



ENROLMENT NO.: 186407279

NAME: Aman Tiwari

ADDRESS: G-284 gazipur village Delhi 110096

PROGRAMME TITLE: BCA 5

COURSE CODE: BCS-052

COURSE TITLE: Network Programming and Administration

ASSIGNMENT CODE:

STUDY CENTRE CODE: 0769 (Shyamla College sahadra)

PHONE NO.: 9891062743 / 8448179216

EMAIL ID: amantiwari8861@gmail.com



Date

SIGNATURE

BCSO52

Ans 1/2 (a) In classful addressing, the address space is divided into five classes, A, B, C, D and E. Each of these classes has a valid range of IP addresses.

IPv4 address is divided into two parts:

- Network ID
- Host ID

	First byte	Second byte	Third byte	Fourth byte	Binary notation
Class A	0				
Class B	10				
Class C	110				
Class D	1110				
Class E	1111				

	First byte	Second byte	Third byte	Fourth byte	Dotted decimal notation
Class A	0-127				
Class B	128-199				
Class C	192-223				
Class D	224-239				
Class E	240-255				

The classful addressing wastes a large part of the address space.

- Class A : $0 \dots 2^7 \dots 2^{12}4$
- Class B : $10 \dots 2^{14} \dots 2^{16}$
- Class C : $110 \dots 2^{21} \dots 2^{10}$
- Class D : $1110 \dots 1 \dots 2^{12}8$

Class	Number of Books	Block Size	Application
A	128	16,777,216	Unicast
B	16,384	65,536	Unicast
C	2,097,152	256	Unicast
D	1	268,435,456	Multicast
E	1	268,435,456	Reserved

Problems with Classful addressing

The problem with this classful addressing method is that millions of Class A addresses are wasted, many of the Class B addresses are wasted, whereas, number of addresses available in Class C is so small that it cannot cater to the needs of organizations.

Class D address are used for routing and are therefore available as a single block only.

Class E address are reserved.

(b) Subnetting is the strategy used to partition a single physical network into more than one smaller logical sub-networks. A IP address includes a network a network segment and host segment.

 A Subnet mask is a number that defines a range of IP addresses available within a network. A Single Subnet mask limits the number of Valid IPs for a Specific network

(c) Address: 205.50.39.56 11001101.00110010 00100111.000111000
 Netmask: 255.255.255.0 = 24 11111111.11111111 00000000
 WildCard: 0.0.0.255 0000000000000000 00000000 11111
 \Rightarrow

Network: 205.50.39.0/24 11001100011111000000111

Broadcast: 205.50.39.255 11001101.00110010 00100111.00000001

Hostmin: 205.50.39.1 ~~255~~ 11001101.00110010.00100111.00000001

Last max: 205.50.39.254 11011001.00110010 00110011.11110000

Hosts/Net: 254

Supernet

Netmask: 255.255.240.0 = 20 11111111110000000000000000

WildCard: 0.0.15.255 00000000000000001111111111

Network 205.50.32.0/20 110010011100011000110000

Broadcast: 205.50.47.255 110011001001111100000000

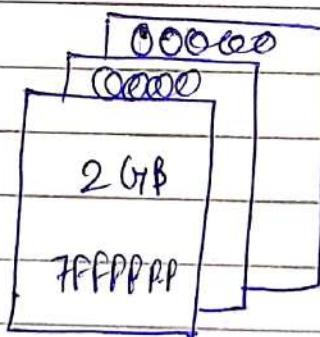
Hostmin: 205.50.32.1 11011000111110000011

Hostmax :- 205.50.47.254 110011000 1110001111110 -

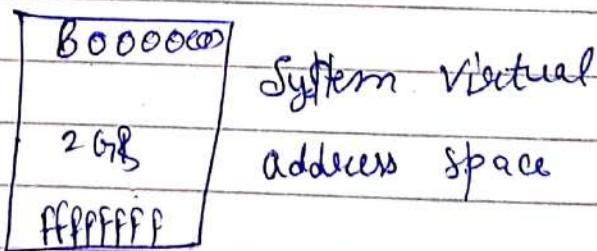
Host/Net : 4094

Ans(a) = Processors like Notepad.exe and MyApp.exe run in User mode. Core operating components and many drivers run in the more privileged Kernel mode. For more information about Processor modes, see User mode and kernel mode.

In 32 bit windows, the total available Virtual Address Space is 2^{32} bytes (4 gigabytes). Usually the lower 2 gigabytes are used for user space, and the upper 2 gigabytes are used for system space.



User-Process Virtual address Space



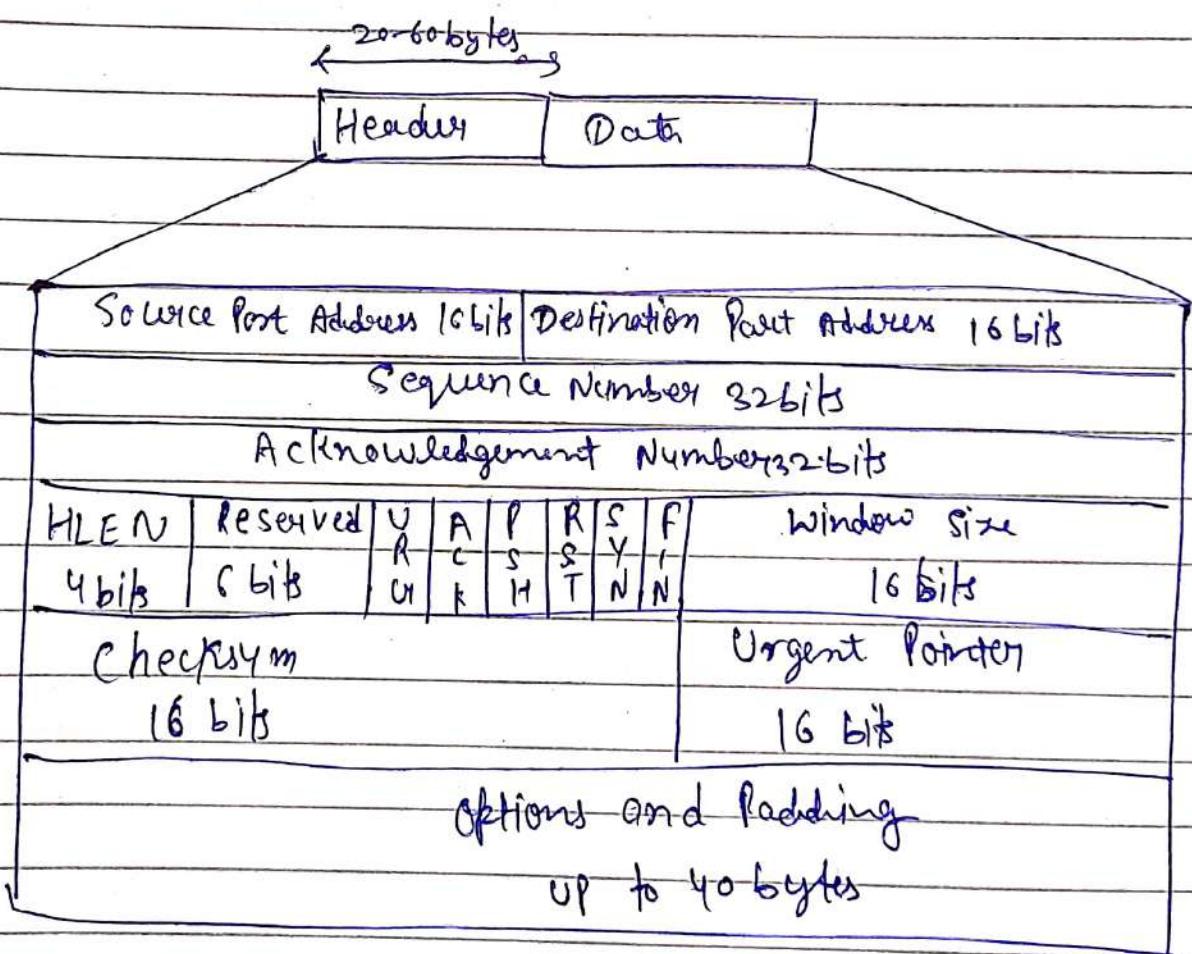
(b) Introduced the concepts of IP address classes and showed how the class related to ranges of IP addresses. Of the five classes, D and E are dedicated to special purposes, so I will leave these alone for now. Classes A, B and C are the ones actually assigned for normal (unicast) addressing purposes on IP internetworks and therefore the primary focus of our continued attention.

IP Address Class	Total # of Bits for Network/Hosts	First Octet of IP Address	# of Network ID Bits Used to Identify Class	Usable # of Network ID Bits	Number of Network IDs
Class A	8/24	0 XXXXXX	1	$8-1=7$	$2^7 = 128$
Class B	16/16	10 XXXXX	2	$16-2=14$	$2^{14} = 16,384$
Class C	24/8	110 XXXXX	3	$24-3=21$	$2^{21} = 2097152$

<u># of Host IDs per network ID</u>
$2^{24}-2 = 16,277,214$
$2^{16}-2 = 65,534$
$2^8-2 = 254$

Ans 3(a) The Transmission Control Protocol is the most common transport layer protocol. It works together with IP and provides a reliable transport service between processes using the network layer service provided by the IP protocol.

TCP Segment consists of data bytes to be sent and a header that is added to the data by TCP as shown:



Source Port address -

16 bits field that holds the Port address of the application that is sending the data segment.

Destination Port Address -

16 bit field holds the Port address of the application in the host that is receiving the data segment.

Sequence Number -

32 bits field that holds the Sequence number, i.e. the byte number of the first bytes that is sent in that particular segment.

Acknowledgement Number:-

32 bit field that holds the acknowledgement number, i.e. byte number that the receiver expects to receive next.

Header length (HLEN) -

This is 4 bit field that indicates the length of the TCP header by number of 4 byte words in the header,

Control flags:-

These are 6 bits control bits that control connection establishment, connection termination, connection abortion, flow control mode of transfer etc.

URG: Urgent pointer is valid

ACK: Acknowledgement number is valid

PSH: Request for Push

RST: Reset the Connection

Syn: Synchronize Sequence numbers

FIN: Terminate the Connection.

Window Size

This field tells the window size of the sending TCP in bytes.

Checksum -

This field holds the checksum for error control.

It is mandatory in TCP as opposed to UDP.

Urgent Pointer:

This field is used to point to data that is urgently required that needs to reach the receiving process at the earliest.

TCP Connection -

TCP is connection oriented. A TCP connection is established by 3 way hand shake.

(b) Sequence Number - A TCP connection is a method of transmitting two byte streams on Stream in each direction. To map the unordered unreliable bytes in IP packets to the ordered bytes in this Stream, each byte in each Stream is identified by a sequence number. Each TCP packet contains a segment of the Stream as its payload.

Reserved bits: Reserved data in TCP headers has a value of zero. This field aligns the total header size as a multiple of four bytes, which is important for the efficiency of computer data processing.

Page fault error Cache bits:

bit 0 == 0: no page found 1: protection fault

bit 1 == 0: read access 1: write access

bit 2 == 0: kernel-mode access 1: user-mode access

bit 3 == 1: use of reserved bit detected

bit 4 == 1: fault was an instruction fetch

Window Size: - TCP window size is one of the most popular options for network troubleshooting or performing an application baseline. I've read many articles and books that can make this option quite overwhelming, but it's actually pretty straightforward.

The TCP window size, or as some call it, the TCP receive window size is simply an advertisement of how much data (in bytes) the receiving device is willing to receive at any point in time. The receiving device can use this value to control the flow of data, or as a flow control mechanism.

Checksum: The TCP/IP checksum is used to detect corruption of data over a TCP or IPv4 connection. If a bit is flipped, a byte mangled, or some other badness happens to a packet, then it is highly likely that the receiver of that broken packet will notice the problem due to a checksum mismatch. This provides end-to-end assurance that the data stream is correct.

Urgent Priority:- TCP offers that ability to flag certain bytes of data as "Urgent". This feature allows an application to process and forward any data that must be dealt with immediately without that data having to sit in the Send Queue for processing. Instead, the data is packaged into a segment, the Urgent flag is set in the TCP header, and a byte offset marking the end of urgent data is specified in the Urgent Point field.

Ques 2: **Role of DNS in Internet:-** The Domain Name System (DNS) is the Phonebook of the Internet. Humans access information online through domain names, like nytimes.com or espn.com. Web browsers interact through Protocol (IP) addresses. DNS translates domain name to IP addresses so browsers can load internet resources.

Domain name resolution:- Suppose you are an employee within XYZ Industries and your client is in charge of the networking department at Googleplex. You type your web browser the address of this department's web server.

1. Your Web browser recognizes the request for a name and invokes your local resolver, passing to it the name "www.netcompsw.googleplex.edu".
2. The resolver checks its Cache to see we are attempting to see if it already has the address for this name.
3. The resolver generates a recursive query and sends it to "ns1.xyzindustries.com"
4. The local DNS Server receives the request and Checks its Cache. Again let's assume it doesn't have the information needed.
5. "ns1.xyzindustries.com" generates an interactive request and sends it to the name server for ".edu".
6. The root name Server does not resolve the name.
7. "ns1.xyzindustries.com" generates an interactive request and sends it to the name server for ".edu".
8. The name server for ".edu" returns the name and address of the name server for the "googleplex.edu" domain.

9. "nl.xyzindustries.com" generates an interactive request, and sends it to the name server for "googleplex.edu".
10. The name server for "googleplex.edu" consults its resource records.

Ans 5 (a) TCP Server

```
#include <sys/types.h>
#include <netdb.h>
#include <netinet/in.h>
#include <stroib.h>
#include <string.h>
#include <sys/socket.h>
#include <sys/types.h>
#define MAX 80
#define PORT 8080
#define SA struct sockaddr
void func(int Sockfd)
{
    Char buff[MAX];
    int n;
    // infinite loop for chat
    for(;;){
        bzero(buff,MAX);
        // read the message from client and copy it in
        // buffer
        read(Sockfd,buff,sizeof(buff));
    }
}
```

// Print buffer which contains the Client
Contents

```
printf("from Client: %s\nTo Client: ", buff);
bzero(buff, MAX);
n = 0;
```

// Copy Server message in the buffer

```
while ((buff[n++]) = getchar() != '\n')
```

// and send that buffer to Client

```
write(sockfd, buff, sizeof(buff));
```

// if msg contains "Exit" then Server exit and chat
ended.

```
if (strcmp("exit", buff, 4) == 0) {
```

```
    printf("Server Exit...\n");
```

```
    break;
```

```
?
```

```
{
```

// Driver function

```
int main()
```

```
{
```

```
    int sockfd, connfd, len;
```

```
    struct sockaddr_in servaddr, cli;
```

// Socket Create and Verification

```
    sockfd = socket(AF_INET, SOCK_STREAM, 0);
```

```
    if (sockfd == -1) {
```

```
        printf("Socket creation failed...\n");
```

```
        exit(0);
```

```
    }
```

```
else
```

// assign IP, PORT

Servaddr.sin_family = AF_INET;

Servaddr.sin_addr.s_addr = htonl(INADDR_ANY);

Servaddr.sin_port = htons(PORT);

// Binding newly created Socket to given IP

and Verification

if ((bind (Sockfd, (SA*) & Servaddr, sizeof(Servaddr)))
= 0) {

printf ("Socket bind failed... \n");
exit(0);

}

else

printf ("Socket successfully binded... \n");

// Now Server is ready to listen and
Verification

if ((listen (Sockfd, 5)) = 0) {

printf ("Listen failed... \n");
exit(0);

}

else

printf ("Server listening... \n");

len = sizeof (cli);

// Accept the data packet from client and
Verification

Connfd = accept (Sockfd, (SA*) & cli, & len);

if (Connfd < 0) {

printf ("Server accept failed... \n");

```

    exit(0);
}
else
    printf("Server accept the client--.\n");
    // Function for Chatting between Client and Server
    func(connfd);
    // After Chatting Close the socket
    close(sockfd);
}

```

TCP Client

```

#include <netdb.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/socket.h>
#define Max 80
#define Port 8080
#define SA struct sockaddr
void func(int sockfd)
{

```

```

    char buff[Max];
    int n;
    for(;;)
        bzero(buff, sizeof(buff));
        printf("Enter the String: ");
        n = 0;
        while ((buff[n++] = getch()) != '\n')

```

```
Write (sockfd, buff, sizeof (buff));
bzero (buff, sizeof (buff));
read (sockfd, buff, sizeof (buff));
printf ("From Server: %s", buff);
if ((strcmp (buff, "exit", 4)) == 0) {
    printf ("Client Exit..\n");
    break;
}
```

```
{ . .
int main()
```

```
{ . .
int sockfd, connfd;
Struct sockaddr_in servaddr, cli;
// Socket Create and verification
sockfd = socket (AF_INET, SOCK_STREAM, 0);
if (sockfd == -1) {
    printf ("Socket creation failed ... \n");
    exit (0);
}
else
    printf ("Socket successfully created. \n");
bzero (&servaddr, sizeof (servaddr));
// assign IP, PORT
servaddr.sin_family = AF_INET;
servaddr.sin_addr.s_addr = inet_addr ("127.0.0.1");
servaddr.sin_port = htons (PORT);
```

```

    // Connect the Client Socket to Server Socket
    if (connect (sockfd, (SA *) & servaddr, sizeof (servaddr)) != 0) {
        exit(0);
    }
    else
        printf ("Connected to the Server... \n");
    // Function for Chat.
    func(sockfd);
    // Close the Socket
    close (sockfd);
}

```

Output -

\$ Server Side
 Socket Successfully Created
 Socket Successfully binded
 Server listening
 Server accept the client
 from Client: hi
 To Client: hello
 from Client: exit
 To Client: exit
 Server exit .

Client Side

Socket successfully created

Connected to the Server

Enter the String : hi
from Server : hello

Enter the String ! exit
from server : exit
Client exit

(b) **Socket Types**:- Sockets are classified according to Communication properties. Processes usually communicate between Sockets of the same type. However, if the underlying communication protocols support the communicate Sockets of different types can communicate.

/* Standard socket types */

```
#define SOCK_STREAM
#define SOCK_DGRAM
#define SOCK_RAW
#define SOCK_RDM
```

/* Virtual Circuit */

```
2/* datagram */
3/* raw socket */
4/* reliably-delivered
   message */
```

```
#define SOCK_CONN_DGRAM 5/* Connection datagram */
```

Socket protocols = A protocol is a standard set of rules for transferring data such as UDP/IP and TCP/IP. An application can specify a protocol only if more than one protocol is supported for this.

Particular socket type in this domain.

The /usr/include/sys/socket.h file contains a list of socket protocol families. The following list provides examples of Protocol families (PF) found in the Socket header file:

PF_UNIX Local communication

PF_INET Internet (TCP/IP)

PF_NS Xerox Network System (XNS) architecture

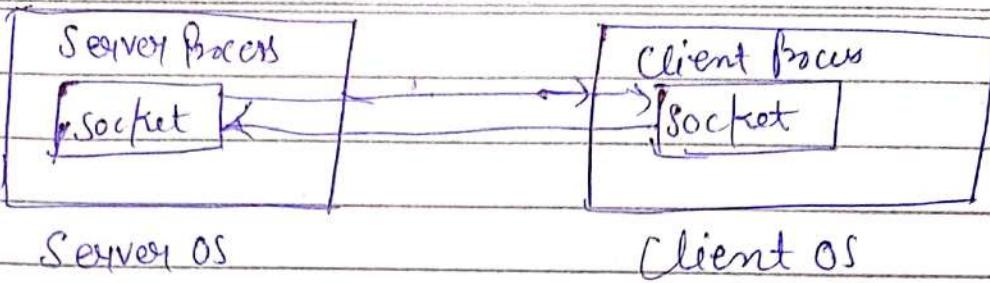
PF_NDD AIX NDD

The System Calls used by the client and a Server to establish connections before data transfer:

Client / Server Communication involves two components namely a client and a server. They are usually multiple clients in communicate with a single server.

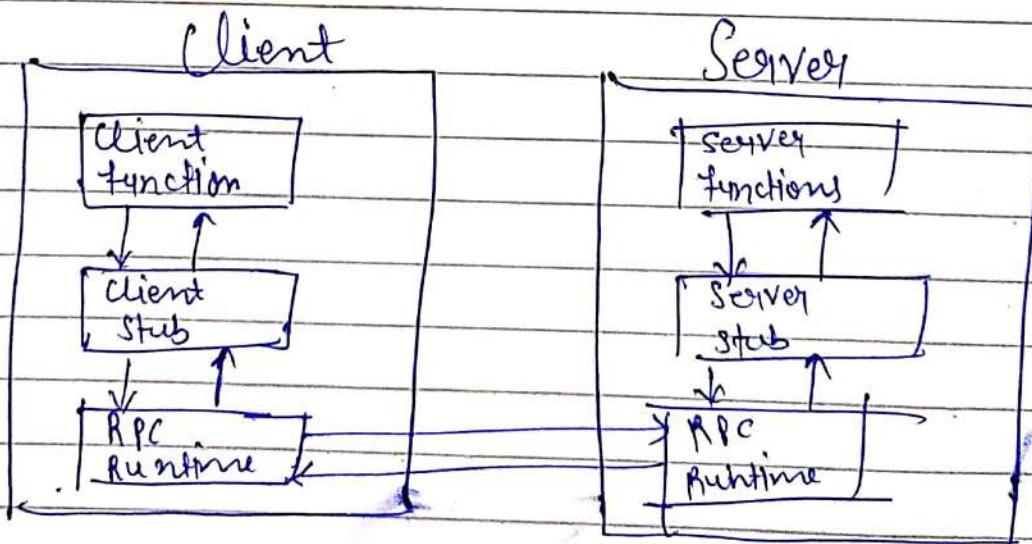
Sockets

Sockets facilitate communication between two processes on the same machine or different machines. They are used in a Client / Server framework and consist of the IP address and port number.



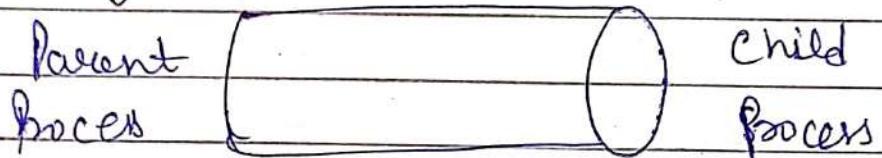
Remote Procedure Calls

These are interprocess communication techniques that are used for client-server based applications. A remote procedure call is also known as a subroutine call or a function call. A Client has a request that the RPC translates and sends to the Server. This request may be a procedure or a function call to a remote Server. When the Server receives the request, it sends the required response back to the client.



Pipes

These are interprocess communication methods that contain two end points. Data is entered from one end of the Pipe by a process and consumed from the other end by the other process.



The Syntax is:-

`Open: devicename; parameters; timeout!`
`namespace.`

- (c) Remote administration: Remote administration refers to any method of controlling a computer from a remote location. Software that allows remote administration is becoming increasingly common and is often used when it is difficult or impractical to be physically near a system in order to use it. A remote location may refer to a computer in the next room or one on the other side of the world.

Common Services for which the remote administration is used:-

Shutdown

- Shutting down or rebooting another computer over a network.

Accessing Peripherals

- Using a network device, like Printer
- Retrieving Streaming data, much like a CCTV system

Modifying

- Editing another computer's Registry setting
- Modifying System Services
- Installing Software on another machine
- Modifying logical groups.

Viewing

- Remotely viewing Others
- Supervising Computer or internet usage
- Access to a remote System's "Computer Management" Snap-in

Hacking

Computers infected with malware such as Trojans sometimes open back doors into Computer Systems which allows malicious users to hack into and control the Computer.

Remote administration tools

- 1) SolarWinds Dameware Remote Support:
 - SolarWinds Dameware Remote Support is an easy-to-use package of remote control and system management tools.
 - It will allow you to reboot systems, start/stop services & processes, copy/delete files, view & clear event logs, etc.
 - It provides system tools and TCP utilities to help you utilities to help you remotely troubleshoot computers without launching a full remote control session.
 - You will be able to remotely access network computer through mobile devices.
 - It will allow you to remotely unlock user accounts, reset password, and edit Group Policy. All properties, system configuration, and software information can be exported in CSV or XML formats.
 - It provides centralized license & user account management and multi-factor authentication.
 - This Remote Support will allow you to remotely access sleeping and powered off computers.

(2) NinjaRamm:

- NinjaRamm directly takes control of windows and Macos devices with integrated Cloud RDP, Teamviewer, and Splashtop.
- Monitors all your windows and Macos Workstations, laptops, and Servers.
- Remotely manage all your devices without interrupting end-user through a robust suite of remote tools.
- Automate OS and third-party application Patching for windows and Macos devices.
- Standardize the deployment, Configuration, and management of devices with Powerful IT automation.

(3) Teamviewer

- Teamviewer is another remote desktop access tool which is used for accessing any desktop system, Android or windows 10 devices.
- This software even supports cross platform such as pc to pc, mobile to mobile pc to mobile to pc.
- Team viewer is well known for online teamwork and the estimated revenue of this tool is around \$125m per annum, in 2016 it was reported as \$200m.

Ans 6: Computers are connected in a network to exchange information or resources each other. Two or more computer connected through network media called Computer network.

1. If Config

ifconfig command is used to initialize an interface, assign IP address to interface and enable or disable interface on demand.

ifconfig

```
eth0      Link encap: Ethernet HWaddr 00:0c:29:28:fd:4c
          inet brdcast 192.168.50.255 mask: 255.255.0.0
                  inet adder: 192.168.50.2 Bcast: 192.168.50.255 Mask:
```

```
inet adder: fe80::20c:29ff:fe28:fd4c/64 Scope:Link
      UP BROADCAST RUNNING MULTICAST MTU:1500 metric:1
```

RX Packets: 6093 errors: 0 dropped: 0 overruns: 0 frame: 0

TX Packets: 4824 errors: 0 dropped: 0 overruns: 0 carrier: 0

Collisions: 0 txqueuelen: 1000

Rx bytes: 6125302 (5.8 MiB) Tx bytes: 536966 (524.3 kB)

interrupt: 18 base address: 0x2000

lo

Link encap: Local Loopback

inet adder: 127.0.0.1 Mask: 255.0.0.0

inet 6 adder: ::1/128 Scope: HOST

UP Loopback RUNNING MTU: 16436 Metric: 1

RX Packets: 0 errors: 0 dropped: 0 overruns: 0 frame: 0

Tx packets: 0 errors: 0 dropped: 0 overruns: 0 carrier: 0
 Collisions: 0 frames: 0

Rx bytes: 480 (480.0 b) Tx bytes: 480 (480.0 b)

Q. PING Command

PING (Packet Internet Groper) Command is the best way to test connectivity between two nodes. Whether it is Local Area Networks (LAN) or wide Area Network (WAN). Ping use ICMP (Internet Control message Protocol) to communicate to other devices. You can ping host name or IP address using below command:

Ping 4.2.2.2

PING 4.2.2.2 (4.2.2.2) 56 (84) bytes of data.

64 bytes from 4.2.2.2: icmp_seq=1 ttl=44 time=203 ms

64 bytes from 4.2.2.2: icmp_seq=2 ttl=44 time=201 ms

64 bytes from 4.2.2.2: icmp_seq=3 ttl=44 time=201 ms

3. TRACEROUTE Command

traceroute is a network troubleshooting utility which shows number of hops taken to reach destination also determine packet traveling path. Below we are tracing route to global DNS server IP Address and able to reach destination also shows path of that packet is traveling

traceroute 4.2.2.2

4. NETSTAT Command

Netstat (Network statistic) Command display Connection
if info routing table information etc. To displays
routing table information use option -r

netstat -r

Destination Gateways Genmask flags mss window
Rttt Free

192.168.50.0 * 255.255.255.0 U 00 0 eth0

link-local * 255.255.0.0 U 00 0 eth0

default 192.168.50.1 0.0.0.0 UH 00 0 eth0

5. DIG Command.

DIG (domain information groper) query DNS related
information like A record, CNAME, MX Record
etc. This Command mainly use to troubleshoot
DNS related query.

dig www.fermint.com; <>> DIG 9.8.2.rcl-Red Hat
- 9.8.2-0.1.0.rcl el6 <>> www.fermint.com

; ; Global Options: +cmd

; ; Got answer:

; ; >> HEADER <

6. NSLOOKUP Command

nslookup Command also use to find out DNS related
query. The following example shows A Record
(IP Address) of fermint.com.

nslookup www.tecmint.com

Server: 4.2.2.2

Address: 4.2.2.2#53

Non-authoritative answer:

www.tecmint.com Canonical name = tecmint.com

Name: tecmint.com

Address: 50.116.66.126

7 ROUTE Command

route command also shows and manipulate ip routing table. To see default routing in Linux, type the following command.

route

Kernel IP routing table

Destination	Gateway	Genmask	Flags	Metric	Ref
192.168.50.0	*	255.255.255.0	U	0	0
clixf-local	*	255.255.0.0	U	1002	0
default	192.168.50.1	0.0.0	UG	0	0

User place

○ eth 0

○ eth 0

○ eth 0

8 Host Command:

Host Command to find name to ip or ip to name in IPv4

Or IPV6 and also query DNS records.

```
# host www.google.com
```

9. ARP Command

ARP (Address Resolution Protocol) is useful to view / add the contents of the kernel's ARP table. To see default use the command as

```
# arp -e
```

Address	Hwtype	Hwaddress	Flags mask	iface
192.168.50.1	ether	00:50:56:c0:00:08	c	eth0

10. ETHTOOL Command

ethtool is a replacement of mii-tool. It is to view, setting speed and duplex of your network interface card (NIC). You can set duplex permanently in /etc/sysconfig/network-scripts/ifcfg-eth0 with ETHTOOL_OPTS variable #

```
# ethtool eth0
```

Setting for eth0

Current message level: 0x00000007 (7)

Link detected: Yes

11. IWCONFIG Command = iwconfig Command in Linux is used to configure a wireless interface. You can see and set the basic wi-fi details like SSID, channel and encryption.

```
# iwconfig [interface]
```

12. HOSTNAME Command

hostname is to identify in a network. Execute hostname Command to see the hostname of your box.

```
# hostname  
feenint.com
```

13. GUI tool System - Config - network

Type System - Config - network in Command prompt to Configure network Setting and You will get nice Graphical User Interface (GUI) which may also use to Configure IP Address, Gateway, DNS etc. as shown below image.

```
# System - Config - network
```

Ans 7-

Function	Description
Get Disk free space	Retrieves information about the specified disk, including the amount of free space on the disk
Get Disk free Space Ex	Retrieves information about the amount of space that is available on a disk volume, which is the total amount of space, the total amount of free space available to the user that is associated with the calling thread

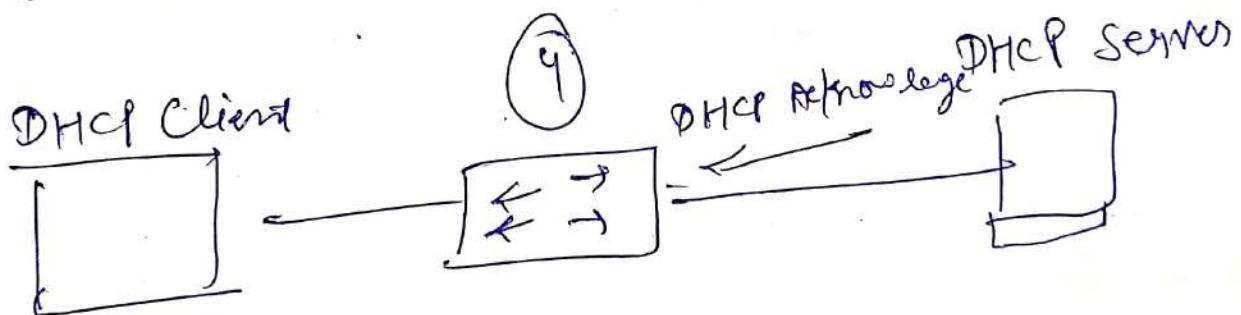
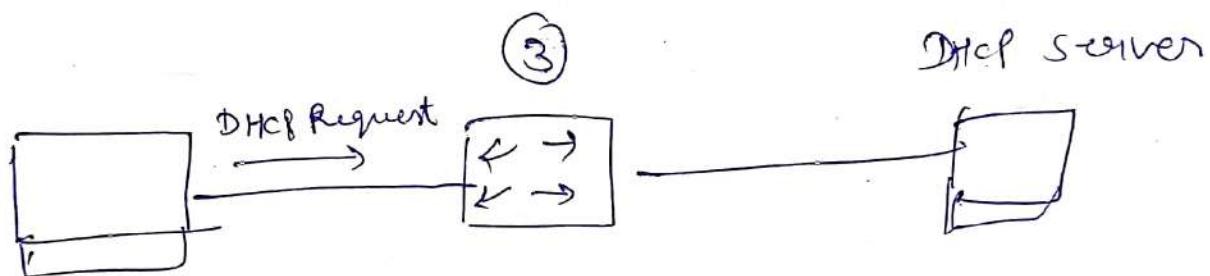
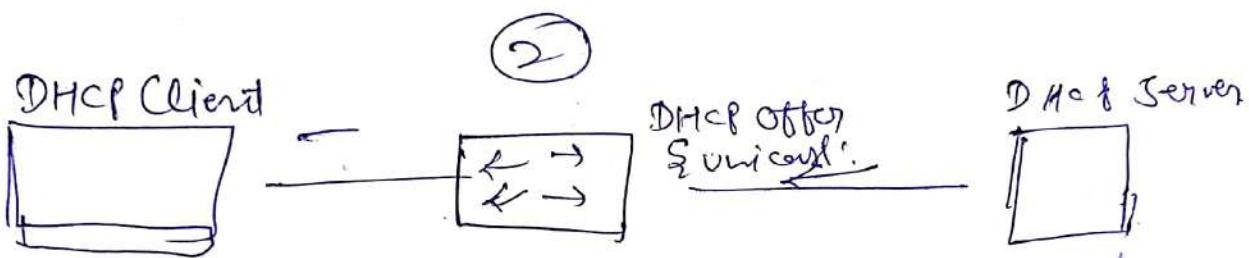
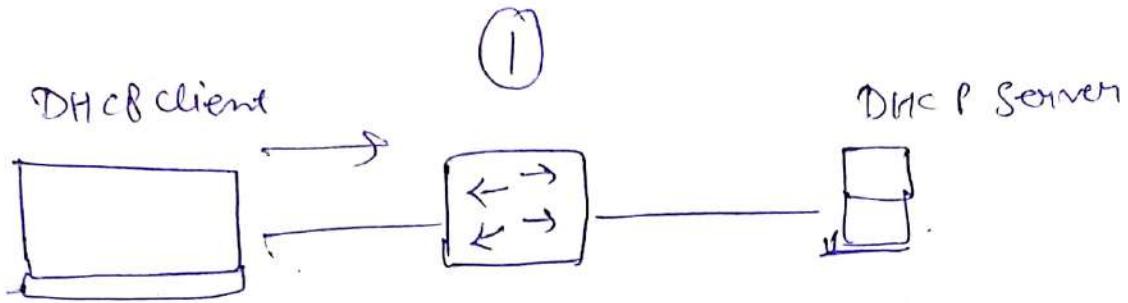
Other functions used in disk management.

In this Section	Description
Function	Description
Create file	Creates or opens a file or I/O device. The most commonly used I/O devices are as follows: file, file stream, directory, physical disk, volume, console buffer, tape device, communication resource, mailslot and pipe.
Delete file	Deletes an existing file.

Ans:- Functions of the Dynamic Host Configuration Protocol:
DHCP stands for Dynamic Host Configuration Protocol. The built in section..

Active leases:- A list of devices that have been provided DHCP leases. The DHCP Server automatically assigns these leases. The list will not include any devices that have static IP addresses on the network.

Reservation:- This is a list of devices with reserved IP addresses. This reservation is almost the same as when a device has a static IP address except that the device must still



request an IP address from the router.

How it work

1. Host connecting to network sends DHCP discover message to all hosts in layer 2 segment. Frame with this discover message lists the DHCP Server.
2. After the DHCP Server receives discover message it suggests the IP addressing offering to the client host by unicast.
3. Now after the client receives the offer it requests the information officially sending REQUEST message to server this time by via unicast.
4. Server sends Acknowledge message confirming the DHCP lease to Client. Now client is allowed to use new IP Setting

DHCP server located. DHCP server should be located on at least one subnet on a LAN in a routed network. When the server needs to support client on remote subnets separated by routers.