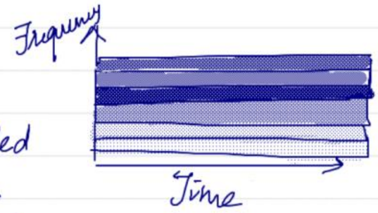
Q.No : 13) **Score : 3.00 / 3.00**

What are the different techniques in which communication medium is shared among several channels in circuit switching? Explain.

Page:1 13

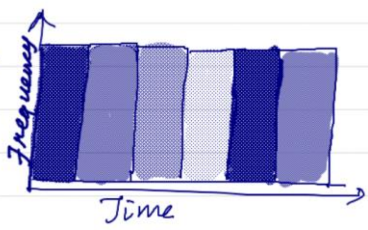
Circuit switching can happen via Time Division Multiplexing or Frequency division multiplexing.

Frequency Division Multiplexing (FDM)



→ The communication medium is divided into several channels based on frequency.
→ Packets can be transmitted across any of the frequency channels
→ They are not transmitted at the same rate. Packets transmitted in the higher frequency channel are faster than in lower transmission channel.
→ Several packets are transmitted at the same time.

Time Division Multiplexing (TDM)
→ The transmission in communication medium is divided into time slots.
→ Only 1 packet is transmitted at a time
→ All packets are transmitted at the same rate – the entire frequency band is used for transmission per packet.



Q.No : 14) **Score : 3.00 / 3.00**

Compare and contrast HTTP persistent and non-persistent connections in detail.

Page:1 14

Persistent HTTP connections

→ After a transfer of content is done from server to client, the HTTP connection stays alive

→ If another transfer of connection needs to happen then connection need not be established again as connection still exists.

→ Transfer of content can happen faster as time is not spent in establishing a connection each time

Non-persistent HTTP Connections

→ Once the server responds to client with the information it needs, the connection is closed

→ If another transfer of contents needs to happen then a new connection has to be established between client & server

→ It is slower as time is spent in establishing connection.

Q.No : 15) **Score : 3.00 / 3.00**

Whether the following applications are tolerant to data loss or not? i) file transfer ii) Email. Also, mention the application-layer and transport-layer protocols used in these applications.

Page:1 15

→ File transfer & Email are not tolerant to data loss. The transport layer protocol used must be reliable.

i) File Transfer :

Application-layer protocol — FTP

Transport-layer protocol — TCP (for reliable data transfer)

ii) Email :

Application-layer protocol — SMTP

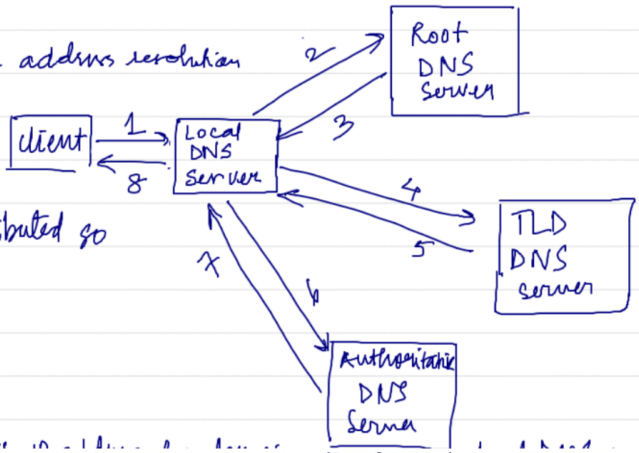
Transport-layer protocol — TCP (for reliable data transfer)

Q.No : 16) **Score : 3.00 / 3.00**

Compare and contrast iterative and recursive name address resolution in domain name systems using appropriate neat diagrams.

Page:1 16

Iterative name address resolution



→ Traffic is distributed so no delays

→ Client requests IP address for domain name from Local DNS server.

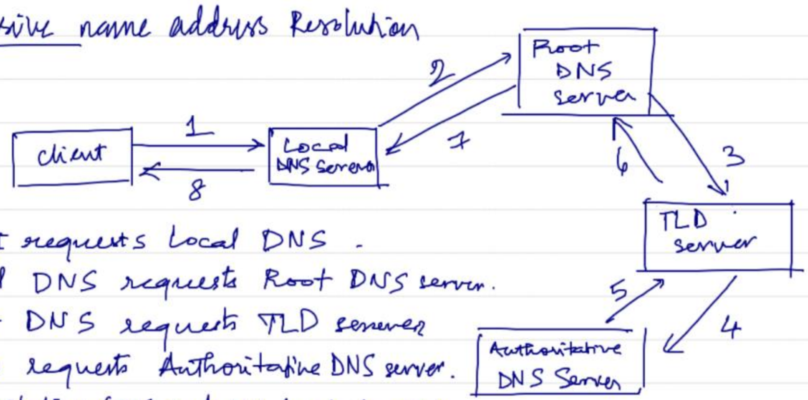
→ Local DNS server makes a request to Root DNS which sends the response back to local DNS (with IP address of TLD server)

→ Local DNS sends request to TLD server which responds back to local DNS with IP address of authoritative DNS server.

→ Local DNS sends request for ip address of domain name to Authoritative DNS server which responds back to local DNS

→ Local DNS forwards IP address of domain name to client.

Recursive name address Resolution



→ Client requests Local DNS.
→ Local DNS requests Root DNS server.
→ Root DNS requests TLD server
→ TLD requests Authoritative DNS server.
→ Authoritative DNS responds back to TLD
→ TLD responds back to Root DNS
→ Root DNS responds back to local DNS
→ DNS responds to client

→ There is a lot of traffic at higher-hierarchy DNS which can cause delays.

Q.No : 17) **Score : 3.00 / 3.00**

How concurrency of server function can be achieved in TCP socket programming? Write a simple TCP server program in C to demonstrate it.

Page:1 17

Concurrency is achieved by creating child processes. Parent process continues to listen for connections & child process communicates with client.

Concurrent TCP server program—

```
#include <stdio.h>
#include <netinet/in.h>
#include <sys/socket.h>
#include <sys/types.h>
#include <string.h>
#include <stdlib.h>
#define PORTNO 7777
int main() {
    int sockfd, newsockfd, n;
    struct sockaddr_in sadd, cadd;
    sockfd = socket(AF_INET, SOCK_STREAM, 0);
    sadd.sin_family = AF_INET;
    sadd.sin_addr.s_addr = inet_addr("172.15.52.64")
    /*server IP address = 172.15.52.64" */
    sadd.sin_port = htons(PORTNO);
    n = bind(sockfd, (struct sockaddr*) &sadd, sizeof(sadd));
    n = listen(sockfd, 5);
    while (1) {
        n = accept(newsockfd, (struct sockaddr*) &cadd, sizeof(cadd));
        /* Parent accepting more connection requests */
        if(fork() == 0) {
            /* child communicating with client */
            char buf[11];
            strcpy(buf, "Hello World");
```

```
        n = write(newsockfd, buf, sizeof(buf));
        close(newsockfd);
    }
}
close(sockfd);
return 0; }
```

Q.No : 18) Score : 2.00 / 2.00

Discuss the disadvantage of Classfull addressing. How these issues are handled in Classless addressing?.

Page:1 18

There are two major disadvantages of classfull addressing –

→ Address utilization – In class A addressing, each network has 2^{24} addresses which is too many for an organization. In class C, each network has $2^8 = 256$ addresses which may be too less for an organization.
Too many addresses are wasted in class A and class B & they may not be sufficient in class C.

→ Scalability – There are very few addresses in classfull addressing which can be used compared to the amount of users. So, if number of users increase, they cannot be accomodated. So, classfull addressing is not scalable.

→ In classless addressing, a network can be assigned a network address based on the number of addresses it wants.
Eg: If an organization needs 250 addresses, then it can have a network address of x.y.z.0/24.
If an organization needs only 10 addresses, then it can have a network address of x.y.z.240/28
This means less wastage of addresses & more users can be accomodated