B.E.

Sixth Semester Examination, 2010

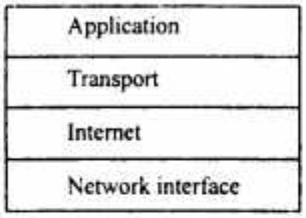
Network Programming (IT-302-E)

Note : Attempt any five questions. All questions carry equal marks.

Q. 1. What is protocol? Describe TCP/IP protocol architecture. Also discuss its merits and demerits.

Ans. Protocol : Protocol is a standard procedure and format that two data communication device must understand, accept and use to be able to talk to each other.

TCP/IP Architecture :



(i) Application Layer :

(a) Provide services that can be used by other applications.

(b) Incorporate the functions of top 3 OSI layers.

(ii) Transport Layer :

(a) Application layer directly run over the transport layer, corresponding to OSI transport layer.

(b) Two kinds of services : TCP & UDP

(c) TCP reliable connect-oriented transfer of a byte stream.

(d) UDP best-effort connectionless transfer of individual messages.

(iii) Internet Layer :

(a) Transfer of information across networks through gateways/routers.

(b) Corresponding to OSI network layer : routing and congestion control.

(c) Global unique IP address and IP packets.

(d) Best-effort connectionless IP packet transfer : no-setup, routed independently, robust, out of order, duplicate, or lose of packet.

(iv) Network Interface Layer :

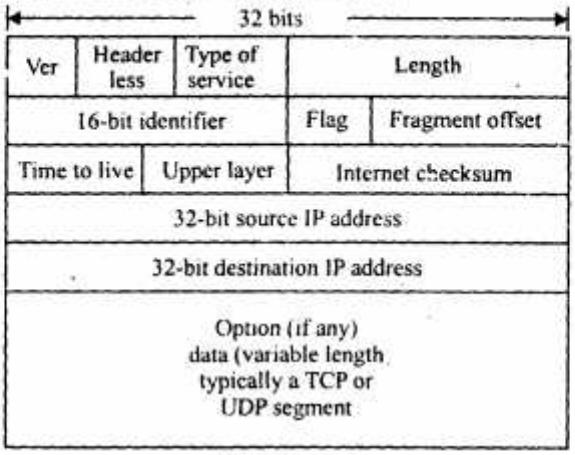
(a) Concerned with network-specific aspects of the transfer of packets.

(b) Corresponding to part of OSI network layer and data link layer.

(c) Different network interfaces : X.25 ATM, frame relay, ethernet etc.

Q. 2. What do you mean by datagram? Draw the IP datagram format. Discuss its various features. How it is different from other datagram? Discuss its merits and demerits.

Ans. Datagram : A datagram is a basic transfer unit associated with a packet-switched network in which the delivery arrival time and order are not guaranteed.



IP datagram

(i) The IP datagram (or Internet datagram) is the basic information unit :

(a) Header area (b) Data area

(ii) IP datagram is transported from one network to another.

(a) Encapsulated in the network frame within a particular network.

(iii) IP allows its datagrams to be fragmented.

(a) Once a datagram is fragmented its fragments travel as separate datagrams all the way to the final destination.

IP Datagram Fields :

(i) VERS : Specifies IP protocol version in use.

(ii) IHL : IP datagram header length.

(iii) Total Length : Specifics total length (data + header)

(iv) Type of Service (TOS) : Limit to network elements.

(v) Identification : Contains a unique integer which identifies the datagram.

(vi) Flags : Contains a do not fragment bit and more fragment bit, the third bit is unused.

(vii) Fragment Offset : Specifies the offset of this fragment into the original datagram.

Q. 3. What is Echo? How echo came in TCP and UDP? How to differentiate Echo services in TCP and UDP? How echo services are activated and deactivated?

Ans. The echo protocol is a service in the internet protocol suite defined in RFC862. It was originally proposed for testing and measurement of round-trip times in IP networks.

A host may connect to a server that supports the echo protocol using the Transmission Control Protocol (TCP) or the User Datagram Protocol (UDP) on the well-known port number 7. The server sends back a copy of the identical data it received.

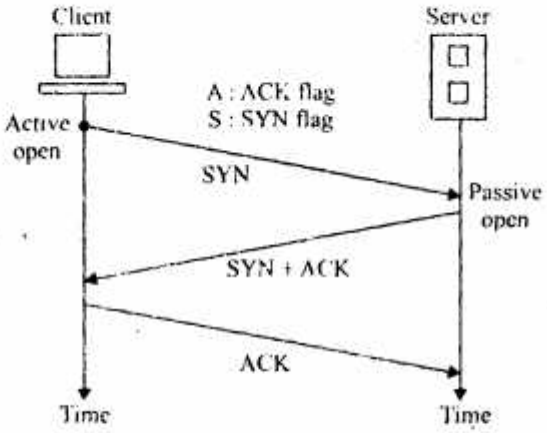
Echo Request and Reply : The echo-request and echo-reply messages are designed for diagnostic purposes. Network managers and users utilize this pair of messages to identify network problems. The combination of echo-request and echo-reply messages determines whether two systems can communicate with each other. The echo-request and echo-replay messages can be used to determine if there is communication at the IP level. Because ICMP messages are encapsulated in IP datagrams, the receipt of an echo-reply message by the machine that sent the echo request is proof that the IP protocols in the sender and receiver are communicating with each other using the IP datagram. Also, it is proof that the intermediate routers are receiving, processing and forwarding IP datagrams. Today, most systems provide a version of the ping command that can create a series of echo-request and echo-reply messages, providing statistical information.

Q. 4. What do you mean by server? Discuss TCP client server. What is role of TCP in server and client? Discuss their separate function with diagram. How the connection takes place between client and server? Explain with example.

Ans. Server : Computer or processes that manage network resources like disk drives, printers, network traffic etc.

TCP is connection-oriented. A connection-oriented transport protocol establishes a virtual path between the source and destination. All the segments belonging to a messages are than sent over this virtual path.

Connection Establishment :



(i) The client sends the first segment, a SYN segment, in which only the SYN flag is set. This segment is for synchronization of sequence numbers. It consumes one sequence number. When the data transfer starts, the sequence number is incremented by 1. We can say that the SYN segment carries no real data, but we can think of it as containing 1 imaginary byte.

(ii) The server sends the second, a SYN + ACK segment, with 2 flag bits set : SYN and ACK. This segment has a dual purpose. It is a SYN segment for communication in the other direction and serves as the acknowledgement for the SYN segment. It consumes one sequence number.

(iii) The client sends the third segment. This is just an ACK segment. It acknowledge the receipt of the second segment with the ACK flag and acknowledgement number field. Note that the number in this

segment is the same as the one in the SYN segment, the ACK segment does not consume any sequence numbers

Q. 5. (a) What is RPC? Why the need of RPC arise in networking? Discuss its model in respect of client and server.

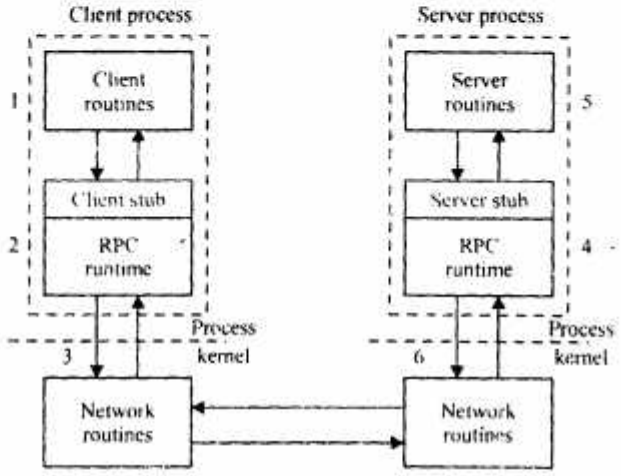
Ans. Remote Procedure Calls (RPC) :

(i) Avoid explicit message exchange between processes.

(ii) Basic idea is to allow a process on a machine to call procedures on a remote machine.

(iii) Make a remote procedure possibly look like a local one.

RPC : The basic mechanism



(i) Client calls a local procedure on the client stub

(ii) The client stub acts as a proxy and marshalls the call and the args.

(iii) The client stub send this to the remote system (via. TCP/UDP).

(iv) The server stub unmarshalls the call and args from the client.

(v) The server stub calls the actual procedure on the server.

(vi) The server stub marshalls the reply and sends it back to the client.

Steps of a Remote Procedure Call (RPC) :

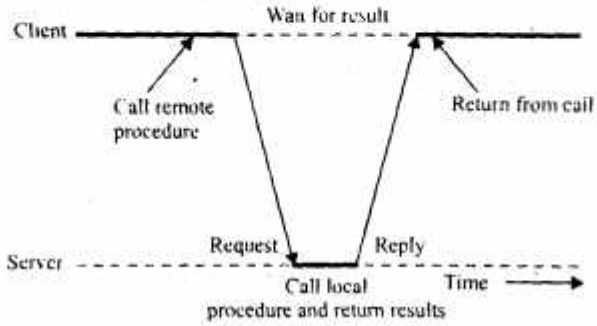


Fig. Principle of RPC between a client and server program

(i) Client procedure calls client stub in normal way.

(ii) Client stub builds message, calls local OS.

(iii) Client's OS sends message to remote OS.

(iv) Remote OS gives message to server stub.

(v) Server stub unpacks parameters, call server.

(vi) Server does work, returns result to the stub.

(vii) Server stub packs it in message, calls local OS.

(viii) Server's OS sends message to client's OS.

(ix) Client's OS gives message to client stub.

(x) Stub unpacks result; returns to client.

Q. 5. (b) Explain Dynamic port mapping with example.

Ans. The data link layer is responsible for delivery of frames between two neighbouring nodes over a link. This is called node-to-node delivery. The network layer is responsible for delivery of datagrams between two hosts. This is called host-to-host delivery. Communication on the internet is not defined as the exchange of data between two nodes or between two hosts. Real communication takes place between two processes. We need process-to-process delivery :

(i) At the network layer, we need an IP address to choose one host among millions. A datagram in the network layer needs a destination IP address for delivery and a source IP address for the destination's reply.

(ii) At the transport layer, we need a transport layer address, called a post number, to choose among multiple process running on the destination host. The destination port number is needed for delivery; the port number is needed for the reply.

Q. 6. What do you mean by Authentication? How Authentication takes place in networking? Discuss its various types with example.

Ans. Authentication :

(i) Positive verification of identify (man or machine)

(ii) Verification of a periods claimed identify

(iii) Who are you? Prove it

(iv) 3 categories :

(a) What you know?

(b) What you have?

(c) Who you are?

(a) What you know :

— Password — Passphrase

— PIN

(b) What you have :

— Digital authentication — Physical devices to aid authentication

(c) Who you are :

— Biometric authentication

— Use of a biometric reading to confirm that a person is who he/she claims to be.

— Biometric reading

— A recording of some physical or behavioural attribute of a person.

Network Authentication : Network authentication confirms the user's identification so any network service that the user is attempting to access. To provide this type of authentication, the security system supports many different authentication mechanisms, including *VS*, secure socket layer/transport layer security (SSL/TLS) and for compatibility with windows NT4.0, NTLM.

Users who use a domain account do not see network authentication Users who use a local computer account must provide credentials every time they access a network resource. By using the domain account, the user has credentials that can be used for single-sign-on.

Q. 7. What is network? How file system works on network? What is specific role of file system at DLL? Discuss its advantages and disadvantages. Discuss various Debugging Technique.

Ans. Network : A network has been defined as "any set of interlinking lines resembling a net, a network of roads or an interconnected system, a network of alliance". This definition suits our purpose will : a computer network is simply a system of interconnected computers. How they are connected is irrelevant.

DLL (Dynamic Link Library) : In a nut shell, a dynamic link library (DLL) is a collection of small programs, which can be called upon when needed by the executable program (EXE) that is running. The DLL lets the executable communicate with a specific device such as a printer or may contain source code to do particular functions.

An example would be if the program (exe) needs to get the free space of your hard drive. It can call the DLL file that contains the function with parameters and a call function. The DLL will then tell the executable the free space. This allows the executable to be smaller in size and not have to write the function has already exists.

This allows any program the information about the free space, without having to write all the source code and it saves space on your drive as well. When a DLL is used in this fashion are also known as shared files.

The advantage of DLL files is that, because they do not get loaded into RAM together with the main program, space is saved in RAM. When and if a DLL file is called, then it is loaded. For example, you are editing a MS word document, the printer DLL file does not need to be loaded into RAM. If you decide to print the document then the printer DLL file is loaded and a call is made to print.

All in all a DLL is an executable file that cannot run on its own, it can only from inside an executable file. This would be like having a car without an engine, where as an executable has an engine.