**Date Communication & Networks**

**Q-1 Write a short note on network layer.**

* The network layer is the third level of the Open Systems Interconnection Model (OSI Model) and the layer that provides data routing paths for network communication. Data is transferred in the form of packets via logical network paths in an ordered format controlled by the network layer.   
    
  Logical connection setup, data forwarding, routing and delivery error reporting are the network layer’s primary responsibilities.
* The network layer is considered the backbone of the OSI Model. It selects and manages the best logical path for data transfer between nodes. This layer contains hardware devices such as routers, bridges, firewalls and switches, but it actually creates a logical image of the most efficient communication route and implements it with a physical medium.
* Network layer protocols exist in every host or router. The router examines the header fields of all the IP packets that pass through it.  
    
  Internet Protocol and Netware IPX/SPX are the most common protocols associated with the network layer.  
    
  In the OSI model, the network layer responds to requests from the layer above it (transport layer) and issues requests to the layer below it (data link layer).

## Network Layer Functionalities

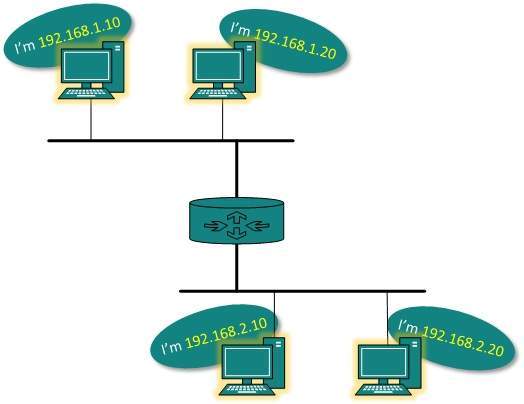
* Devices which work on Network Layer mainly focus on routing. Routing may include various tasks aimed to achieve a single goal. These can be:
* Addressing devices and networks.
* Populating routing tables or static routes.
* Queuing incoming and outgoing data and then forwarding them according to quality of service constraints set for those packets.
* Internetworking between two different subnets.
* Delivering packets to destination with best efforts.
* Provides connection oriented and connection less mechanism.

## Network Layer Features

* With its standard functionalities, Layer 3 can provide various features as:
* Quality of service management
* Load balancing and link management
* Security
* Interrelation of different protocols and subnets with different schema.
* Different logical network design over the physical network design.
* L3 VPN and tunnels can be used to provide end to end dedicated connectivity.
* Internet protocol is widely respected and deployed Network Layer protocol which helps to communicate end to end devices over the internet. It comes in two flavors. IPv4 which has ruled the world for decades but now is running out of address space. IPv6 is created to replace IPv4 and hopefully mitigates limitations of IPv4 too.

**Network Addressing:**

* Layer 3 network addressing is one of the major tasks of Network Layer. Network Addresses are always logical i.e. these are software based addresses which can be changed by appropriate configurations.
* A network address always points to host / node / server or it can represent a whole network. Network address is always configured on network interface card and is generally mapped by system with the MAC address (hardware address or layer-2 address) of the machine for Layer-2 communication.
* There are different kinds of network addresses in existence:
* IP
* IPX
* AppleTalk
* We are discussing IP here as it is the only one we use in practice these days.

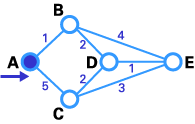


* IP addressing provides mechanism to differentiate between hosts and network. Because IP addresses are assigned in hierarchical manner, a host always resides under a specific network.The host which needs to communicate outside its subnet, needs to know destination network address, where the packet/data is to be sent.
* Hosts in different subnet need a mechanism to locate each other. This task can be done by DNS. DNS is a server which provides Layer-3 address of remote host mapped with its domain name or FQDN. When a host acquires the Layer-3 Address (IP Address) of the remote host, it forwards all its packet to its gateway. A gateway is a router equipped with all the information which leads to route packets to the destination host.
* Routers take help of routing tables, which has the following information:
* Method to reach the network
* Routers upon receiving a forwarding request, forwards packet to its next hop (adjacent router) towards the destination.
* The next router on the path follows the same thing and eventually the data packet reaches its destination.
* Network address can be of one of the following:
* Unicast (destined to one host)
* Multicast (destined to group)
* Broadcast (destined to all)
* Anycast (destined to nearest one)
* A router never forwards broadcast traffic by default. Multicast traffic uses special treatment as it is most a video stream or audio with highest priority. Anycast is just similar to unicast, except that the packets are delivered to the nearest destination when multiple destinations are available.

**Q-2 Explain Dijkstra’s Routing Algorithm in detail.**

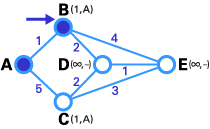
* **Dijkstra's algorithm** is an [algorithm](https://en.wikipedia.org/wiki/Algorithm) for finding the [shortest paths](https://en.wikipedia.org/wiki/Shortest_path_problem) between [nodes](https://en.wikipedia.org/wiki/Vertex_%28graph_theory%29) in a network [graph](https://en.wikipedia.org/wiki/Graph_%28abstract_data_type%29). It was developed by [computer scientist](https://en.wikipedia.org/wiki/Computer_scientist) [Edsger W. Dijkstra](https://en.wikipedia.org/wiki/Edsger_W._Dijkstra) in 1956 and it was published three years later.
* As we know Dijkstra's algorithms use to find the shortest path between two nodes. For this from the single node (the "source" node)it finds shortest paths from the source to all other nodes in the graph, producing a [shortest-path tree](https://en.wikipedia.org/wiki/Shortest-path_tree).
* For a given source vertex (node) in the graph, the algorithm finds the path with lowest cost (i.e. the shortest path) between that vertex and every other vertex. It can also be used for finding costs of shortest paths from a single vertex to a single destination vertex by stopping the algorithm once the shortest path to the destination vertex has been determined. For example, if the vertices of the graph represent cities and edge path costs represent driving distances between pairs of cities connected by a direct road, Dijkstra's algorithm can be used to find the shortest route between one city and all other cities. As a result, the shortest path algorithm is widely used in network routing protocols, most notably IS-IS and OSPF (Open Shortest Path First).

**Example:**



* Here we want to find the best route between A and E (see below). You can see that there are six possible routes between A and E (ABE, ACE, ABDE, ACDE, ABDCE, ACDBE), and it's obvious that ABDE is the best route because its weight is the lowest. But life is not always so easy, and there are some complicated cases in which we have to use algorithms to find the best route.

**Step 1:**



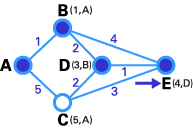
* As you see in the first image, the source node (A) has been chosen as T-node, and so its label is permanent (we show permanent nodes with filled circles and T-nodes with the --> symbol).

**Step 2:**



* In the next step, you see that the status record set of tentative nodes directly linked to T-node (B, C) has been changed. Also, since B has less weight, it has been chosen as T-node and its label has changed to permanent (see below).

**Step 3:**



* In step 3, like in step 2, the status record set of tentative nodes that have a direct link to T-node (D, E), has been changed. Also, since D has less weight, it has been chosen as T-node and its label has changed to permanent.

**Step 4:**

* In step 4, we don't have any tentative nodes, so we just identify the next T-node. Since E has the least weight, it has been chosen as T-node.
* Lastly, E is the destination, so we stop here.

**Q-3 Explain Data Link Layer in detail.**

* The data link layer is used for the encoding, decoding and logical organization of data bits. Data packets are framed and addressed by this layerThe data link layer's first sublayer is the **media** Data link layer hides the details of underlying hardware and represents itself to upper layer as the medium to communicate.
* Data link layer works between two hosts which are directly connected in some sense. This direct connection could be point to point or broadcast. Systems on broadcast network are said to be on same link. The work of data link layer tends to get more complex when it is dealing with multiple hosts on single collision domain.
* Data link layer is responsible for converting data stream to signals bit by bit and to send that over the underlying hardware. At the receiving end, Data link layer picks up data from hardware which are in the form of electrical signals, assembles them in a recognizable frame format, and hands over to upper layer.
* Data link layer has two sub-layers:
  + **Logical Link Control:** It deals with protocols, flow-control, and error control
  + **Media Access Control:** It deals with actual control of media

**Functionality of Data-link Layer**

* Data link layer does many tasks on behalf of upper layer. These are:
* **Framing**

Data-link layer takes packets from Network Layer and encapsulates them into Frames.Then, it sends each frame bit-by-bit on the hardware. At receiver’ end, data link layer picks up signals from hardware and assembles them into frames.

* **Addressing**

Data-link layer provides layer-2 hardware addressing mechanism. Hardware address is assumed to be unique on the link. It is encoded into hardware at the time of manufacturing.

* **Synchronization**

When data frames are sent on the link, both machines must be synchronized in order to transfer to take place.

* **Error Control**

Sometimes signals may have encountered problem in transition and the bits are flipped.These errors are detected and attempted to recover actual data bits. It also provides error reporting mechanism to the sender.

* **Flow Control**

Stations on same link may have different speed or capacity. Data-link layer ensures flow control that enables both machine to exchange data on same speed.

* **Multi-Access**

When host on the shared link tries to transfer the data, it has a high probability of collision. Data-link layer provides mechanism such as CSMA/CD to equip capability of accessing a shared media among multiple Systems.

**Q-4 Write a short note on error detection and correction.**

* There are many reasons such as noise, cross-talk etc., which may help data to get corrupted during transmission. The upper layers work on some generalized view of network architecture and are not aware of actual hardware data processing.Hence, the upper layers expect error-free transmission between the systems.
* Most of the applications would not function expectedly if they receive erroneous data. Applications such as voice and video may not be that affected and with some errors they may still function well.
* Data-link layer uses some error control mechanism to ensure that frames (data bit streams) are transmitted with certain level of accuracy. But to understand how errors is controlled, it is essential to know what types of errors may occur.

## Types of Errors

* There may be three types of errors:
* **Single bit error**



In a frame, there is only one bit, anywhere though, which is corrupt.

* **Multiple bits error**



Frame is received with more than one bits in corrupted state.

* **Burst error**



Frame contains more than1 consecutive bits corrupted.

* Error control mechanism may involve two possible ways:
* Error detection
* Error correction

## Error Detection

* Errors in the received frames are detected by means of Parity Check and Cyclic Redundancy Check (CRC). In both cases, few extra bits are sent along with actual data to confirm that bits received at other end are same as they were sent. If the counter-check at receiver’ end fails, the bits are considered corrupted.

### Parity Check

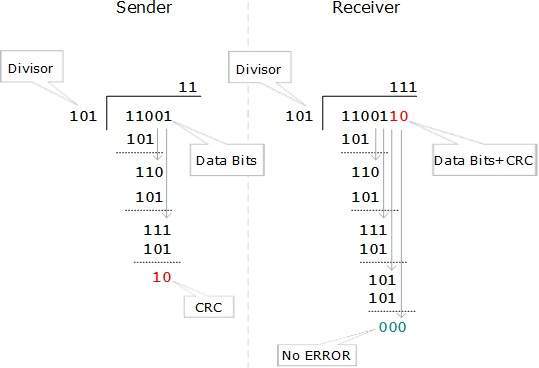
* One extra bit is sent along with the original bits to make number of 1s either even in case of even parity, or odd in case of odd parity.
* The sender while creating a frame counts the number of 1s in it. For example, if even parity is used and number of 1s is even then one bit with value 0 is added. This way number of 1s remains even.If the number of 1s is odd, to make it even a bit with value 1 is added.

Even Parity

* The receiver simply counts the number of 1s in a frame. If the count of 1s is even and even parity is used, the frame is considered to be not-corrupted and is accepted. If the count of 1s is odd and odd parity is used, the frame is still not corrupted.
* If a single bit flips in transit, the receiver can detect it by counting the number of 1s. But when more than one bits are erro neous, then it is very hard for the receiver to detect the error.

### Cyclic Redundancy Check (CRC)

* CRC is a different approach to detect if the received frame contains valid data. This technique involves binary division of the data bits being sent. The divisor is generated using polynomials. The sender performs a division operation on the bits being sent and calculates the remainder. Before sending the actual bits, the sender adds the remainder at the end of the actual bits. Actual data bits plus the remainder is called a codeword. The sender transmits data bits as codewords.



* At the other end, the receiver performs division operation on codewords using the same CRC divisor. If the remainder contains all zeros the data bits are accepted, otherwise it is considered as there some data corruption occurred in transit.

## Error Correction

* In the digital world, error correction can be done in two ways:
* **Backward Error Correction**  When the receiver detects an error in the data received, it requests back the sender to retransmit the data unit.
* **Forward Error Correction**  When the receiver detects some error in the data received, it executes error-correcting code, which helps it to auto-recover and to correct some kinds of errors.
* The first one, Backward Error Correction, is simple and can only be efficiently used where retransmitting is not expensive. For example, fiber optics. But in case of wireless transmission retransmitting may cost too much. In the latter case, Forward Error Correction is used.
* To correct the error in data frame, the receiver must know exactly which bit in the frame is corrupted. To locate the bit in error, redundant bits are used as parity bits for error detection.For example, we take ASCII words (7 bits data), then there could be 8 kind of information we need: first seven bits to tell us which bit is error and one more bit to tell that there is no error.
* For m data bits, r redundant bits are used. r bits can provide 2r combinations of information. In m+r bit codeword, there is possibility that the r bits themselves may get corrupted. So the number of r bits used must inform about m+r bit locations plus no-error information, i.e. m+r+1.

Required bits

**Q-5 Explain Network Service Model.**

* The network layer moves transport-layer segments from one host to nother. At the sending host, the transport-layer segment is passed to the network layer. It is then the job of the network layer to get the segment to the destination host and pass the segment up the protocol stack to the transport layer.
* When the transport layer at a sending host transmits a packet into the network (that is, passes it down to the network layer at the sending host), can the transport layer count on the network layer to deliver the packet to the destination?
* When multiple packets are sent, will they be delivered to the transport layer in the receiving host in the order in which they were sent? Will the amount of time between the sending of two sequential packet transmissions be the same as the amount of time between their reception? Will the network provide any feedback about congestion in the network? What is the abstract view (properties) of the channel connecting the transport layer in the sending and receiving hosts? The answers to these questions and others are determined by the service model provided by the network layer.
* The network-service model defines the characteristics of end-to-end transport of data between one "edge" of the network and the other, that is, between sending and receiving end systems.

**Datagram or Virtual Circuit?**

* Perhaps the most important abstraction provided by the network layer to the upper layers is whether or not the network layer uses virtual circuits (VCs). You may recall from Chapter 1 that a virtual-circuit packet network behaves much like a telephone network, which uses "real circuits" as opposed to "virtual circuits." There are three identifiable phases in a virtual circuit: •

**VC setup.**

During the setup phase, the sender contacts the network layer, specifies the receiver address, and waits for the network to set up the VC. The network layer determines the path between sender and receiver, that is, the series of links and packet switches through which all packets of the VC will travel. typically involves updating tables in each of the packet switches in the path. During VC setup, the network layer may also reserve resources (for example, bandwidth) along the path of the VC.

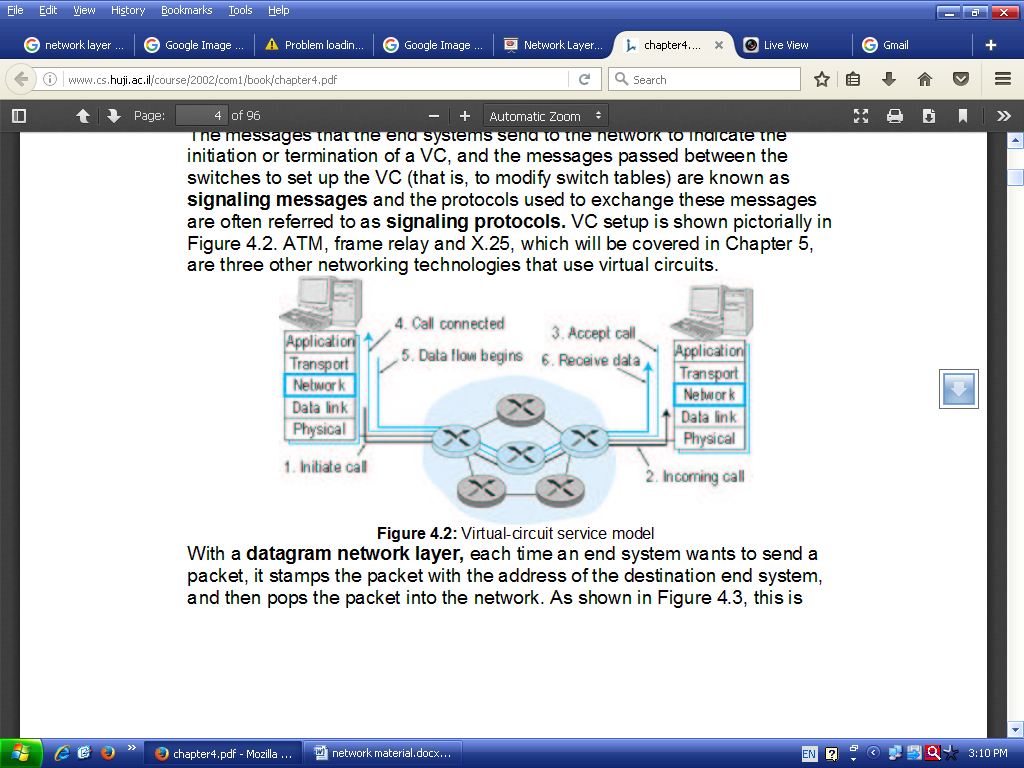
**Data transfer.**

Once the VC has been established, data can begin to flow along the VC.

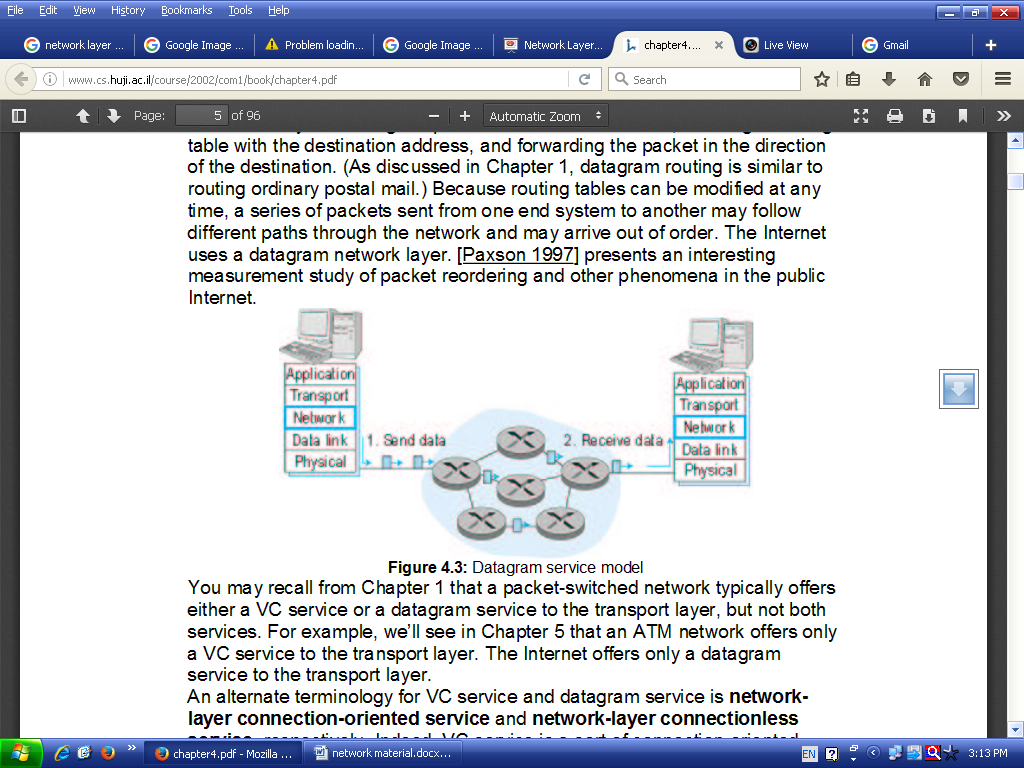
**Virtual-circuit teardown**.

This is initiated when the sender (or receiver) informs the network layer of its desire to terminate the VC. The network layer will then typically inform the end system on the other side of the network of the call termination and update the tables in each of the packet switches on the path to indicate that the VC no longer exists.

* There is a subtle but important distinction between VC setup at the network layer and connection setup at the transport layer (for example, the TCP three- way handshake we studied in Chapter 3). Connection setup at the transport layer involves only the two end systems.
* The two end systems agree to communicate and together determine the parameters (for example, initial sequence number, flow-control window size) of their transport-layer connection before data actually begins to flow on the transport-level connection. Although the two end systems are aware of the transport-layer connection, the switches within the network are completely oblivious to it.
* On the other hand, with a virtual-circuit network layer, packet switches along the path between the two end systems are involved in virtual-circuit setup, and each packet switch is fully aware of all the VCs passing through it.
* The messages that the end systems send to the network to indicate the initiation or termination of a VC, and the messages passed between the switches to set up the VC (that is, to modify switch tables) are known as signaling messages and the protocols used to exchange these messages are often referred to as signaling protocols. VC setup is shown pictorially in following Figure.
* ATM, frame relay and X.25, which will be covered in Chapter 5, are three other networking technologies that use virtual circuits.



* With a datagram network layer, each time an end system wants to send a packet, it stamps the packet with the address of the destination end system, and then pops the packet into the network. As shown in following figure, this is done without any VC setup. Packet switches in a datagram network (called "routers" in the Internet) do not maintain any state information about VCs because there are no VCs! Instead, packet switches route a packet toward its destination by examining the packet’s destination address, indexing a routing table with the destination address, and forwarding the packet in the direction of the destination.
* Because routing tables can be modified at any time, a series of packets sent from one end system to another may follow different paths through the network and may arrive out of order. The Internet
* uses a datagram network layer. [Paxson 1997] presents an interesting measurement study of packet reordering and other phenomena in the public Internet.



* Datagram service is a sort of connectionless service in that it does not employ connection-like entities. Both sets of terminology have advantages and disadvantages, and both sets are commonly used in the networking literature. In this book we decided to use the "VC service" and "datagram service" terminology for the network layer, and reserve the "connection-oriented service" and "connectionless service" terminology for the transport layer. We believe this distinction will be useful in helping the reader delineate the services offered by the two layers.

**Q-6 Explain various measures to be taken in order to run network smoothly.**

* There are following networking measures to be taken to run network smoothly.

**Create a network** [**documentation policy**](http://itknowledgeexchange.techtarget.com/network-technologies/sample-campus-network-documentation-policy)

* A network documentation policy should detail what aspects of a network need to be documented, especially each server. A documentation policy also communicates to each administrator exactly what is expected of them regarding the documentation process.

**Create a network topology diagram**

* Ideally, you want this map of the [network's topology](http://searchsecurity.techtarget.com/answer/Network-topology-mapping-How-to-automate-network-documentation) to include each network segment, the routers connecting the various segments, and the servers, gateways and other major pieces of networking hardware that are connected to each segment. For larger networks, you may have to create a general segment map and make more specific maps of each individual segment.

**Document server names, roles and IP addresses**

* While the information included in a network topology diagram is not necessarily specific, there is certain information that you should include for each server, even if that information has to be placed in an appendix. For each server, list the server's name, its IP address and the role that the server is performing (DNS, DHCP, mail server, etc.). Keep in mind that a server may be assigned multiple IP addresses or have multiple NICs, so you should document that information too.

**Create a change log for each server**

* When a server fails, the failure can often be traced to a recent change. As a part of the network documentation, consider making a [log book](http://searchdatacenter.techtarget.com/tip/Managing-and-monitoring-server-log-data) for each server for documenting changes such as patch and application installations and modified security settings. Not only will the log help you troubleshoot future problems, it can help you rebuild the server in the event of a catastrophic failure.

**Document software versions and proof of licenses**

* Document the applications and their versions running on each server. You might also include a copy of the software license or a receipt within this documentation just in case your customer becomes involved in a software audit.

**Document hardware components**

* I have talked about documenting individual servers, but it's equally important to document switches, routers, gateways and other networking hardware. The documentation should include information such as:
  + How is the device connected to the network?
  + How is the device configured?
  + Does a backup of the configuration exist?
  + What firmware revision is the device running?
  + Is the device configured to use a password? (Don't include the actual password, but you can include a password hint or a reference to the password being written in a notebook that is stored in the safe.)