CN Assignment 3

Date 13.10.2020

Roll No: B032	Name: Naman Garg
Sem/Year: 5	Batch: B2
Grade:	

1. Explain the various issues involved in connection establishment and connection release in transport protocols

2. Discuss the Dijkstra's Shortest Path Algorithm with the help of a suitable example

3. Write a note on SMTP and POP protocols

4. Compare Classful addressing vs Classless addressing

A1.)

The Transmission Control Protocol (TCP) provides full transport layer services to applications. TCP is a reliable stream transport port-to-port protocol. The term stream means connection-oriented: a connection must be established between both ends of a transmission before either may transmit data. Connection is creating between sender and receiver through a virtual circuit.
TCP is a connection – oriented protocol establishes a virtual path for segment transfer between the source and the destination requires two procedures.

1. **Connection establishment** – Connections for the duration of an entire exchange are different, and are handled by session functions in individual applications.
2. **Connection termination** – Ends each transmission. **Connection Establishment:**

Connection establishment is performed by the concept called **Three-way Handshake.** To establish a connection, TCP uses a three-way handshake. Before a client attempts to connect with a server, the server must first bind to a port to open it up for connections: this is called a passive open. Once the passive open is established, a client may initiate an active open. To establish a connection, the three-way handshake occurs:

1. The active open is performed by the client sending a SYN to the server.
2. In response, the server replies with a SYN-ACK.
3. Finally the client sends an ACK back to the server.

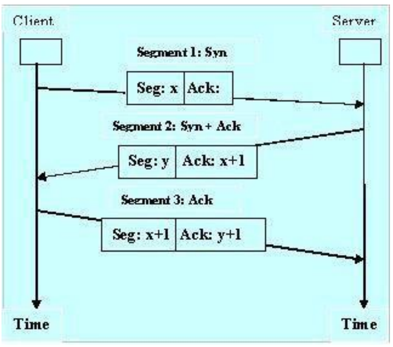
At this point, both the client and server have received an acknowledgement of the connection.

Example:

The initiating host (client) sends a synchronization packet (SYN flag set to
to initiate a connection. It sets the packet's sequence number to a random value x.

The other host receives the packet, records the sequence number x from the client, and replies with an acknowledgement and synchronization (SYN-ACK). The Acknowledgement is a 32-bit field in TCP segment header. It contains the next sequence number that this host is expecting to receive (x+1). The host also initiates a return session. This includes a TCP segment with its own initial Sequence Number of value y.

The initiating host responds with the next Sequence Number (x+1) and a simple Acknowledgement Number value of y+1, which is the Sequence Number value of the other host +1.

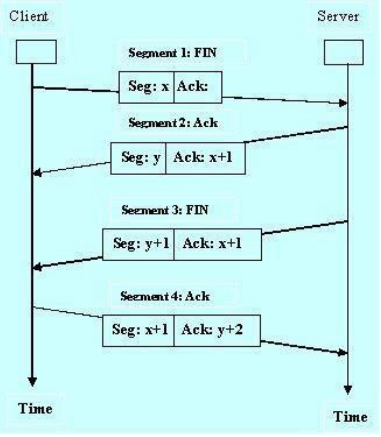


Connection Termination:

Connection Termination is performed by a concept called Four-way hand shake.] The server as well as client both should participate in the connection termination. When connection in one direction is terminated, the other party can continue sending data in the other direction. Four steps need to perform the connection termination from both server and client.

The four steps are as follows,

1. The client TCP sends the FIN segment first.
2. The server TCP sends the ACK segment to confirm the receipt of the FIN from the client. It increments the sequence number of FIN by 1 and no other user data will add with the ACK segment.
3. Server does not have any data for transmission, then it sends the FIN segment to Client side.
4. Then client sends the ACK segment again to the server side. The connection termination fulfilled.



Connection Resetting:

TCP may request the resetting of a connection at some condition.

- When a TCP at client side has requested a non-exist port application server, then the TCP on the other side may send a segment with its RST bit set to annual to request.
- At some abnormal situation RST segment is sent to close the connection. • TCP on server side may discover that the TCP on the client side has been idle for a long time. It may send an RST segment to destroy the connection.

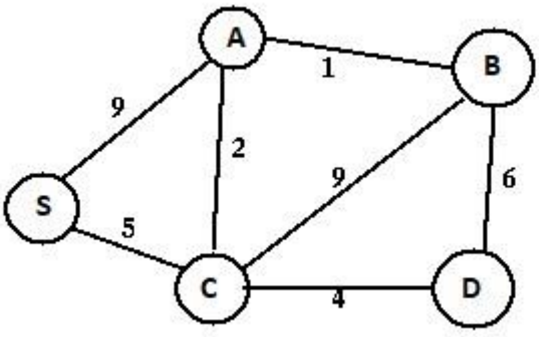
A2.)

Dijkstra's Algorithm

It is a greedy algorithm that solves the single-source shortest path problem for a directed graph G = (V, E) with nonnegative edge weights, i.e., $w(u, v) \geq 0$ for each edge $(u, v) \in E$. Dijkstra's Algorithm maintains a set S of vertices whose final shortest - path weights from the source s have already been determined. That's for all vertices $v \in S$; we have $d[v] = \delta(s, v)$. The algorithm repeatedly selects the vertex $u \in V - S$ with the minimum shortest - path estimate, insert u into S and relaxes all edges leaving u.
Because it always chooses the "lightest" or "closest" vertex in V - S to insert into set S, it is called as the **greedy strategy**.

Dijkstra's Algorithm (G, w, s)

```
INITIALIZE - SINGLE - SOURCE (G, s)
$S \leftarrow \emptyset$
$Q \leftarrow V[G]$
while $Q \neq \emptyset$
    do u ← EXTRACT - MIN (Q)
        $S \leftarrow S \cup \{u\}$
        for each vertex $v \in Adj[u]$
            do RELAX (u, v, w)
```



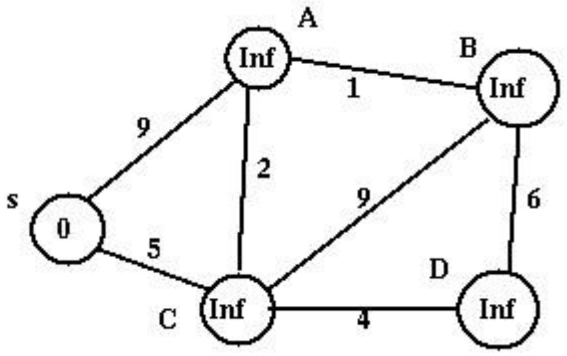
Given the above weighted and connected graph and source vertex s, following steps are used for finding the tree representing shortest path between s and all other vertices-

Step A- Initialize the distance array (dist) using the following steps of algorithm –

- **Step 1**- Set dist[s]=0, S=ϕ // u is the source vertex and S is a 1-D array having all the visited vertices

- **Step 2**- For all nodes v except s, set dist[v]= ∞

Set of visited vertices (S)	S	A	B	C	D
	0	∞	∞	∞	∞



Step B-

a) Choose the source vertex s as dist[s] is minimum and s is not in S.

- **Step 3**-
  Find q not in S such that dist[q] is minimum. // vertex should not be visited Visit s by adding it to S.
- **Step 4**-
  Add q to S // add vertex q to S since it has now been visited.

Step C-
For all adjacent vertices of s which have not been visited yet (are not in S) i.e A and C, update the distance array using the following steps of algorithm –

- **Step 5**- update dist[r] for all r adjacent to q such that r is not in S //vertex r should not be visited.
  dist[r]=min(dist[r], dist[q]+cost[q][r]) //Greedy and Dynamic
  approach dist[A]= min(dist[A], dist[s]+cost(s, A)) = min(∞, 0+9)
  = 9 dist[C] = min(dist[C], dist[s]+cost(s, C)) = min(∞, 0+5) = 5

Thus dist[] gets updated as follows-

Set of visited vertices (S)	S	A	B	C	D
[s]	0	9	∞	5	∞

Step D- Repeat Step B by

a. Choosing and visiting vertex C since it has not been visited (not in S) and dist[C] is minimum
b. Updating the distance array for adjacent vertices of C i.e. A, B and D

• **Step 6**- Repeat Steps 3 to 5 until all the nodes are in
S dist[A]=min(dist[A], dist[C]+cost(C,A)) = min(9, 5+2)= 7
dist[B]= min(dist[B], dist[C]+cost(C,B)) = min(∞, 5+9)= 14

dist[D]= min(dist[D], dist[C]+cost(C,D))= min((∞,5+4)=9

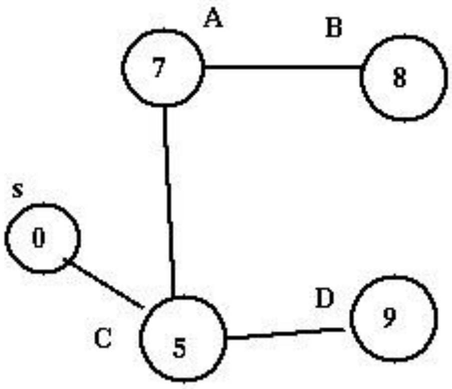
This updates dist[] as follows-

Set of visited vertices (S)	S	A	B	C	D
[s]	0	9	∞	5	∞
[s,C]	0	7	14	5	9

Continuing on similar lines, Step B gets repeated till all the vertices are visited (added to S). dist[] also gets updated in every iteration, resulting in the following –

Set of visited vertices (S)	S	A	B	C	D
[s]	0	9	∞	5	∞
[s,C]	0	7	14	5	9
[s, C, A]	0	7	8	5	9
[s, C, A, B]	0	7	8	5	9
[s, C, A, B, D]	0	7	8	5	9

The last updation of dist[] gives the shortest path values from s to all other vertices **The resultant shortest path spanning tree for the given graph is as follows-**



Disadvantage of Dijkstra's Algorithm:

1. It does a blind search, so wastes a lot of time while processing.
2. It can't handle negative edges.
3. It leads to the acyclic graph and most often cannot obtain the right shortest path. 4. We need to keep track of vertices that have been visited.

A3.)

Email is emerging as one of the most valuable services on the internet today. Most of the internet systems use SMTP as a method to transfer mail from one user to another. SMTP is a push protocol and is used to send the mail whereas POP (post office protocol) or IMAP (internet message access protocol) are used to retrieve those mails at the receiver's side.

SMTP Fundamentals

SMTP is an application layer protocol. The client who wants to send the mail opens a TCP connection to the SMTP server and then sends the mail across the connection. The SMTP server is always on listening mode. As soon as it listens for a TCP connection from any client, the SMTP process initiates a connection on that port (25). After successfully establishing the TCP connection the client process sends the mail instantly.

SMTP Protocol

The SMTP model is of two type :

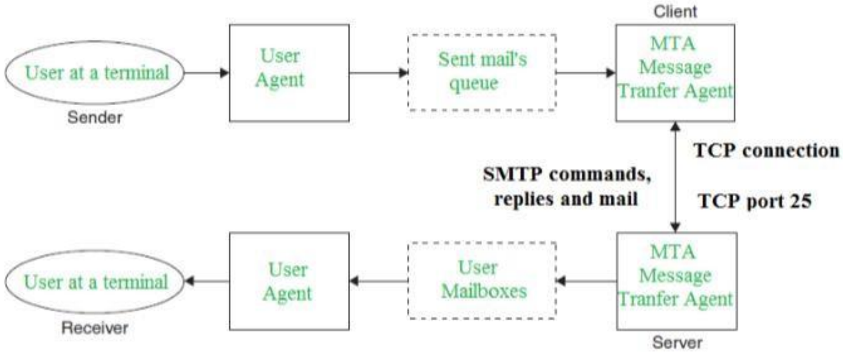
1. End-to- end method
2. Store-and- forward method

The end to end model is used to communicate between different organizations whereas the store and forward method are used within an organization. A SMTP client who wants to send the mail will contact the destination's host SMTP directly in order to send the mail to the destination. The SMTP server will keep the mail to itself until it is successfully copied to the receiver's SMTP.

The client SMTP is the one which initiates the session let us call it as the client-SMTP and the server SMTP is the one which responds to the session request and let us call it as receiver-SMTP. The client- SMTP will start the session and the receiver-SMTP will respond to the request.

Model of SMTP system

In the SMTP model user deals with the user agent (UA) for example Microsoft Outlook, Netscape, Mozilla, etc. In order to exchange the mail using TCP, MTA is used. The users sending the mail do not have to deal with the MTA it is the responsibility of the system admin to set up the local MTA. The MTA maintains a small queue of mails so that it can schedule repeat delivery of mail in case the receiver is not available. The MTA delivers the mail to the mailboxes and the information can later be downloaded by the user agents.



Both the SMTP-client and SMTP-server should have 2 components:

1. User agent (UA)
2. Local MTA

Communication between sender and the receiver :
The senders, user agent prepare the message and send it to the MTA. The MTA functioning is to transfer the mail across the network to the receivers MTA. To send mail, a system must have the client MTA, and to receive mail, a system must have a server MTA.

SENDING EMAIL:
Mail is sent by a series of request and response messages between the client and a server. The message which is sent across consists of a header and the body. A null line is used to terminate the mail header. Everything which is after the null line is considered as the body of the message which is a sequence of ASCII characters. The message body contains the actual information read by the receipt.

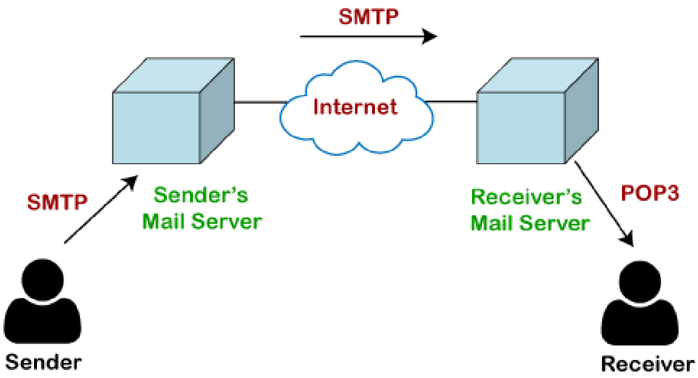
RECEIVING EMAIL:
The user agent at the server-side checks the mailboxes at a particular time of intervals. If any information is received it informs the user about the mail. When the user tries to read the mail it displays a list of mails with a short description of each mail in the mailbox. By selecting any of the mail user can view its contents on the terminal.

Some SMTP Commands:

- HELO – Identifies the client to the server, fully qualified domain name, only sent once per session
- MAIL – Initiate a message transfer, fully qualified domain of originator
- RCPT – Follows MAIL, identifies an addressee, typically the fully qualified name of the addressee and for multiple addressees use one RCPT for each addressee
- DATA – send data line by line

POP Protocol

The POP protocol stands for Post Office Protocol. As we know that SMTP is used as a message transfer agent. When the message is sent, then SMPT is used to deliver the message from the client to the server and then to the recipient server. But the message is sent from the recipient server to the actual server with the help of the Message Access Agent. The Message Access Agent contains two types of protocols, i.e., POP3 and IMAP. **How is mail transmitted?**



Suppose sender wants to send the mail to receiver. First mail is transmitted to the sender's mail server. Then, the mail is transmitted from the sender's mail server to the receiver's mail server over the internet. On receiving the mail at the receiver's mail server, the mail is then sent to the user. The whole process is done with the help of Email protocols. The transmission of mail from the sender to the sender's mail server and then to the receiver's mail server is done with the help of the

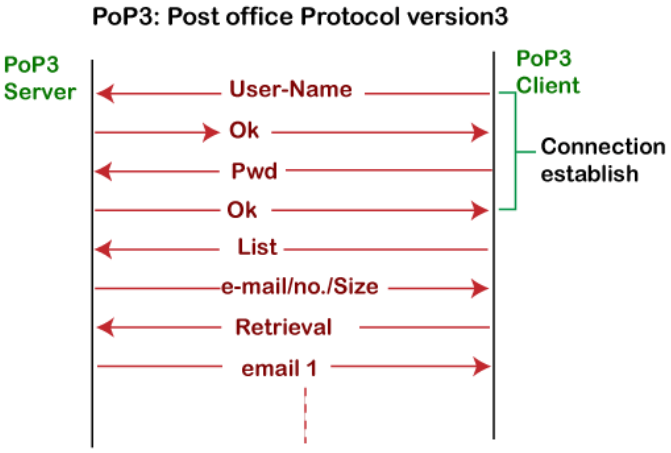
SMTP protocol. At the receiver's mail server, the POP or IMAP protocol takes the data and transmits to the actual user.

Since SMTP is a push protocol so it pushes the message from the client to the server. As we can observe in the above figure that SMTP pushes the message from the client to the recipient's mail server. The third stage of email communication requires a pull protocol, and POP is a pull protocol. When the mail is transmitted from the recipient mail server to the client which means that the client is pulling the mail from the server.

What is POP3?

The POP3 is a simple protocol and having very limited functionalities. In the case of the POP3 protocol, the POP3 client is installed on the recipient system while the POP3 server is installed on the recipient's mail server.

Let's understand the working of the POP3 protocol.



To establish the connection between the POP3 server and the POP3 client, the POP3 server asks for the user name to the POP3 client. If the username is found in the POP3 server, then it sends the ok message. It then asks for the password from the POP3 client; then the POP3 client sends the password to the POP3 server. If the password is matched, then the POP3 server sends the OK message, and the connection gets established. After the establishment of a connection, the client can see the list of mails on the POP3 mail server. In the list of mails, the user will get the email numbers and sizes from the server. Out of this list, the user can start the retrieval of mail.

Once the client retrieves all the emails from the server, all the emails from the server are deleted. Therefore, we can say that the emails are restricted to a particular machine, so it would not be possible to access the same mails on another machine. This situation can be overcome by configuring the email settings to leave a copy of mail on the mail server.

Advantages of POP3 protocol

The following are the advantages of a POP3 protocol:

- It allows the users to read the email offline. It requires an internet connection only at the time of downloading emails from the server. Once the mails are downloaded from the server, then all the downloaded mails reside on our PC or hard disk of our computer, which can be accessed without the internet. Therefore, we can say that the POP3 protocol does not require permanent internet connectivity.
- It provides easy and fast access to the emails as they are already stored on our PC.
- There is no limit on the size of the email which we receive or send. o It requires less server storage space as all the mails are stored on the local machine. o There is maximum size on the mailbox, but it is limited by the size of the hard disk. o It is a simple protocol so it is one of the most popular protocols used today.
- It is easy to configure and use.

Disadvantages of POP3 protocol

The following are the advantages of a POP3 protocol:

- If the emails are downloaded from the server, then all the mails are deleted from the server by default. So, mails cannot be accessed from other machines unless they are configured to leave a copy of the mail on the server.
- Transferring the mail folder from the local machine to another machine can be difficult.
- Since all the attachments are stored on your local machine, there is a high risk of a virus attack if the virus scanner does not scan them. The virus attack can harm the computer.
- The email folder which is downloaded from the mail server can also become corrupted.
- The mails are stored on the local machine, so anyone who sits on your machine can access the email folder.

A4.)

Classful Addressing: Introduced in 1981, with classful routing, IP v4 addresses were divided into 5 classes(A to E).

- Classes A-C: unicast addresses
- Class D: multicast addresses
- Class E: reserved for future use

Class A In a class A address, the first bit of the first octet is always ‘0’. Thus, class A addresses range from 0.0.0.0 to 127.255.255.255(as 01111111 in binary converts to 127 in decimal). The first 8 bits or the first octet denote the network portion and the rest 24 bits or the 3 octets belong to the host portion. Example: 10.1.1.1 Exception:

- 127.X.X.X is reserved for loopback

- 0.X.X.X is reserved for default network

Therefore, the actual range of class A addresses is: 1.0.0.0 to 126.255.255.255

Class B

In a class B address, the first octet would always start with ’10’. Thus, class B addresses range from 128.0.0.0 to 191.255.255.255. The first 16 bits or the first two octets denote the network portion and the remaining 16 bits or two octets belong to the host portion. Example: 172.16.1.1

Class C In a class C address, the first octet would always start with ‘110’. Thus, class C addresses range from 192.0.0.0 to 223.255.255.255. The first 24 bits or the first three octets denote the network portion and the rest 8 bits or the remaining one octet belong to the host portion.

Example: 192.168.1.1

Class D

Class D is used for multicast addressing and in a class D address the first octet would always start with ‘1110’. Thus, class D addresses range from 224.0.0.0 to 239.255.255.255.

Example: 239.2.2.2

Class D addresses are used by routing protocols like OSPF, RIP, etc.

Class E

Class E addresses are reserved for research purposes and future use. The first octet in a class E address starts with ‘1111’. Thus, class E addresses range from 240.0.0.0 to 255.255.255.255.

Disadvantage of Classful Addressing:

Class A with a mask of 255.0.0.0 can support 16, 777, 214 addresses
Class B with a mask of 255.255.0.0 can support 65, 534
addresses Class C with a mask of 255.255.255.0 can
support 254 addresses But what if someone requires 2000
addresses ?

One way to address this situation would be to provide the person with class B network.

But that would result in a waste of so many addresses.

Another possible way is to provide multiple class C networks, but that too can cause a problem as there would be too many networks to handle.

To resolve problems like the one mentioned above CIDR was introduced.

Classless Inter-Domain Routing (CIDR):

CIDR or Class Inter-Domain Routing was introduced in 1993 to replace classfull addressing. It allows the user to use VLSM or Variable Length Subnet Masks.

CIDR notation:
In CIDR subnet masks are denoted by /X. For example a subnet of 255.255.255.0 would be denoted by /24. To work a subnet mask in CIDR, we have to first convert each octet into its respective binary value. For example, if the subnet is of 255.255.255.0. then :

First Octet:
255 has 8 binary 1's when converted to binary
Second Octet:
255 has 8 binary 1's when converted to
binary Third Octet:
255 has 8 binary 1's when converted to binary
Fourth Octet:
0 has 0 binary 1's when converted to binary
Therefore, in total there are 24 binary 1’s, so the subnet mask is /24.

While creating a network in CIDR, a person has to make sure that the masks are contagious, i.e. a subnet mask like 10111111.X.X.X can’t exist.

With CIDR, we can create Variable Length Subnet Masks, leading to less wastage of IP addresses. It is not necessary that the divider between the network and the host portions is at an octet boundary. For example, in CIDR a subnet mask like 255.224.0.0 or 11111111.11100000.00000000.00000000 can exist.

S.NO	CLASSFUL ROUTING	CLASSLESS ROUTING
1.	In classful routing, VLMS(Variable Length Subnet Mask) is not supported.	While in classless routing, VLMS(Variable Length Subnet Mask) is supported.
2.	Classful routing requires more bandwidth.	While it requires less bandwidth.
3.	In classful routing, hello messages are not used.	While in classless routing, hello messages are used.
4.	Classful routing does not import subnet mask.	Whereas it imports subnet mask.
5.	In classful routing, address is divided into three parts which are: Network, Subnet and Host.	While in classless routing, address is divided into two parts which are: Subnet and Host.
6.	In classful routing, regular or periodic updates are used.	Whereas in this, triggered updates are used.
7.	In classful routing, CIDR(Classless InterDomain Routing) is not supported.	While in classless routing, CIDR(Classless Inter-Domain Routing) is supported.
8.	In classful routing, subnets are not displayed in other major subnets.	While in classless routing, subnets are displayed in other major subnets.

9.	In classful routing, fault can be detected easily.	While in classless routing, fault detection is little tough.