17.1

When we type a URL into a browser, something will happen as follows:

1. Browser contacts the DNS server to find the IP address of URL.

2. DNS returns back the IP address of the site.

3. Browser opens TCP connection to the web server at port 80.

4. Browser fetches the html code of the page requested.

5. Browser renders the HTML in the display window.

6. Browser terminates the connection when window is closed.

17.2

BGP: Border Gateway Protocol

BGP is the core routing protocol of the Internet. When a BGP router first comes up on the Internet, either for the first time or after being turned off, it establishes connections with the other BGP routers with which it directly communicates. The first thing it does is download the entire routing table of each neighboring router. After that it only exchanges much shorter update messages with other routers.

BGP routers send and receive update messages to indicate a change in the preferred path to reach a computer with a given IP address If the router decides to update its own routing tables because this new path is better, then it will subsequently propagate this information to all of the other neighboring BGP routers to which it is connected, and they will in turn decide whether to update their own tables and propagate the information further.

RIP: Routing Information Protocol RIP provides the standard IGP protocol for local area networks, and provides great network stability, guaranteeing that if one network connection goes down the network can quickly adapt to send packets through another connection.

What makes RIP work is a routing database that stores information on the fastest route from computer to computer, an update process that enables each router to tell other routers which route is the fastest from its point of view, and an update algorithm that enables each router to update its database with the fastest route communicated from neighboring routers.

OSPF: Open Shortest Path First

Open Shortest Path First (OSPF) is a particularly efficient IGP routing protocol that is faster than RIP, but also more complex.

The main difference between OSPF and RIP is that RIP only keeps track of the closest router for each destination address, while OSPF keeps track of a complete topological database of all connections in the local network.

17.3

IPv4 and IPv6 are the internet protocols applied at the network layer IPv4 is the most widely used protocol right now and IPv6 is the next generation protocol for internet.

» IPv4 is the fourth version of Internet protocol which uses 32 bit addressing whereas IPv6 is a next generation internet protocol which uses 128 bits addressing.

» IPv4 allows 4,294,967,296 unique addresses where as IPv6 can hold 340-undecillion (34, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000) unique IP addresses.

» IPv4 has different class types: A, B, C, D and E. Class A, Class B, and Class C are the three classes of addresses used on IP networks in common practice. Class D addresses are reserved for multicast. Class E addresses are simply reserved, meaning they should not be used on IP networks (used on a limited basis by some research organizations for experimental purposes).

» IPv6 addresses are broadly classified into three categories:

1. Unicast addresses: A Unicast address acts as an identifier for a single interface. An IPv6 packet sent to a Unicast address is delivered to the interface identified by that address.

2. Multicast addresses: A Multicast address acts as an identifier for a group / set of interfaces that may belong to the different nodes. An IPv6 packet delivered to a multicast address is delivered to the multiple interfaces.

3. Anycast addresses: Anycast addresses act as identifiers for a set of interfaces that may belong to the different nodes. An IPv6 packet destined for an Anycast address is delivered to one of the interfaces identified by the address.

» IPv4 address notation: 239 255 255 255, 255 255 255 0.

» IPv6 addresses are denoted by eight groups of hexadecimal quartets separated by colons in between them.

» An example of a valid IPv6 address: 2001:cdba:0000:0000:0000:0000:3257:9652

17.4

A mask is a bit pattern used to identify the network/subnet address. The IP address consists of two components: the network address and the host address.

The IP addresses are categorized into different classes which are used to identify the network address.

Example: Consider IP address 152 210 011 002. This address belongs to Class B, so:

Network Mask: 11111111.11111111.00000000.00000000

Given Address: 10011000.11010101.00001011.00000010

By ANDing Network Mask and IP Address, we get the following network address:

10011000.11010101.00000000.00000000 (152.210.0.0)

Host address: 00001011.00000010

Similarly, a network administrator can divide any network into sub-networks by using subnet mask. To do this, we further divide the host address into two or more subnets.

For example, if the above network is divided into 18 subnets (requiring a minimum of 5 bits to represent 18 subnets), the first 5 bits will be used to identify the subnet address.

Subnet Mask: 11111111 11111111 11111000 00000000 (255 255 248 0)

Given Address: 10011000 11010101 00001011 00000010

So, by ANDing the subnet mask and the given address, we get the following subnet address:

10011000 11010101 00001000 00000000 (152 210 1 0)

How Host A sends a message/packet to Host B:

When both are on same network: the host address bits are used to identify the host within the network.

Both are on different networks: the router uses the network mask to identify the network and route the packet. The host can be identified using the network host address.

The network layer is responsible for making routing decisions. A routing table is used to store the path information and the cost involved with that path, while a routing algorithm uses the routing table to decide the path on which to route the packets.

Routing is broadly classified into Static and Dynamic Routing based on whether the table is fixed or it changes based on the current network condition.

17.5

TCP (Transmission Control Protocol) : TCP is a connection-oriented protocol. A connection can be made from client to server, and from then on any data can be sent along that connection.

» Reliable - when you send a message along a TCP socket, you know it will get there unless the connection fails completely. If it gets lost along the way, the server will re-request the lost part. This means complete integrity; data will not get corrupted

» Ordered - if you send two messages along a connection, one after the other, you know the first message will get there first. You don’t have to worry about data arriving in the wrong order.

» Heavyweight - when the low level parts of the TCP “stream” arrive in the wrong order, resend requests have to be sent. All the out of sequence parts must be put back together, which requires a bit of work.

UDP(User Datagram Protocol): UDP is connectionless protocol.

With UDP you send messages (packets) across the network in chunk.

» Unreliable - When you send a message, you don’t know if it’ll get there; it could get lost on the way.

» Not ordered - If you send two messages out, you don’t know what order they’ll arrive in » Lightweight - No ordering of messages, no tracking connections, etc. It’s just fire and forget! This means it’s a lot quicker, and the network card / OS have to do very little work to translate the data back from the packets.

Explain how TCP handles reliable delivery (explain ACK mechanism), flow control (explain TCP sender’s/receiver’s window):

For each TCP packet, the receiver of a packet must acknowledge that the packet is received. If there is no acknowledgement, the packet is sent again. These guarantee that every single packet is delivered. ACK is a packet used in TCP to acknowledge receipt of a packet. A TCP window is the amount of outstanding (unacknowledged by the recipient) data a sender can send on a particular connection before it gets an acknowledgment back from the receiver that it has gotten some of it.

For example, if a pair of hosts are talking over a TCP connection that has a TCP window with a size of 64 KB, the sender can only send 64 KB of data and then it must wait for an acknowledgment from the receiver that some or all of the data has been received. If the receiver acknowledges that all the data has been received, then the sender is free to send another 64 KB. If the sender gets back an acknowledgment from the receiver that it received the first 32 KB (which could happen if the second 32 KB was still in transit or it could happen if the second 32 KB got lost), then the sender can only send another additional 32 KB since it can’t have more than 64 KB of unacknowledged data outstanding (the second 32 KB of data plus the third).

Congestion Control

The TCP uses a network congestion avoidance algorithm that includes various aspects of an additive-increase-multiplicative-decrease scheme, with other schemes such as slow-start in order to achieve congestion avoidance.

There are different algorithms to solve the problem; Tahoe and Reno are the most well-known. To avoid congestion collapse, TCP uses a multi-faceted congestion control strategy. For each connection, TCP maintains a congestion window, limiting the total number of unacknowledged packets that may be in transit end-to-end. This is somewhat analogous to TCP’s sliding window used for flow control. TCP uses a mechanism called slow start to increase the congestion window after a connection is initialized and after a timeout. It starts with a window of two times the maximum segment size (MSS). Although the initial rate is low, the rate of increase is very rapid: for every packet acknowledged, the congestion window increases by 1 MSS so that for every round trip time (RTT), the congestion window has doubled. When the congestion window exceeds a threshold ssthresh the algorithm enters a new state, called congestion avoidance. In some implementations (i.e., Linux), the initial ssthresh is large, and so the first slow start usually ends after a loss. However, ssthresh is updated at the end of each slow start, and will often affect subsequent slow starts triggered by timeouts.