1) **How to reach Amazon?**

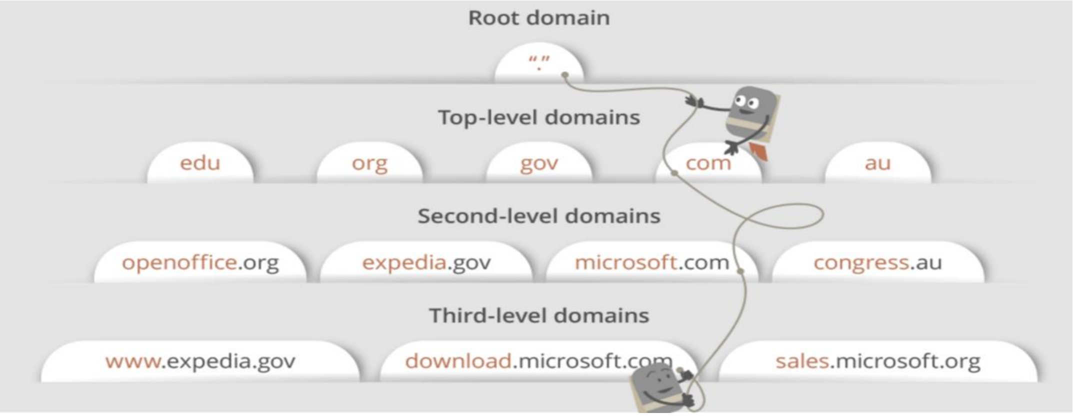
https://medium.com/@maneesha.wijesinghe1/what-happens-when-you-type-an-url-in-the-browser-and-press-enter-bb0aa2449c1a
https://stackoverflow.com/questions/2092527/what-happens-when-you-type-in-a-url-in-browser
https://www.quora.com/What-are-the-series-of-steps-that-happen-when-a-URL-is-requested-from-the-address-field-of-a-browser
http://edusagar.com/articles/view/70/What-happens-when-you-type-a-URL-in-browser
https://www.linkedin.com/pulse/what-happens-when-you-type-url-browser-press-enter-wijesinghe/

Steps:

1. Firstly, When a pc boots up it obtains its own IP address, IP address of local DNS server and Default gateway and subnet block using **DHCP (Dynamic Host Control Protocol) DORA process.**
2. Then The browser checks the cache for a DNS record to find the corresponding IP address of maps.google.com. In order to find the DNS record, the browser checks four caches.
   a) First, it checks the **browser cache**. The browser maintains a repository of DNS records for a fixed duration for websites you have previously visited. So, it is the first place to run a DNS query.
   b) Second, the browser checks the **OS cache**. If it is not found in the browser cache, the browser would make a system call (i.e. *gethostname* on Windows) to your underlying computer OS to fetch the record since the OS also maintains a cache of DNS records.
   c)Third, it checks **the router cache**. If it's not found on your computer, the browser would communicate with the router that maintains its' own cache of DNS records.
   d)Fourth, it checks the **ISP cache**. If all steps fail, the browser would move on to the ISP. Your ISP maintains its' own DNS server which includes a cache of DNS records which the browser would check with the last hope of finding your requested URL.

3. If the requested URL is not in the cache, ISP's DNS server initiates a DNS query to find the IP address of the server that hosts maps.google.com. (You may wonder why there are so many caches maintained at so many levels. Although our information being cached somewhere doesn't make us feel very comfortable when it comes to privacy, **caches are important for regulating network traffic and improving data transfer times**.)
4. (As mentioned earlier, in order for my computer to connect with the server that hosts maps.google.com, I need the IP address of maps.google.com.) The purpose of a DNS query is to search multiple DNS servers on the internet until it finds the correct IP address for the website. This type of search is called a recursive search since the search will continue repeatedly from DNS server to DNS server until it either finds the IP address we need or returns an error response saying it was unable to find it.

5. In this situation, we would call the ISP's DNS server a DNS recursor whose responsibility is to find the proper IP address of the intended domain name by asking other DNS servers on the internet for an answer. The other DNS servers are called name servers since they perform a DNS search based on the domain architecture of the website domain name. **Diagram to explain the domain architecture.**



https://webhostinggeeks.com/guides/dns/ Many website URLs we encounter today contain a third-level domain, a second-level domain, and a top-level domain. Each of these levels contains their own name server which is queried during the DNS lookup process.

6. For maps.google.com, first, the DNS recursor will contact the root name server. The root name server will redirect it to **.com** domain name server. **.com** name server will redirect it to **google.com** name server. **google.com** name server will find the matching IP address for maps.google.com in its' DNS records and return it to your DNS recursor which will send it back to your browser.

7. These requests are sent using small data packets which contain information such as the content of the request and the IP address it is destined for (IP address of the DNS recursor). These packets travel through multiple networking equipment between the client and the server before it reaches the correct DNS server. This equipment use routing tables to figure out which way is the fastest possible way for the packet to reach its' destination. If these packets get lost you'll get a request failed error. Otherwise, they will reach the correct DNS server, grab the correct IP address, and come back to your browser.
8. Browser initiates a TCP connection with the server.

(Once the browser receives the correct IP address it will build a connection with the server that matches IP address to transfer information. Browsers use internet protocols to build such connections. There are a number of different internet protocols which can be used but TCP is the most common protocol used for any type of HTTP request.)

9. In order to transfer data packets between your computer(client) and the server, it is important to have a TCP connection established. This connection is established using a process called the TCP/IP three-way handshake. This is a three step process where the client and the server exchange SYN(synchronize) and ACK(acknowledge) messages to establish a connection.

a)Client machine sends a SYN packet to the server over the internet asking if it is open for new connections.

b)If the server has open ports that can accept and initiate new connections, it'll respond with an ACKnowledgment of the SYN packet using a SYN/ACK packet.

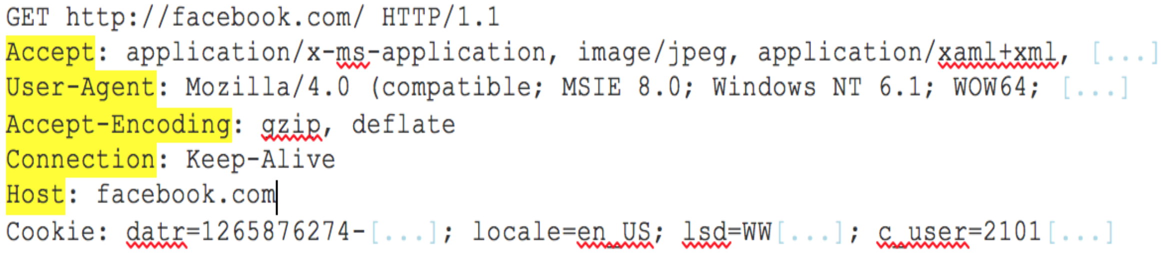
c)The client will receive the SYN/ACK packet from the server and will acknowledge it by sending an ACK packet.

Then a TCP connection is established for data transmission!

10.The browser sends an HTTP request to the web server.

(Once the TCP connection is established, it is time to start transferring data! The browser will send a GET request asking for maps.google.com web page. If you're entering credentials or submitting a form this could be a POST request. This request will also contain additional information such as browser identification (*User-Agent* header), types of requests that it will accept (*Accept*header), and connection headers asking it to keep the TCP connection alive for additional requests. It will also pass information taken from cookies the browser has in store for this domain.)

Sample GET request (Headers are highlighted):



```
GET http://facebook.com/ HTTP/1.1
Accept: application/x-ms-application, image/jpeg, application/xaml+xml, [...]
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; WOW64; [...]
Accept-Encoding: gzip, deflate
Connection: Keep-Alive
Host: facebook.com
Cookie: datr=1265876274-[...]; locale=en_US; lsd=WW[...]; c_user=2101[...]
```

(If you're curious about what's going on behind the scenes you can use tools such as Firebug to take a look at HTTP requests. It is always fun to see the information passed between clients and servers without us knowing).

10. The server handles the request and sends back a response.

11. The server contains a web server (i.e Apache, IIS) which receives the request from the browser and passes it to a request handler to read and generate a response. The request handler is a program (written in ASP.NET, PHP, Ruby, etc.) that reads the request, its' headers, and cookies to check what is being requested and also update the information on the server if needed. Then it will assemble a response in a particular format (JSON, XML, HTML)

12. **The server sends out an HTTP response.**
    The server response contains the web page you requested as well as the status code, compression type (*Content-Encoding)*, how to cache the page (*Cache-Control*), any cookies to set, privacy information, etc.
    Example HTTP server response:

```
HTTP/1.1 200 OK
Cache-Control: private, no-store, no-cache, must-revalidate, post-check=0,
    pre-check=0
Expires: Sat, 01 Jan 2000 00:00:00 GMT
P3P: CP="DSP LAW"
Pragma: no-cache
Content-Encoding: gzip
Content-Type: text/html; charset=utf-8
X-Cnection: close
Transfer-Encoding: chunked
Date: Fri, 12 Feb 2010 09:05:55 GMT
```

13. If you look at the above response the first line shows a status code. This is quite important as it tells us the status of the response. There are five types of statuses detailed using a numerical code.

    1xx indicates an informational message only
    2xx indicates success of some kind
    3xx redirects the client to another URL
    4xx indicates an error on the client's part
    5xx indicates an error on the server's part

14. So, if you encountered an error you can take a look at the HTTP response to check what type of status code you have received.The browser displays the HTML content (for HTML responses which is the most common).

15. The browser displays the HTML content in phases. First, it will render the bare bone HTML skeleton. Then it will check the HTML tags and sends out GET requests for additional elements on the web page, such as images, CSS stylesheets, JavaScript files etc. These static files are cached by the browser so it doesn't have to fetch them again the next time you visit the page. At the end, you'll see maps.google.com appearing on your browser. (though this seems like a very tedious prolonged process we know that it takes less than seconds for a web page to render after we hit enter on our keyboard.)

2) Packet Flow at each layer:

Downstream (on source side):

1)The Application layer encapsulate the HTTP request into HTTP data header and handled the header to transport layer for further processing.

2 & 3)The presentation layer provides format of the data like asci for text msg or JPEG for image file. And Then the session layer establishes, controls and ends the sessions between local and remote applications.

4)After that, the transport layer will recives and encapsulate the data supplied by application layer inside a TCP/UDP header forming a segment, Adds DESTINATION AND SOURCE TCP OR UDP PORT to segment and finally handle the segment to network layer for packing procedure.

5)Now, the network layer will recives and encapsulates TCP/UDP header inside an ip header forming an IP packet ,it will translate the destination domain name into IP address using DNS and add the destination and source IP address to IP packet and finally routes the IP packet using outgoing NIC or interface through the data link layer.

6)Data link layer will receives the IP packet from n/w layer and encapsulates the IP Packet inside the Ethernet header and trailer to forming a frame, adds destination and source MAC address to the frame based on ARP table,and finally forwards the frame to next hop using the device outgoing NIC or interface through physical layer.

7)Now, physical layer will receive a frame and covert it into bits, then encodes the bits into signals based on the type of media used (cu,wireless,fiber) and transmits the signals 1 at a time using the devices outgoing NIC or through the transmission media used.

Upstream(on destination side):

1)Physical layer will capture the signals and it will convert the electrical signals to bit stream as frame and handles the frame to data link layer.

2)now, data link layer receives the frame and applies Frame Check Sequence on each frame for error detection ,if no error found, it will deencapsulate the frame and discard the MAC address and handle the Ip packet to Network layer.

3) Now, the network layer will recives and reads the packets destination and sorce ip address, and compare the destination address to known IP routes or subnets by reading the routing table entries. Now, it will route the ip packet to given interface in routing table.

4)receives and encapsulated the segment supplied by n/w layer and analyze any destination TCP or UDP source port numbers and finally handle the data to the application layers service after analysis of port number. example, If the port number is 80 , it handles the data to HTTP servies that is associated with that port and responsible to respond and process web pages data.

5)and at last application layer, will check the request in the packet and according to that open new data connection to transfer data or do further process to fulfill the client's demand.

3) What Linux command to check open ports?

- Netstat
- **Which process is using a particular port:**

- **# netstat -an | grep ':80' :** To get the process
- which is using the given port.
- -I for interfaces
- -a for all ports

4) How will you troubleshoot slow website?

- I would first ask the user whether every application on the computer is slow or is it just one particular application. if it is just one, try and sort that out.
  If the answer is all applications are running slow, look at common causes like, virus scans scheduled, free disk space on the home drive, RAM, lastly if the deployment team is rolling out any updates (which should not be done during business hours, but i've seen that happen too ;)
  If you suspect network latency, a simple ping test from the user's computer will tell you how many milli seconds network traffic is taking to reach the server.
- At a high level there are 3 things one or more of which may be slow:
  01. Your computer, 02. internet connection, 03. the server
  01. Your computer it can be slow because of some anti-virus scan, some process using high processor/network, HDD may be full, your computer may be virus infected, and thousands other possible reasons.
  02. Internet: Your proxy setting may be using a slow proxy, your ISP may be slow, the server may have failed over to a different Geo, so it is taking long time to route the calls, etc.
  03. Slow server: Server might be under DOS attack, All the instances may not be up, because of festival season/deal/new offering much more than usual # of people are using the site, servers running out of space, Networking issue in data center, slow or dead partner servers, etc. to name a few.
- DO "tracert" command in cmd and If there are a large number of failed hops it may be an indication that some ISPs along the route are having some network issues which in turn are causing your site to load slowly.
  Network drop:
  1)check the physical connection of LAN or Router or check wifi is not off
  2) run the network troubleshooter
  3) check for valid ip address using ipconfig
  4) try a ping and tracert command

Ping 8.8.8.8
→this will send four packets to Google. If they fail to send, you'll be told what the problem was. For more information, type this line to trace the route between your computer and google's dns servers.

```
tracert 8.8.8.8
```

The above command gives you a step-by-step breakdown of the path that the information takes to reach the destination you specify. Watch it, and if it fails, check to see where the problem occurs. If an error pops up early in the route, the issue is likely with your local network.

Packet Loss:

1) Try to ping –n 20 8.8.8.8 ,if successful, increase n by 100 to test for longer period of time.(*This only tests for packet loss impacting ICMP or all traffic. Protocol specific loss may not be reflected*.)
2) If packet loss was seen, the next step is to identify where the packet loss begins to occur. 'tracert' can be used to check each layer 3 device along the path to the destination."tracert -d 8.8.8.8".
This will perform a trace route to 8.8.8.8 and present each hop as an IP address. Substitute 8.8.8.8 with whatever address must be tested to.
3) As a more robust test, the tool MTR can be used to perform a continuous series of traces and present a % of loss at each hop in the path to more clearly identify where the loss is occurring.

Tracert only provides information for layer 3 devices in the path, such as routers. However, in the case where packet loss is occurring at the first hop, and must pass through a wireless access point and switch to get there, additional testing is required to isolate the problem. In this case, testing will need to be done multiple times, while getting progressively closer to the layer 3 device. The following steps are illustrated in the image below:

1. Ping the access point to test wireless quality. If using a Cisco Meraki AP, ping my.meraki.com.
If loss begins occurring here troubleshooting wireless performance.
2. Ping a client connected to the same VLAN (if configured) on the switch that the wireless client is connected to. If multiple switches exist along the path, repeat this step as needed.
If loss begins occurring here, the issue is most likely:
- Duplex/speed settings mismatch on the link between the AP and the switch, or switch and wired client
- Bad cable between the AP and switch, or switch and wired client
3. Connect a client directly to the router/firewall, on the same VLAN as the wireless client, and ping it from the wireless client.
If loss begins occurring here, the issue is most likely:
- Duplex/speed settings mismatch on the link between the switch and the router/firewall, or router/firewall and wired client
- Bad cable between the AP and switch, or router/firewall and wired client

Duplex mismatch

This occurs when two ends of a link are using different speed/duplex settings, such as 100Mbps/half-duplex and 1000Mbps/full-duplex. When this occurs, some or all traffic will be lost on the link. To correct this, ensure both sides of the link have identical settings. Ideally, both ends of the connection should be set to "Auto" for both speed and duplex. If a speed or duplex setting must be manually set of one end, ensure that it has been set to the same values on the other end as well.

Link congestion (too much traffic)

This occurs when more traffic is attempting to go over a network link than it can support. Such as 60Mbps of traffic passing over the same 20Mbps link. This creates a bottleneck, resulting in some traffic being dropped.

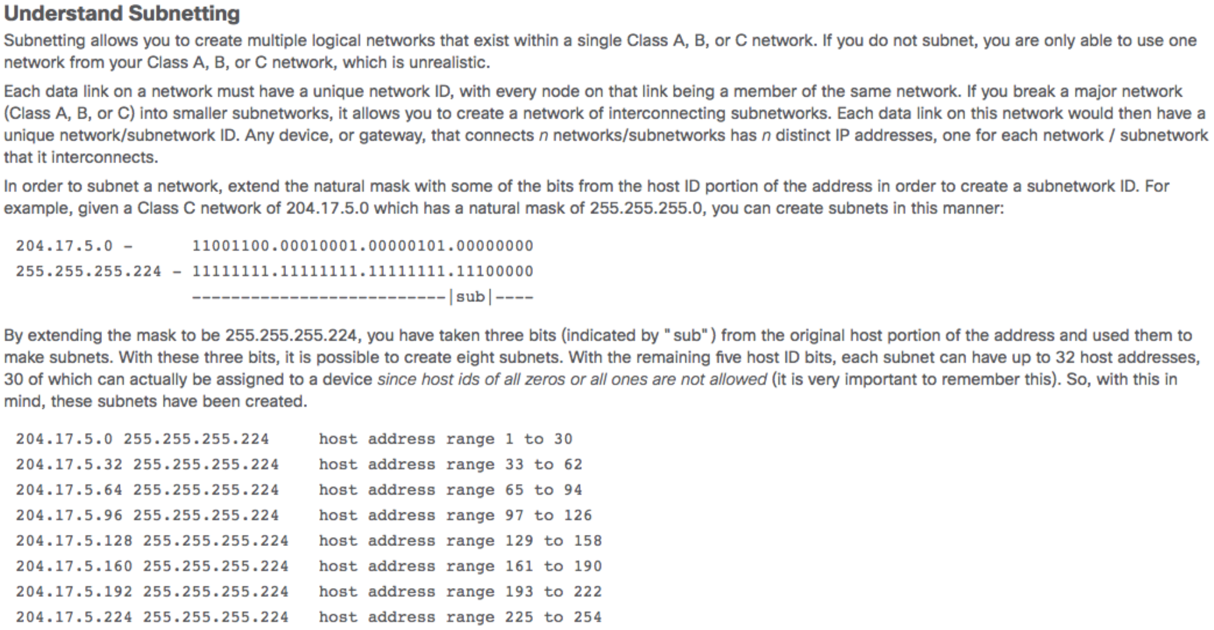
Firewall blocking certain traffic

Even if packet loss isn't occurring for all types of traffic, an upstream firewall may be filtering certain types of traffic. This can result in some websites loading and others failing, or some services being accessible, while others are not. If a firewall exists between two devices/locations experiencing these symptoms, ensure that the firewall is not blocking the traffic that is experiencing the problem.

Bad cable or loose connection

A cable that has been poorly/incorrectly terminated or damaged can result in an incomplete or inaccurate electrical signal passing between devices. Swapping a cable with a new one, or performing a cable test on the one in question, can help to eliminate this as a possibility.

5)Subnetting:



Understand Subnetting

Subnetting allows you to create multiple logical networks that exist within a single Class A, B, or C network. If you do not subnet, you are only able to use one network from your Class A, B, or C network, which is unrealistic.

Each data link on a network must have a unique network ID, with every node on that link being a member of the same network. If you break a major network (Class A, B, or C) into smaller subnetworks, it allows you to create a network of interconnecting subnetworks. Each data link on this network would then have a unique network/subnetwork ID. Any device, or gateway, that connects *n* networks/subnetworks has *n* distinct IP addresses, one for each network / subnetwork that it interconnects.

In order to subnet a network, extend the natural mask with some of the bits from the host ID portion of the address in order to create a subnetwork ID. For example, given a Class C network of 204.17.5.0 which has a natural mask of 255.255.255.0, you can create subnets in this manner:

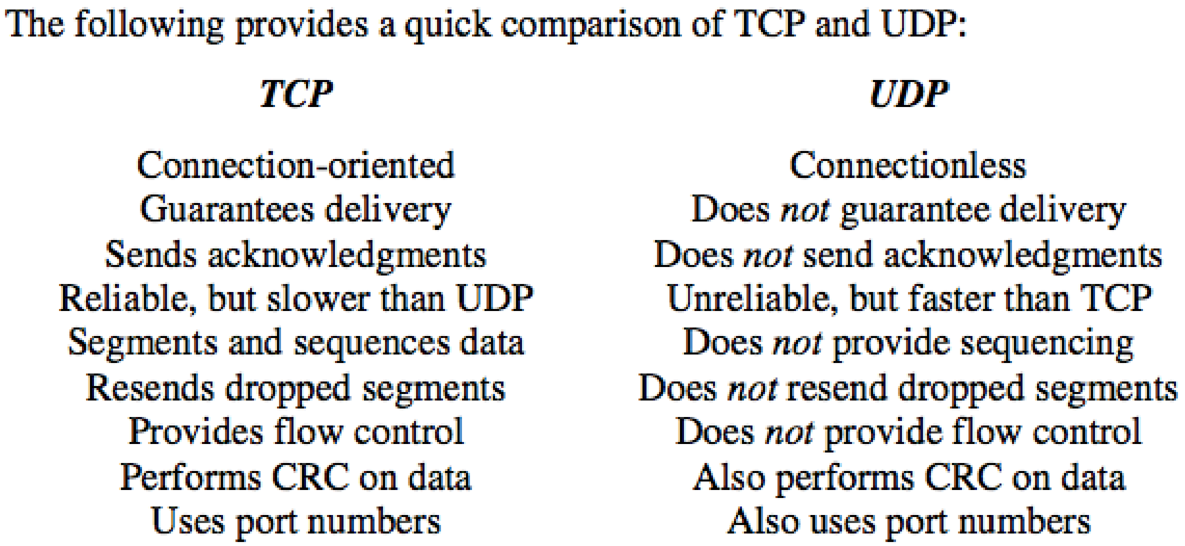
```
204.17.5.0 -      11001100.00010001.00000101.00000000
255.255.255.224 - 11111111.11111111.11111111.11100000
                  --------------------------|sub|----
```

By extending the mask to be 255.255.255.224, you have taken three bits (indicated by "sub") from the original host portion of the address and used them to make subnets. With these three bits, it is possible to create eight subnets. With the remaining five host ID bits, each subnet can have up to 32 host addresses, 30 of which can actually be assigned to a device *since host ids of all zeros or all ones are not allowed* (it is very important to remember this). So, with this in mind, these subnets have been created.

```
204.17.5.0 255.255.255.224     host address range 1 to 30
204.17.5.32 255.255.255.224    host address range 33 to 62
204.17.5.64 255.255.255.224    host address range 65 to 94
204.17.5.96 255.255.255.224    host address range 97 to 126
204.17.5.128 255.255.255.224   host address range 129 to 158
204.17.5.160 255.255.255.224   host address range 161 to 190
204.17.5.192 255.255.255.224   host address range 193 to 222
204.17.5.224 255.255.255.224   host address range 225 to 254
```

https://www.cisco.com/c/en/us/support/docs/ip/routing-information-protocol-rip/13788-3.html

6)Difference between TCP and UDP?



The following provides a quick comparison of TCP and UDP:

TCP	*UDP*
Connection-oriented	Connectionless
Guarantees delivery	Does *not* guarantee delivery
Sends acknowledgments	Does *not* send acknowledgments
Reliable, but slower than UDP	Unreliable, but faster than TCP
Segments and sequences data	Does *not* provide sequencing
Resends dropped segments	Does *not* resend dropped segments
Provides flow control	Does *not* provide flow control
Performs CRC on data	Also performs CRC on data
Uses port numbers	Also uses port numbers

7) Difference between static and dynamic routing
https://techdifferences.com/difference-between-static-and-dynamic-routing.html

1. The routers are configured manually, and the table is also created manually in static routing whereas in dynamic routing the configuration and table creation is automatic and router driven.
2. In static routing, the routes are user-defined while in dynamic routing the routes are updated as topology changes.
3. Static routing does not employ complex algorithms. As against, dynamic routing uses the complex algorithm for calculating shortest path or route.
4. Dynamic routing is suitable for large networks where the number of hosts is high. Conversely, static routing can be implemented in a small network.
5. When a link fails in static routing, the rerouting is discontinued and requires manual intervention to route traffic. In contrast, link failure in dynamic routing does not disrupt rerouting.

6. The message broadcast and multicast in dynamic routing makes it less secure. On the other hand, static routing does not involve advertisement which makes it more secure.
7. Dynamic routing involves protocols such as RIP, EIGRP, BGP, etc. Inversely, static routing does not require such protocols.
8. Static routing does not need any additional resources while dynamic routing requires additional resources such as memory, bandwidth, etc.

8)Traceroute

When host is trying to ping server, but if it is not working then we can check the problems in between routers, for that we can use traceroute, which gives us the complete network devices list in between with their ip addresses, so we can ping each of them individually & check where's the problem.
How does traceroute work? Traceroute uses the TTL (Time to Live) field in the IP packet header. Normally, TTL is used to prevent packets from being forwarded forever when there is a routing loop. Whenever an IP packet is forwarded by a router, the **TTL is decreased by one**. When the **TTL is zero, the IP packet will be discarded**.

Traceroute uses the TTL (Time to Live) field in the IP packet to send probes to the destination, allowing us to discover the path from the source to the destination. You have also seen how Windows uses ICMP and Linux uses UDP for traceroute. TTL is used to prevent packets from being forwarded forever when there is a routing loop.

Tracert –d : do not resolve hostname to ip address.

Options for Linux (traceroute):
-i It specifies the interface through which the traceroute will go. By default is selected according to the routing table.
-I Use ICMP ECHO instead of UDP datagrams. (A synonym for "-P imcp").
-s It chooses an alternative source address. By default, the address of the outgoing interface is used.
-t Type of service. The value must be a decimal integer in the range from 0 to 255. You can use it to check if different types-of-service result in different paths.

https://networklessons.com/cisco/ccna-routing-switching-icnd1-100-105/traceroute/

9)**ICMP**

ICMP is a vital part of the IP protocol it is typically considered a layer 3 protocol. While ICMP is not used regularly in end-user applications, it is used by network administrators to troubleshoot Internet connections in diagnostic utilities including ping and traceroute.
ICMP messages are transmitted as datagramsand consist of an IP header that encapsulates the ICMP data. ICMP packets are IP packets with ICMP in the IP data portion. ICMP messages also contain the entire IP header from the original message, so the end system knows which packet failed
ICMP Header: Type, Code, Checksum
Type 8 is used for an ICMP request and type 0 is used for an ICMP reply. We use type 3 for destination unreachable messages.
https://networklessons.com/cisco/ccnp-route/icmp-internet-control-message-protocol/
https://www.ibm.com/support/knowledgecenter/en/SS42VS_7.3.1/com.ibm.qradar.doc/c_DefAppCfg_guide_ICMP_intro.html
The second byte called **Code** specifies what kind of ICMP message it is. For example, the destination unreachable message has 16 different codes.

0 = Network Unreachable - generated by a router if a forwarding path (route) to the destination network is not available;

1 = Host Unreachable - generated by a router if a forwarding path (route) to the destination host on a directly connected network is not available (does not respond to ARP);

6 = Destination Network Unknown - This code SHOULD NOT be generated since it would imply on the part of the router that the destination network does not exist (net unreachable code 0 SHOULD be used in place of code 6);

7 = Destination Host Unknown - generated only when a router can determine (from link layer advice) that the destination host does not exist

The third field are 2 bytes that are used for the checksum to see if the ICMP header is corrupt or not.

ICMP has been used to execute denial-of-service attacks (also called the ping of death) by sending an IP packet larger than the number of bytes allowed by the IP protocol.

10)Ping

If ping is working, it tells me that everything on the physical layer, data link layer, and network layer in between H1 and H2 are functional.
https://networklessons.com/cisco/ccna-routing-switching-icnd1-100-105/ping-troubleshooting-on-cisco-ios/
https://www.lifewire.com/ping-command-2618099
ping is working, doesn't mean that an application would work. TCP traffic for certain ports could be blocked by an access-list or a web server is not listening on TCP port 80. Firewall can be a problem.

11) MTR:Mytraceroute

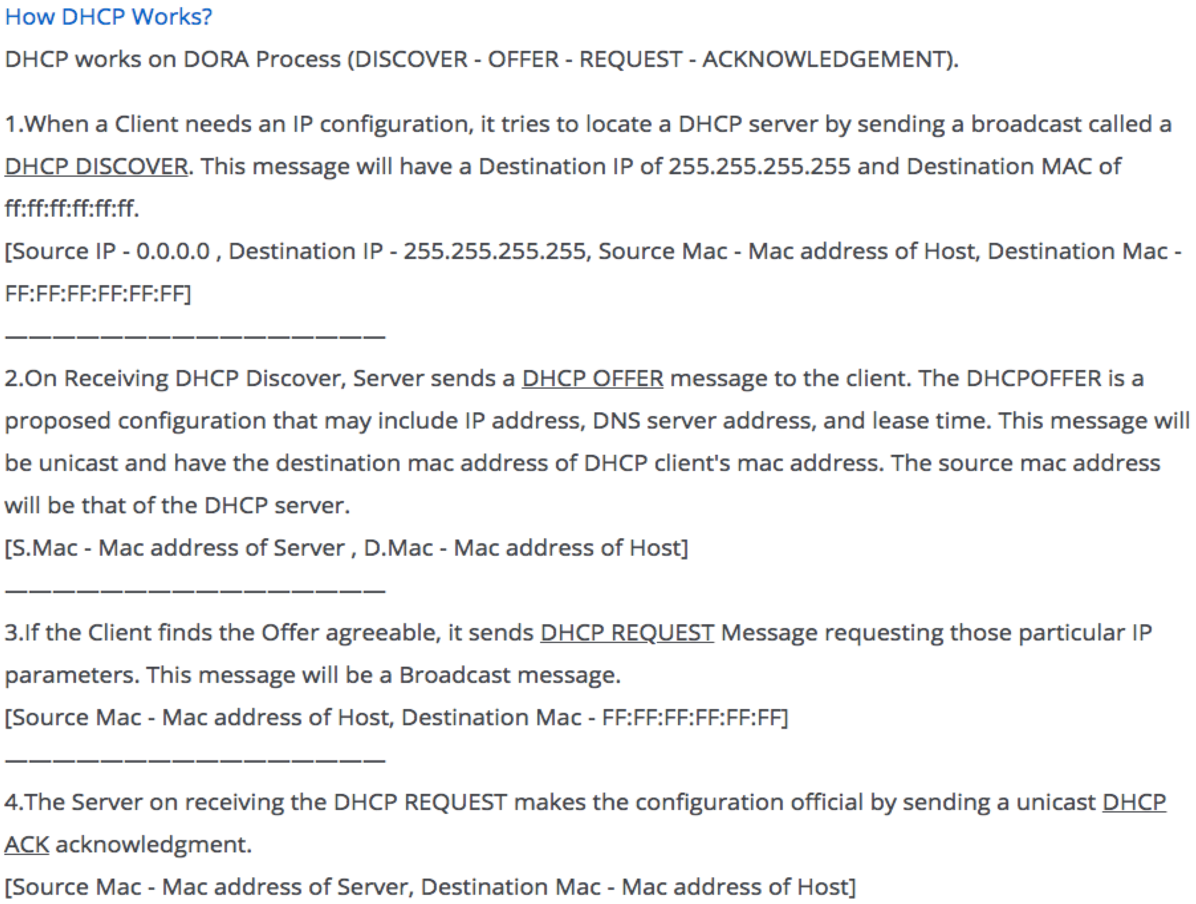
MTR combines the functionality of both the Ping and traceroute Utilities (or alternatives of traceroute), providing a robust tool for troubleshooting purposes.
MTR can provide a complete overview of the connection between two hosts on the Internet.
1)Connectivity to destination device
2)packet loss
3)Roundtrip time
By sending a series of packets and causing them to return after one hop, then two, then three, MTR is able to assemble the route that traffic takes between hosts on the Internet.
So, it gives a simple outline of the route that traffic takes across the Internet, MTR collects additional information regarding the state, connection, and responsiveness of the intermediate hosts which can be used for network dignostics.

How report generated? sudo apt-get install mtr , sudo **mtr –report google.com**
Using MTR command.
because MTR provides an image of the route traffic takes from one host to another, it is essentially a directional tool. The route taken between two points on the Internet can vary a great deal based on location and the routers that are located upstream. For this reason,it is a good idea to collect MTR reports in both directions for all hosts that are experiencing connectivity issues.

Linode Customer Support will often request MTR reports both **to** and **from** your Linode if you are experiencing networking issues. This is because, from time to time, MTR reports will not detect errors from one direction when there is still packet loss from the opposite direction.

12)DHCP

In all steps the destination ip address is 255.255.255.255. DHCP Uses udp ports 67 and 68. The DHCP server listens/ operates on ***udp*** port 67 while the Clients use port 68.
https://www.cloudshark.org/captures/291dfdfb602d



How DHCP Works?

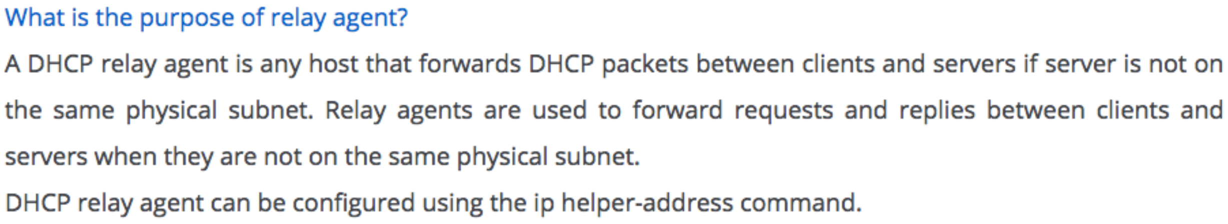
DHCP works on DORA Process (DISCOVER - OFFER - REQUEST - ACKNOWLEDGEMENT).

1.When a Client needs an IP configuration, it tries to locate a DHCP server by sending a broadcast called a DHCP DISCOVER. This message will have a Destination IP of 255.255.255.255 and Destination MAC of ff:ff:ff:ff:ff:ff.
[Source IP - 0.0.0.0 , Destination IP - 255.255.255.255, Source Mac - Mac address of Host, Destination Mac - FF:FF:FF:FF:FF:FF]
———————————————————

2.On Receiving DHCP Discover, Server sends a DHCP OFFER message to the client. The DHCPOFFER is a proposed configuration that may include IP address, DNS server address, and lease time. This message will be unicast and have the destination mac address of DHCP client's mac address. The source mac address will be that of the DHCP server.
[S.Mac - Mac address of Server , D.Mac - Mac address of Host]
———————————————————

3.If the Client finds the Offer agreeable, it sends DHCP REQUEST Message requesting those particular IP parameters. This message will be a Broadcast message.
[Source Mac - Mac address of Host, Destination Mac - FF:FF:FF:FF:FF:FF]
———————————————————

4.The Server on receiving the DHCP REQUEST makes the configuration official by sending a unicast DHCP ACK acknowledgment.
[Source Mac - Mac address of Server, Destination Mac - Mac address of Host]



What is the purpose of relay agent?

A DHCP relay agent is any host that forwards DHCP packets between clients and servers if server is not on the same physical subnet. Relay agents are used to forward requests and replies between clients and servers when they are not on the same physical subnet.
DHCP relay agent can be configured using the ip helper-address command.

13)ARP

https://www-sop.inria.fr/members/Vincenzo.Mancuso/ReteInternet/09_arp.pdf

14)Bandwidth, Delay and Latency

What is the difference between Bandwidth, Delay and Latency?

“Latency” is just the time that it takes us to do an operation. For example, the time that we spend waiting at a stoplight is a latency.
“Bandwidth” refers to how much data you can move in a given amount of time.

Bandwidth

Rated throughput capacity of a given network medium or protocol. Bandwidth is the different between the highest and lowest frequencies available for network signals.
https://searchnetworking.techtarget.com/definition/bandwidth

Latency

Delay between the time when a device receives a frame and the time that frame is forwarded out the destination port
Data latency is the time between a query and the result displaying on the screen.
https://whatis.techtarget.com/definition/latency

Delay

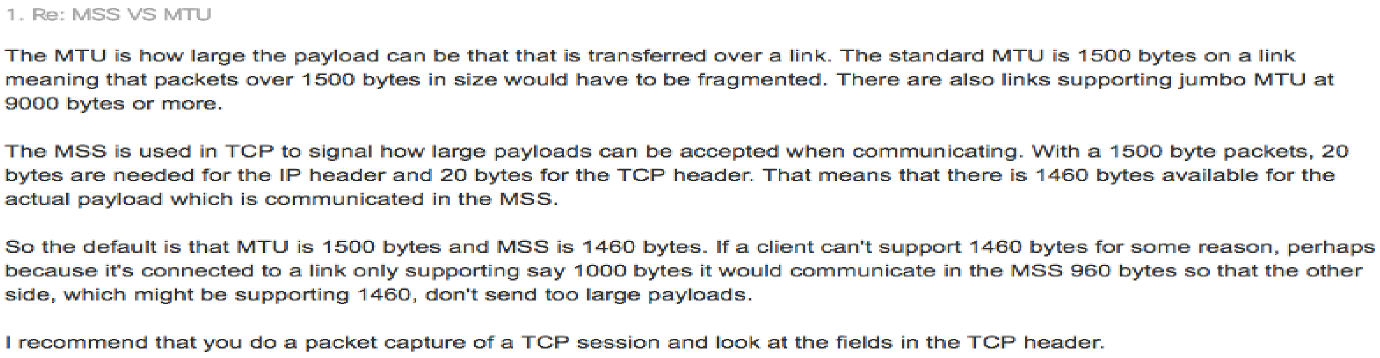
Length of time between the initiation of a transaction by a sender and the first response received by the sender.
Length of time required to move a packet from source to destination over a given path.

15)MTU, Window Size, MSS

What is MTU, Window Size, Segment (MSS)?
UDP does not have an MSS. UDP has datagrams, and maximum size of udp datagram is limited by many factor (maximum to the length of UDP datagram length header field =16 bits, that is including header). The hard limit is the size IP can carry.

MTU is always layer 1 and represents capacity of a physical link. But there are situations where protocols/software need to define MTU manually, eg IP MTU or MPLS MTU.

IP MTU is used to determine whether an ip packet needs to be fragmented or not.
MSS is always calculated from MTU to avoid any further fragmentation. In case no MTU value is found MSS with minimum size (576) will be send (as you know MSS = MTU - layer3 header + layer 2 header). and MTU is maximum packet size an interface can support.
Offset bit is set to Zero if there are more data to be sent. And it is set to one if there is no more fragmented packet to follow.



1. Re: MSS VS MTU

The MTU is how large the payload can be that that is transferred over a link. The standard MTU is 1500 bytes on a link meaning that packets over 1500 bytes in size would have to be fragmented. There are also links supporting jumbo MTU at 9000 bytes or more.

The MSS is used in TCP to signal how large payloads can be accepted when communicating. With a 1500 byte packets, 20 bytes are needed for the IP header and 20 bytes for the TCP header. That means that there is 1460 bytes available for the actual payload which is communicated in the MSS.

So the default is that MTU is 1500 bytes and MSS is 1460 bytes. If a client can't support 1460 bytes for some reason, perhaps because it's connected to a link only supporting say 1000 bytes it would communicate in the MSS 960 bytes so that the other side, which might be supporting 1460, don't send too large payloads.

I recommend that you do a packet capture of a TCP session and look at the fields in the TCP header.

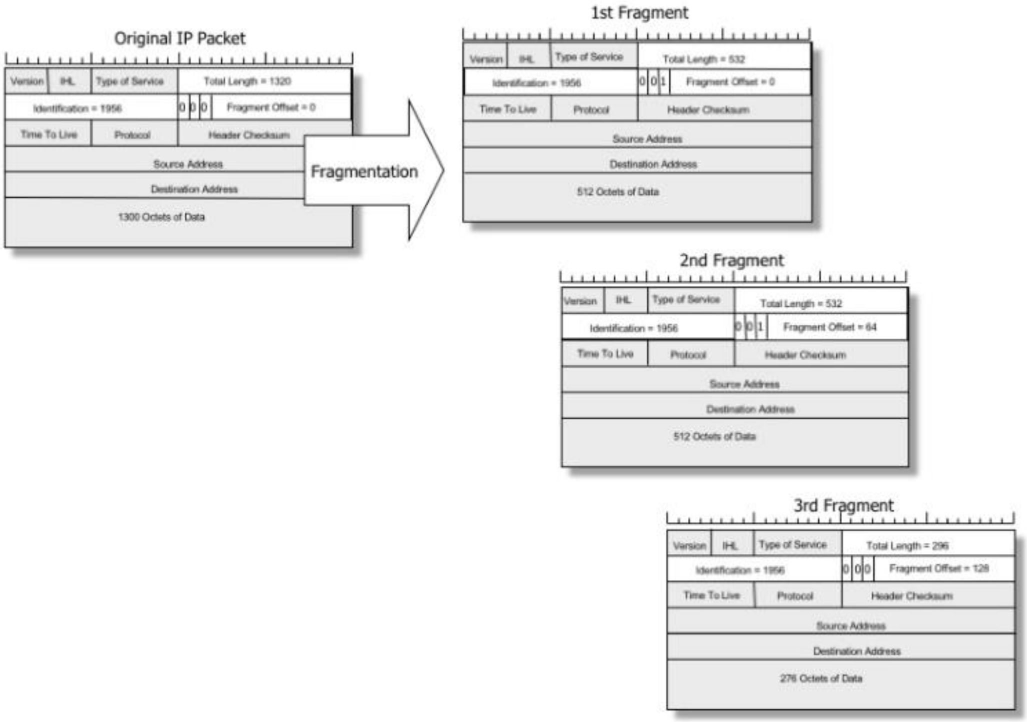
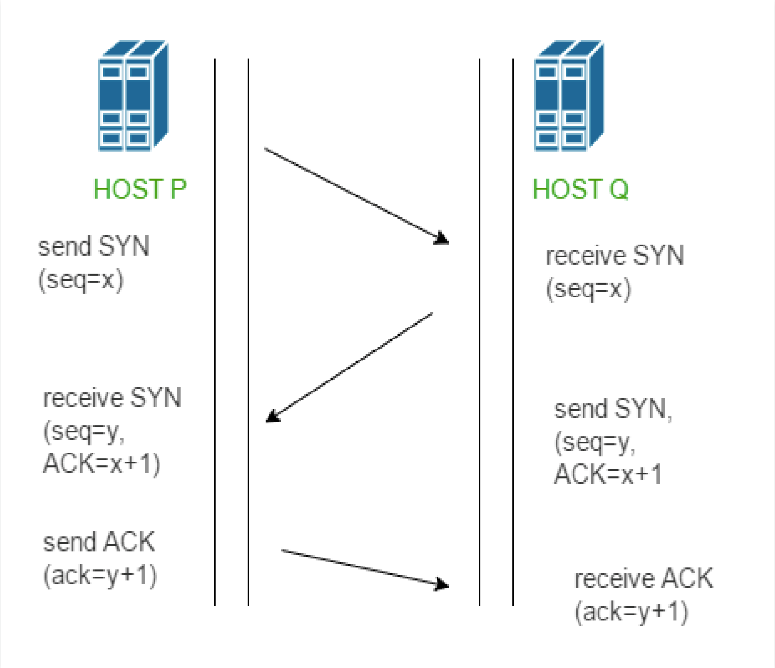


Figure 2 – Example of IPv4 Packet Fragmentation

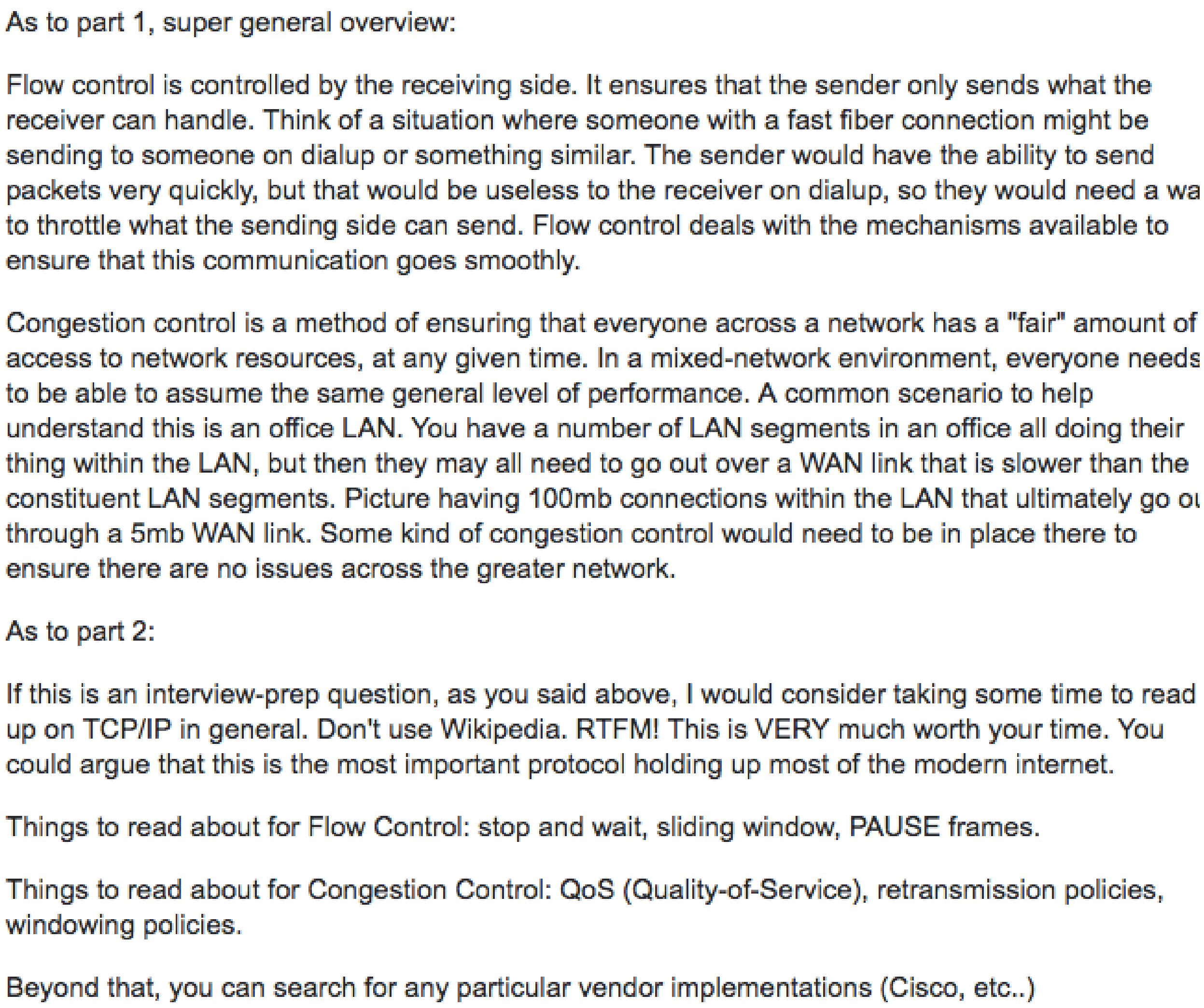
4) TCP and UDP



- **Step 1 (SYN) :** In the first step, client wants to establish a connection with server, so it sends a segment with SYN(Synchronize Sequence Number) which informs server that client is likely to start communication and with what sequence number it starts segments with
- **Step 2 (SYN + ACK):** Server responds to the client request with SYN-ACK signal bits set. Acknowledgement(ACK) signifies the response of segment it received and SYN signifies with what sequence number it is likely to start the segments with
- **Step 3 (ACK) :** In the final part client acknowledges the response of server and they both establish a reliable connection with which they will start eh actual data transfer

The steps 1, 2 establish the connection parameter (sequence number) for one direction and it is acknowledged. The steps 2, 3 establish the connection parameter (sequence number) for the other direction and it is acknowledged. With these, a full-duplex communication is established.

Note – Initial sequence numbers are randomly selected while establishing connections between client and server.



As to part 1, super general overview:

Flow control is controlled by the receiving side. It ensures that the sender only sends what the receiver can handle. Think of a situation where someone with a fast fiber connection might be sending to someone on dialup or something similar. The sender would have the ability to send packets very quickly, but that would be useless to the receiver on dialup, so they would need a wa to throttle what the sending side can send. Flow control deals with the mechanisms available to ensure that this communication goes smoothly.

Congestion control is a method of ensuring that everyone across a network has a "fair" amount of access to network resources, at any given time. In a mixed-network environment, everyone needs to be able to assume the same general level of performance. A common scenario to help understand this is an office LAN. You have a number of LAN segments in an office all doing their thing within the LAN, but then they may all need to go out over a WAN link that is slower than the constituent LAN segments. Picture having 100mb connections within the LAN that ultimately go ou through a 5mb WAN link. Some kind of congestion control would need to be in place there to ensure there are no issues across the greater network.

As to part 2:

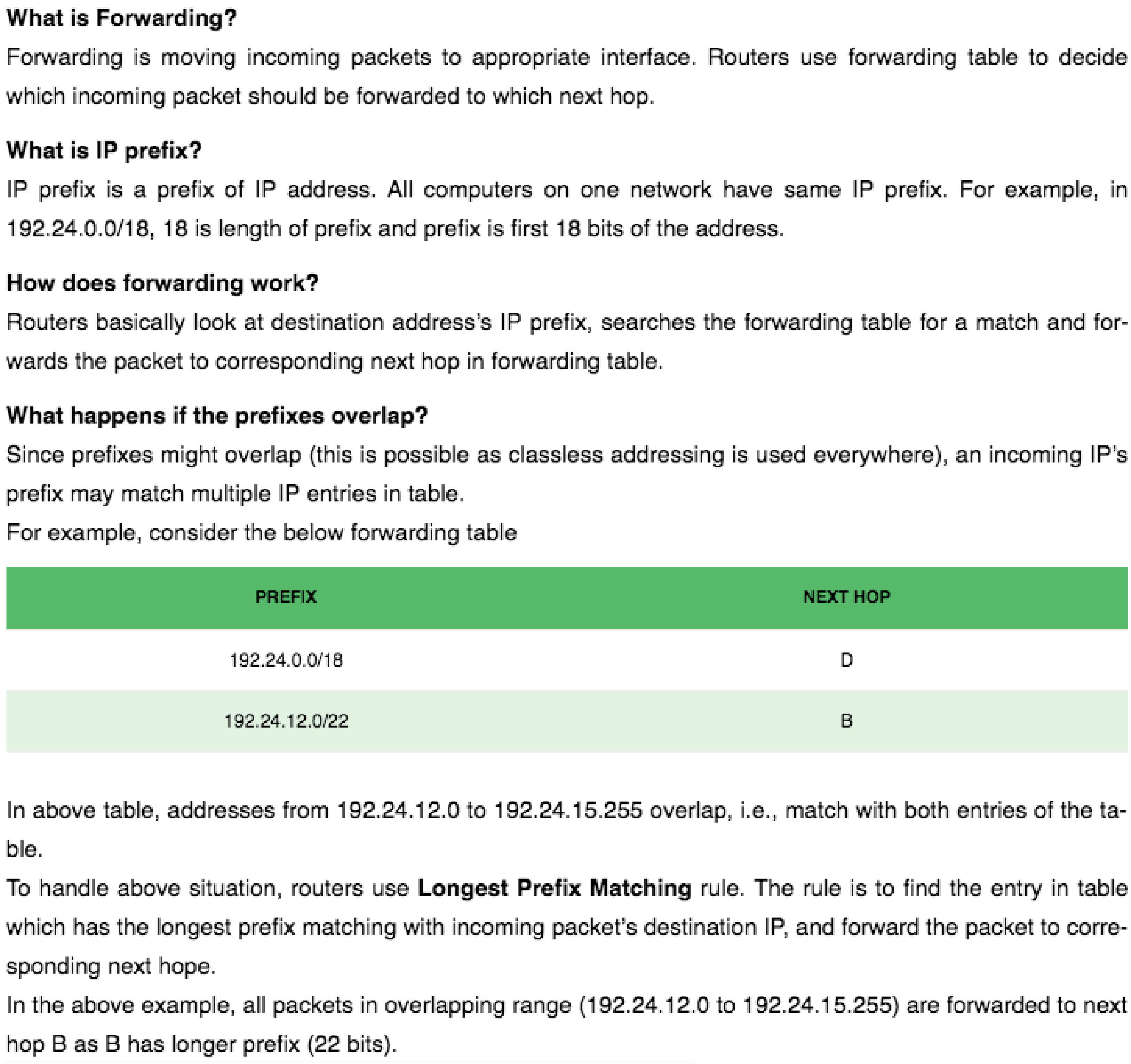
If this is an interview-prep question, as you said above, I would consider taking some time to read up on TCP/IP in general. Don't use Wikipedia. RTFM! This is VERY much worth your time. You could argue that this is the most important protocol holding up most of the modern internet.

Things to read about for Flow Control: stop and wait, sliding window, PAUSE frames.

Things to read about for Congestion Control: QoS (Quality-of-Service), retransmission policies, windowing policies.

Beyond that, you can search for any particular vendor implementations (Cisco, etc..)

5) Prefix Match and Route Summarization
https://www.youtube.com/watch?v=QqEcCzhlWis



What is Forwarding?

Forwarding is moving incoming packets to appropriate interface. Routers use forwarding table to decide which incoming packet should be forwarded to which next hop.

What is IP prefix?

IP prefix is a prefix of IP address. All computers on one network have same IP prefix. For example, in 192.24.0.0/18, 18 is length of prefix and prefix is first 18 bits of the address.

How does forwarding work?

Routers basically look at destination address's IP prefix, searches the forwarding table for a match and forwards the packet to corresponding next hop in forwarding table.

What happens if the prefixes overlap?

Since prefixes might overlap (this is possible as classless addressing is used everywhere), an incoming IP's prefix may match multiple IP entries in table.

For example, consider the below forwarding table

PREFIX	NEXT HOP
192.24.0.0/18	D
192.24.12.0/22	B

In above table, addresses from 192.24.12.0 to 192.24.15.255 overlap, i.e., match with both entries of the table.

To handle above situation, routers use **Longest Prefix Matching** rule. The rule is to find the entry in table which has the longest prefix matching with incoming packet's destination IP, and forward the packet to corresponding next hope.

In the above example, all packets in overlapping range (192.24.12.0 to 192.24.15.255) are forwarded to next hop B as B has longer prefix (22 bits).

6) State full and Stateless Firewall

A firewall can be described as being either Stateful or Stateless.

STATELESS Firewalls

Stateless firewalls watch network traffic and restrict or block packets based on source and destination addresses or other static values. They're not 'aware' of traffic patterns or data flows. A stateless firewall uses simple rule-sets that do not account for the possibility that a packet might be received by the firewall 'pretending' to be something you asked for.

A stateless firewall filter, also known as an access control list (ACL), does not statefully inspect traffic. Instead, it evaluates packet contents statically and does not keep track of the state of network connections.

Purpose of Stateless Firewall Filters

The basic purpose of a stateless firewall filter is to enhance security through the use of packet filtering. Packet filtering enables you to inspect the components of incoming or outgoing packets and then perform the actions you specify on packets that match the criteria you specify. The typical use of a stateless firewall filter is to protect the Routing Engine processes and resources from malicious or untrusted packets.

STATEFUL Firewall

Stateful firewalls can watch traffic streams from end to end. They are aware of communication paths and can implement various IP Security (IPsec) functions such as tunnels and encryption. In technical terms, this means that stateful firewalls can tell what stage a TCP connection is in (open, open sent, synchronized, synchronization acknowledge or established). It can tell if the MTU has changed and whether packets have fragmented. etc.

Neither is really superior and there are good arguments for both types of firewalls. Stateless firewalls are typically faster and perform better under heavier traffic loads. Stateful firewalls are better at identifying unauthorized and forged communications.

7) DNS
https://www.youtube.com/watch?v=-4F5TctHI4c
https://www.youtube.com/watch?v=833Qnc-7-ug
DNS data is divided into small parts so that it can be managed easily, which is known as DNS zones. DNS zones are DSN data base's file, which usually are in text format stored in various multiple DNS servers.

Two types of DNS zones types: Forward and reverse lookup zones
Forward is used for name to ip resolution. Reverse lookup zones are used for ip to name resolution. (used for troubleshooting)
We can create various sub zones in these zones which are: Primary, Secondary, Active directory integrated zones, and stub zones.

- Primary zones are DNS databases' read write copy. We can add/edit/remove entries from these files. It is installed on a server called primary server. Only one primary server.
- Secondary zones are read only copy of DNS's database, where we cannot modify the entries. **It is used for redundancy for name resolution**. It can take the data from 3 ways: from the primary zone, or from its peer (secondary zone) or from active directory integrated zone. Problem? Contains too much information which is never going to be used by a particular network device.
- Stub zones is like secondary zone (read only). It's a pointer towards DNS server of another zone. Only name server records of that zone are updated (not the host or some other's data). **Stub zones are used to make the query process faster.** Contains only partial data.
- AD integrated zones are joint with the Active directory. An active directory zone is simply a primary zone stored in active directory. We can only see these files where the active directory is installed (usually it is installed in Domain Controllers). Read write copy. **Used for fault tolerance, high availability, redundancy.**

DNS records
https://www.youtube.com/watch?v=6uEwzkfViSM

DNS records are basically mapping files that tell the DNS server which IP address each domain is associated with, and how to handle requests sent to each domain. When someone visits a web site, a request is sent to the DNS server and then forwarded to the web server provided by a web hosting company, which contain the data contained on the site.

Various strings of letters are used as commands that dictate the actions of the DNS server, and these strings of commands are called DNS syntax. Some DNS records syntax that are commonly used in nearly all DNS record configurations are A, AAAA, CNAME, MX, PTR, NS, SOA, SRV, TXT, and NAPTR. The following paragraph details the meaning and usage of each of these syntax.

DNS Syntax Types Explained

An "A" record, which stands for "address" is the most basic type of syntax used in DNS records, indicating the actual IP address of the domain. The "AAAA" record is an IPV6 address record that maps a hostname to a 128-bit Ipv6 address. Regular DNS addresses are mapped for 32-bit IPv4 addresses.

The "CNAME" record stands for "canonical name" and serves to make one domain an alias of another domain. CNAME is often used to associate new subdomains with an existing domain's DNS records.

The "MX" record stands for "mail exchange" and is basically a list of mail exchange servers that are to be used for the domain.

The “PTR” record stands for “pointer record” and maps an Ipv4 address to the CNAME on the host.

The “NS” record stands for “name server” and indicates which Name Server is authoritative for the domain.

An “SOA” record stands for “State of Authority” and is easily one of the most essential DSN records because it stores important information like when the domain was last updated and much more.

An “SRV” record stands for “service” and is used to define a TCP service on which the domain operates.

A “TXT” record lets the administrator insert any text they'd like into the DNS record, and it is often used for denoting facts about the domain.

https://my.bluehost.com/hosting/help/508

8) Iptables

https://www.booleanworld.com/depth-guide-iptables-linux-firewall/

https://www.tecmint.com/basic-guide-on-iptables-linux-firewall-tips-commands/

Firewall decides fate of packets incoming and outgoing in system. IPTables is a rule based firewall and it is pre-installed on most of Linux operating system. By default it runs without any rules. IPTables was included in Kernel 2.4, prior it was called ipchains or ipfwadm. IPTables is a front-end tool to talk to the kernel and decides the packets to filter. This guide may help you to rough idea and basic commands of IPTables where we are going to describe practical iptables rules which you may refer and customized as per your need.
Different services is used for different protocols as:

- iptables applies to IPv4.
- ip6tables applies to IPv6.
- arptables applies to ARP.
- ebtables applies to Ethernet frames..

IPTables main files are:

- /etc/init.d/iptables – init script to start|stop|restart and save rulesets.
- /etc/sysconfig/iptables – where Rulesets are saved.
- /sbin/iptables – binary.

There are at present three tables.

- Filter
- NAT
- Mangle

At present, there are total four chains:

- INPUT : Default chain originating to system.
- OUTPUT : Default chain generating from system.
- FORWARD : Default chain packets are send through another interface.
- RH-Firewall-1-INPUT : The user-defined custom chain.

Note: Above main files may slightly differ in Ubuntu Linux.
How to start, stop and restart Iptabe Firewall.

```
# /etc/init.d/iptables start

# /etc/init.d/iptables stop

# /etc/init.d/iptables restart
```

To start IPTables on system boot, use the following command.

```
#chkconfig --level 345 iptables on
```

Saving IPTables rulesets with below command. Whenever system rebooted and restarted the IPTables service, the exsiting rules flushed out or reset. Below command save TPTables rulesets in /etc/sysconfig/iptables file by default and rules are applied or restored in case of IPTables flushes out.

```
#service iptables save
```

Checking the status of IPTables / Firewall. Options “-L” (List ruleset), “-v” (Verbose) and “-n” (Displays in numeric format).

```
[root@tecmint ~]# iptables -L -n -v



Chain INPUT (policy ACCEPT 0 packets, 0 bytes)

 pkts bytes target     prot opt in     out     source
destination

    6   396 ACCEPT     all  --  *      *       0.0.0.0/0
0.0.0.0/0           state RELATED,ESTABLISHED

    0     0 ACCEPT     icmp --  *      *       0.0.0.0/0
0.0.0.0/0

    0     0 ACCEPT     all  --  lo     *       0.0.0.0/0
0.0.0.0/0

    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0
0.0.0.0/0           state NEW tcp dpt:22
```

```
    0     0 REJECT     all  --  *      *       0.0.0.0/0
0.0.0.0/0           reject-with icmp-host-prohibited



Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)

 pkts bytes target     prot opt in     out     source
destination

    0     0 REJECT     all  --  *      *       0.0.0.0/0
0.0.0.0/0           reject-with icmp-host-prohibited



Chain OUTPUT (policy ACCEPT 5 packets, 588 bytes)

 pkts bytes target     prot opt in     out     source
destination
```

Display IPTables rules with numbers. With the help of argument “–line-numbers” you can append or remove rules.

```
[root@tecmint ~]# iptables -n -L -v --line-numbers



Chain INPUT (policy ACCEPT 0 packets, 0 bytes)

num   pkts bytes target     prot opt in     out     source
destination

1        51  4080 ACCEPT     all  --  *      *       0.0.0.0/0
0.0.0.0/0           state RELATED,ESTABLISHED

2         0     0 ACCEPT     icmp --  *      *       0.0.0.0/0
0.0.0.0/0

3         0     0 ACCEPT     all  --  lo     *       0.0.0.0/0
0.0.0.0/0
```

```
4        0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0
0.0.0.0/0           state NEW tcp dpt:22


5        0     0 REJECT     all  --  *      *       0.0.0.0/0
0.0.0.0/0           reject-with icmp-host-prohibited




Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)


num   pkts bytes target     prot opt in     out     source
destination


1        0     0 REJECT     all  --  *      *       0.0.0.0/0
0.0.0.0/0           reject-with icmp-host-prohibited




Chain OUTPUT (policy ACCEPT 45 packets, 5384 bytes)


num   pkts bytes target     prot opt in     out     source
destination
```

Flushing or deleting IPTables rules. Below command will remove all the rules from tables. Take rulesets backup before executing above command.

```
[root@tecmint ~]# iptables -F
```

Deleting or appending rules, let us first see the rules in chains. Below commands shall display rulesets in INPUT and OUTPUT chains with rule numbers which will help us to add or delete rules

```
[root@tecmint ~]# iptables -L INPUT -n --line-numbers

Chain INPUT (policy ACCEPT)
num  target     prot opt source               destination
1    ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0           state
RELATED,ESTABLISHED
2    ACCEPT     icmp --  0.0.0.0/0            0.0.0.0/0
3    ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0
4    ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           state NEW
tcp dpt:22
```

```
5    REJECT     all  --  0.0.0.0/0            0.0.0.0/0           reject-
with icmp-host-prohibited
[root@tecmint ~]# iptables -L OUTPUT -n --line-numbers
Chain OUTPUT (policy ACCEPT)
num  target     prot opt source               destination
```

Let’s say if you want to delete rule no 5 from INPUT chain. Use the following command.

```
[root@tecmint ~]# iptables -D INPUT 5
```

To insert or append rule to INPUT chain in between 4 and 5 ruleset.

```
[root@tecmint ~]# iptables -I INPUT 5 -s ipaddress -j DROP
```

We have just tried to cover basic usages and functions of IPTables for begineer. You may create complex rules once you have complete understanding of TCP/IP and good knowledge of your setup.

9) HTTP

https://www.jmarshall.com/easy/http/

HTTP is the network protocol of the Web. It is both simple and powerful. Knowing HTTP enables you to write Web browsers, Web servers, automatic page downloaders, link-checkers, and other useful tools.

HTTP stands for **Hypertext Transfer Protocol**. It's the network protocol used to deliver virtually all files and other data (collectively called *resources*) on the World Wide Web, whether they're HTML files, image files, query results, or anything else. Usually, HTTP takes place through TCP/IP sockets (and this tutorial ignores other possibilities).

A browser is an *HTTP client* because it sends requests to an *HTTP server* (Web server), which then sends responses back to the client. The standard (and default) port for HTTP servers to listen on is 80, though they can use any port.

What are "Resources"?

HTTP is used to transmit *resources*, not just files. A resource is some chunk of information that can be identified by a URL (it's the **R** in **URL**). The most common kind of resource is a file, but a resource may also be a dynamically-generated query result, the output of a CGI script, a document that is available in several languages, or something else.

While learning HTTP, it may help to think of a resource as similar to a file, but more general. As a practical matter, almost all HTTP resources are currently either files or server-side script output.

10) VLAN

A VTP client can overwrite a VTP server if the revision number is higher because a VTP server is also a VTP client.

11) Load balancers

https://kemptechnologies.com/load-balancer/load-balancing-algorithms-techniques/

https://www.digitalocean.com/community/tutorials/what-is-load-balancing

→Load balancing is a technique used to distribute workloads uniformly across servers or other compute resources to optimize network efficiency, reliability and capacity. Load balancing is performed by an appliance -- either physical or virtual -- that identifies in real time which server in a pool can best meet a given client request, while ensuring heavy network traffic doesn't unduly overwhelm a single server.

→load balancing provides failover. If one server fails, a load balancer immediately redirects its workloads to a backup server, thus mitigating the impact on end users.

→Load balancing is usually categorized as supporting either Layer 4 or Layer 7. Layer 4 load balancers distribute traffic based on transport data, such as IP addresses and Transmission Control Protocol (TCP) port numbers. Layer 7 load-balancing devices make routing decisions based on application-level characteristics that include HTTP header information and the actual contents of the message, such as URLs and cookies. Layer 7 load balancers are more common, but Layer 4 load balancers remain popular, particularly in edge deployments.

How load balancing works

Load balancers handle incoming requests from users for information and other services. They sit between the servers that handle those requests and the internet. Once a request is received, the load balancer first determines which server in a pool is available and online and then routes the request to that server. During times of heavy loads, a load balancer can dynamically add servers in response to spikes in traffic. Conversely, they can drop servers if demand is low.

A load balancer can be a physical appliance, a software instance or a combination of both. Traditionally, vendors have loaded proprietary software onto dedicated hardware and sold them to users as stand-alone appliances -- usually in pairs, to provide failover if one goes down. Growing networks require purchasing additional and/or bigger appliances.

In contrast, software load balancing runs on virtual machines (VMs) or white box servers, most likely as a function of an application delivery controller (ADC). ADCs typically offer additional features, like caching, compression, traffic shaping, etc. Popular in cloud environments, virtual load balancing can offer a high degree of flexibility -- for example, enabling users to automatically scale up or down to mirror traffic spikes or decreased network activity.

Methods:

1)Round Robin

2)least connections- assign the server haing less current conections

3)Least time algorithm – considers both server response time and active connection. Servers that have a better balance of fewest connections and fastest response time receive a greater proportion of the connections.

4) predictive -The servers in the specified pool with better performance rankings that are currently improving, rather than declining, receive a higher proportion of the connections.

5) fastest method: The Fastest method passes a new connection based on the fastest response time of all servers. This method may be particularly useful in environments where servers are distributed across different logical networks. On the BIG-IP, only servers that are active will be selected.

12) Common Port Number

FTP-20,21

SSH-22

Telnet-23

IPSec-50,51

DNS-53

DHCP-67,68

HTTP-80

HTTPS-443

BGP-179

SMTP-25

13) Host A and B are unable to communicate with each other

https://www.computerworld.com/article/2494764/data-center/server-s-down--how-do-i-find-out-what-s-wrong-.html

i) First check on the network from host A, whether a computer from the same network as of host A is able to communicate with the host B or not? If it is able to communicate then there is a problem with host A. If unable to communicate then there is a problem with host B or somewhere in the network.

ii) I will then use ethtool on host A to actually verify whether host A is actually configured on the network or not. With ethtool we can see the speed and duplex too which should match on the interfaces to work. So, we can do speed and duplex to auto.

iii) Next, I will check to see whether the interface and configured correctly and is up and running which I will do with the help of eth 0 command.

iv) Next I will check whether a default gateway is setup between them or not? Using route command in linux I can check to see the routing table which will show me the default gateway.

v) Once I have seen this I will try to ping the Ip address and if a reply comes back then it's working fine and if it dosent then it can mean that the fireall may be blocking the icmp packets.

- Help! My network is down: https://netbeez.net/blog/troubleshooting-network-problems/
- IP address in CIDR: http://droptips.com/cidr-subnet-masks-and-usable-ip-addresses-quick-reference-guide-cheat-sheet

14) ACL :

ACL is a set of rules which will allow or deny the specific traffic moving through the router.

It is Layer3 Security which controls the flow of traffic from 1 router to another.Also called PACKET FILTERING FIREWALL.

1)filtering

2)classification

If we have a VPN that encrypts traffic between the two routers. Whenever we create a VPN we can use an access-list to "select" what traffic should be encrypted. We can use an access-list to "select" traffic, this is called classification.

Wildcard mask: tells the router which portion of the bits to match/ignore.

Global subnet mask-255.255.255.255

Customized subnet mask-255.255.255.0

Wild card mask(from abv 2)-0.0.0.255(G-C)

Wild card mask for host-0.0.0.0

FTP:

FTP is an application protocol ,which use to transfer files to/from remote file system. Its based on client-server model. FTP client contacts FTP server on port 21 using TCP and browse remote directoy by sending different commands. For example if FTP client wants to have list of all files and folders present on FTP server, it issue "list" command on port 21, and FTP server sends the list on port 20,which is used for data connection. (USER,PASS,LIST,RETR,STOR)

15) SNMP:

Simple Network Management Protocol (SNMP) is an application-layer protocol used to manage and monitor network devices and their functions using NMS. Provides common language for network . SNMP is supported on routers,switches,ap,printers,scanners etc. SNMP can be used to monitor services such as Dynamic Host Configuration Protocol (DHCP).

SNMP performs many functions rely on PUSH/PULL communications between N/W device and management system .

SNMP issue read/write commands .ex: resetting password or change configuration settings.

Give bandwidth, cpu, memory in use with some SNMP mangers to admin.

GET: Generated by the SNMP manager and sent to an agent to obtain the value of a variable, identified by its OID, in an MIB(management info base) .
RESPONSE: Sent by the agent to the SNMP manager, issued in reply to a GET request. Contains the values of the requested variables.
GETNEXT: Sent by the SNMP manager to agent to retrieve the values of the next OID in the MIB's hierarchy.
GETBULK: Sent by the SNMP manager to the agent to obtain large tables of data by performing multiple GETNEXT commands.
SET: Sent by the SNMP manager to the agent to issue configurations or commands.
TRAP: An asynchronous alert sent by the agent to the SNMP manager to indicate a significant event, such as an error or failure, has occurred.

16) Simple Mail Transfer Protocol (SMTP)

Email is emerging as the one of the most valuable service in 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.

Working: 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 has persistant connections.

3 phase: handshaking, transfer, closure.

17) GRE (Generic routing Encapsulation):

Generic Routing Encapsulation (GRE) is a protocol that encapsulates packets in order to route other protocols over IP networks.
GRE was developed as a tunneling tool meant to carry any OSI Layer 3 protocol over an IP network. In essence, GRE creates a private point-to-point connection like virtual private network (VPN).
In contrast to IP-to-IP tunneling, GRE tunneling can transport multicast and IPv6 traffic between networks.
Advantages of GRE tunnels include the following:
GRE tunnels encase multiple protocols over a single-protocol backbone.
GRE tunnels provide workarounds for networks with limited hops.
GRE tunnels connect discontinuous sub-networks.
GRE tunnels allow VPNs across wide area networks (WANs).
While GRE provides a stateless, private connection, it is not considered a secure protocol because it does not use encryption like the IP Security (IPsec)
Cookies: An id sent from the web server to be stored in a client's web browser for tracking and security. As for example, when we are doing online shopping on amazon and adding something in our shopping cart, it will be saved in cookies from amazon server, so next time when we again login through the same account, it will have the last session information.
18)CD*N*: *is short for **c**ontent **d**elivery **n**etwork.* It is a system of distributed servers (network) that deliver pages and other Web content to a user, based on the geographic locations of the user, the origin of the webpage and the content delivery server.
→How CDNs Work
Servers nearest to the website visitor respond to the request. The content delivery network copies the pages of a website to a network of servers that are dispersed at geographically different locations, caching the contents of the page. When a user requests a webpage that is part of a content delivery network, the CDN will redirect the request from the originating site's server to a server in the CDN that is closest to the user and deliver the cached content. CDNs will also communicate with the originating server to deliver any content that has not been previously cached.
The process of bouncing through CDNs is nearly transparent to the user. The only way a user would know if a CDN has been accessed is if the delivered URL is different than the URL that has been requested.

26) **Reverse Proxy:** A **reverse proxy** accepts a request from a client, forwards it to a server that can fulfill it, and returns the server's response to the client.

Whereas deploying a load balancer makes sense only when you have multiple servers, it often makes sense to deploy a reverse proxy even with just one web server or application server. You can think of the reverse proxy as a website's "public face." Its address is the one advertised for the website, and it sits at the edge of the site's network to accept requests from web browsers and mobile apps for the content hosted at the website. The benefits are two-fold:

Increased security – No information about your backend servers is visible outside your internal network, so malicious clients cannot access them directly to exploit any vulnerabilities. Many reverse proxy servers include features that help protect backend servers from distributed denial-of-service (DDoS) attacks, for example by rejecting traffic from particular client IP addresses (blacklisting), or limiting the number of connections accepted from each client.

Increased scalability and flexibility – Because clients see only the reverse proxy's IP address, you are free to change the configuration of your backend infrastructure. This is particularly useful In a load-balanced environment, where you can scale the number of servers up and down to match fluctuations in traffic volume.

Another reason to deploy a reverse proxy is for web acceleration – reducing the time it takes to generate a response and return it to the client. Techniques for web acceleration include the following:

Compression – Compressing server responses before returning them to the client (for instance, with gzip) reduces the amount of bandwidth they require, which speeds their transit over the network.

SSL termination – Encrypting the traffic between clients and servers protects it as it crosses a public network like the Internet. But decryption and encryption can be computationally expensive. By decrypting incoming requests and encrypting server responses, the reverse proxy frees up resources on backend servers which they can then devote to their main purpose, serving content.

Caching – Before returning the backend server's response to the client, the reverse proxy stores a copy of it locally. When the client (or any client) makes the same request, the reverse proxy can provide the response itself from the cache instead of forwarding the request to the backend server. This both decreases response time to the client and reduces the load on the backend server

27)Proxy server:

- it acts as intermediary between Server and client(requesting machine).A client connects to the proxy server, requesting some service, such as a file, connection, web page, or other resource available from a different server and the proxy server evaluates the request as a way to simplify and control its complexity.

proxies reduces overload of main server by serving requests by themselves. Proxy servers can make a network virtually invisible to external users.

An advantage of a proxy server is that its cache can serve all users. If one or more Internet sites are frequently requested, these are likely to be in the proxy's cache, which will improve user response time.

28) Cloud vs traditional and how will you improve:

In an era of tight budgets many businesses, from enterprise level to small and medium sized business, are looking for efficient new ways to manage their web hosting needs. The hosting environment is changing, and some are now looking beyond traditional setups, and into the possibilities of cloud hosting.

Traditional Hosting

Traditional hosting comes mainly in two forms, *dedicated* and *shared*. With dedicated hosting, a company pays for the complete resources of one or more servers from a service provider. The client has a set amount of dedicated bandwidth, CPU, RAM, and drive space, and the client has full control over the server's resources.

With shared hosting, which is more common among small and medium sized businesses, the client pays for a set amount of space (storage) on a single server, and that server's resources are shared by a number of other websites. It's a cost-efficient, low-maintenance way to host a website or application, and the hosting company is responsible for managing, maintaining, and updating the units.

Traditional hosting, especially shared hosting, has its drawbacks though. Because the resources of a single server are shared among a number of different websites, spikes in traffic to those websites can mean decreased performance for your own. Security breaches and other performance issues on other sites make take yours down as well. And there's a single point of failure. If the server itself experiences technical problems, everyone hosted on that server will be affected.

With shared hosting, you're also paying for a set amount of storage and processing power. If you have a predictable flow of traffic, this may be a good solution for you. But if your traffic is increasing rapidly, or if you see sudden spikes in traffic due to a new product or feature, you may be constrained the amount of storage you currently have.

You will need to adapt by purchasing additional server space to add to your storage space and processing power. But if traffic falls again, you will be paying for resources that you aren't using.

Cloud Hosting : Cloud hosting offers a level of scalability that traditional hosting can't. Cloud hosting companies provide virtual space on an on-demand, as-needed basis. Instead of paying for a set amount of space upfront on a single server, the user pays as they go for what they actually use.

With cloud hosting, the load is balanced across a cluster of multiple servers. The information and applications contained on those servers are mirrored across the whole cluster, meaning that if an individual server goes down, there is no lost information or downtime. Because of this redundancy, cloud hosting is much more elastic and resilient. Problems with one website or application are unlikely to affect your bandwidth or performance.

Cloud hosting companies provide Infrastructure-as-a-Service (IaaS). They house, run, and maintain all of the necessary hardware, and the customer pays for the resources the use, similar to how we pay for utilities like electricity.

IT departments don't need to invest in in-house server hardware. And customers don't need to pay for up front for extra storage or processing capacity that they don't use. Cloud hosting is more quickly scalable than traditional hosting. If an application or website receives more or less traffic, the cloud servers scale up and down automatically. With cloud hosting, there's no need to manually add or remove server space as there is in shared hosting.

Cloud hosting is still a relatively new technology, and many who have experience with traditional hosting are hesitant to move to something different. Shared hosting provides consumers with a convenient, low-entry hosting solution, and many users never experience problems. But if you're looking for a low-cost, flexible, easily scalable hosting solution, it may be time to move to the cloud.

29-A) Browsing 2 different things on the same browser, how it will be defined which packets of data should be sent where :

Packet has destination ip address in IP Packet, so each router on the next hop will know which route to take to reach to that destination.

29-B) is every website use same port for same website on browser?

Local ports are going to be different ,but the remote ports are gonna be 80 for all http request or 443 for all https.

30) POP,IMAP,EMAil:

Both POP (Post Office Protocol) and IMAP (Internet Message access protocol) allow people to get access to their email from a remote server; however, that is where most similarities end. POP simply downloads email to your computer, and usually (but not always) deletes the email from the remote server. The problems arise if you have more than one device where you read your mail (desktop, laptop, tablet or phone). Here's why it's bad: You have to delete or file the same email on every device

Logging into each device, you will see lots of unread emails with no indication of which you deleted, read, flagged or filed Any folders you created and organize on one device won't be replicated on the other devices

IMAP allows users to store their email on remote servers. This two-way protocol also allows the user to synchronize their email among multiple devices, which is extremely important today, when most people have at least two devices - their laptop and smartphone.

IMAP (Internet Messaging Access Protocol)
- Emails are stored on the server.
- Sent messages are stored on the server.
- Messages can be synced and accessed across multiple devices.

POP3 (Post Office Protocol)
- Emails are stored on a single device.
- Sent messages are stored on a single device.
- Emails can only be accessed from a single device.
- If you want to keep messages on the server, make sure the setting "Keep email on server" is enabled or all messages are deleted from the server once downloaded to the app or software.

31) BGP: BGP is inter AS routing and exterior gateway protocol. BGP runs on the OSI Application Layer and get reliability with the use of TCP (port 179).
Border Gateway Protocol advertises, learns and chooses the best paths inside the global Internet. When two ISPs connect, they typically use BGP to exchange routing information. Enterprises also sometimes uses BGP to exchange routing information with ISPs, allowing the Enterprise routers to learn Internet routes. when we have multiple Internet connections and we want to influence some packets to take one path and some packets to take another we use BGP.

BGP Tables / Databas :

Neighbor Table: This contains the list of all configured BGP neighbors. The 'show ip bgp summary' command would show the neighbor table

Forwarding Table / Routing Information Base (RIB): This contains a list of networks along their path and attributes which are known by BGP. The 'show ip bgp' command would display the information.

Routing Table: This table lists the best path to the destination networks and also the next hop for each network. Like other protocols, the 'show ip route' will show the routing table

Multihoming :A site with a single ISP connection is called Singled-homed. That is making use of more than one ISPs for your network i.e. at a time you connect to more than one ISP. Multihoming is used for redundancy and backup, in case of failure of one of the ISPs the other is still active and also for better performance

BGP Peers (neighbors) :The routers used in BGP (called speakers) must form neighbor relationship (called peers) with other routers to function properly.

iBGP (internal BGP) : The routers that form neighbor relationship with the AS. The iBGP neighbors need not be directly connected.

eBGP (external BGP): The routers that form neighbor relationship between different AS. The eBGP neighbors need to be directly connected.

BGP Message Types

Open: This message is sent after BGP neighbor is configured. This message is sent to establish or form peering with that neighbor. This message contains information such as the AS number, router ID and the hold-time.

Update: Routing information between peers is transferred using this message. This message would include, new routes, withdrawn routes and the path attributes.

Keepalive: Similar to Hellos in the other routing protocols, BGP uses keepalives which help in keeping the peering session active. The BGP peers exchange keepalive messages every 60 seconds.

Notification: In the event of a problem which causes the router to end the BGP peering session, the BGP neighbor is send a notification message and the connection is closed. Also report errors in previous messages.

BGP Peering State

When a BGP peer session is formed, it passes through some states. This process is called the BGP FSM (Finite State Machine).

Idle - The BGP process is either administratively down or waiting for the the next retry attempt. At this state there is no peering. The router would be searching the routing table to see if a route exists to reach a neighbor.

Connect - The BGP process is waiting for the TCP connection to be completed. If it is successful, it will continue to the OpenSent state. In case it fails, it will continue to the Active state.

Active - BGP will try another TCP three-way handshake to establish a connection with the remote BGP neighbor. If it is successful, it will move to the OpenSent state.

Opensent - The TCP connection exists, and a BGP Open message has been sent to the peer, but the matching Open message has not yet been received from the other router.

Openconfirm - An Open message has been both sent to and received from the other router. Next step is to receive a BGP Keepalive message (to confirm that all neighbor-related parameters match) or a BGP Notification message (to learn that there is some mismatch in neighbor parameters).The router may also go into ACTIVE state if there is no response to the Open message.

Established: When the peering has been established. At this state the routing begins. This is the desired state.

The command '**show ip bgp neighbors**' and '**show ip bgp summary'** can help us verify BGP peering status.

BGP Attributes

AS Path: Is a Well-known, Mandatory attribute. It contains the list of AS though which the updates have traversed. The path that has the shortest AS Path is most preferred or desirable.

Next Hop: Is a Well-known, Mandatory attribute. As BGP is an AS to AS protocol, Next hop would NOT mean next router but instead Next Hop is equal to the IP to reach the next AS. This it is the IP address of the next AS to reach a given network.

Origin: Is a Well-known, Mandatory attribute. Tells some information about a network i.e. it tells how BGP learned about that network.

– IGP 'i' means it is advertised in BGP using a network command
– EGP 'e' means it is redistributed from some other source (EGP)
– Incomplete '?' means redistributed into BGP from an IGP or static

Weight: Is a partial attribute. This is a Cisco proprietary attribute. It gives information on how to exit an AS. The path with the highest weight is more desirable. This is of local significance. The default weight is 0 for learned routes and 32768 for locally injected routes.

Local Preference: Is a Well-known discretionary attribute. It is a value that tells the iBGP peers which path needs to be used to exit or leave the AS. The path with the highest preference is considered more desirable. These are advertised only the iBGP neighbor within the AS and the default value is 100.

MED (Multi-Exit Discriminator): Is an Optional, Non-Transitive attribute. It define how the data traffic should enter an AS i.e. it shows the neighboring AS the available paths to select if some data need to arrive to your AS.

32) OSPF:

OSPF is link state routing protocol which works on digikrtr algoritham , to initially construct the shortest path and then populate routing table with best path.its a classless protocol, so supports vlsm.

The main advantage of OSPF is it supports unlimited hop counts. It supports equal load balancing. Updates through multicast 224.0.0.5/6.

	BGP	OSPF	IS-IS	EIGRP	IGRP	RIPv2
Metric	Attributes	Cost, BW	Arbitrary Cost	BW,Dealy,Reliability,Load,MTU	Distance(BW,Delay)	Hop count
Class	Classless	classless	classless	classless	classful	Classless(v1:classful)
	Inter AS	Link state, Dijkstra	Link state, dijkstra	Hybrid	Distance vector, bellman ford	Distance vector, bellman ford
Updates	Unicast	multicast	multicast	multicast	broadcast	Broadcast via
Updates via	-	224.0.0.5/6	-	224.0.0.10	-	255.255.255.255 **RIPV2-** 224.0.0.9(multicast)
Type	open	Open	Enterprise Ver.of Cisco	Cisco Prprietary,IP,IPX,Appletalk	IP Protocol 9	Open standard
LB	NO LB	ELB	ELB	Unequal LB	ELB	equal load balancing
Max hop count	-	unlimited	unlimited	224	255	15(max routers-16)
AD	20 (external), 200(internal)	110	115	90(inter AS), 170(outer AS)	100	120
Update Timer	-	10 sec		No update, hello every 5 sec (fast LAN/WAN), 60 sec slow WAN	90 sec	30 sec
Invalid timer						180 (30+150)
Flush		Dead 40 sec				240(180+60)
Hold on				15 sec(UP*3)(fast) 180(slow)		180
Convergence time		40 sec		15 sec		>=4 min
	TCP Port 179	VLSM,CIDR		RTP for reliability, VLSM	ASN number in configuration	

Computer Network | IP security (IPSec)

The **IP security (IPSec)** is an Internet Engineering Task Force (IETF) standard suite of protocols between 2 communication points across the IP network that provide data authentication, integrity, and confidentiality. It also defines the encrypted, decrypted and authenticated packets. Management of cryptographic keys and Security Associations can be either manual or dynamic using an IETF-defined key management protocol called Internet Key Exchange (IKE). To participate in a virtual private network (VPN), a host must encrypt and authenticate individual IP packets between itself and another communicating host. IPSec is one of several mechanisms for achieving this, and one of the more versatile.

IPSec uses IP filtering to determine which traffic should be protected by IPSec.

Uses of IP Security –

IPsec can be used to do the following things:

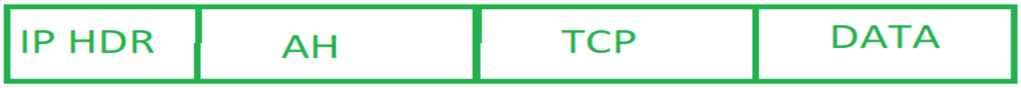
- To encrypt application layer data.

- To provide security for routers sending routing data across the public internet.
- To provide authentication without encryption, like to authenticate that the data originates from a known sender.
- To protect network data by setting up circuits using IPsec tunneling in which all data is being sent between the two endpoints is encrypted, as with a Virtual Private Network(VPN) connection.

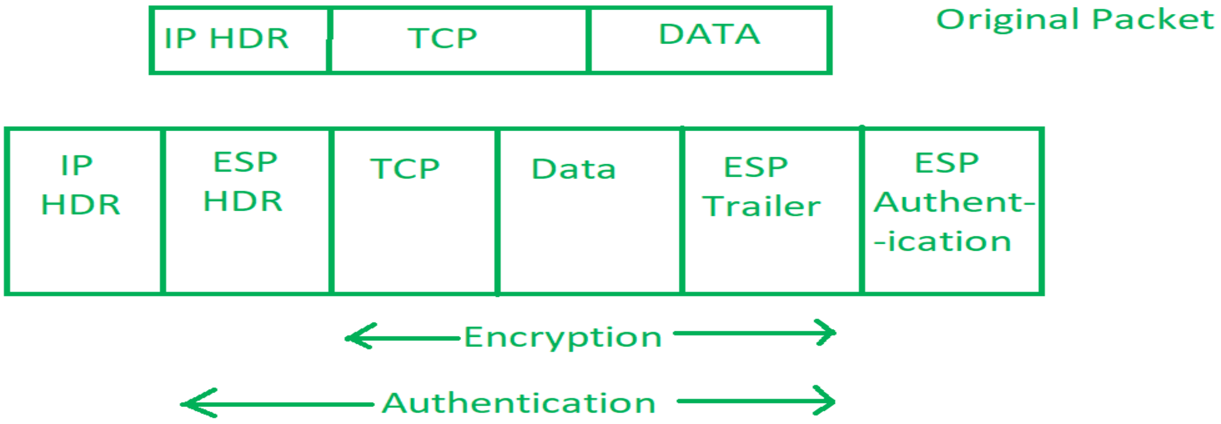
Components of IP Security –

It has the following components:

1. **Encapsulating Security Payload (ESP) –**
   It provides data integrity, encryption, authentication and anti replay. It also provides authentication for payload.
2. **Authentication Header (AH) –**
   It also provides data integrity, authentication and anti replay and it does not provide encryption. The anti replay protection, protects against unauthorized transmission of packets. It does not protect data's confidentiality.



3. **Internet Key Exchange (IKE) –**
   It is a network security protocol designed to dynamically exchange encryption keys and find a way over Security Assocaition (SA) between 2 devices. The Security Association (SA) establishes shared security attributes between 2 network entities to support secure communication. The Key Management Protocol (ISAKMP) and Internet Security Association which provides a framework for authentication and key exchange. ISAKMP tells how the set up of the Security Associations (SAs) and how direct connections between two hosts that are using IPsec. Internet Key Exchange (IKE) provides message content protection and also an open frame for implementing standard algorithms such as SHA and MD5. The algorithm's IP sec users produces a unique identifier for each packet. This identifier then allows a device to determine whether a packet has been correct or not. Packets which are not authorized are discarded and not given to receiver.



Working of IP Security –

1. The host checks if the packet should be trasmitted using IPsec or not. These packet traffic triggers the security policy for themselves. This is done when the system sending the packet apply an appropriate encryption. The incoming packets are also checked by the host that they are encrypted properly or not.
2. Then the **IKE Phase 1** starts in which the 2 hosts(using IPsec) authenticate themselves to each other to start a secure channel. It has 2 modes. The **Main mode** which provides the greater security and the **Aggressive mode** which enables the host to establish an IPsec circuit more quickly.
3. The channel created in the last step is then used to securely negotiate the way the IP circuit will encrypt data accross the IP circuit.
4. Now, the **IKE Phase 2** is conducted over the secure channel in which the two hosts negotiate the type of cryptographic algorithms to use on the session and agreeing on secret keying material to be used with those algorithms.
5. Then the data is exchanged accros the newly created IPsec encrypted tunnel. These packets are encrypted and decrypted by the hosts using IPsec SAs.

6. When the communication between the hosts is completed or the session times out then the IPsec tunnel is terminated by discarding the keys by both the hosts.

Que) disk space df –l

1. `df command` – Shows the amount of disk space used and available on Linux file systems.
2. `du command` – Display the amount of disk space used by the specified files and for each subdirectory.
3. You can print all available fields, enter:
   `$ df –o`
4. `$ df -h ### Human format`
   `$ df -m ### Show output size in one-megabyte`
   `$ df -k ### Show output size in one-kilobyte blocks (default)`

```
Df -l -l, --local           limit listing to local file systems


Mandatory arguments to long options are mandatory for short options too.


  -a, --all             include dummy file systems


  -B, --block-size=SIZE use SIZE-byte blocks


  -h, --human-readable  print sizes in human readable format


  -H, --si              likewise, but use powers of 1000 not 1024


  -i, --inodes          list inode information instead of block usage


  -k                    like --block-size=1K


  -l, --local           limit listing to local file systems


      --no-sync         do not invoke sync before getting usage info
(default)


  -P, --portability     use the POSIX output format


      --sync            invoke sync before getting usage info


  -t, --type=TYPE       limit listing to file systems of type TYPE


  -T, --print-type      print file system type


  -x, --exclude-type=TYPE   limit listing to file systems not of type TYPE
  -v                     (ignored)

      --help     display this help and exit

      --version  output version information and exit
```

Display output using inode usage instead of block usage

An inode is a data structure on a Linux file system that stores all information about file. To list inode information, enter:

```
$ df -i
$ df -i -h
```

6. Find out the type of each file system displayed

Pass the -T option to display the type of each filesystems listed such as ext4, btrfs, ext2, nfs4, fuse, cgroup, cputset, and more:

```
$ df -T
$ df -T -h
$ df -T -h /data/
```

Linux Server troubleshooting:

1) **check the hardware:**this shell command tells you if your Ethernet device link is detectable:

   $ sudo ethtool eth0 → If the answer is yes, you know the port is talking to the network.

Other common hardware problems can't be spotted by a mark one eyeball. For example, bad RAM causes all kinds of problems. VMs and containers can hide these problems, but if you see a pattern of failures linked to a specific bare-metal server, check its memory.

To see what a server's BIOS/UEFI reports about its hardware, including memory, use the dmidecode command:

```
$ sudo dmidecode --type memory
```

If this looks right—it may not be, as SMBIOS data isn't always accurate—and you still suspect a memory problem, it's time to deploy Memtest86. This is the essential memory checking program, but it's slow. If you're running it on a server, don't expect to use that machine for anything else while the checks are running.

2) check the server is running or not: You should also determine whether the problem is with the server per se or the server application.

There are numerous ways to check to see if an application is running. Two of my favorites are:

```
$ sudo ps -ef | grep apache2

$ sudo netstat -plunt | grep apache2
```

If it turns out that, say, the Apache web server isn't running, you can start it with this:

```
$ sudo service apache2 start
```

In short, before jumping in to work out what's wrong, make sure you work out which element is at fault. Only once you're sure you know what a problem is do you know the right questions to ask or the next level of troubleshooting to investigate.

3) **Use TOP command to see all running processes:**

system debugging step is top, to check load average, swap, and which processes are using resources. Top shows all of a Linux server's currently running processes.
To find the process consuming the most memory, sort the process list by pressing the `M`key. To see which applications are using the most CPU, press `P`; and to sort by running time, press `T`. To more easily see which column you're using for sorting, press the b key.

with **htop** you can use the mouse and scroll the process list vertically and horizontally to see all processes and complete command lines.

4) **check the disk space:** using df (disk filesystem) command to view a full summary of available and used disk space.

5) **check the server log:** For instance, if you have an Apache server running on an Ubuntu server, by default the logs will be kept in `/var/log/apache2`.One useful troubleshooting tool is **dmesg.** This displays all the kernel messages. That's usually way too many, so use this simple shell script to display the last 10 messages:

```
$ dmesg | tail.
```