B.E.

Sixth Semester Examination, Dec-2008

Networking Programming (IT-302-E)

Note : Attempt any ***FIVE*** questions.

Q. 1. (a) Define computer network? List out various benefits of computer networking? How networking is performed in daily use? Explain with example.

Ans. Computer Network : The term "computer network" mean a collection of autonomous computers interconnected by a single technology. Two computers are said to be interconnected if they are able to exchange information. The connection need not be via a copper wire; fiber optics, microwave, infrared & communication satellites can also be used. Networks come in many sizes, shapes & forms.

Various Benefits of Computer Networking : Many organisations have a large number of computers in operation. These computers may be within the same building, campus, city or different cities. Even though the computers are located in different locations, the organizations want to keep tracks of inventories, monitor productivity, do the ordering & billing etc. The computer networks are useful to the organizations in the following ways :

(i) Resource sharing

(ii) For providing high reliability

(iii) To save money

(iv) It can provide a powerful communication medium.

Networking Performed in Daily Use : Starting in 1990s, the computer networks began to start delivering services to the private individual at home. The computer networks offer the following services to an individual person :

1. Access to remote information
2. Person to person communication
3. Interactive entertainment.

1. Access to Remote Information : Access to remote information involves interaction between a person & remote database. Access to remote information comes in many forms like :

Home shopping, paying telephone, electricity bills, e-banking, on the share market etc.

2. Person to Person Communication : Person to person communication includes the following :

Electronic-mail (e-mail), Real time e-mail, i.e., video conferencing allows remote users to communicate with no delay by seeing & hearing each other.

3. Interactive Entertainment : Interactive entertainment includes : Multiperson real-time simulation games; video on demand; participation in line T.V. programmes like quiz, contest, discussion, etc.

Q. 1. (b) Define Datagram? How it works? Explain format of the following datagram :

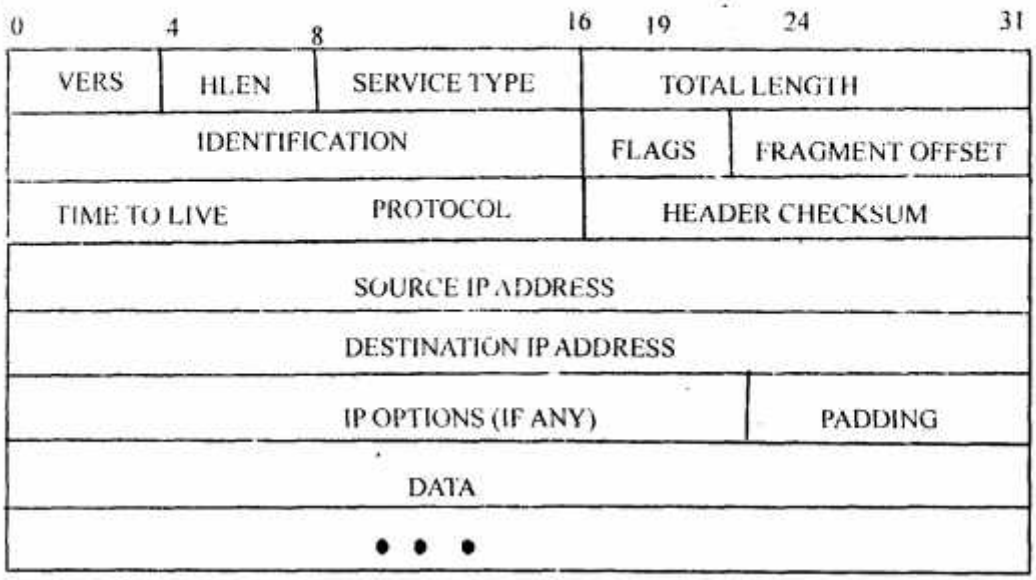
(i) IP datagram

(ii) UDP datagram

Ans. Datagram : The analogy between a physical network & a TCP/IP internet is strong. On a physical network, the unit of transfer is a frame that contains a header & data, where the header gives information such as the (physical) source & destination addresses. The internal calls its basic transfer unit an Internet datagram,

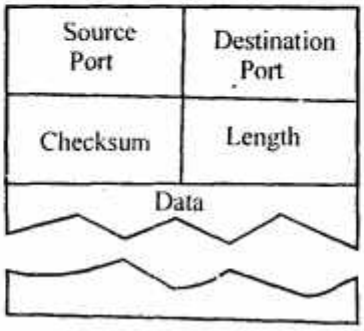
sometimes referred to as an IP datagram or merely a datagram. Like a typical physical network frame, a datagram is divided into header & data areas. Also like a frame, the datagram header contains the source & destination addresses & a type field that identifies the contents of the diagram. The differences, of course, is that the datagram header contains IP addresses whereas the frame header contains physical addresses.

(i) Layout of IP Datagram : The below figure shows the layout of IP datagram :



1. 4-bit field in a datagram (VERS) contains the version of the IP protocol that was used to create the datagram.
2. If header length field (HLEN), also 4-bits, gives the datagram header length measured in 32-bit words.
3. All fields in the header have fixed length except for the IP OPTIONS & corresponding PADDING fields.
4. The TOTAL LENGTH field gives the length of the IP datagram measured in octets, including octets in the header & data. The total length field in 16 bits long, the maximum possible size of an IP datagrams is 2^{16} or 65,535 octets.

(ii) Layout of UDP Datagram : The below figure shows the layout of UDP datagram :



The UDP header, port addressing is the primary function of UDP. The header is mostly port-addressing

fields. A checksum is used to detect corrupted packets. If an application requires more reliable service that this, TCP or supplemental protocols must be used.

The endpoint of a connection is called a socket. It consists of a host's IP address & a port number. Therefore, a socket identifies a specific application running on a specific computer & is the end point of a connection.

Q. 2. (a) What is ICMP? How it works? What is its purpose? Explain advantages and disadvantages of ICMP?

Ans. Internet Control Message Protocol (ICMP) : To allow routers in an internet to report errors or provide information about unexpected circumstances the designers added a special purpose message mechanism to the TCP/IP protocols. The mechanism, known as the Internet Control Message Protocol (ICMP), is considered a required part of IP & must be included in every IP implementation.

Like all other traffic, ICMP messages travel across the internet in the data portion of IP datagrams. The ultimate destination of an ICMP message is not an application program or user on the destination machine, however, but the Internet Protocol software on that machine. That is, when an ICMP error message arrives, the ICMP software, module handles it. If ICMP determines that a particular higher-level protocol or application program has caused a problem, it will inform the appropriate module.

ICMP is an error reporting mechanism. It provides a way for routers that encounters an error to report the error to the original source. Although the protocol specification outlines intended uses of ICMP suggests possible actions to take in response to error reports. ICMP does not fully specify the action to be taken for each possible error.

Most errors stem from the original source, but others do not. Because ICMP reports problems to the original source, however, it cannot be used to inform intermediate routers about problems. For example, suppose a datagram follows a path through a sequence of routers, R_1, R_2, ..., . If R_4 has incorrect routing information & mistakenly routes the datagram to router R_E. R_E cannot use ICMP to report the error back to router R_E, ICMP can only send a report back to the original source. The original source has no responsibility for the problem or control over the misbehaving routers. Infact, the source may not be able to determine which router caused the problem.

A datagram only contains fields that specify the original source & the ultimate destination, it does not contain a complete record of its trip through the internet. Furthermore, because routers can establish & change their own routing tables, there is no global knowledge of routes. Thus, when a datagram reaches a given router, it is impossible to know the path it has token to arrive there. If the router detects a problem, it cannot know the set of intermediate machines that processed the datagram, so it cannot inform them of the problem. Instead of silently discarding the datagram, the router uses ICMP to inform the original source that a problem has occurred, & trusts that host administrators will cooperate with network administrators to locate & repair the problem.

Q. 2. (b) What is Socket? Draw structure of Socket? Where Socket are used? Explain with example.

Ans. Socket : Sockets are mechanism that allows programs to communicate either on the same machine or across a network.

Most socket functions require a pointer to a socket address structure as an argument. Each supported protocol suite defines its own socket address structure. The names of these structure begin with sockaddr_ & end with a unique suffix for each protocol suite.

An IPV4 socket address structure, commonly called an "Internet socket address structure," is named sockaddr_in and is defined by including the <netinet/in.h> header. The below shows the POSIX definition :

```
Struct in_addr {
    in_addr_t s_addr;                    /* 32-bit IPV4 address */
    /* network byte ordered */
    },
    struct sockaddr_in {
      init 8_t sin_len;                  /*length of structure (16) */
     sa_family_t sin_family,             /* AF_INET */
      in-port_t  sin_port;               /* 16-bit TCP or UDP port number */
                                         /* network byte ordered */
   .struct in_addr sin_addr;             /*32-bit IPV4 address */
                                         /* network byte ordered */
      char sin_zero [8];                 /*  unused*/
};
```

The socket functions that pass a socket address structure from they process to the kernel, bind, connect, send to, & sending, all go through the sockarge function in a Berkeley-derived implementation. This function copies the socket address structure from the process & explicitly sets its sin_less member to the size of the structure that was passed as an argument to these functions. The socket functions that pass a socket address structure from the kernel to the process, accept, recuform, recumsg, getpername, & get socknames, all set the sin_len member before returning to the process.

Q. 3. What is Broadcasting? Where Broadcasting is used and how? How broadcasting is performed in mobile networks? List out various advantages and disadvantages of Broadcasting.

Ans. Broadcasting : Process by which a message is sent from a single host to all hosts on the network, without regard to the kind of data being sent or the destination of the data.

Usage of Broadcasting : Traditionally, broadcasting has not been a part of universal access and service (VAS), but is now regarded as part of ICTs, in particular as underlying technology and delivery mechanism between telecommunications and broadcasting are converging. First models of how to include broadcasting in VAS policies are explored.

Interestingly, there are fewer radio and television than telephony subscriptions in many regions of the world. In some cases this is due to the fact that free-to-air radio and TV does not require subscriptions, and the number of actual radio and TV users is much higher than subscription numbers imply. These numbers might increase as people take up phones that also support mobile radio or television services. The number of radios is higher than the number of televisions by a factor of two in many regions of the world, and the number of televisions is much higher than the number of personal computers in regions made up of developing countries. While broadcasting has been available for much longer time than the Internet figures for usage of broadcasting are not always comparable between countries because in many countries the number of radios relate only to stand alone radios and the number of subscribers may not accountfully for free-to-air users.

Broadcasting in Mobile Network : In mobile network the broadcast control channel is a continuous stream of output, from the base station containing the base station's identity and the channel status. All mobile stations monitor their signal strength to see when they have moved into a new cell. The dedicated control channel is used for location updating, registration and call setup. In particular, each base station maintains a database of mobile stations currently under its jurisdiction. Information needed to maintain this database is sent on the dedicated control channel.

Finally, there is a common control channel which is split up into three logical subchannels. The first of these subchannels is the paging channel, which the base station uses to announce incoming calls. Each mobile station monitors it continuously to watch for calls it should answer. The second is the random access channel, which allows users to request a slot on the dedicated control channel. If two requests colloide, they are garbled and have to be retried later. Using the dedicated control channel slot, the station can set up a call. The assigned slot is announced on the third subchannel, the access grant channel.

Advantages of Broadcasting :

1. Help to Know MAC Address : Broadcasting helps to know the MAC address of a system by sending a packet to its IP address, which returns its MAC address.

2. Faster Message Delivery : If same message is to be sent to all the users in the network then broadcasting is the best method for the this purpose.

Disadvantages of Broadcasting :

1. Wastage of Bandwidth : Broadcasting results in wastage of bandwidth.

2. Flooding : Broadcasting also results in network floading ie., the network is full of frames that may not be processed.

3. Congestion : Broadcasting also results in network congestion.

Q. 4. Explain the following Socket functions with example :

(i) I/O function **(ii) Select function**

(iii) Poll function.

How echo services are handled in TCP and UDP? Explain.

Ans. I/O Functions : I/O function is typically used in networking applications in the following scenarious :

When a client is handling multiple descriptors, I/O function should be used.

It is possible, but rare, for a client to handle multiple sockets at the same time.

If a TCP server handles both a listening socket & its connected sockets, I/O function is normally used.

If a server handles both TCP & UDP, I/O function is normally used.

If a server handles multiple services & perhaps multiple protocols, I/O function is normally used.

I/O function is not limited to network programming. Many non-trivial applications find a need for these techniques.

Types of I/O functions :

1. Blocking I/O.
2. Non-blocking I/O.
3. I/O function (select & poll).
4. Single driven I/O (SIGIO).
5. Asynchronous I/O (the POSIX aio_functions).

(ii) Select Function : This function allows the process to instruct the kernel to wait for any one of multiple events to occur & to wake up the process only when one or more of these events occurs or when a specified amount of time has passed.

That is, we tell the kernel what descriptors we are interested in & how long to wait. The descriptors in which we are interested are not restricted to sockets; any description can be tested using select.

* Include < sys/select.h>
* Include <sys/time.h>

  int select (int) maxfdp1, fd_set*readset, fd_set * writes ct, fd_set * exceptset ,

const strut timeval *timeout);

Returns; positive count of ready descriptors, 0 on timeout, –1 on error.

A timeval structure specifies the number of second & microseconds.

```
struct tiemval {
long tv–sec;            /* seconds */
logn tv-user;           /*microseconds */
};
```

There are three possibilities :

1. Wait Forever : Return only when one of the specified descriptors in ready for I/O. For this, we specify the timeout argument as a null pointer.

2. Wait up to a Fixed Amount of Time : Return when one of the specified descriptors is ready for I/O, but do not wait beyond the number of seconds & microseconds specified in the timeval structure pointed to by the timeout argument.

3. Do not Wait at All : Return immediately after checking the descriptors. This is called palling. To specify this, the time out argument must point to a timeval structure & the timer value must be O.

(iii) Pool Function : The poll function originated with SVR3 & was originally limited to STREAMS devices. SVR4 removed this limitation, allowing poll to work with any descriptor. Poll provides functionality that is similar to select, but poll provides additional information when dealing with STREAMS devices.

```
#include <poll.h>
int poll (struct pollfd *fdarray, unsigned long nfds, int timeout);
```

Returns : count of ready descriptions, 0 ontimeout, –1on error.

The first argument is a pointer to the first element of an array of structures. Each element of the array is a pollfd structure that specifies the conditions to be tested for a given descriptors, fd.

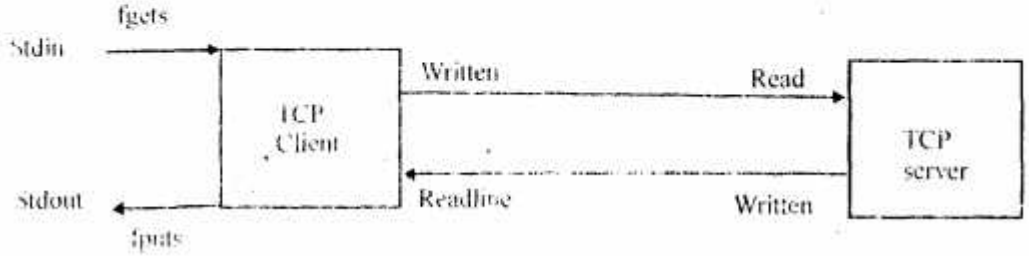
The conditions to be tested are specified by the events member & the function returns the status for that descriptor in the corresponding revents number. The constants used to specify the events flag & to test the revents flag against.

Constant	Input to events?	Results from revents?	Description
POLLIN	•	•	Normal on priority band data can be read
POLLRDNORM	•	•	Normal data can be read
POLLRDBAND	•	•	Priority band data can be read
POLLPRI	•	•	High-priority data can be read
POLLOUT	•	•	Normal data can be written
POLLWRNORM	•	•	Normal data can be written
POLLWRBAND	•	•	Priority band data can be written
POLLERR		•	Error has occurred
POLLHUP		•	Error has occured
POLLNVAL		•	Descriptor is not an open file

Echo Services Handled in TCP & UDP : An echo server performs the following steps :

1. The client reads a line of text from its standard input & writes the line to the server.
2. The server reads the line from its network input & echoes the line back to the client.
3. The client reads the echoed line prints it on its standard output.

The below figure depicts this simple client server along with the functions used for input & output.



We show two arrows between client & server, but this is really one full-duplex TCP connection. The fgets & fputs functions are from the standard I/O library & the written and readline functions.

While we will develop our own implementation of an echo server, most TCP/IP implementations such a [illegible] using both TCP & UDP. We will also use this servers with our own client.

TCP Echo Server :

1. Create Socket, Bind Server's Well-known Port : A TCP socket is created. An Internet socket address structure is filled in with wildcard address & the server's well-known port.

2. Wait for Client Connection to Complete : The server blocks in the call to accept, waiting for a client connection to complete

3. Read a Buffer & Echo the Buffer : Read reads data from the socket & the line is echoed back to the client by written.

4. Create Socket, Fill in Internet Socket Address Structure : A TCP socket is created & an Internet socket address structure is filled in with the server's IP address & port number.

5. Connect to Server : Connect establishes the connection with the server.

6. Read a Line, Write to Server : fgets reads a line of text & written sends the line to the server.

7. Read Echoed Line from Server, Write to Standard Output : Readline reads the line echoed back from the server & fputs writes it to standard output.

8. Return to Main : The loop terminates when fgets returns a null pointer, which occurs when it encounters either an end-of-file (EOF) or an error.

Q. 5. What is server? How it works? How servers are different from TCP/IP client server? Explain in brief various iterative and concurrent servers.

Ans. Server : A computer or a software package, that provides a specific kind of service to client software running on other computers. The term can refer to a particular piece of software, such as a WWW server, or to the machine on which the software is running e.g., "Our mail server is down today, that's why e-mail isn't getting out." A single server machine can have several different server software packages running on it, thus providing many different servers to client on the network. Sometimes server software is designed so that additional capabilities can be added to the main program by adding small programs known as servlets.

The client-server paradigm uses the direction of initiation to categorize whether a program is a client or server. In general, an application that initiates peer-to-peer communication is called a client. End users usually invoke client software when they use a network service. Most client software consists of conventional application programs. Each time a client application executes, it contacts a server, sends a request & a waits a response. When the response arrives, the client continues processing, clients are often easier to build than servers & usually require no special system privileges to operate.

By comparison, a server is any program that waits for incoming communication request from a client. The server receives a client's reqest, performs the necessary computation & returns the result to the client.

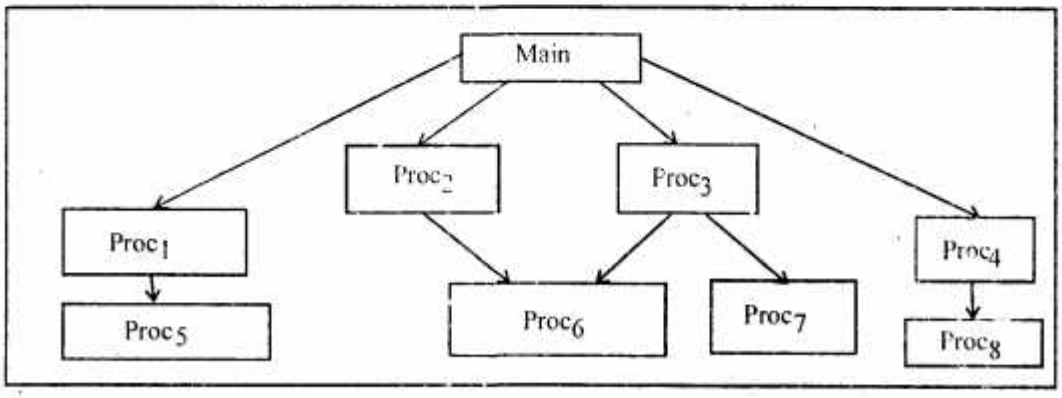
The terms concurrent server refers to a server that handles multiple request at one time, not to the underlying implementation uses multiple con-current process. In general, con-current servers are more difficult to design & build & the resulting code is more complex & difficult to modify.

The term interactive server refers to a server that process one request at a time. Iterative servers cause unnecessary delays in distributed applications & can become a performance bottleneck that affects many client applications.

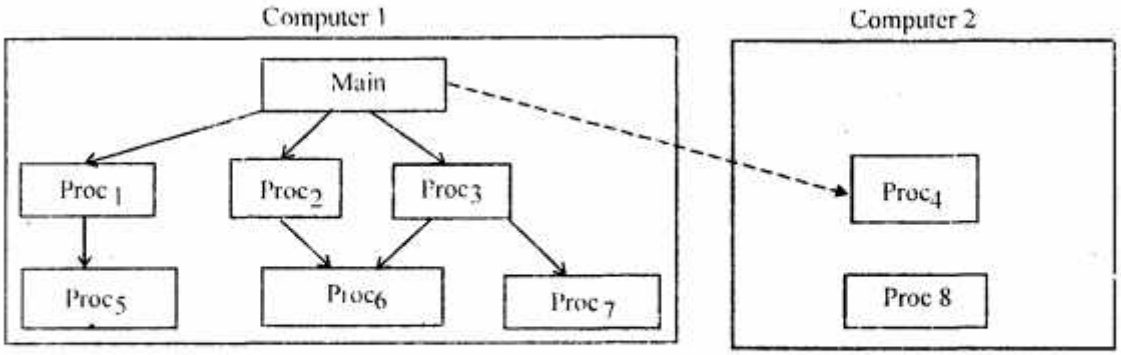
"Iterative server implementation which are easier to build & understand, may result in poor performance because they make clients wait for services. In contrast, concurrent server implementations, which are more difficult to design & build, yield better performance."

Q. 6. (a) Explain RPC with its models? How RPC of client and server is different in terms of analogy?

Ans. A Conceptual Model : The remote procedure call model draws heavily from the procedure call mechanism found in conventional programming languages. Procedures offer a powerful abstraction that allows programmers to divide programs into small, manageable, easily-understood pieces. Procedures are especially useful because they have a straight-forward implementation that provides a conceptual model of program execution. The below figure illustrates the concept :



An Extension of the Procedural Model : The remote procedure call model uses the same procedural abstraction as or conventional program, but allows a procedure call to span the boundary between two computers. The below figure shows how the remote procedure call paradigm can be used to divide a program into two pieces that each execute on a separate computer. Of course, a conventional procedure call cannot pass from one computer to another. Before a program can use remote procedure calls, it must be augmented with protocol software that allows it do communicate with the remote procedure.

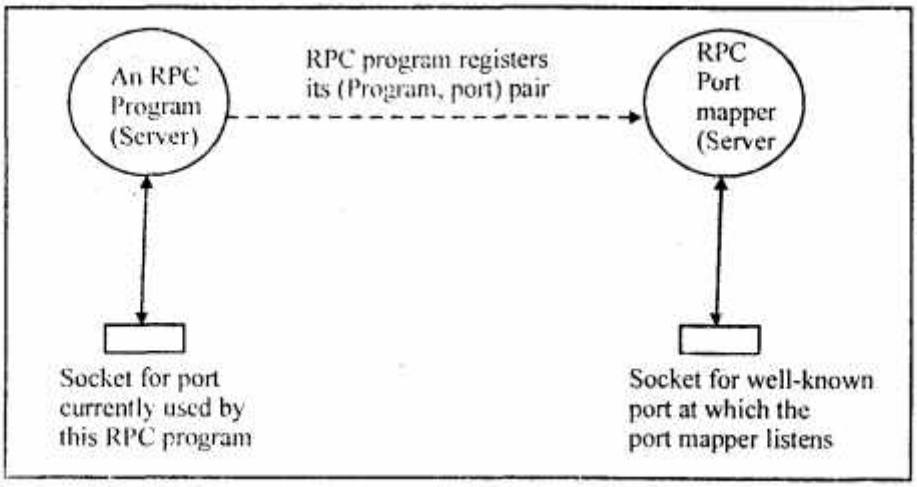


The analogy between remote procedure call & client-server interaction does not explain all the details. For example, we know that a conventional procedure remains completely inactive until the flow of control passes to it. In contrast, a server process must exist in the remote system & be waiting to compute a response before it receives the first request from a client. Further differences arise in the way data flows to a remote procedure conventional procedures usually accept a few arguments & return only a few results. However, a server can accept or return arbitrary of data.

Although it would be ideal if local & remote procedure calls behaved identically, several practical constraints prevent it. First, network delays can make a remote procedure call several orders of magnitude more expensive than a conventional procedure call. Second, because the called procedure operates in the same address space as the calling procedure, conventional programs can pass pointers as arguments. A remote procedure call cannot have pointers as arguments because the remote procedure operates in a completely different address space than the caller. Thrid, because a remote procedure does not share the callers environment, it does not have direct access to the caller's I/O descriptors or operating system functions. For example, a remote procedure cannot write error messages directly to the caller's standard error file.

Q. 6. (b) Explain, how Dynamic port mapping performed in RPC.

Ans.



To solve the port identification problem, a client must be able to map from an RPC program number & machine address to the protocol port that the server obtained on the destination machine when it started. The mapping must be dynamic because it can change if the machine reboots on if the RPC program starts execution again.

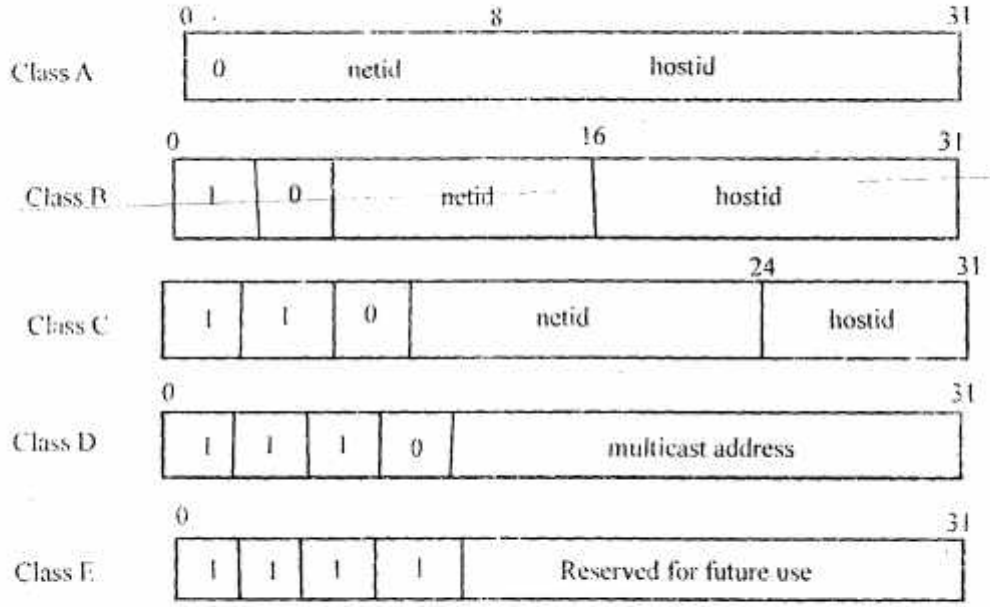
To allow clients to contact remote programs, the sun RPC mechanism includes a dynamic mapping service. Each machine that offers an RPC program maintains a database of port mappings & provides a mechanism that allows a caller to map RPC program numbers to protocol ports called the RPC port mapper or sometimes simply the port mapper, the RPC port mapping mechanism uses a server to maintain a small database on each machine. The below figure shows that the port mapper operates as a separate server process :

Q. 7. What is internet? How internet addresses are classified? How these address are resoluted? Explain.

Ans. Internet : An Internet is a network of networks, which consists of millions of computers connected with each other all over the world.

Internet Addresses (IP) address are classified as :

1. Classful Addresses : Each host on the internet is assigned a 32-bit integer address called its internet address or IP address. Each address is a pair (netid, hostid) where network is identified by netid and hostid identifies a host on that network. Each IP address had one of the three forms i.e., Class A, B or C.



2. Classless Addressing : The classless scheme called CIDR {Classless Inter-Domain Routing} was implemented in the early 1990s to help overcome an impending address shortage. Organizations were supposed to return their class-based addresses to the address pool and receive a continguous set of classless addresses is return. This wasn't entirely successful since many organizations were not willing to give up their address assignments, especially since it meant renumbering internal networks.

They are two ways to resolver the addresses :

1. Resolution Through Direct Mapping : This solution is used for proNET token ring network. In proNET administrator assigns IP addresses with the hostid portion equal to 1, 2, 3 & so on and while installing network interface hardware, selects & physical address that corresponds to the IP address. Conceptually, choosing a numbering scheme that makes address resolution efficient means selecting a function 'f' that maps IP addresses to physical addresses. Resolving IP address I_A means computing :

$$\boxed{P_A = f(I_A)}$$

Here computation of 'f' must be efficient.

2. Resolution Through Dynamic Binding : This solution is used for Ethernet technology. In this solution a low level protocol termed as ARP {Address Resolution Protocol} is used to bind address dynamically. The ARP protocol provides a mechanism that is both efficient and easy to maintain.

Q. 8. Explain the following :

(i) Routing Sockets **(ii) RARP**

(iii) RPC Transmission **(iv) Address Conversions.**

Ans. (i) Routing Sockets : Routing sockets are created with domain AF_ROUTE and type SOCK_RAW. This type of socket is used with the following system calls ; socket( ), close( ), write( ), and setsockopt( ). A routing socket is a special type of socket that is not specific to any particular network protocol, but allows "privileged" user processes to write information into the kernel. User processes use this type of socket to add & remove information from the routing table. This is done by filling in the rt_msghdr structure & passing it to write( ). The rt_msghdr structure is defined in $PDIR/include/route.h.

The following values of he rtm_type field of the rt_msghdr structure are supported :

* **RTM_ADD :** Add an entry to the routing table;

* **RTM_DELETE :** Delete an entry from the routing table;

* **RTM_CHANGE :** Change an entry in the routing table. The implementation of this command is equivalent to performing an RTM_DELETE followed by an RTM_ADD.

These constants are defined in $PDIR/include/route.h.

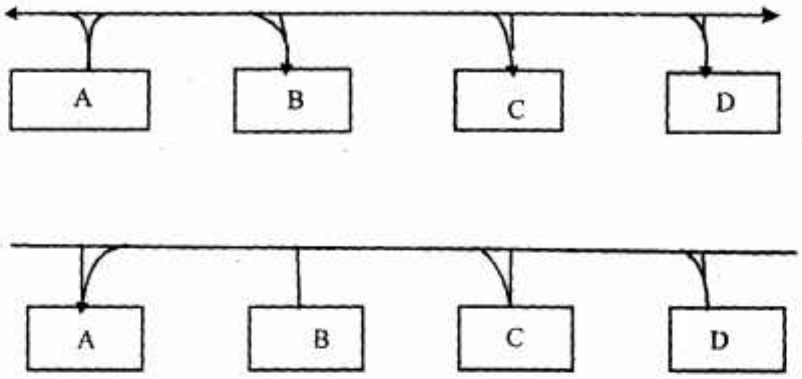
To look up a route within the kernel, call rt_lookup_dest( ). This function will return the address of the gateway to which a packet with the given destination address should be sent. It also returns the index of the interface over which the packet should be sent. Note that the current implementation of the simulator supports only host routes, there is no support for network masks or prefix matching. If there is no host route available in the routing table, the function returns the default gateway. The default gateway route is defined as a route that has a destination address of 0.0.0.0. It can be inserted into the routing table in the same way as host routes.

(ii) Reverse Address Resolution Protocol (RARP) : The TCP/IP protocol that allows a computer to obtain its IP address from a server is known as the Reverse Address Resolution Protocol.

Like an ARP msg, a RARP msg; is sent from one machine to another encapsulated in the data portion of a network frame. The frame type contains the value 8035_{16} to identity the contents of the frame as a RARP msg. The data portion of the frame contains the 28-Octet RARP msg.

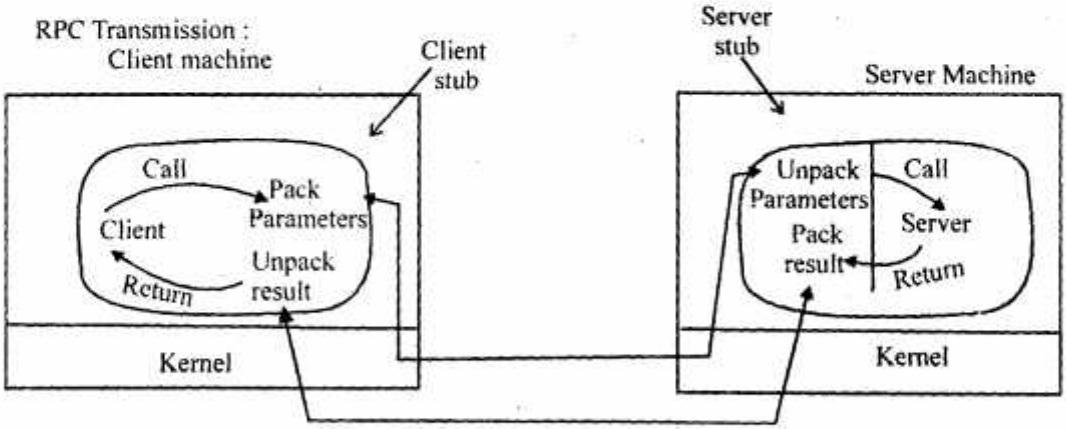
The sender broadcasts a RARP request that specifies itself as both the sender & target machine & supplies its physical network address in the target hardware address field. All computers on the network

receive the request, but only those authorized to supply the RARP service process the request & send a reply, such computers are known as RARP servers.



Servers answer requests by filling in the target protocol address field, changing the msg type from request to reply & sending the reply back directly to the machine making the request. The original machine receives replies from all RARP servers, even though only that first as needed.

(iii) RPC Transmission :



Remote procedure call occurs in the following steps :

1. The client procedure calls the client stub in the normal way.
2. The client stub builds a message & traps to the kernel.
3. The kernel sends the message to the remote kernel.
4. The remote kernel gives the message to the server stub.
5. The server stub unpacks the parameters & calls the server.
6. The server does the work & returns the result to the stub.
7. The server stub packs it in a message & traps to the kernel.
8. The remote kernel sends the message to the clients kernel.

9. The clients kernel gives the message to the client stub.
10. The stub unpacks the result & returns to the client.

(iv) Address Conversions : The DNS is used primarily to map between host names & IP address. A hostname can be either a simplename, such as salaries or feedsd, on a fully qualified domain name (FQDN), such as salaries, unpbook.com.

Entries in the DNS are known as resource records (RRs). There are only a few types of RRs that we are interested in.

A : An A record maps a hostname into a 32-bit IPV4 address. For example, here are the four DNS records for the host freebsd in the unpbook.com domain.

AAAA : A AAAA record, called a "quad A" record, maps a hostname into a 128-bit IPV6 address. The term "quad A" was chosen because a 128-bit address is four times larger than a 32-bit address.

PTR : PTR records map IP addresses into hostnames. For an IPV4 address, the 4 bytes of the 32-bit address are reversed, each byte is converted to its decimal ASCII value (0–255), & in-addr. arpa is then appended. The resulting string is used in the PTR query.

MX : An MX record specifies a host to act as a "mail exchanger" for the specified host. In the example for the host freebsd above, two MX records are provided. The first has a preference value of 5 & the second has a preference value of 10. Wher. multiple MX records exist, they are used in order of preference, starting with the smallest value.

CNAME : CNAME stands for "canonical name." A common use is to assign CNAME records for common services, such as ftp & www. If people use these service names instead of the actual hostnames, it is transparent when a service is moved to another host.