Computer Networks

Contents

1 IP addresses:

- IP address in IPv4 are of 32 bits, represented as 8.8.8.8 bits.
- Primarily consist of 2 parts, first n bits as network ID and rest $32 - n$ as Host ID.
- IP address can represent 2^n different networks of 2^{32-n} hosts each.
- Divided into 5 classes:
  - Class A:
    - ∗ Addresses starting with 0.
    - ∗ Range: 0.0.0.0-127.255.255.255, total 2^{31} addresses.
    - ∗ First 8 bits are network ID, of which first bit has to be 0. So, $2^7 = 128$ combinations are possible for network ID part.
    - ∗ Last 24 bits are Host ID. i.e. 16M IP addresses per network are possible.
  - Class B:
    - ∗ Addresses starting with 10.
    - ∗ Range: 128.0.0.0-191.255.255.255, total 2^{30} addresses.
    - ∗ First 16 bits are network ID, of which first 2 bits have to be 10. So, $2^{14} = 16K$ combinations are possible for network ID part.
    - ∗ Last 16 bits are Host ID. i.e. 64K IP addresses per network are possible.
  - Class C:
    - ∗ Addresses starting with 110.
    - ∗ Range: 192.0.0.0-223.255.255.255, total 2^{29} addresses.
    - ∗ First 24 bits are network ID, of which first 3 bits have to be 110. So, $2^{21} = 2M$ combinations are possible for network ID part.
    - ∗ Last 8 bits are Host ID. i.e. 255 IP addresses per network are possible.
  - Class D and E:
    - ∗ Addresses starting with 1110 and 1111 repsectively.

    - Range: 224.0.0.0-239.255.255.255, and 240.0.0.0 to 255.255.255.255. total 2^{28} addresses each.
    - No number of bits are specified as network ID or host ID.

- 1st IP address of each network is used to denote the ID of network as a whole. i.e. all host ID bits as 0.
  e.g. 10.0.0.0 is used to represent a class A network. 130.25.0.0 is used to represent a class B network, etc.

- If a message is to be sent to all the hosts in a network, it is called a broadcast.
  - If a host inside the network is to do a broadcast, it is called a **Limited Broadcast**. Done by setting destination address to 255.255.255.255.
  - If a host outside the network is to do a broadcast to a network, it is called a **Directed Broadcast**. If message is to be broadcasted to e.g. 130.20.0.0, destination address is set as 130.20.255.255. that is, set all the host ID bits as 1.

- This is why, if k bits represent host ID, $2^k - 2$ hosts can be configured, as first and last IP address is reserved for network ID and directed broadcast respectively.

1.1 Subnetting:

- Subnetting is done to divide the networks into multiple parts, and to address the different parts independently.

- If a network is to be divided into 2 different parts, one bit from host ID part is reserved to denote the subnet. For 4 parts, 2 bits are reserved and so on.

- E.g. network 130.20.0.0 is to be divided into 2 parts, 16^{th} bit is used to denote the subnet ID.

- Subnets are:
  1. From 130.20.0.0 to 130.20.127.255.
  2. From 130.20.128.0 to 130.20.255.255.

  2^{16} IPs are divided into 2 subnets of 2^{15} IPs each.

- Similar to networks, first IP in a subnet denotes that subnet, and last ID is the directed broadcast address. That is, for the first subnet, 130.20.0.0 is subnet ID, and 130.20.127.255 is directed broadcast address.

- Here, 130.20.255.255 is directed broadcast address for both the network 130.20.0.0, and the subnet 130.20.128.0. This may result in ambiguity. To resolve this, if a message with destination 130.20.255.255 is received
  - from outside the network, it is sent to all the hosts inside the network, as external networks are not aware of the subnets that exist inside the network.
  - from within the network itself, it is sent to hosts inside the subnet 130.20.128.0 only.

- Here, first and last IP of each subnet is reserved for the ID and the directed broadcast address. So, total $2^{16} - 4$ hosts can be configured (for 2 subnets).

- To get the subnet ID of a particular host, the IP address of that host is anded with a *subnet mask*. This mask contains ones in the network ID and subnet ID part, and zeros in the host ID part.
  E.g. for a host 130.20.138.23, subnet mask is 255.255.128.0, i.e. 17 ones followed by 15 zeros. Result after anding with the mask will be 130.20.128.0, which is the subnet ID of the subnet to which this IP belongs.

- By choosing 2 bits, network can be divided into 4 subnets of 2^{14} IPs each.
- If, for example, we want 3 subnets of size $2^{15}, 2^{14}$, and 2^{14} IPs, subnet ID is of variable length, and follows the same approach as division into classes.
- E.g.
  1. If host ID starts with 0, IP belongs to subnet 1, 130.20.0.0 to 130.20.127.255, total 2^{15} IPs.
     Mask: 255.255.128.0
  2. If host ID starts with 10, IP belongs to subnet 2, 130.20.128.0 to 130.20.191.255, total 2^{14} IPs.
     Mask: 255.255.192.0
  3. If host ID starts with 11, IP belongs to subnet 3, 130.20.192.0 to 130.20.255.255, total 2^{14} IPs.
     Mask: 255.255.192.0
- Subnet ID is determined by checking if the IP anded with the mask matches any present subnet ID. In case of variable length subnet ID, multiple matches are possible. In that case, ID where mask contains more ones is chosen.

1.2 Classless Inter Domain Routing:

- $2^{24}, 2^{16}$, and 2^{8} IPs are available in class A, B, and C type network respectively. So, if the requirement is, say 300, we'll have to go for the class B network which has a capacity of 64K, which is way too much for our application.
- This is why there was a need of a better system. So, CIDR was introduced. There are no classes in CIDR, so it is not possible to tell how much is the network ID part, and how much is the host ID part. So in CIDR, IP addresses are represented as a.b.c.d/n where n is no. of bits in the network ID.
- In CIDR system, one can ask for a "block" of IP addresses. A block can be of variable size. For the block to be qualified as a valid block, there are 3 rules:
  1. All the IPs in one block must be continuous.
  2. No. of IPs in one block should be a power of 2. This is because it is easier to break the IP into N_ID and H_ID parts.
  3. First IP in every block should be evenly divisible by the size of the block (2^n). This basically means all bits of first IP address after block ID should be 0. This is how 2^n IP addresses are formed from n zeros to n ones.
- A CIDR block can be represented in a.b.c.d/n form, where a.b.c.d is an IP in the block, and n is number of bits in N_ID part. First IP in the block can be found out by making the last $32 - n$ bits 0, and the last IP can be found by making those bits 1.
- Every CIDR block can also be subnetted using the same logic used in class system, with constant length subnet masks or variable length subnet masks.
- If a block is divided into 2 subnets, 1 bit will be used to indicate the subnet. In that case, the IP will be represented as a.b.c.d/n+1, as one more bit is used to indicate the block ID.

1.3 Supernetting:

- Combining 2 or more networks to decrease the contents of routing table.
- For 2 or more networks to be combined, they should satisfy 3 conditions:
  1. All the networks should be contigious.
  2. Size of all the networks should be same and a power of 2, so that no. of bits representing H_ID are same.
  3. 1st IP address in the first network should be evenly divisible by the total size of the supernet.
- Combined mask of the resulting supernet contains 0s in H_ID part and part of the network ID that is changing, and 1s at all other positions.
  E.g. if the networks 130.20.0.0, 130.20.1.0, 130.20.2.0, 130.20.3.0 are to me aggregated, new supernet mask will be 255.255.252.0.
- Supernet ID can be found by anding any IP in the supernet by supernet mask.

2 Delays:

- Transmission delay (T_t): time taken by host to put the data packet onto the transmission link. If available bandwidth is B bits/sec, and the size of data packet is L bits, transmission delay is $\frac{L}{B}$.
- Propagation delay (T_p): time taken by a bit to reach from sender to the receiver through the transmission link.
- Queueing delay: time spent by the message in receiver's queue.
- Processing delay: time talen by receiver to process the message.

3 Flow Control:

- Synchronization betweeen data
- Stop and wait:
  - Send a packet, and wait for the acknowledgement.
  - Will take $T_t + 2T_p$ time, in which T_t is used to transmit the data, and we wait for $2T_p$ time.
  - Efficiency is $\frac{useful\ time}{Total\ time}$, which is $\frac{T_t}{T_t+2T_p}$.
  - $\eta = \frac{1}{1+2a}$ where $a = \frac{T_p}{T_t}$.
  - A timeout timer is also there at sender side. If ack is not received within a specific time, sender will assume that the packet was lost and send it again.
- Capacity of channel - no. of bits that can be held by the channel at a time. Equal to $B * T_p$.
- To increase η, pipelining is used, i.e. instead of waiting for $2T_p$ time, keep sending the packets.
- In $T_t + 2T_p$ time, $1 + 2a$ packets can be sent, when the last packet is sent, ack for first packet will be received. Sending $1 + 2a$ packets will result in 100% efficiency.
- In case some data is lost and ack is not received, it'll need to be resent. So, $1 + 2a$ packets will be stored in a "window". Once the ack for earliest packet is received, that packet will be removed from window, and next packet will be added.
  This is also called sliding window protocol.

- To indicate packet number, a sequence number is included in the packet. If there are n bits, upto 2^n packets can be sent in one window.
- Sequence numbers are generally reused, i.e. for example if there are 2 bits, sequence numbers are 0, 1, 2, 3, 0, 1, 2...and so on.
- If number of bits don't allow $1 + 2a$ packets, we will have to wait till the ack for packet 0 is received. In this case, efficiency is not 100%, it is $\frac{2^n}{1+2a}$ where n is number of bits.
- Ways to Implement sliding window protocol:
  1. Go Back N (GBN):
     - Window size is N. e.g. GB10 has W_S of 10.
     - Sender will send packet number $N + 1$ only when ack for packet 0 is received.
     - If some packet is lost, timeout timer for that packet will be triggered, and sender will send all the packets including and following the lost packet. i.e. sender will "Go Back N" packets.
     - Uses cumulative ack, i.e. multiple packets are acknowledged at once. Whenever a packet is received, ack timer starts, and at the end of the timer, if last packet received is N, ackN+1 is sent.
     - Acknowledgement timer at receiver should be greater than timeout timer at sender.
     - Available sequence numbers should be greater than W_S, to avoid data duplication problem.
     - Cannot accept out of order packets.
  2. Selective Repeat (SR):
     - Receiver window size is same as sender window size.
     - Whenever a packet is lost, only that packet is retransmitted. This reduces the number of total transmitted packets.
     - Acknowledges every packet separately. If some packet is received but it is corrupted, a 'nack' is sent. Whenever a nack is received, that packet is retransmitted immediately.
     - Can accept out of order packets.

4 Access Control:

- Needed to avoid collisions in broadcasting channel.
- **Time Division Multiplexing:**
  - Slots of time are reserved for every host in the network. It is of length $T_t + T_p$.
  - Not very efficient as many slots may go wasted.
- **Polling:**
  - Pole is conducted and slot is given to the winner.
  - May result in one host gettting too many slots, and others no slots at all.
- **Carrier Sensing Multiple Access/Collision Detection (CSMA/CD):**
  - Every host senses the carrier, whether it is empty or not. If it is empty, host sends the data. If it's not, host waits.
  - There are no acknowledgements, so the sender has to detect if there is any collision in the channel.

- Packet will be in the channel for T_p time, and if the collision is at the end of the channel, it'll take another T_p time to reach to the original sender. Sender keeps looking for collisions for as long as the data is being transmitted. $\therefore T_t > 2T_p$, or $L > 2 * T_p * BW$.
- If the data is less than the required length, additional bits are padded.
- Backoff Algorithm:
  - If a collision is detected by 2 hosts, and both stop sending the data, and both start sending at the same time, there will be an infinite collision loop. This algorithm is used to avoid this situation.
  - A waiting time T_S is decided, which is approximate time taken for collision signal to leave the channel.
  - Sender counts the number of collisions faced by the packet it is about to send. If there are n collisions, it generates a random number k between 0 and $(2^n - 1)$, and it waits for $k * T_S$ time before starting transmission.
  - For example, if 2 hosts, A and B start transmitting data, and collision is detected after some time, Both stop and generate a number between 0 and $2^1 - 1 = 1$. In 01 and 10 cases, one host sends the data and the other one waits. In 00 and 11 cases, collision happens.
  - If suppose A wins and sends the data, and now A wants to send another packet, and the 2^{nd} packet collides with 1^{st} packet of B. Now $n_A = 1$, and $n_B = 2$ as packet 1 from B has collided 2 times.

---

5 Topologies

1. **Mesh**: every device in a network is connected to eachother.
   - ${}^nC_2$ cables.
   - $n - 1$ ports per device, $n(n - 1)$ total.
   - High reliability as even if one connection fails, there are many more options available.
   - Cost is high, as many cables are required.
   - Security is also high as there is point to point connection.
   - Generally used for LANs where n is small.
2. **Star**: all devices are connected to a central device named *hub*.
   - n cables, from each device to hub.
   - 1 port per device, N total.
   - Low reliability, as there is *Single Point of Failure*(SPOF) i.e. if hub fails, communication stops.
   - Cost is lower than mesh.
   - Security is high as there is point to point connection.
3. **Bus**: there is one thick wire, and all devices are connected to that wire.
   - Thick central wire is called the backbone cable.
   - $n + 1$ cables, one per device + backbone cable.
   - 1 port per device, n total.
   - Low reliability as backbone cable is SPOF.

- Cost is low.
- Less secure as every message can be accessed by all devices in a network.
- Has a limit to how long the backbone cable can be, repeaters are used if longer connection is required.
- As all data is flowing through same cable, there is high chance of collision. So, access control methods are used.

4. **Ring**: same as bus, but the ends of backbone cable are joined to eachother.
    - All points in bus topology apply to ring as well, except the repeater one.

6 Devices

1. **Repeater**:
    - Purely hardware, works only on physical layer.
    - Has 2 ports only.
    - Reads the signal, and restores the strength to the original value.
    - Cannot filter the messages, whatever is received will be forwarded.
2. **Hub**:
    - Also purely hardware, works on physical layer.
    - Multiport repeater.
    - Cannot filter, whatever is received will be forwarded to rest all ports.
3. **Bridge**:
    - Has 2 ports, used to connect 2 different LANs.
    - Works on physical and data link layer.
    - Message can be filtered if both sender and receiver are in the same network. This is checked using a table where record of where every device is connected is kept. This is of 2 types,
        * Static: network administrator has to manually enter the data, and update if some connection is changed.
        * Dynamic: Contents are updated automatically based on which port of the bridge receives a message from which device. e.g. if a message is received from device $d1$ at port 1, entry $|d1|\text{port1}|$ is added to table.
          If the entry is not present in the table, the message is forwarded by default.

      Filtering takes place based on MAC addresses, as IPads are only accessible from network layer and above.
    - A buffer is present inside the bridge, so even if multiple messages are received at the same time, collision doesn't occur.
4. **Router:**
    - Has many ports, used to connect 2 or more different networks.
    - Works on physical, data link, and network layer.
    - Routing table is used to decide where to send the data after reception, based on the destination IP address.
    - Routing table contains the list of all N_IDs and ports that are connected to those N_IDs.

  - If a suitable port cannot be decided, data is sent to all the networks connected to it. This is called ”flooding”.
  - A port of a router connected to the network A has an IP address that is present within the network A. Different ports of a router have different IP addresses.
  - A buffer is present, so there are no collisions.

7 Packet Switching

- Data is divided into multiple packets and sent.
- Datagrams (connectionless):
  - Each packet is sent independantly, with sequence numbers.
  - Every packet has a separate header.
  - May be received out of order, then it is arranged in correct sequence.
  - Resources are not reserved, allocation is done on demand. This increases delays, but also increases efficiency.
  - Sliding window protocols are used.
  - Used in internet.
- Virtual circuit (connection oriented):
  - A packet called global header is sent first which reserves some resources along the path for the following message packets.
  - A virtual *connection* is established, then all the message packets are sent sequentially.
  - Data received is in order, no need of rearrangement.
  - As resources are reserved till the connection is not ended, there are no delays, but it decreases efficiency.
  - Used in VoIP.

Cyclic Redundancy Check (CRC)

- Take message bits and append n zeros to the end, where n is degree of divisor polynomial.
- Perform the division and replace the zeros at the end with the obtained remainder.
- To check if a code is valid or not, perform the same operation. If 0 remainder is obtained, it is a valid code.