**Question-15**

a.) Address in binary – 11000000.10101000.00000001.11111110

starting bit is 110 which is class C.

The subnet mask for class C, IP Address is 255.255.255.0

b.) 192.168.1.254

Converting the numbers 192,168,1,254 into 8 bits binary numbers from decimal form we get,

(192)10 = ( 11000000 )2

(168)10 = ( 10101000 )2

(1)10 = ( 00000001)2

(254)10 = ( 1111110)2

the subnet mask is 11000000.10101000.00000001.11111110

c.) The base IP address is 192.168.1.254/26

here 26 is netmask meaning the sum of the total of the binary 1’s in the IP address.

Address – 255.255.255.xxx = 11111111.11111111.11111111.xxx

sum unril now is 24, but we need 26 so we need only 2 now

xxx = 11000000

xxx in decimal is 192.

so the subnet mask is now 255.255.255.192.

so we have the choice of choosing hosts from 255.255.255.193 to 255.255.255.255.

Total we have 62 hosts.

**Question-11**

a.) In a bus topology the clients are usually connected to each other. I believe in the early days of networking they essentially connected 1 coaxial cable that was half duplex to all the computers in the network and so all data packets have to travel on that one cable. The cable can only handle one client transmitting at any time. If more than one client put a packet on the cable at the same time then a collision occurs because of signal interference. This one coaxial cable creates what’s called a collision domain with all those clients belonging to it. So, collisions occur.

b.) In the scenario, 20 host computers are connected to eachother via a bus topology with 10 hosts on either side. In this scenario, the bandwidth is shared among the nodes equally i.e, they get a bandwodth of 10mbps divided by 20 hosts which is equal to 0.5mbps. The throughput will be even less depending on the signals and the collisions taking place.

**Question-14**

a.) The HTTP protocol is based on a request/response paradigm. The client establishes connection with a server and sends a request to the server in the form of a request method, URI, and protocol version, followed by a MIME-like message containing request modifiers, client information, and possible body content. The server responds with a status line, including the message's protocol version and a success or error code, followed by a MIME-like message containing server information, entity metainformation, and possible body content.

b.) Most HTTP communication is initiated by a user agent and consists of a request to be applied to a resource on some origin server. In the simplest case, this may be accomplished via a single connection (v) between the user agent (UA) and the origin server (O).

request chain ( Client) ------------------------>

UA -------------------v------------------- O

<----------------------- response chain (Google Server)

c.) [HTTP headers](https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers) make the protocol easy to extend and experiment with. New functionality can even be introduced by a simple agreement between a client and a server about a new header's semantics. HTTP messages can be read and understood by humans, providing easier testing for developers, and reduced complexity for newcomers.

**Question-13**

**Network delay** is a design and performance characteristic of the network. It specifies the latency for a bit of data to travel across the network from one node to another. It is typically measured in multiples or fractions of a second. Delay may differ slightly, depending on the location of the specific pair of communicating endpoints. Engineers usually report both the maximum and average delay, and they divide the delay into several parts:

* Processing Delay - time it takes a router to process the packet header
* Queing Delay - time the packet spends in routing queues
* Transmission Delay - time it takes to push the packet's bits onto the link
* Propagation Delay - time for a signal to reach its destination

**Processing Delay** - During processing of a packet, routers may check for bit-level errors in the packet that occurred during transmission as well as determining where the packet's next destination is. Processing delays in high-speed routers are typically on the order of microseconds or less.

**Queuing Delay** - is the time between the completion of signaling by the call originator and the arrival of a ringing signal at the call receiver. Queuing delay may be caused by delays at the originating switch, intermediate switches, or the call receiver servicing switch.

**Transmission Delay** - is the amount of time required to push all the packet's bits into the wire. In other words, this is the delay caused by the data-rate of the link. It is a function of the packet's length and has nothing to do with the distance between the two nodes.

**Propagation Delay** - is the length of time taken for a signal to reach its destination. It can relate to [networking](https://en.wikipedia.org/wiki/Computer_network), [electronics](https://en.wikipedia.org/wiki/Electronics) or [physics](https://en.wikipedia.org/wiki/Physics). Hold time is the minimum interval required for the logic level to remain on the input after triggering edge of the clock pulse.

**Question-12**

a.) In [cryptography](https://en.wikipedia.org/wiki/Cryptography), to confirm that the message came from the stated sender (its authenticity) and has not been changed, a **message authentication code** (**MAC**) is used. MACs differ from [digital signatures](https://en.wikipedia.org/wiki/Digital_signature) as MAC values are both generated and verified using the same secret key( only sender and receiver is supposed have this ). This implies that the sender and receiver of a message must agree on the same key before initiating communications, as is the case with [symmetric encryption](https://en.wikipedia.org/wiki/Symmetric_encryption). Hence the symmetric encryption.

b.) To validate the MAC, we need the secret key to validate the mac. Without it we cannot do so. And so we have a successful authentication to encrypt the message.

c.) If the MAC is successfully verified, then a certificate will be generated that the source can be trusted by the receiver. The certificate of trust is a must along with the secret key for decryption.

There are two assumptions made always while trusting a source.

i.) Firstly, the sources are divided into two states, secure and insecure.

ii.) Secondly, it is ensured that the system sjould not go into insecure state.

These two assumptions are must and ensure that no harm arises to the reciever. And with these, the validated MAC is trusted as a trusted source to receive and send data.

**MCQ**

**10 - option-4**

**9 - option-3**

**8 - option-4**

**5 - option 4**

**7 - option -1**

**6 - option -3**

**4 - option -3**

**3 - option -4**

**2 - option -4**

**1 - option-1**