Date : / /
Page No. 1

BCS052

Ans 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
Class A	0			
Class B	10			
Class C	110			
Class D	1110			
Class E	1111			

Binary notation

	First byte	Second byte	Third byte	Fourth byte
Class A	0-127			
Class B	128-191			
Class C	192-223			
Class D	224-239			
Class E	240-255			

Dotted decimal notation

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

- Class A : 0 ... 2^7 ... 2^{24}
- Class B : 10 ... 2^{14} ... 2^{16}
- Class C : 110 ... 2^{21} ... 2^8
- Class D : 1110 ... 1 ... 2^{28}

Class	Number of Blocks	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 address are wasted, many of the Class B address 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. An 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 00100111.00011000
Netmask : 255.255.255.0 =24 11111111.11111111.11111111.00000000
Wildcard : 0.0.0.255 00000000.00000000.00000000.11111111
⇒
Network : 205.50.39.0/24 11001101.00110010.00100111.00000000
Broadcast : 205.50.39.255 11001101.00110010.00100111.11111111
Host min : 205.50.39.1 11001101.00110010.00100111.00000001
Host max : 205.50.39.254 11001101.00110010.00100111.11111110
Hosts/Net : 254

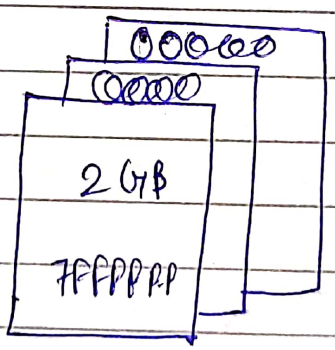
Supernet
Netmask : 25.5255.240.0 =20 11111111.11111111.11110000.00000000
Wildcard : 0.015.255 00000000.00000000.00001111.11111111
Network 205.5032.0/20 11001101.11000101.00010000.00000000
Broad cast : 205.50.47.255 11001101.00100111.11110000.11111111
Host min : 205.50.32.1 11001101.11000101.00010000.00000001

Date : / /
Page No. 4

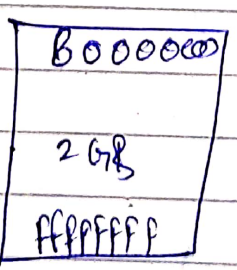
Hostmax :- 205.50.47.254 11001100 11100 11111110.
Host/Net : 4094

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



Per-Process Virtual address spaces



System virtual address space

Date : / /
Page No. 5

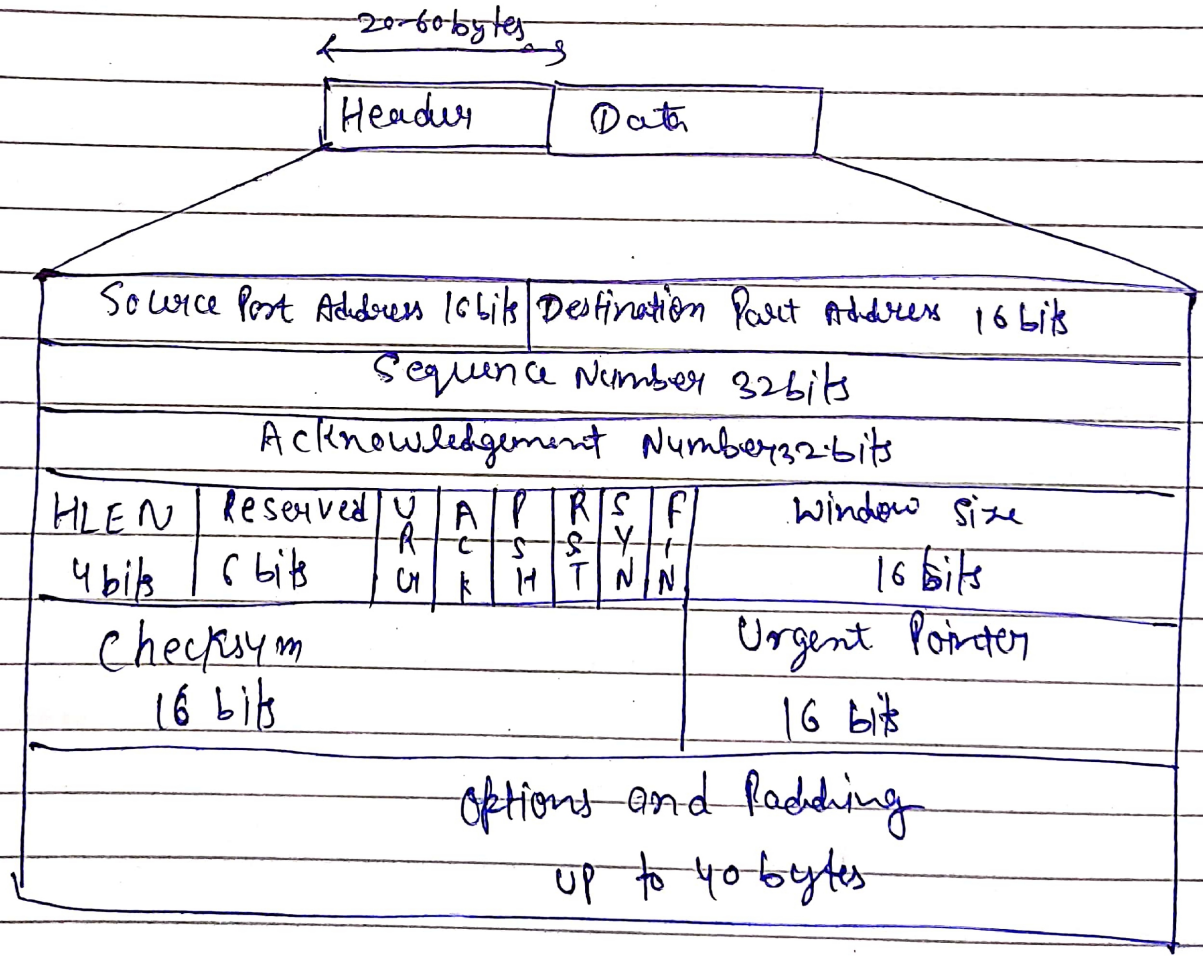
(b) Introduced the concepts of IP address Classes and showed how the class related to ranges of IP addresses. Of the five Classes, D and E are dedicated to special purposes, so I will leave these alone for now. Class A, B and C are the ones actually assigned for normal (unicast) addressing purposes on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

IP Address Class	Total # of Bits for Network ID/Hosts	First octet of IP Address	# of Network ID Bits Used to identify class	Usable # of Network ID Bits	Number of Possible Network IDs
Class A	8/24	0XXXXXX	1	8-1=7	$2^7-2=126$
Class B	16/16	10XXXXX	2	16-2=14	$2^{14}=16,384$
Class C	24/8	110XXXXX	3	24-3=21	$2^{21}=2097152$

# of Host IDs Per network ID
$2^{24}-2=$ 16277214
$2^{16}-2=65,534$
$2^8-2=254$

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, ie. the byte numbers 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 handshake.

Date : / /
Page No. 9

(b) Sequences 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 Steam 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 Code bits:

bit 0 == 0: no Page found 1: Protection fault
bit 1 == 0: read accen 1: write accen
bit2 = 0: kernel-mode access 1: User-mode accen
bit3 == 1: Use of reserved bit detected
bit 4 == 1: fault was an instruction fetch

Date : / /
Page No.

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 bytes 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 Pointer:- 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 is set in the TCP header, and a byte off-set making the end of urgent data is specified in the Urgent Point field.

Ans 4: 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 Botocol (IP) address. 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 of yours clients is in charge of the networking department at Googleplex. You type your web brower the address of this department's web Server.

Date : / /
Page No. 12

1. Your web browser recognizes the request for a name and invokes your local resolver, passing to it the name "www.networkcomputing.googleplex.edu".
2. The resolver checks its cache to see (we are assuming 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 interative request and sends it to the name servers for "edu"
6. The root name server does not resolve the name.
7. "ns1.xyzindustries.com". Generates an interative 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. "ns1.xyzindustries.com" generates an interactive request and Sends it to the nameServer for "googleplex.edu".

10. The name Server for "googleplex.edu" consults its resource records.

Ans 5 (a) TCP Server

```
#include <stdio.h>
#include <netdb.h>
#include <netinet/in.h>
#include <stdlib.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\t To 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 Creat 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");
```

Date : / /
Page No. 16

```
    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++] = getchar()) != '\n')
```

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

   int main()
     {
       int Sockfd, Connfd;
    Struct Sockaddr_in Servaddr, cli;
    // Socket create and varification
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);
```

Date : / /
Page No. 10

```
// 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

Date : / /
Page No. 19

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        1/* Virtual Circuit*/
# define Sock_DGRAM        2/* datagram*/
# define Sock_RAW          3/* raw Socket*/
# define SOCK_RDM          4/*reliably-delivered
                             message*/
# define Sock_CONN_DGRAM   5/* Connection datagram */
```

Socket Protocols = A Protocal is a Standard Set of rules for transferring data such as UDP/IP and TCP/IP. An application can specify a Protocal only if more than one Protocal is supported for this

particular socket type in this domain.

The /usr/include/Sys/socket.h file contains a list of socket protocal families. The following list provides examples of protocal families (PF) found in the socket header file.

PF_UNIX Local Communicate
PF_INET Internet (TCP/IP)
PF_NS Xerox Network System (XNS) architecture
PF_NDD AIX NDD

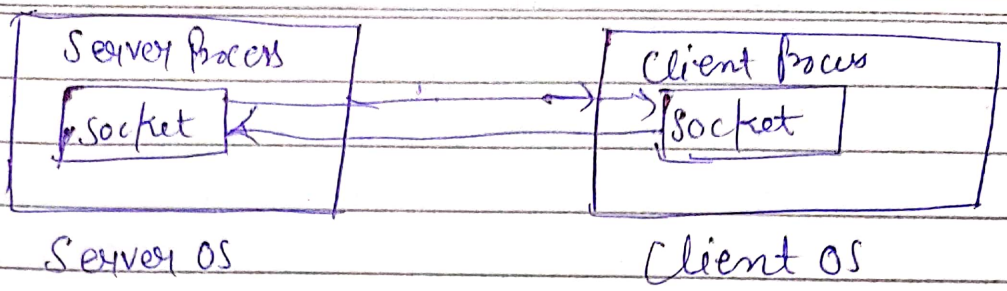
The System Calls used by the client and a Server to establish connections before data transfer.

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

Sockets

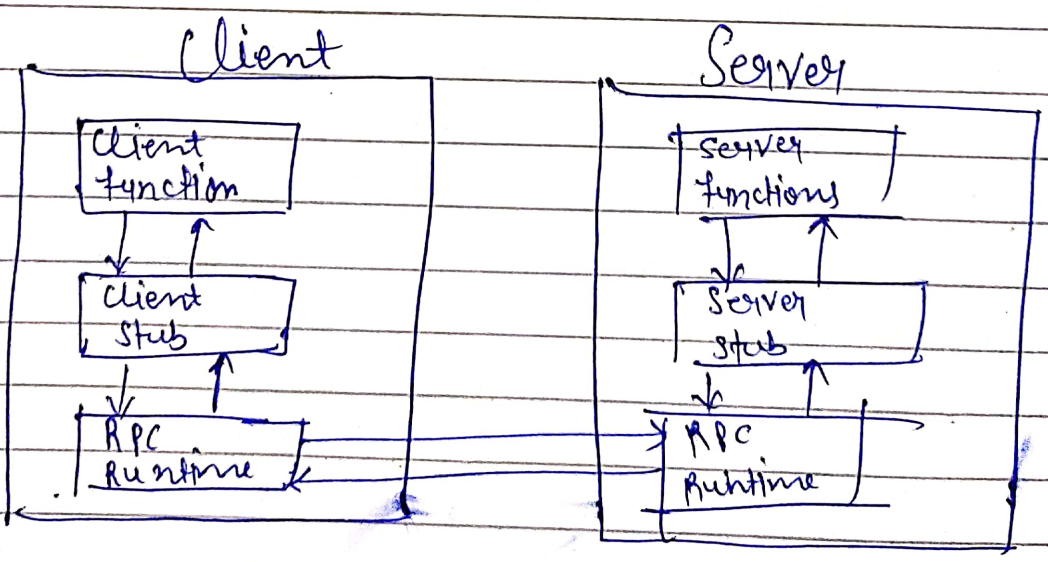
Sockets facilitate Communicate between two processes on the same machine or different machines. They are used in a client/Server framwork and consist of the IP address and Port Number.

Date : / /
Page No. 1



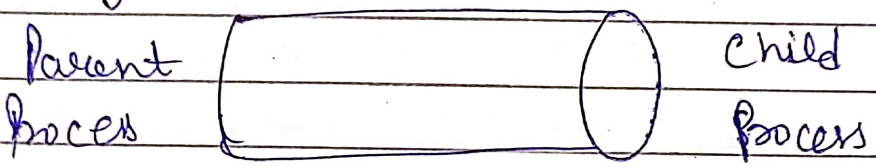
Remote Procedure Calls

There are interprocess communicate 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 translate 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 reponse 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 : mnemonic.

(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 assisting 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 door into Computer Systems which allows malicious users to hack into and control the computer.

Date : / /
Page No. 24

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 allows you to reboot Systems, Start/Stop Services & Process, 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.
- Ad 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 acces Sleeping and Powered off Computers.

(2) NinjaRemm.

- NinjaRMM directly takes control of windows and macos devices with integerated cloud RDP, Teamviewer, and Splashtop.
- Monitor all your windows and macos workstations, laptops, and Serves.
- 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 window to devices
- This Software even Supports Cross platform such as Pc to Pc, mobile to mobile Pc to mobile to Pc.
- Teamviewer 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.

Date : / /
Page No. 26

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. IfConfig

ifConfig Command is used to initalize 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 addr: 192.168.50.2 BCast: 192.168.50.255 Mask: 255.255.255.0
inet6 addr: 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 KiB)
Interrupt: 18 Base address: 0x2000

lo

Link encap: Local Loopback
inet addr: 127.0.0.1 Mask: 255.0.0.0
inet6 addr: ::1/128 Scope: HOST
UP Loopback RUNNING MTU: 16436 Metric: 1
RX Packets: 8 errors: 0 dropped: 0 overruns: 0 frame: 0

Tx Packets:8 errors:0 dropped:0 overruns:0 carries:0
Collisions:0 txqueuelen:0
Rx bytes:480 (480.0b) Tx bytes:480 (480.0b)

2. PING Command

PING (Packet INternet Gropher) Command is the best way to test Connectivity between two nodes. Whether it is Local Area Network (LAN) or wide Area Network (WAN). Ping use ICMP (Internet Control message protocol) to Communicate to other devices. You can Ping host name of 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 id info routing table information etc. To displays routing table information use option as -r

~~route~~/ netstat -r

Destination	Gateways	Genmask	flags	MSS	Windows	Irtt	Iface
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	UG	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.tecmint.com; <<>> DiG 9.8.2rc1-Red Hat 9.8.2-0.10.rc1.el6 <<>> www.tecmint.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 tecmint.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    Use Iface
192.168.50.0    *               255.255.255.0   U     0      0        0 eth0
link-local      *               255.255.0.0     U     1002   0        0 eth0
default         192.168.50.1    0.0.0.0         UG    0      0        0 eth0
```

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 tables. 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 permantly 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 use to configure a wireless network 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
  fedoraint.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 desk
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

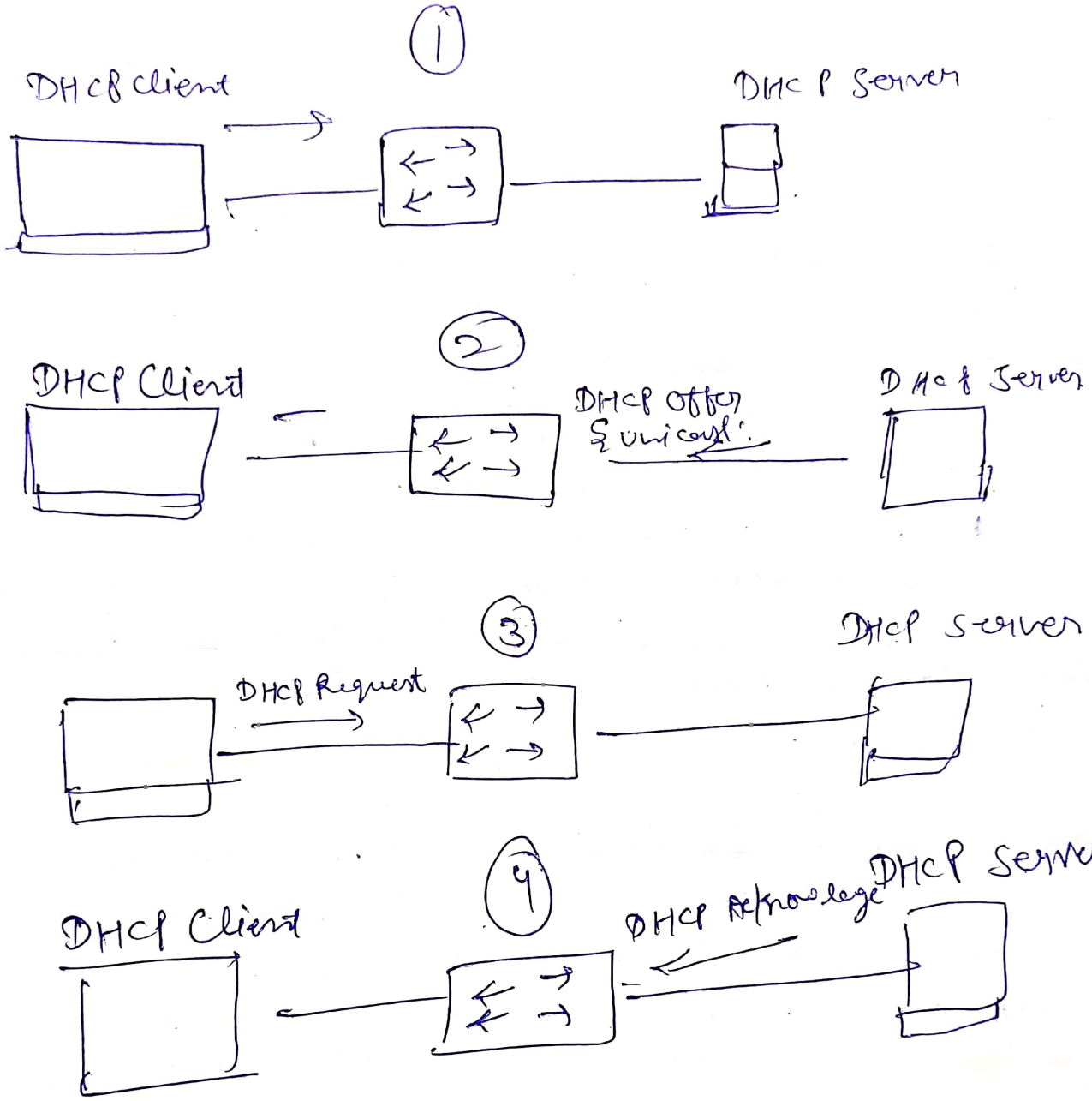
other functions used in disk management.

In this Section	
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 drive, Communications resource, mailslot and Pipe
Delete file	Deletes an existing file.

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

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

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


1
DHCP client
DHCP Server
2
DHCP Client
DHCP offer
& unicast.
DHCP Server
3
DHCP Request
DHCP server
4
DHCP Client
DHCP Acknowledge
DHCP Server

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 list the DHCP Serve.

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 ~~Se~~ceives the offer it requests the information officially Sending REQUEST message to Server this time by ~~mic~~ unicast.

4. Server Sends Acknowledge message confirming the DHCP leases to Client. Now client is allowed to use new IP Setting

DHCP server location. 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.