Computer Networks - Review Notes

Aditya Soni

Computer Science and Engineering, NIT Silchar

Reference: Data Communications and Networking (4th ed.)
By Behrouz A Forouzan

Contents

Chapter 1

Introduction

Network is a set of devices(or nodes) connected by communication links. Networks must meet certain criteria:

(a) Performance: usually measured by metrics *throughput* and *delay*. More throughput and less delay is desirable.

(b) Reliability: measured by frequency of failure, time it takes to recover from failure, and robustness of the network.

(c) Security: protecting data from unauthorized access, damage and development, and implementing policies for recovery from breaches.

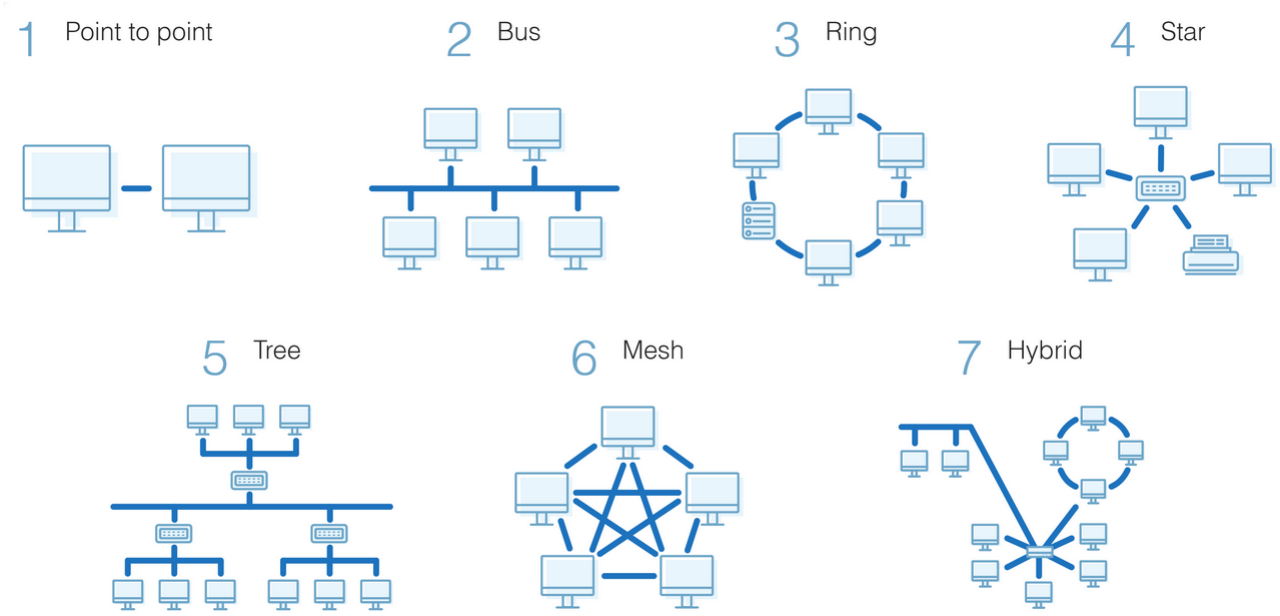
Basic terminology:

- Link: communication pathway that transfers data from one device to another.
- Point to point: provides dedicated link between two devices, means that entire capacity of link is reserved for communication between those two devices.
- Multipoint: more than two devices share a specific link. If several devices can use the link simultaneously, it is *spacially shared* connection, else if devices must take turn, then it is *time-shared* connection.
- Topology: It is geometric representation of relationship of all links and linking devices to one another.

1.0.1 Basic Topologies

(a) **Mesh**: Every device has dedicated point to point link to every other device. For a network with n nodes, number of links are $n(n-1)/2$. Advantages include elimination of traffic due to point-to-point connections, privacy and security due to dedicated links, entire system doesn't fail if some link fails. Also, point-to-point links make fault detection and identification easy. Disadvantages include difficulty in installation and reconnection, unavailability of space for such huge wiring, and expensive due to requirement of more hardware to establish links.

(b) **Star**: Each device has dedicated point-to-point link only to a central controller (called hub). If device want to send data to another device, it sends data to hub, which relays data to the desired connected device. For a network with n nodes, number of links are n. Advantages include ease of installation and configuration as only one link needs to be added or altered, all other links remain active if one link fails, ease of fault identification and isolation. Disadvantages include dependency of entire network on the hub i.e. failure of the hub results in failure of the entire network.



(c) **Bus**: Multipoint connections are used. One long cable acts as backbone to connect all devices in network (called bus). Nodes are connected to the bus by drop lines and taps (connector). For a network with n nodes, number of links are $(n + 1)$. Advantages include ease of installation as backbone cable is laid in an efficient path and devices are connected using variable length drop lines. Also, redundancy os eliminated as drop lines only need to reach closest point on the bus. Disadvantages include difficulty in identifying and isolating faults, degradation of signal quality if taps are distant, fault or break in bus cable stops all transmission, and spacing between taps limit the number of devices.

(d) **Ring**: Each device has dedicated point-to-point connections with only two devices on either side of it. Signal is passed along the ring in one direction until it reaches destination. Each device incorporates repeater that regenerates signal intended for another device. For a network with n nodes, number of links are n. Advantages include relative ease of installation and reconfiguration since it only requires changing two cables, fault isolation is simplified as when a signal is passed around and if a devie doesn't receive the signal in stipulated time, it is faulty. Disadvantages include, break or fault in single device can disable the entire network.

(e) **Hybrid**: Above basic topologies can be combined to create hybrid topologies.

1.0.2 Classification of Networks (based of range)

(a) LAN (Local Area Network): usually privately owned and connects device in single room, building or campus. Range is about $10m - 1km$.

(b) MAN (Metropolitan Area Network): covers a town or city for high speed connectivity. Range is about $10km - 50km$.

(c) WAN (Wide Area Network): provides long distance transmission of data over large geographic areas. Range is about $100km - 1000km$.

(d) Internet: stands for internetwork i.e. connection of two or more networks to create a large scale network.

1.1 OSI Model

OSI – Open Systems Interconnection

Protocol consists of a set of rules that govern data communication. It determines what, how and when communication takes place. Key elements of protocol are: (i)Syntax which is structure and format of data, (ii)Semantics which interprets the meaning of bits, and (iii)Timing i.e. when and at what speed data should be sent.

An open system is a set of protocols that allows any two different systems to communicate regardless of their underlying architecture. Purpose of OSI model is to show how to facilitate communication between different systems without requiring changes to the logic of the underlying hardware and software. OSI model has 7 ordered layers. Principles behind creating these layers are:

- A layer should be created where a different abstraction is needed.
- Each layer should perform a well defined function.
- Function of each layer should be chosen by keeping eye on internationally standardized protocol.
- Layer boundaries should by chosen to minimize the information flow across the interfaces.
- Number of layers should be large enough that distinct functions need not be thrown together in same layer out of necessity.

1.1.1 Breakdown of OSI Model

At the physical layer, communication is direct i.e. device A sends stream of bits to B. At higher layers, communication must move through the layers and interfaces. Communication moves down from A. Each layer in sending device adds its own information to the message it receives from layer just above it and passes whole package down. At the receiving machine, the message is unwrapped layer by layer with each layer recieving and removing data meant for it, and then passing the processed message to the layer above it.

1.1.2 Brief description of Layers

(1) **Physical layer**: It is responsible for transmission and reception of the unstructured raw data over network. Data encoding is also done in this layer. Data is sent and received in form of bits.

(2) **Data Link layer**: It divides stream of bits into managable data units called frames. It also imposes flow control mechanisms to avoid overwhelming sender or receiver. It makes raw transmission facility into a reliable link.

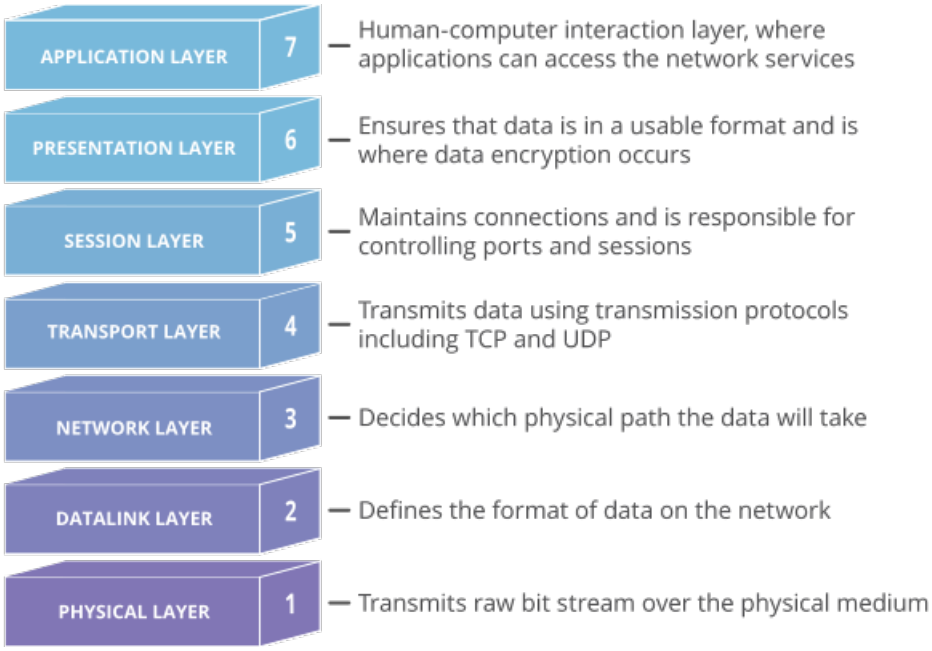
(3) **Network layer**: It is responsible for source-to-destination delivery of a packet, possibly across multiple networks. It makes sure packet reaches to the destination through optimal route.

(4) **Transport layer**: It is responsible for process-to-process delivery of the message. While network layer treats each package as individual message, transport layer ensures whole message is received intact and in order.

(5) **Session layer**: It is the network dialogue container which establishes, maintains, and synchronizes interaction among communicating systems.

(6) **Presentation layer**: It is concerned with syntax and semantics of the information exchanged between two systems. It translates data into sender-dependent or receiver-dependent format. Also, it carries out encryption and decryption of data.

(7) **Application layer**: It enables user (human or software) to access the network. It provides user with interfaces and support for various services and software.



1.2 TCP/IP Model

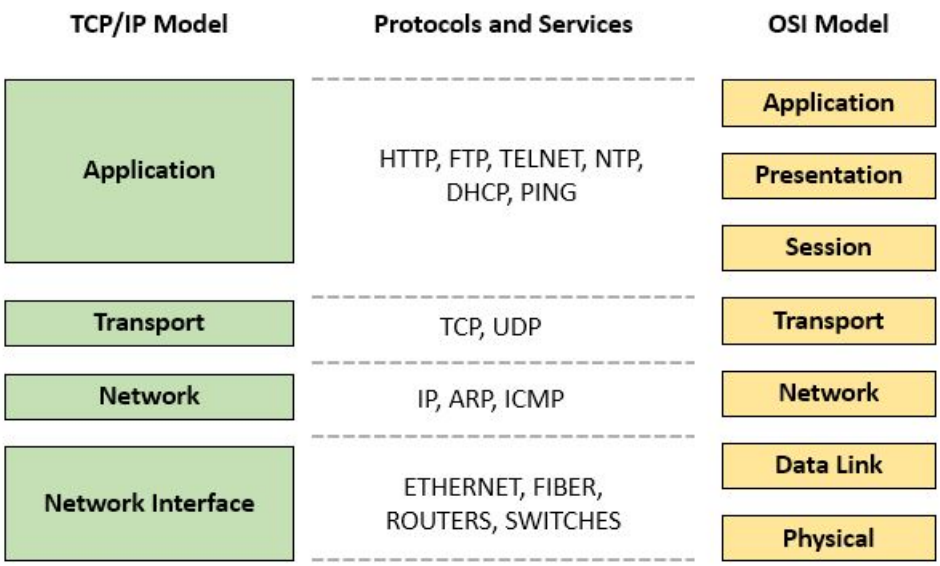
It has 5 (sometimes 4) layers which are physical, data link, network (or *internet*), transport, and application layer. Here, *host-to-network* or *network access* layer is equivalent to combination of physical and data link layers.

- Application layer: This layer is combined equivalent of session, presentation, and application layers of the OSI model. Actually, experience with OSI model has proved that session or presentation layers are of little use to most applications. Example of protocol of this layer is HTTP.

- Transport layer: allows peer hosts to carry on conversation. Two end-to-end transport protocols used are:

  (a) Transmission Control Protocol (TCP): A reliable connection oriented protocol that allows a byte-stream originating from one machine to other without error. TCP also handles flow control to avoid overwhelming the receiver.

  (b) User Datagram Protocol (UDP): An unreliable connectionless protocol for applications that do not want TCP's sequencing or flow control. It is a process-to-process protocol that only adds port addresses, checksum, error control, and length information to the data. UDP does not specify which packet is lost, if any.



1.2.1 Distinction between OSI Model and TCP/IP Model

- OSI model provides guidelines on how communication needs to be done. It is a generic model for specifying connection procedures, layered architecture. OSI model makes clear distinction between concept of protocol, services and interfaces (i.e. protocol is set of rules for communication within a layer, service is what a layer provides to the layer above it through an interface).
- OSI model is protocol independent hence can be implemented based on network specification. Whereas, TCP/IP Model was developed later and is protocol specific, so it can only solve certain set of problems.
- Network layer in OSI model provides both connection-oriented and connectionless service and transport layer only has connection oriented communication. Whereas, network layer in TCP/IP model provides only connectionless service.
- TCP/IP model provies reliable end-to-end communication which is secure.

Chapter 2

Physical Layer

2.1 Signals

Bandwidth The difference between highest and lowest frequencies contained in a (composite) signal is the bandwidth value.

2.1.1 Digital Signal Transmission

1. Baseband Transmission: sending a digital signal without changing the digital signal to analog signal. It requires *low-pass* channel i.e. a channel with a bandwidth that starts from 0.

2. Broadband Transmission or Modulation: changing the digital signal to analog signal for transmission. Modulation, a process of varying one or more properties of periodic waveform, allows us to use *bandpass* channel i.e. a channel with bandwidth that does not start from 0.

2.1.2 Transmission Impairment

Imperfect transmission medium causes impairment in the signal. The causes are:

(a) **Attenuation**: means loss of energy. When signal travels through a medium, it loses energy in overcoming the resistance of the medium. **Amplifiers** are used to compensate for the loss by amplifying the signal.
Attenuation is measured in decibels (dB) by measuring relative strength of signal at two different points.If the value is $+ve$, means signal is attenuated, and if the value is $-ve$, signal is amplified. If P_1 and P_2 are powers of signal at points 1 and 2, the decibel is computed as $10 \log_{10} \left(\frac{P_2}{P_1}\right)$.

(b) **Distortion**: means changing of form or shape of the signal. Distortion occurs in composite signal as each frequency component travels through the medium with distinct propagation speeds which causes delay for some signals, that means signals have different phases at receiver that what they had at source.

(c) **Noise**: means unwanted disturbance in an electrical signal. Types of noise are:

- Thermal: random noise of electrons in the conductor which creates extra signal.
- Induced: when more than one signal share a single transmission channel, noise is generated.
- Crosstalk: caused by one wire on the other (as electromagnetic interference). One wire acts as a sending antenna and transmission medium acts as receiving antenna.
- Impulse: random spikes that comes from power lines, lightning etc.

Signal-to-Noise Ratio (SNR) is the measure of signal strength relative to background noise. The higher its value, the better.

$$SNR = \frac{Avg\ signal\ power}{Avg\ noise\ power} = \frac{S}{N} \qquad and,$$

$$SNR_{dB} = 10 \log_{10} SNR$$

2.1.3 Data Rate Limits

It is important to consider how fast we can send data, in bits per second, over a channel. The factors responsible are: (i)Bandwidth available, (ii)Level of signals used, if L levels exist, $\log_2 L$ bits exist in every level, and (iii)Quality of channel (level of noise).
Two theoretical formulae are generally used for calculating data rate:

- **Nyquist Bit Rate** (for a noiseless channel) defines the theoretical maximum bit rate (where, B is the bandwidth of channel, and L is number of signal levels used to represent data):

  $$BitRate = 2 \times B \times \log_2 L\ bits/sec$$

  Also, increasing the levels of signal may reduce the reliability of the system, as receiver has to distinguish each levels. If number of levels from the formula is not a power of 2, either number of levels is increased or bit rate is reduced.

- **Shannon Bit Rate or Shannon Capacity** (for noisy channel) defines the bit rate as (where, B is the bandwidth and SNR is the signal-to-noise ratio):

  $$Capacity = B \times \log_2 (1 + SNR)\ bits/sec$$

2.1.4 Network Performance

The following metrics are used:

(a) **Bandwidth**: can be used in two different contexts, (i) in Hertz, for range of frequencies of component signal or a channel, and (ii) in bits/sec, for the number of bits that a channel, link or network can transmit per second.

(b) **Throughput**: measure of how much data can be transferred from source to destination in a given timeframe.

(c) **Latency** (delay): defines how long it takes for an entire message to completely arrive at the destination from the time first bit is sent out. $Latency\ time = propagation\ time + transmission\ time + queuing\ time + processing\ time.$

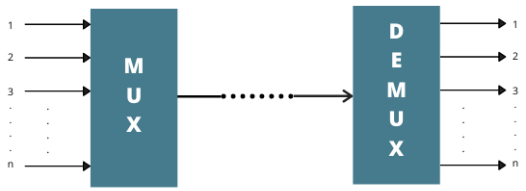
- Propagation time: The time required for bit to travel from sender to receiver, depends on medium and frequency of signal.
- Transmission time: The time required to push message's data bits into the wire.
  $Transmission\ time = Message\ size\ /\ bandwidth.$
- Queuing time: The time needed for each intermediate or end device to hold the message before it can proceed. It varies with load on the network.
- Processing time: The time required for devices to determine headers and process.

(d) **Bandwidth-Delay product**: defines the number of bits that can fill the link. The value is calculated as $Bandwidth \times Latency$.

2.2 Multiplexing

Multiplexing techniques allows simultaneous transmission of multiple signals across single data link. If bandwidth of a link is greater than bandwidth needs of the devices connected to it, the link can be shared so that bandwidth utilization is maximum.

In a multiplexed system, n channels share one link. MUX (multiplexer) combines the lines into one single stream while DEMUX (demultiplexer) separates the stream back into component transmissions.



Multiplexing & Demultiplexing

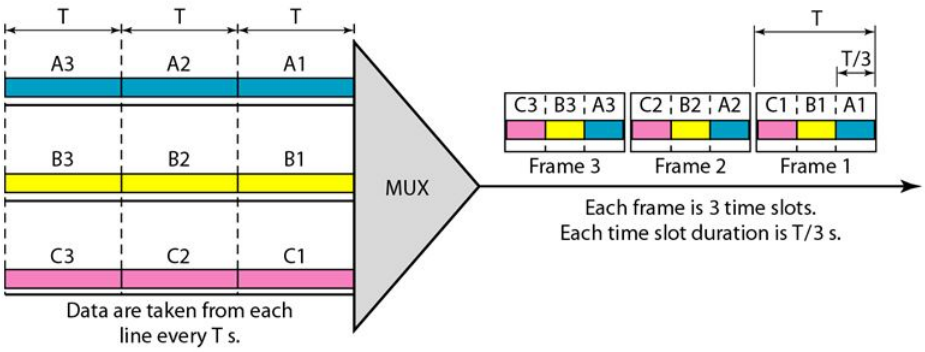
Basic types -

(a) **Frequency-Division Multiplexing** is an analog technique used when bandwidth of the link (in hertz) is greater than combined frequencies of signals to be transmitted. Each signal is modulated with different carrier frequencies and these modulated signals are then combined into one composite signal to be sent via the link that has enough bandwidth to accommodate it. The demux filters out the constituent signals and demodulator separates signals from their carrier frequencies and passes then to output lines.

(b) **Wavelength-Division Multiplexing** is designed to utilize high data-rate of fiber optic cables. Conceptually same as FDM, except it involves optical signals. Basic idea is to combine multiple light sources into one single light at mux and vice-versa at demux.

(c) **Time-Division Multiplexing** is a digital process where each connection occupies a portion of time in the link i.e. time is shared. It combines several low-rate channels into one high-rate one.

- **Synchronous TDM** where each input connection has an allotment in the output frame even if it's not sending any data. Here, a round of data units from each input connection is collected and interleaved into a frame with each unit having it's own slot in the frame. This technique can be inefficient if some input lines have no data to send.
- **Statistical TDM** where slots are dynamically allocated to improve bandwidth efficiency. Mux checks input lines in round-robin fashion, if any line has no data to send, it skips the line and checks the next line. Here, no slot is left empty as long as there is data to be sent by any input line. Of course, slots need to carry address of the destination to which the data belongs.



(d) **Code-Division Multiplexing -** The technique used is called Code Division Multiple Access, further explained in *Multiple Access* part.

2.3 Transmission Media

2.3.1 Guided Media

They provide a conduit from one device to another, include twisted-pair cable, coaxial cable, and fiber-optic cable. A signal traveling along any of these media is directed and contained by the physical limits of the medium.

- Twisted Pair Cable, consists of two conductors, each with its own insulation, twisted together. One wire is used to carry signals to the receiver, and the other is used only as a ground reference. The reciever uses the difference of the two. There are two categories: unshielded and shielded twisted pair cable.

- Coaxial Cable, has a central core conductor of solid or stranded wire enclosed in an insulating sheath, which is, in turn, encased in an outer conductor of metal foil or braid.

- Fibre Optic Cable, transmits signals in the form of light using total internal reflection.

2.3.2 Unguided Media

Also called wireless communication, it transports electromagnetic waves without using a physical conductor. Signals are broadcasted through free space and are available to anyone who has a device capable of receiving them. Wireless transmission can be divided into three broad groups: Radio waves, Microwaves, and Infrared waves.

Chapter 3

Data Link Layer

Specific responsibilities of data link layer include framing, addressing, flow control, error control, and media access control.

- The data link layer divides the stream of bits received from the network into manageable data units called frames.
- It adds header to the frame to define the addresses of sender and receiver of the frame.
- If rate at which data in received is less than rate of sending data, data link layer imposes flow control mechanism to prevent overwhelming the receiver.
- When two or more devices are connected to the same link, data link layer protocols are necessary to determine which device has control over the link.
- Data link layer adds reliability to the physical layer by adding mechanism to detect and retransmit lost or damaged frames.

3.1 Error Detection and Correction

Whenever bits flow from one point to another, they are subject to unpredictable changes because of interference. In a single-bit error, a 0 is changed to a 1 or a 1 to a 0. In a burst error, multiple bits are changed. Burst error length is measured from the first corrupted bit to last corrupted bit, though some bits in between may not have been corrupted.

Two main methods of error correction are

(a) Forward Error Correction - Receiver tries to guess the message using redundant bits.

(b) Retransmission - Receiver detects the occurrence of error and asks sender to resend the message.

3.1.1 Block Coding

Message is divided into blocks, each of k bits, called datawords. Redundant bits r is added to each block to make the length $n = k+r$. The resulting $n-bit$ blocks are called codewords. With k bits, 2^k datawords are possible; with n bits, we 2^n codewords can be created. Hence, $2^n - 2^k$ are invalid or illegal.

- Error can be detected when receiver has a list (or can find) valid codewords, and original codeword has changed to an invalid one. However, if the codeword is corrupted during transmission but the received word still matches a valid codeword, the error remains detected.
- Error can be corrected if receiver can guess (or find) the original codeword sent by adding more redundant bits.

Hamming Distance between two words is the number of differences between corresponding bits. $d(x, y)$ can be easily found by XORing the words $(x \oplus y)$ and counting the number of $1s$ in it. Also, hamming distance between the received codeword and send codeword is the number of bits that are corrupted during transmission.

Minimum Hamming Distance is the smallest hamming distance between all possible pairs in a set of words.

- To guarantee error detection of upto s errors, the minimum hamming distance in a block code must be $d_{min} = s + 1$. That is, if minimum distance between all valid codewords is $s + 1$, the received codeword cannot be erroneously mistaken for another codeword.
- To guarantee error correction of upto t errors, the minimum hamming distance in a block code must be $d_{min} = 2t + 1$.

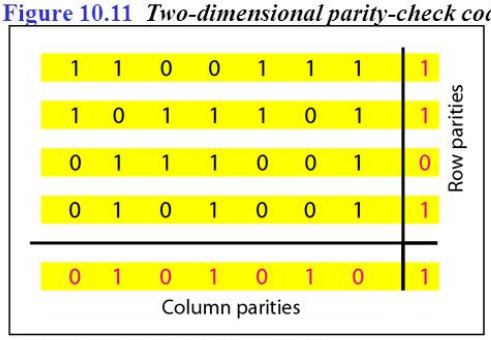
3.1.2 Linear Block Coding

In a linear block code, the XOR of any two valid codewords creates another valid codeword.

Simple Parity Check is single-bit error-detecting code in which $n = k + 1$ with $d_{min} = 2$. The extra bit, called parity bit is selected to make the total number of $1s$ in the codeword even. Simple parity check can detect an odd number of errors. At the receiver, the number of $1s$ modulus 2, called syndrome, in the word is checked and if it's 1 then recieved codeword is corrupted.

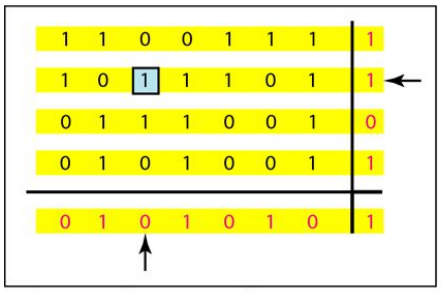
Two-dimensional Parity Check where dataword is organized in a table (rows and columns). For each row and column, parity-check bit is calculated. The table is the sent to the receiver which finds syndrome for each row and column. Upto three errors can be detected anywhere in the table.

Figure 10.11 *Two-dimensional parity-check code*

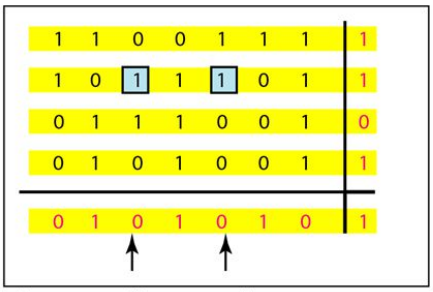


a. Design of row and column parities

Figure 10.11 *Two-dimensional parity-check code*



b. One error affects two parities



c. Two errors affect two parities

3.1.3 Hamming Codes

Hamming codes are set error-correcting codes using redundant bits (parity bits). The number of redundant bits are calculated using $2^r = k + r + 1$, where k is number of bits in dataword.

(1) Mark all bit positions that are powers of 2 as parity bits. All other bit positions are for the data to be encoded.

(2) The position of the parity bit determines the sequence of bits that it alternately checks and skips.

   (a) Position 1: Check 1 bit, skip 1 bit and so on. (1,3,5,7...)
   (b) Position 2: Check 2 bits, skip 2 bits and so on. (2,3,6,7,10,11...)
   (c) Position 4: Check 4 bits, skip 4 bits and so on. (4,5,6,7,12,13,14,15...)

(3) Set parity bit to 1 if number of 1s in the positions it checks is odd. Similarly, set to 0 if even.

For example, if dataword is 10011010,

- Codeword is created _ _ 1 _ 0 0 1 _ 1 0 1 0.
- Calculating parity for each parity bit.
   (a) Position 1 checks: **?** _ **1** _ **0** 0 **1** _ **1** 0 **1** 0 which sets the parity bit to 0 as **0** _ **1** _ **0** 0 **1** _ **1** 0 **1** 0
   (b) Position 2 checks: 0 **?** **1** _ 0 **0** **1** _ 1 **0** **1** 0 which sets the parity bit to 1 as 0 **1** **1** _ 0 **0** **1** _ 1 **0** **1** 0
   (c) Position 4 checks: 0 1 1 **?** **0** **0** **1** _ 1 0 1 **0** which sets the parity bit to 1 as 0 1 1 **1** **0** **0** **1** _ 1 0 1 **0**
   (d) Position 8 checks: 0 1 1 1 0 0 1 **?** **1** **0** **1** **0** which sets the parity bit to 0 as 0 1 1 1 0 0 1 **0** **1** **0** **1** **0**

Hence the final codeword is 011100101010.

The error can be detected by checking each parity bit and adding the positions of incorrect parity bits gives the position of error bit. Hamming code can correct single bit and detect upto double bit error.

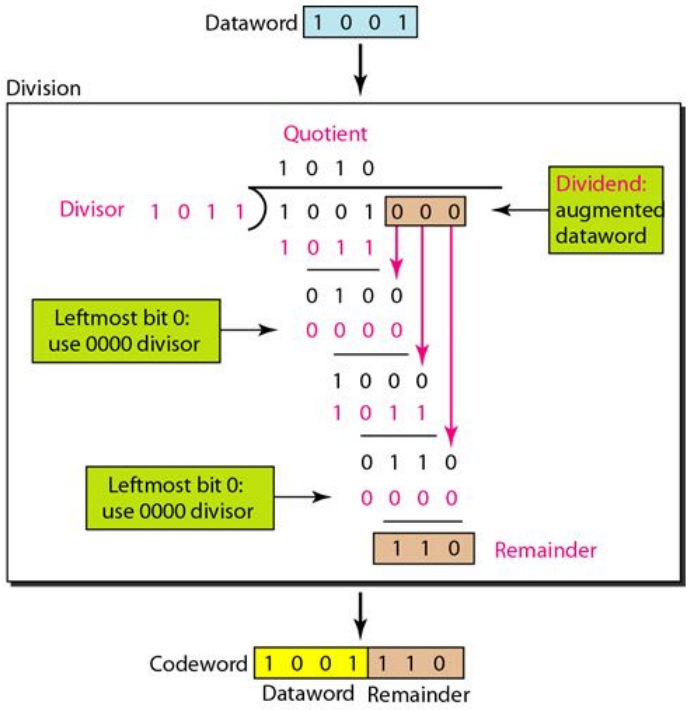
Suppose the codeword received is 0111**1**0101010. Clearly, parity bits 1 and 4 are incorrect so $1 + 4 = 5$ gives the position of error bit.

Burst error correction using hamming code can be done. The key is to split a burst error between several codewords, one error for each codeword. Normally, a packet or frame of data is sent, and for hamming code to respond to burst error of size N, one frame is split into N codewords. These codewords are sent one by one and compiled into one at the receiver.

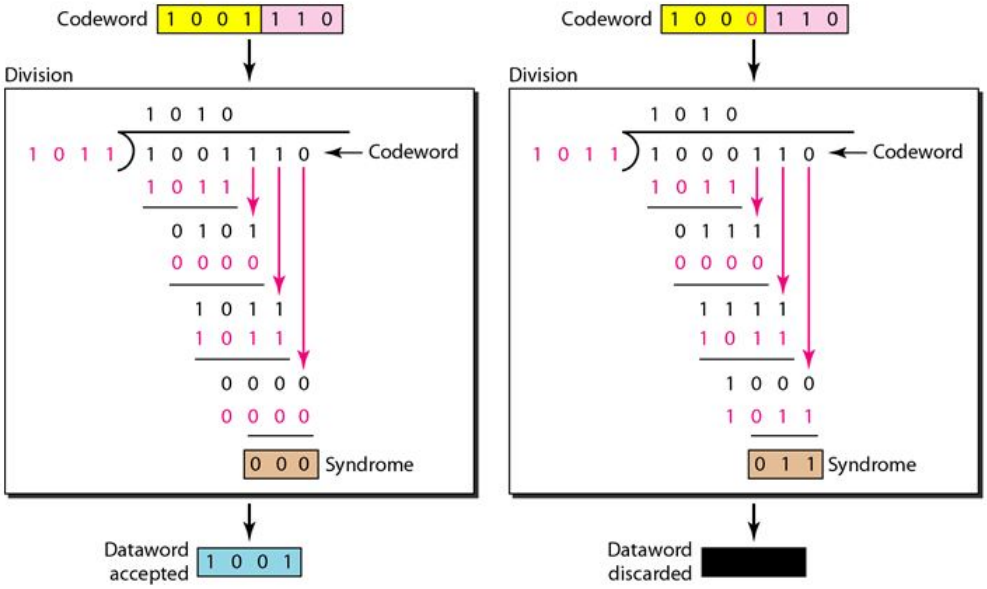
3.1.4 Cyclic Codes - Cyclic Redundancy Check

Codewords can change during transmission. Say, dataword has k bits and codeword has n bits.

- *Encoder* takes the dataword and augments it with $n-k$ number of 0s. It then divides the augmented dataword by the divisor which is predefined of size $n - k + 1$. Steps for modulo-2 binary division is shown in the figure. After division, the 3-bit remainder forms the check bits (r2, r1 and r0) and are appended to the dataword to create the codeword.



- *Decoder* does the same division process as the encoder. The remainder of the division is the syndrome. If the syndrome is all 0s, there is no error, the dataword is separated from the received codeword and accepted. Otherwise, everything is discarded. Division is shown in the figure.

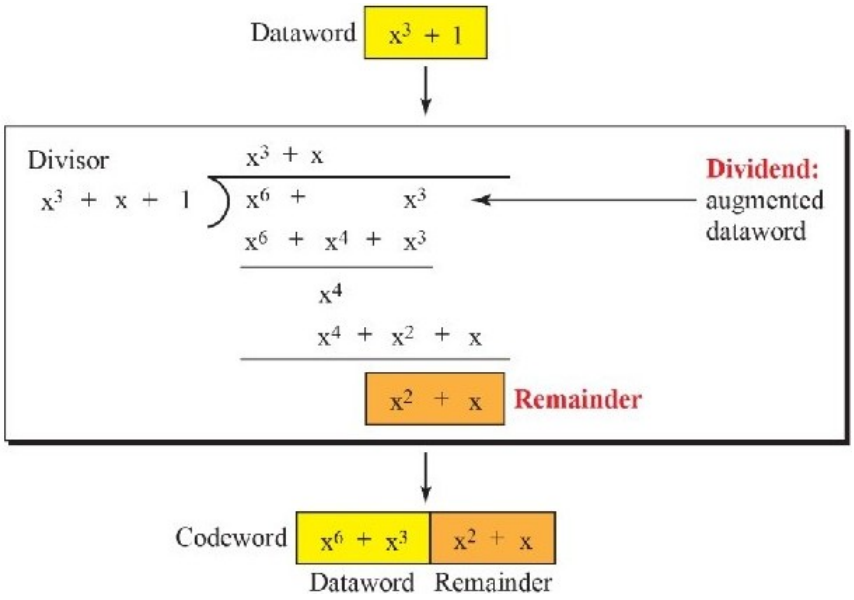


Using Polynomials - A pattern of 0s and 1s can be represented as a polynomial with coefficients 0 and 1. The power of each term shows the position of the bit; the coefficient shows the value of the bit. The divisor in cyclic code is normally called generator polynomial.

Operations on Polynomials

- Addition - combining the common terms and final coefficient is number of terms with same power modulo-2.

- Subtraction - deleting the common terms.
- Multiplication - powers are added as usual and the addition principle is used.
- Division - powers are subtracted as usual. Polynomial division is shown in encoding part below.



Cyclic Code Analysis - Say, $f(x)$ is a polynomial with binary coefficients. Then, following are defined, Dataword: $d(x)$, Codeword: $c(x)$, Generator: $g(x)$, Syndrome: $s(x)$, and Error: $e(x)$. If $s(x)$ is not zero, then one or more bit is corrupted. However, if $s(x)$ is zero, either no bit is corrupted or decoder failed to detect any errors. Now,

$$Received\ Codeword = c(x) + e(x)$$

$$\frac{Received\ Codeword}{g(x)} = \frac{c(x)}{g(x)} + \frac{e(x)}{g(x)}$$

Single-Bit Error If generator has at least two terms and the coefficient of x^0 is 1, then all single bit errors can be caught.

Two Isolated Single-Bit Errors Let the error be $e(x) = x^j + x^i$, where i and j define positions of errors and $j - i$ defines distance between the two errors. $e(x)$ can be written as $x^i(x^{j-i} + 1)$. If $g(x)$ has more than one term and one term is x^0, it cannot divide x^i. So, if $g(x)$ is to divide $e(x)$, it must divide $x^{j-i} + 1$. So, $g(x)$ must not divide $x^t + 1$, where t is between 0 and $n - 1$. However, $t = 0$ is meaningless and $t = 1$ is required in catching odd number of errors. That means essentially, t should be between 2 and $n - 1$.

3.1.5 CheckSum

The idea is to send the negative of the sum of the data along with the data, so that the receiver can add all the data including checksum to check if it adds upto 0 or not. 1's Complement is used for binary data. Any number from 0 to $2^n - 1$ can be represented in n bits. If the number has more than n bits, the extra leftmost bits are taken and addition is done with the n rightmost bits (wrapping). In 1's complement arithmetic, a negative number can be represented by inverting all bits. Internet uses a 16-bit checksum. Following are the steps:

- Sender's site:
  (a) Sender divides the message into 16-bit words.
  (b) All words are added using 1's complement addition.
  (c) The sum is complemented and becomes the checksum.
  (d) The checksum is sent with the data.
- Receiver's site:
  (a) At receiver's site, the message (including checksum) is divided into 16-bit words.
  (b) All words are added using 1's complement addition.
  (c) The sum is complemented and becomes the checksum.
  (d) If the value of checksum is 0, the message is accepted, otherwise rejected.

Note: Checksum is not as strong as CRC in error-checking capability.

3.2 Data Link Control

The functions include framing, flow and error control, and software-implemented protocols that provide smooth and reliable transmission of frames between nodes.

3.2.1 Framing

The data link layer packs bits into frames, so that each frame is distinguishable from the other, by adding sender address (helps receiver acknowledge the receipt) and destination address (where packet will go). The header contains source and destination addresses and control information, while the trailer carries error detection or error correction redundant bits.

- **Byte Stuffing,** where a special byte is added to the data section of the frame. This byte is usually called the escape character (ESC), which has a predefined bit pattern. Whenever the receiver encounters the ESC character, it removes it from the data section and treats the next character as data. Basically, byte stuffing is the process of adding 1 extra byte whenever there is a flag or escape character in the text.
- **Bit Stuffing** is the process of adding one extra 0 whenever five consecutive $1s$ follow a 0 in the data, so that receiver does not mistake the pattern 0111110 for a flag.

Flow Control is a set of procedures used to restrict the amount of data that the sender can send before waiting for acknowledgement.

Error Control is based on automatic repeat request, which is the retransmission of data.

3.2.2 Protocols

Stop and Wait (for noiseless channels) - To prevent receiver from becoming overwhelmed with frames, the sender needs to be notified somehow to slow down. There must be feedback from the receiver to the sender. In stop-and-wait, sender sends one frame, stops until it receives confirmation from the receiver, and then sends the next frame.

Stop and Wait Automatic Repeat Request (Stop and Wait ARQ) is done by keeping a copy of the sent frame and keeping a timer. If there is no ACK (auxiliary acknowledgement frame) from the receiver before the timer expires, the frame is retransmitted and again timer is started. Stop and Wait ARQ is very inefficient if the channel is long and has high bandwidth.

Go-Back-N ARQ where several frames can be sent before receiving acknowledgements. Here, frames from the sender are numbered sequentially. If m is the size of sequence number bits in header, the sequence numbers are modulo-2^m. **Sliding window** is used, which consists of series of frames which are awaiting ACK (called outstanding frames), series of frames which are to be sent. The window slides as ACK is recieved for outstanding frames and new frames that are yet to be sent are added to the window. Meanwhile, the receiver window contains a single frame, that is, the frame to be expected next. If the frame arrives safely, ACK is sent and window slides to the next expected frame.

Selective Repeat ARQ - Here, the size of window must be at most 2^{m-1} where m is the size of sequence number bits. The window size of sender and receiver is the same. Selective Repeat ARQ allows as many frames as the size of the receive window to arrive out of order and stored until they can be delivered.

- **Reason for 2^{m-1} max window size -** For $m = 2$, taking window size as 2. If all ACK are lost and timer for frame 0 expires, frame 0 is resent. However, receiver is now expecting frame 2, not frame 0, so the duplicate frame is correctly discarded. Now if the size of window is 3 (i.e. $> 2^{m-1}$ and all ACK are lost, the sender sends duplicate frame 0. However, receiver window expects frame 0, so it accepts frame 0, not a duplicate, but as the first frame in the next cycle, which is clearly an error.

Concept of Piggybacking is used to improve efficiency of the bidirectional protocols. When a frame is carrying data from node A to node B, it carries control information arrived (or lost) frames from B. Similarly, for frame carrying data from B to A, it carries control information about arrived (or lost) frames from A.

3.2.3 High-level Data Link Control (HDLC)

HDLC is a bit-oriented protocol for communication over point-to-point and multipoint links. It implements ARQ mechanisms.

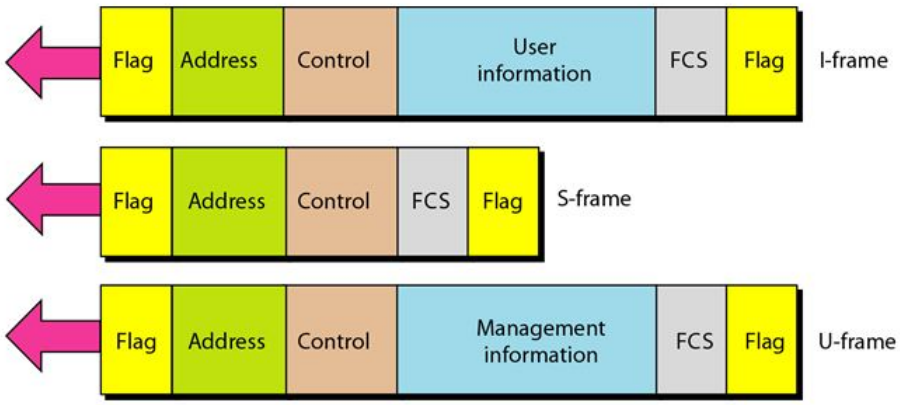
Transfer Modes - HDLC provides two common transfer modes that can be used in different configurations.

- **Normal Response Mode (NRM) -** In NRM, the station configuration is unbalanced. There's one primary station that can send commands, and multiple secondary stations that can only respond. NRM is used for both point-to-point and multipoint links.
- **Asynchronous Balanced Mode (ABM) -** In ABM, the configuration is balanced. The link is point-to-point, and each station can function as a primary and a secondary station.

Frames - To provide flexibility to support all options posssible in modes and configurations, HDLC defines three types of frames:

- Information frames (I-frames) are used to transport user data and control information relating to user data (piggybacking).
- Supervisory frames (S-frames) are used only to transport control information.

- Unnumbered frames (V-frames) are reserved for system management. Information carried is intended for managing the link itself.



HDLC frames

Each frame in HDLC may contain upto six fields:

(1) Flag field of an HDLC frame is 8-bit sequence with the bit pattern 0111110 that identifies both the beginning and the end of a frame and serves as synchronization patter for the receiver.

(2) Address field. If primary station created the frame, it contains a *to* address. If secondary station creates the frame, it contains *afrom* address. Address field can be 1 byte or several bytes long. One byte can address upto 128 (2^7) stations. If address field is only 1 byte, the last bit is always 1. If address is more than 1 byte, all bytes other than the last one will end with 0, and the last byte will end with 1. Ending each byte with 0 indicates to the receiver that there are more address bytes to come.

(3) Control field is a 1 or 2 byte segment of the frame used for flow or error control. The interpretation of bits in this field depends on the frame type.

(4) Information field contains user data from the network layer or management information.

(5) FCS field. Frame check sequence field is HDLC error detection field. It can contain either 2 or 4 byte CRC.

Control field determines the type of frame and defines its functionality.

- Control Field for *I-frames* - I-frames are designed to carry user data from network layer. The first bit defines the type (0 if an I-frame). Next 3 bits, called $N(S)$, define sequence number of the frame (from 0 to 7). The last 3 bits, called $N(R)$, correspond to the acknowledgement number when piggybacking is used. The single bit between $N(S)$ and $N(R)$ is called the *PIF* bit. If *PIF* bit is set, it means poll or final. It means poll when frame is sent by primary station to a secondary station, and final when frame is sent by secondary station to primary.

- Control Field for *S-frames* - S-frames are used for flow or error control whenever piggybacking is either impossible or inappropriate. If first 2 bits of control field are 10, it means frame is an S-frame. The last three bits, called, $N(R)$ corresponds to ACK or NAK(negative ACK). The 2 bits called code is used to define the type of S-frame itself. With 2 bits, four types are possible:

(a) Receive Ready (RR) when value of code field is 00. RR S-frame acknowledges the receipt of a safe and sound frame. Here, $N(R)$ field defines the acknowledgement number.

(b) Recieve Not Ready (RNR) when value of code field is 10. RNR S-frame acknowledges the receipt of frame and announces that receiver is busy and cannot receive more frames (kind of asking sender to slow down).

(c) Reject (REJ) when value of code field is 01. REJ S-frame is NAK frame, the one used in Go-Back-N ARQ to improve efficiency by informing the sender beforehand (before timer expiry) that frame is lost or damaged.

(d) Selective Reject (SREJ) when value of code field is 11. SREJ S-frame is NAK frame used in Selective Repeat ARQ.

- Control Field for *V-frames* - V-frame codes are divided into two sections i.e. 2-bit prefix before the *PIF* bit and a 3-bit suffix after *PIF* bit. Together, the 5 bits can create upto 32 different types of V-frames.

3.2.4 Point-to-Point Protocol

PPP is byte-oriented protocol and provides several services:

(a) PPP defines format of frame to be exchanged between devices.

(b) PPP defines how two devices can negotiate the establishment of the link and exchange of data.

(c) PPP defines how network layer data are encapsulated in the data link frame.

(d) PPP defines how two devices can authenticate each other.

(e) PPP provides connections over multiple links.

(f) PPP provides network address configuration.

Frame format in PPP

(1) Flag - PPP frame starts with 1-byte flag with the bit pattern 01111110. The flag is treated as a byte and not individual bits.

(2) Address - Address field is constant value and set to 11111111.

(3) Control - The field is set to constant value 11000000. This field is not needed at all as PPP does not provide any flow control or error correction.

(4) Protocol - The field defines what is being carried in the data field. The field is by default 2-byte long.

(5) Payload - The field carries either user data or other information. The field is default of maximum 1500 bytes. The data field is byte-stuffed if flag appears in the field. The escape byte is 01111101. Also, padding is needed if the size is less than the maximum default value.

(6) FCS - The frame check sequence is simply 2-byte or 4-byte standard CRC.

Transition Phases that PPP connection goes through are as follows:

- Dead phase where the link is not being used. There is no active carrier.
- Establish phase when one of the nodes starts communication. In this phase, options are negotiated between the two nodes. If it's successful, connection goes into authentication phase (if required), or directly to networking phase.
- Authenticate phase is optional. Nodes send several authentication packets, if result is successful, connection goes into networking phase, otherwise, it goes in termination phase.
- Network layer, where negotiations for network layer protocols takes place. Since PPP supports multiple protocols at network layer, the receiver needs to know which protocol will receive the data.
- Open phase where data transfer takes place. The exchange of data packets can be started and takes place until one of the endpoints terminates the connection.
- Terminate phase where connection is terminated and packets are exchanged for closing the link.

Three set of protocols make PPP powerful. The Link Control Protocol (LCP), two Authentication Protocols (APs), and several Network Control Protocols (NCPs). At any moment, PPP packet can carry data from one of these protocols in its data field.

- Link Control Protocol (LCP) is responsible for establishing, maintaining, configuring, and terminating links. All LCP packets are carried in the payload field of the PPP frame.
- Authentication Protocols (APs) in PPP are two:
  (a) Password Authentication Protocol (PAP) is simple authentication process with two steps: (i) user who wants access to system sends an identification and password. (ii) System checks validity of the user and either accepts or denies connection.
  (b) Challenge Handshake Authentication Protocol (CHAP) provides greater security as user password is never sent online. The system sends challenge packet containing a challenge value. User enters their password and a result is created from challenge value and the password using a predefined function. The result is sent to the system and system does the opposite to decode and verify the user.
- Network Control Protocols (NCPs) are what is defined in PPP for each network protocol.

3.3 Multiple Access

Data link layer can be considered as two sublayers. Upper sublayer is responsible for data link control and is called logical link control (LLC) layer. Lower sublayer is reponsible for resolving access to the shared media and is called medium access control (MAC) layer. If channel is dedicated, then the lower sublayer is not needed. When nodes or stations are connected and use a common link, called a multipoint or broadcast link, multiple-access protocol is needed to coordinate access to the link.

3.3.1 Random Access

There is no scheduled time for a station to transmit. Transmission is random among the stations. Also, no rules specify which station should send next. Stations compete with one another to access the medium. However, if more than one stations tries to send data, there is an access conflict (collision) and the frames will be either destroyed or modified.There are two common methods: ALOHA and CSMA

Carrier Sense Multiple Access (CSMA) - The chance of collision can be reduced if a station senses the medium before trying to use it. CSMA requires that each station first listen to the medium before sending. CSMA can reduce the possibility of collision, but it cannot eliminate it. Even then, possibility of collision still exists due to propagation delay.

Persistence Methods

- **I-Persistent -** After the station finds the line idle, it sends the frame immediately (with probability I). This method has the highest chance of collision because two or more stations may find the line idle and send their frames immediately.
- **Non-Persistent -** A station that has a frame to send senses the line. If the line is idle, it sends immediately. If the line is not idle, it waits a random amount of time and then senses the line again. This approach reduces the chance of collision because it's unlikely that two or more stations will wait the same amount of time and retry to send simultaneously. However, this method reduces the efficiency of the network because the medium remains idle when there may be stations with frames to send.
- **P-Persistent -** If the channel has time slots with a slot duration equal to or greater than the maximum propagation time. It reduces the chance of collision and improves efficiency. Here, if the station finds the line idle, following steps are followed: With probability p, the station sends its frame. With probability $q = 1 - p$, the station waits for the beginning of the next time slot and checks the line again, if the line is idle, it goes to previous step; if the line is busy, it acts as though a collision has occurred and uses the back-off procedure.

Carrier Sense Multiple Access with Collision Detection (CSMA/CD) - CSMA method does not specify the procedure following a collision. In CSMA/CD, a station monitors the medium after it sends a frame to see if the transmission is successful, if yes, the station is finished, else if there's a collision, the frame is sent again. When there is no collision, the station receives only one signal: its own signal. If there is a collision, the station receives two signals: its own signal and signal transmitted by a second station.
How it works - Collision detection is a continuous process. Frame size needs to be specified such that when the bits of the frame are transmitted, and a collision is detected, the information of collision can come back to the sender before the last bit of the frame is sent. So, the sender does not need to keep a copy of the frame once entire frame is transmitted without any collision. Here, the frame transmission time T_{fr} must be at least two times the maximum propagation time T_P.

Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) - It was developed for wireless networks where most of the energy is lost in transmission and a collision may only add upto 5 or 10 percent additional energy in the sender's signal. There are three strategies in CSMA/CA:

(a) Interframe Space (IFS) - When an idle channel is found, station does not transmit immediately. It waits for a period of time called IFS since even though channel appears to be idle, another station must've started transmitting but the signal hasn't reached the station. If after IFS, the channel is still found idle, the station can send but it still needs to wait for a time period of contention time.

(b) Contention Window, is an amount of time divided into slots. A station that is ready to send chooses a random number of slots as its wait time. It is set to one slot the first time and then doubles each time the station cannot sense an idle channel after IFS time. If station finds the channel busy after a time slot, it does not restart the process, it just stops the timer and starts back again when the channel is sensed idle. This gives priority to the station with the longest waiting time.

(c) Acknowledgements can help guarentee that receiver received the frames.

3.3.2 Controlled Access

The stations consult one another to find which station has the right to send. There are three popular methods:

(a) **Reservation -** A station needs to make a reservation before sending data. Time is divided into intervals and in each interval, a reservation frame precedes the data frames sent in that interval.

(b) **Polling** works with topologies in which one station is primary station and others are secondary stations. The primary device controls the link and gets to determine which station is allowed to use the channel at a given time.

(c) **Token Passing -** The stations in the network are organized in a logical ring i.e. for each station, there is a *predecessor* and a *successor*. The current station is one that is accessing the channel right now. The right to access the channel is done by a special packet called *token* that circulates through the ring. The station possessing the token has the access to the channel and sends its data. When its done, the token is passed to its successor and so on.

3.3.3 Channelization

- **FDMA** or Frequency Division Multiple Access where the available bandwidth of the common channel is divided into bands that are separated by guard bands.

- **TDMA** or Time Division Multiple Access where the bandwidth is just one channel that is time-shared between different stations.

- **CDMA** or Code Division Multiple Access where one channel carries all transmissions simultaneously.

  – *Idea:* Let's assume there are four staions A, B, C, and D connected to the same channel. Bit 0 is represented by -1, bit 1 by +1, and 0 if the channel is silent. The data from corresponding stations are d_A, d_B, d_C, and d_D. Each station is assigned a code (generated by Walsh table), which is sequence of numbers called chips. Chips have two properties: Multiplying any two chips results 0 and multiplying any chip with itself gives number of stations or number of bits in the chip. Let the chip codes be c_A, c_B, c_C, c_D. While sending, data from each station is encoded with spreading chip using multiplication/XNOR and then added and the sum is sent through the channel, which is $[d_A.c_A \quad d_B.c_B \quad d_C.c_C \quad d_D.c_D]$. Now, if station C wants to hear what station A is saying, C multiplies the data on the channel with chip code of A (c_A) and divides by the number of stations or number of bits in the chip code to get the bit that A sent.

  – *Example:*

    * Assuming stations A, B, C and D want to send 0, 0, nothing, and 1 respectively. So, $d_A = -1$, $d_B = -1$, $d_c = 0$ and $d_D = +1$.
    * Let chip codes be $c_A = [+1 \ +1 \ +1 \ +1]$, $c_B = [+1 \ -1 \ +1 \ -1]$, $c_C = [+1 \ +1 \ -1 \ -1]$, and $c_D = [+1 \ -1 \ -1 \ +1]$. Each station multiplies the data with chip code which gives $d_A.c_A = [-1 \ -1 \ -1 \ -1]$, $d_B.c_B = [-1 \ +1 \ -1 \ +1]$, $d_C.c_C = [0 \ 0 \ 0 \ 0]$, and $d_D.c_D = [+1 \ -1 \ -1 \ +1]$.
    * Now, the data sent in the common channel is the sum which is $[-1 \ -1 \ -3 \ +1]$.
    * If station C who is silent wants to listen to station B, C multiplies data on the channel by chip code of station B, which gives $[-1 \ -1 \ -3 \ +1].[+1 \ -1 \ +1 \ -1] = -4$. Dividing the result gives -1 which tells C that B sent bit 0.

Chapter 4

Network Layer

The network layer is responsible for host-to-host delivery and for routing the packets through the routers or switches. Network layer at the switch or router is responsible for routing the packet. When a packet arrives, the router or switch consults its routing table and finds the interface from which the packet must be sent. The packet, after some changes in header (routing information) is passed to data link layer again. The network layer at the destination is responsible for address verification. If packet is a fragment, the network layer waits for all fragments to arrive, then reassembles them and delivers to packet to transport layer.

4.1 Logical Addressing

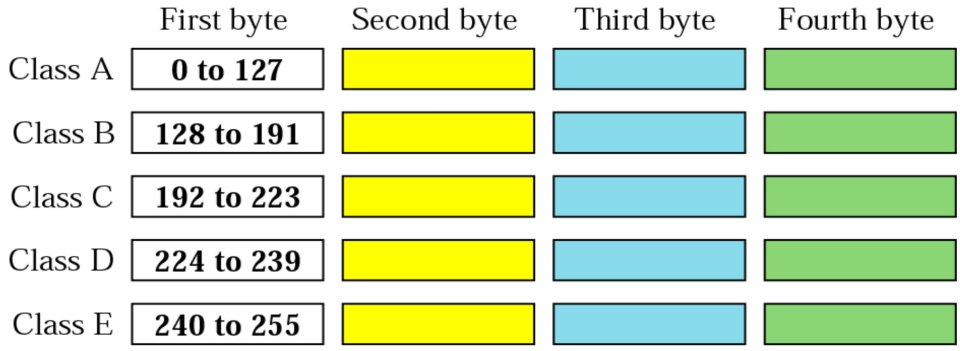
For a computer somewhere in the world to communicate to another computer somewhere else in the world, a global addressing scheme is required.

4.1.1 IPv4 Address

An IPv4 address is a 32-bit address that uniquely and universally defines the connection of a device to the Internet. Since IPv4 is 32-bit, the address space is 2^{32} or 4,294,967,296. There are two ways to represent addresses:

1. *Binary Notation*, where 32 bits can be divided into octets which makes IPv4 addresses 4-byte long. An example, 01110101 10010101 00011101 00000010
2. *Decimal Notation*, which is the dotted-decimal representation of the address. An example, 117.149.29.2

Classful Addressing where the address space is divided into five classes: A, B, C, D, E.



In classful addressing, the address in class A, B, and C is divided into *net-id* and *host-id*.

(a) For class A, net-id is defined by the first byte (where first bit is always 0). The rest 7 bits are used to define blocks, i.e. $2^7 = 128$ blocks. The host-id is defined by rest 3 bytes i.e. 24 bits which gives the number of hosts for each block as $2^{24} = 16,777,216$. So, 128 blocks can be assigned to 128 organisations where each organisation can have 16,777,216 hosts connected to the network.

(b) For class B, net-id is defined by first two bytes (where first two bits of first byte is always 10). The rest 14 bits can calculate the number of blocks in class B i.e. $2^{14} = 16,384$. The next two bytes denotes the host-id which is 16 bits that gives the number of hosts as $2^{16} = 65,536$.

(c) For class C, net-id is defined by first three bytes (where first 3 bits always stays 110). The 21 bits define the number of blocks as $2^{21} = 2,097,152$. The last byte defines the host-id which is upto $2^8 = 256$.

(d) Class D where all the addresses are of one single block. The class D addresses are designed for multicasting. The first four bits always stays as 1110.

(e) Class E addresses are one block addresses which are reserved for future use. The first four bytes are always 1111.

In classful addressing, a large part of available addresses are wasted.

Default mask can be used to find net-id for classes A,B, and C. The net-id can be derived from the IPv4 address by using logical AND on the address and the default mask.

Class	*Binary*	*Dotted-Decimal*
A	11111111 00000000 00000000 00000000	255.0.0.0
B	11111111 11111111 00000000 00000000	255.255.0.0
C	11111111 11111111 11111111 00000000	255.255.255.0

Subnetting was introduced during times of classful addressing. If the organization was granted a large block in class A or B, it could divide the addresses into several contigous groups and assign each group to smaller networks (called subnets). Subnetting increases the number of 1s in the mask.

Supernetting where an organisation can combine several C class blocks to create a larger range of addresses. This way, several networks are combined to create a super-network or supernet.

Classless Addressing where addresses are granted in blocks but no classes exist. To simplify handling of addresses, following three restrictions are imposed on classless address blocks:

(a) The addresses in a block must be contiguous.

(b) The number of addresses in a block must a power of 2.

(c) The first address must be evenly divisible by the number of addresses.

Mask is a 32-bit number in which n leftmost bits into $1s$ and the $32 - n$ rightmost bits are $0s$. In classless addressing, it is convenient to give just the value of n preceded by slash (Classless Interdomain Routing or CIDR notation).

In classless addressing, a block of address can be defined as *x.y.z.t/n*. The first address can be found by setting the $32 - n$ rightmost bits in the binary notation of the address to 0s. The last address can be found by setting the rightmost $32 - n$ bits to 1s. The number of addresses in the block can be found as 2^{32-n}.

If subnets are required within a block, the value n_{sub} for subnet mask is given by $n_{sub} = n + \log_2 \frac{N}{N_{sub}}$, where N is number of addresses in the block and N_{sub} is the size of sub-block.

Network Address Translation (NAT) enables a user to have a large set of addresses internally (private IP) and a small set of addresses externally (public IP). The following set of addresses are reserved as private addresses, which are unique inside the organisation, but not unique globally (no router will forward a packet that has one of these as destination addresses):

Range			*Total*
10.0.0.0	to	10.255.255.255	2^{24}
172.16.0.0	to	172.31.255.255	2^{20}
192.168.0.0	to	192.168.255.255	2^{16}

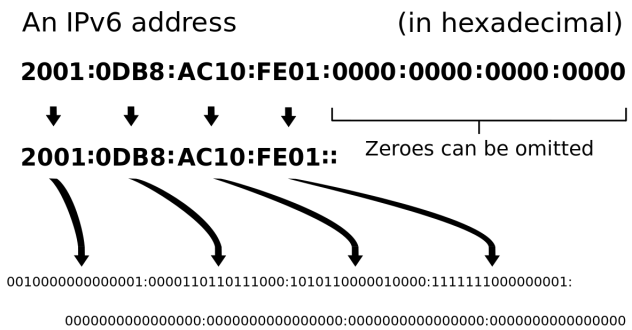
NAT and ISP An ISP that serves dial-up customers use NAT to conserve address. Suppose an ISP is granted 1000 addresses and serves 100,000 customers. Each customer is assigned a private address. The ISP translates 100,000 outgoing packets to one of the 1000 global addresses, similarly, it translates the global destination address in incoming packets to the corresponding private address.

4.1.2 IPv6 Address

An IPv6 address is 128 bits long i.e. 16 bytes (octets). IPv6 has an address space of 2^{128}.

Hexadecimal Colon Notation makes the addresses more readable. Here, 128 bits is divided into eight sections, each 2 bytes in length. Two bytes in hexadecimal notation requires four hexadecimal digits. Therefore, address consists of 32 hexadecimal digits, with every four digits separated by a colon.

Abbreviation is used since even in hexadecimal notation, the address is long and many digits are zeros. The leading zeros of a section (four digits between two colons) can be omitted. Only the leading zeros can be dropped, not the trailing zeros. Further abbreviation are possible if there are consecutive sections consisting of zeros only, which can be removed altogether and replaced with double semicolon. This type of abbreviation is allowed only once per address.



Unicast Address defines a single computer. The packet sent to unicast address must be delivered to that specific computer. IPv6 defines two types of unicast address: geographically based and provider-based.

Multicast Address are used to define a group of hosts instead of one. A packet sent to multicast address must be delivered to each member of the group.

4.2 Internet Protocol

Connection Oriented protocol is when the source makes connection with the destination before sending a packet. When connection is established, packets are sent in a sequence. There is a relationship between packets i.e. a packet is logically connected to the packet traveling before and packet traveling after it. When all packets of a message is sent, connection is terminated.

Connectionless protocol is when network layer protocol treats each packet independently, with each packet having no other relationship to any other packet. The packets in a message may or may not travel the same path to the destination. *Communication at the network layer in the Internet is connectionless.*

4.2.1 IPv4 Datagram

A **datagram** is a variable-length packet consisting of header and data. The header is 20 to 60 bytes in length and contains information about routing and delivery. It is customary in TCP/IP to show the header in 4-byte sections. The fields are as follows:

(a) **Version (VER)** is a 4-bit field that defines the version of IPv4 protocol. Currently, the version is 4. If machine is using some other version of IPv4, the packet is discarded.

(b) **Header Length (HLEN)** is a 4-bit field that defines the total length of the datagram header in 4-byte words. Say if header length is 20 bytes, the value of this field is 5 ($5x4 = 20$).

(c) **Services** is an 8-bit field. The field, previously called service type, is now called differentiated services.

(d) **Total Length** is a 16-bit field that defines the total length (header + data) of the IPv4 datagram in bytes. Length of data can be easily found by subtracting header length from total length.

(e) **Identification** is a field used in fragmentation.

(f) **Flags**

(g) **Fragmentation Offset**

(h) **Time to Live** which stores the counter to countdown the lifespan of a datagram. Everytime the datagram visits a router, the counter is decreased. If the counter becomes 0 at a point, the router discards the datagram. The field is also used to intentionally limit the journey of a packet.

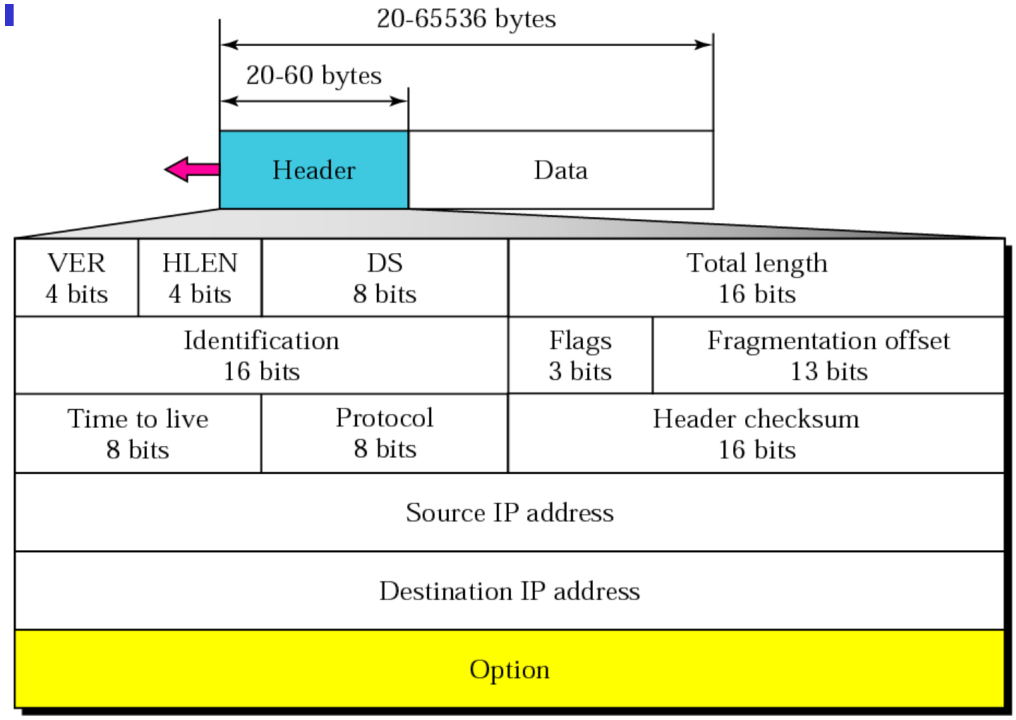
(i) **Protocol** is an 8-bit field that defines the higher-level protocol that uses the services of the IPv4 layer. Since IPv4 packet carries data from several different protocols (like TCP,UDP,ICMP), the value of this field helps the receiving network layer know to which protocol does the data belong.

(j) **CheckSum**

(k) **Source Address** is a 32-bit field that defines the IPv4 address of the source host.

(l) **Destination Address** is a 32-bit field that defines the IPv4 address of the destination host.

Options in IPv4 Datagram The header of the IPv4 is made of two parts: a fixed part and a variable part. The fixed part is 20 bytes long. The variable part comprises of the options that can be maximum of 40 bytes. Options, as the name implies, are not required for a datagram. They can be used for network testing and debugging.



IPv4 Datagram Format

4.2.2 Fragmentation (IPv4)

A datagram can travel through different networks and each router decapsulates IPv4 datagram from the frame it receives, processes it, and the encapsulates in another frame. The format and size of the frame depend on the protocol used by the physical network through which the frame has just traveled.

Maximum Transfer Unit (MTU) field defines the maximum size of data field in a frame. To make IPv4 protocol independent of the physical network, the designers have decided to make the maximum length of IPv4 datagram equal to 65,535 bytes. If the maximum size of datagram exceeds the MTU, the packet is divided into several small packets and then sent (called **fragmentation**).

The source usually does not fragment the datagram. The transport layer will instead segment the data into a size that can be accommodated by IPv4 and data link layer in use. When the datagram is fragmented, required parts of header must be copied by all fragments. The host or router that fragments the datagram must change the values of three fields: flags, fragmentation offset and total length. The rest of the fields must be copied. Of course, the value of checksum must be recalculated regardless of fragmentation.

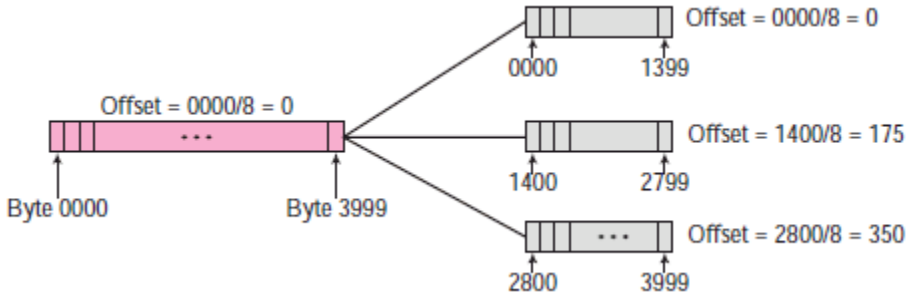
Fields Related to Fragmentation

- **Identification** is a 16-bit field that identifies a datagram originating from the source host. The combination of identification and source IPv4 address must uniquely define a datagram as it leaves

the source host. When a datagram is fragmented, identification value is copied to all fragments which helps destination in reassembling the datagram.

- **Flags** is a 3-bit field Reserve, DoNotFragment, MoreFragment. If value of DoNotFragment is 1, the machine must not fragment the datagram. If it cannot pass it through any available physical network, the datagram is discarded and ICMP error message is sent to host. If the value is 0, the datagram can be fragmented if necessary. If value of MoreFragment is 1, it means that the datagram is not the last fragment, there are more fragments after this one, and otherwise, if the value is 0.
- **Fragmentation Offset** is a 13-bit field that shows the relative position of this fragment with respect to the whole datagram. It is the offset of the data in the original datagram measured in units of 8 bytes.

For example, if the datagram has data size of 4000 bytes, it's fragmented into three fragments. The bytes in the original datagram is numbered from 0 to 3999. The first fragment carries bytes 0 to 1399. The offset for this datagram is $0/8 = 0$. The second fragment carries bytes 1400 to 2799 and the offset for this fragment is $1400/8 = 175$. Finally, the third fragment carries bytes 2800 to 3999 and the offset value for this fragment is $2800/8 = 350$.
Since the value of offset is measured in 8 bytes and size of offset field is 13 bits, routers and hosts that fragment datagrams are forced to choose a fragment size so that the first byte number is divisible by 8.



4.2.3 IPv6 Datagram

IPv6 has advantages over IPv4 that are summarized as follows:

- Larger address space of 2^{128} as compared to 2^{32} addresses in IPv4.
- Better header format in which options are separated from base header and inserted, when needed, between the based header and the upper-layer data. This simplifies and speeds up the routing process.
- New options to allow for additional functionalities.
- Allowance for extension, allows for extension of the protocol if required by new technologies.
- Support for resource allocation to enable the source to request special handling of the packet (can be used to support real-time audio and video).
- Support for more security where encryption and authentication options provide confidentiality and integrity of the packet.

Each packet in IPv6 is composed of a mandatory base header (40 bytes), followed by payload consisting of optional extensional headers and data from an upper layer, of size upto 65,535 bytes. Fields in base header:

(a) **Version** is 4-bit field that defines the version number of IP, in case of IPv6, the value of the field is 6.

(b) **Priority** is 4-bit field that defines priority of the packet with respect to traffic congestion.

(c) **Flow Label** is 3-byte (24-bit) field that is designed to provide special handling for a particular flow of data.

(d) **Payload Length** is 2-byte field that defines the length of datagram excluding the base header.

(e) **Next Header** is 8-bit field defining the header that follows the base header in the datagram. The next header is either one of the optimal extension headers used by IP or the header of an encapsulated packet such as UDP or TCP.

(f) **Hop Limit** is 8-bit field similar to time-to-live field of IPv4.

(g) **Source Address** is 16-byte field that identifies source of the datagram.

(h) **Destination Address** is 16-byte field that usually identifies final destination of the datagram. If source routing is used, this field contains the address of the next router.

IPv4 and IPv6 Header Comparison

- Header length field is eliminated in IPv6 because the length of the header is fixed.
- Service type field is eliminated in IPv6. Priority and flow label fields together take over its function.
- Total length field is eliminated in IPv6 and replaced by the payload length field.
- Identification, flag, and offset fields are eliminated from the base header in IPv6. They are included in fragmentation extension header.
- Time-to-live field is called hop limit in IPv6.
- Protocol field is replaced by the next header field in IPv6.
- Header checksum is eliminated because the checksum is provided by upper-layer protocols and therefore not needed in this level.
- Options fields of IPv4 are implemented as extension headers in IPv6.

4.2.4 Transition from IPv4 to IPv6

The transition must be smooth to prevent any problems between IPv4 and IPv6 systems. Following are the three strategies devised for the transition:

- **Dual Stack:** A station must run IPv4 and IPv6 simultaneously until all the internet uses IPv6. To determine which version to use when sending a packet to the destination, the source host queries the DNS. If the DNS returns an IPv4 address, source host sends an IPv4 packet and otherwise, it sends an IPv6 packet.
- **Tunneling** is a strategy used when two stations using IPv6 want to communicate with each other and packet must pass through a region that used IPv4. To pass through the region, the packet must have an IPv4 address. So the IPv6 packet is encapsulated in an IPv4 packet when it enters the region, and it leaves its capsule when it exits the region. It seems as if the IPv6 packet goes through a tunnel at one end and emerges at the other end. To make it clear that IPv4 packet is carrying an IPv6 packet as data, the protocol value is set to 41.

- **Header Translation** is necessary when majority of internet has moved to IPv6 but some systems still use IPv4. The sender wants to use IPv6, but the receiver does not understand IPv6. Tunneling does not work in this situation because the packet must be in IPv4 format to be understood by the receiver. In this case, the header format must be totally changed through header translation. The header of the IPv6 packet is converted to an IPv4 header. Header translation process is as follows:

  (a) IPv6 mapped address is changed to an IPv4 address by extracting the rightmost 32 bits.

  (b) Value of the IPv6 priority field is discarded.

  (c) Type of service field in IPv4 is set to zero.

  (d) Checksum of IPv4 is calculated and inserted into corresponding field.

  (e) IPv6 flow label is ignored.

  (f) Compatible extension headers are converted to options and inserted in the IPv4 header. Some may be dropped.

  (g) Length of IPv4 header is calculated and inserted into the corresponding field.

  (h) Total length of the IPv4 packet is calculated and inserted in the corresponding field.

4.3 Routing

Routing table can be static or dynamic. *Static table* is one with manual entries. *Dynamic table* is updated automatically when there is change somewhere in the internet. Today, internet needs dynamic routing for efficient delivery of the IP packets.

Autonomous System is a group of networks and routers under the authority of a single administration. Routing inside an autonomous system is referred to as *intra-domain routing*. Routing between autonomous systems is referred to as *inter-domain* routing. Each autonomous system can choose one or more intra-domain routing protocols to handle routing inside the autonomous system. However, only one interdomain routing protocol handles routing between autonomous systems.

4.3.1 Intra-Domain Routing Protocols

1) Distance Vector Routing - Here, the algorithm used is based on the working of Bellman Ford algorithm. The least-cost route between any two nodes is the route with minimum distance. Each node maintains a vector of minimum distances to every node. The table contains information: To (destination), Cost, Next (next hop).

- *Initialization:* Initially, each node can know only the distance between itself and its immediate neighbours. Assuming that each node can send a message to the immediate neighbours and find the distance between itself and these neighbours. The distance for any entry that is not a neighbour is marked as infinite.

- *Sharing:* Each node shares its routing table with its immediate neighbours periodically and when there is a change.

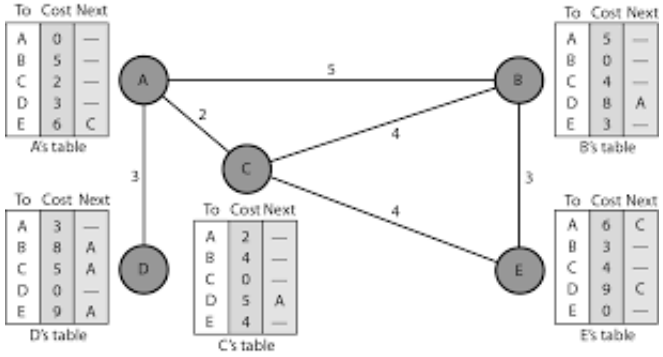
- *Update:* When a node receives a two-column table from a neighbour, it needs to update its routing table.

  (1) The receiving node needs to add the cost between itself and the sending node to each value in the second column.

(2) The receiving node needs to add the name of the sending node to each row as the third column if the receiving node uses information from any row. The sending node is the next node in the route.

(3) The receiving node needs to compare each row of its old table with the corresponding row of the modified received table. (3a) If the next node entry is different, the receiving node chooses the node with the smaller cost. If there is a tie, the old one is kept. (3b) If next node entry is the same, the receiving node chooses the new row.

- *When to share* the table to its immediate neighbours. It can be:
  - Periodic Update - A node sends its routing table, normally every 30 secs or any other defined period.
  - Triggered Update - A node sends its routing table to its neighbours anytime there is a change in its routing table.
- *Issues in DVR*
  - *Two-Node Loop Instability* - A network using this protocol can become unstable if some entries of both the tables changes continuously forming an infinite loop.
  - *Three-Node Instability* - The two-node instability can be avoided by using the split horizon strategy combined with poison reverse (further reading suggested). However, if there is instability between three nodes, stability cannot be guaranteed.



Routing Information Protocol (RIP) implements distance vector routing directly with some considerations. The metric used is hop count, that is, the distance is defined as the number of links (networks) to reach the destination, and infinity is defined as 16. The next node column defines the address of the router to which the packet is to be sent to reach its destination.

2) Link State Routing - Here, the nodes creates its routing table based on Dijkstra's algorithm. Each node in the autonomous system has the entire topology of that autonomous system, which includes list of nodes and links, how they are connected, type, cost and condition of links. Link state routing is based on the assumption that even though global knowledge about the topology is not clear, each node has partial knowledge, and the whole topology is combined based on the partial knowledge of each node. Each node creates routing tables considering global view of the topology.

Building Routing Tables requires four sets of actions to ensure that each node has the routing table showing the least-cost node to every other node.

(a) Creation of states of the links for each node, called link state packet *(LSP)*.

- LSPs are generated either when there is a change in the topology of the domain or on a periodic basis.
- LSP contains a lot of information, including the node identity, list of links, sequence number (to distinguish LSPs and facilitate flooding), and age.

(b) Dissemination of LSPs to every other router, called flooding, in an efficient and reliable way. The process is based on the following:

- Creating node sends a copy of LSP out of each interface.
- Node that receives the LSP compares it to the copy it may already have. If newer LSP is received, the node discards the old LSP and sends a copy of it out of each interface except the one from which the packet arrived.

(c) Formation of a shortest path tree for each node using Dijkstra's algorithm.

(d) Calculation of a routing table based on the shortest path tree.

Open Shortest Path First (OSPF) protocol implements link state routing. To handle routing efficiently, OSPF divides an autonomous system into areas. A special area, called *backbone*, such that all the areas inside an autonomous system must be connected to the backbone. At the border of an area, special routers called *area border routers* summarize the information about the area and send it to other areas. Four types of links are defined in OSPF: *(i)* Point-to-point, which connects two routers without any other host or router in between. *(ii)* Transient link, which is a network with several routers attached to it such that data can enter through one router and leave through any router. *(iii)* Stub link, where the network is connected to only one router, and data packets enter the network through this single router and leaves through this same router. *(iv)* Virtual link, that the administrator may create if link between two routers is broken which is likely to be a longer path.

4.3.2 Inter-Domain Routing Protocol

Path Vector Routing - Here, it is assumed that there is one node (can be more), called the speaker node, in each autonomous system that acts on behalf of the entire autonomous system. The speaker node in the AS creates a routing table and advertises it to speaker nodes in the neighbouring ASs. The speaker node advertises the path, not the metric of the nodes, in its AS.

(a) *Initialization*, that is, in the beginning, each speaker node can know only the reachability of nodes inside its autonomous system.

(b) *Sharing* - A speaker in an AS shares its tables with immediate neighbours.

(c) *Updating* - When a speaker node receives a two-column table from a neighbour, it updates its own table by adding the nodes that are not in its routing table and adding its own AS and the AS that sent the table.

- Loop Prevention - When a router receives a message, it checks to see if its autonomous system is in the path list to the destination. If it is, looping is involved and the message is ignored.
- Policy Routing - When a router receives a message, it can check the path. If one of the autonomous systems listed in the path is against its policy, it can ignore that path and that destination. It does not update its routing table with this path and does not send this message to its neighbours.
- Optimum Path - Path that is best for the organisation. Criteria such as security, safety, and reliability can be applied. Metrics can not included in the consideration because one system may be using RIP which uses hop count as metric, while the other might be using OSPF which is based on minimum delay as metric.

Border Gateway Protocol (BGP) protocol implements path vector routing. Autonomous systems can be *stub* (has only one connection to another AS), *Multihomed* (has more than one connection to other ASs, but still acts as the source or sink for traffic and does not allow transient traffic), and *Transit* (multihomed AS that allows transient traffic). Path Attributes are used to make a more informed decision about the path, which can be either well-known or optional. *External-BGP* is used to exchange information between two speaker nodes belonging to two different ASs. *Internal-BGP* is used to exchange information between two routers inside an AS.

Chapter 5

Transport Layer

The transport layer is responsible for the delivery of a message from one process to another. It ensures that whole message arrives intact and in order, overseeing both error control and flow control at the source-to-destination level.

5.1 Addressing

At the data link layer, a MAC address is needed to choose one node among several nodes if the connection is not point-to-point. A frame in the data link layer needs a destination MAC address for delivery and a source address for the next node's reply.
At the network layer, an IP address is needed to choose one host among millions. A datagram in the network layer needs a destination IP address for delivery and a source IP address for reply.
At the transport layer, a transport layer address is needed, called *port number* to choose among multiple processes running on the destination host. In the Internet model, port numbers are 16-bit integers between 0 and 65,535. The client program defines itself with a port number, chosen randomly by the transport layer software running on the client host, called the *ephemeral port number*.

Port Number Ranges have been defined by IANA (Internet Assigned Number Authority):

(a) Well-known ports: Ports ranging from 0 to 1023 are assigned and controlled by IANA.

(b) Registered ports: Ports ranging from 1024 to 49,151 are not assigned or controlled by IANA.

(c) Dynamic ports: Ports ranging form 49,152 to 65,535 are neither controlled nor registered. They can be used by any process. These are the ephemeral ports.

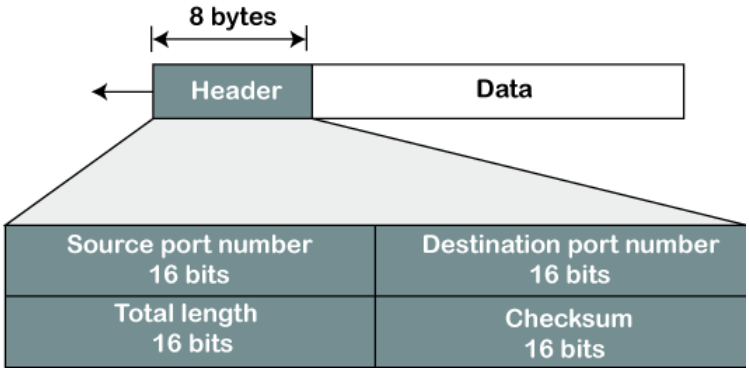
Socket Address - Process-to-process delivery requires two identifiers: IP address and the port number, at each end to make a connection. The combination of IP address and port number is called the socket address.

5.2 Transport Layer Protocols

5.2.1 User Datagram Protocol (UDP)

UDP is an unreliable connectionless protocol used for its simplicity and efficiency in applications where limited error control is required.

User Datagrams (or UDP packets), have a fixed-size header of 8 bytes. The fields are *(i)* Source port number, which is 16 bits long, *(ii)* Destination port number, which is also 16 bits long, *(iii)* Length, which is a 16-bit field that defines the total length of the user datagram, header and data, and *(iv)* Checksum.



Uses of UDP

- Suitable for processes that require simple request-response mechanism with little concern for flow and error control.
- Suitable for processes that have their internal flow and error control mechanisms. For example, Trivial File Transfer Protocol (TFTP).
- Suitable protocol for multicasting.
- Used for management processes such as SNMP.
- Used for some route updating protocols, such as RIP.

5.2.2 Transmission Control Protocol (TCP)

TCP is a reliable connection-oriented protocol.

TCP Services

- *Process-to-Process Communication*
- *Stream Delivery Service* that allows the sending process to deliver data as stream of bytes and the receiving process to obtain data as a stream of bytes.
- *Full-Duplex Communication*
- *Connection-Oriented Service*
- *Reliable Service*

TCP Segment which is a packet in TCP. The segment consists of 20 to 60 bytes header, followed by data from application program. The header fields are as follows:

(a) Source port address, is a 16-bit field.

(b) Destination port address, is a 16-bit field.

(c) Sequence number, is a 32-bit field that defines the number assigned to the first byte of data contained in this segment. To ensure connectivity, each byte to be sent is numbered.

(d) Acknowledgement number, is a 32-bit field that defines the byte number that the receiver of the segment is expecting to receive from the other party.

(e) Header length, is a 4-bit field that indicates the number of 4-byte words in the TCP header.

(f) Reserved, is a 6-bit field that is reserved for future use.

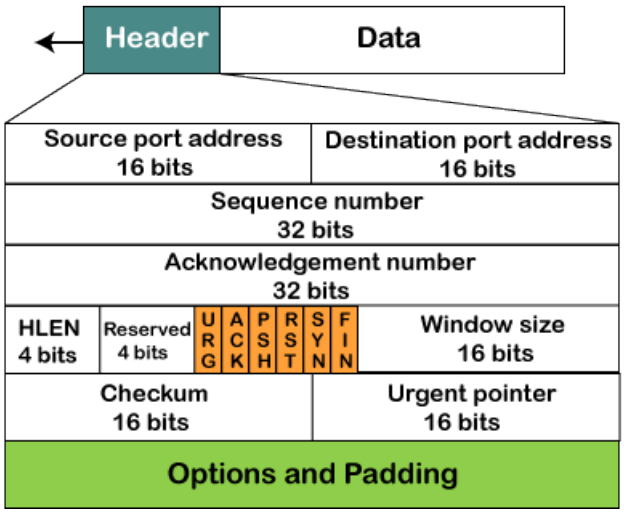
(g) Control, is a field that defines 6 different control bits or flags.

(h) Window size, is a 16-bit field that defines the size of the window, in bytes. This value if normally determined by the receiver.

(i) Checksum, is a 16-bit field.

(j) Urgent pointer, is a 16-bit field, which is valid only if urgent flag is set, is used when the segment contains urgent data.

(k) Options, which can be upto 40 bytes.



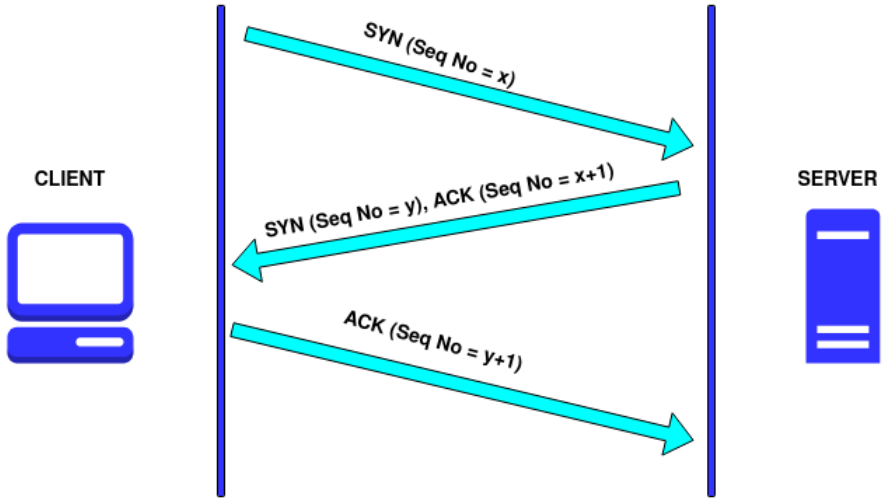
TCP Connection

(a) **Connection Establishment -** The process starts with the server telling its TCP that it is ready to accept a connection. This is called a request for *passive open*. The client program issues a request for an *active open*. A client that wishes to connect to an open server tells its TCP that it needs to be connected to that particular server. TCP can now start *Three-Way Handshaking* -

(1) The client sends the first segment, a SYN segment, in which only the SYN flag is set. This segment is for synchronization of segment numbers. A SYN segment cannot carry data, but it consumes one sequence number.

(2) The server sends the second segment, a SYN+ACK segment, with 2 flag bits set. It's a SYN segment for communication in the other direction and serves as acknowledgement for SYN segment. It cannot carry data, but does consume one sequence number.

(3) The client sends the third segment, an ACK segment. It acknowledges the receipt of the second segment with the ACK flag and acknowledgement number field. An ACK segment, if carrying no data, does not consume any sequence number.



(b) **Data Transfer**

(c) **Connection Termination -**

(1) Normally, the client TCP, after receiving a close command from the client process, sends the first segment, a FIN segment in which FIN flag is set. The FIN segment consumes one sequence number if it does not carry data.

(2) Server TCP, after receiving the FIN segment, informs the process and sends the second segment, a FIN+ACK segment, to confirm the receipt of FIN segment from the client and announce closing of connection. This segment can also contain last chunk of data from the server. FIN+ACK segment consumes one sequence number if it does not carry data.

(3) Client TCP sends the last segment, an ACK segment, to confirm the receipt of segment from the server. The segment contains acknowledgement number, carries no data and consumes no sequence number.

TCP Flow Control uses a sliding window, which is something like Go-Back-N because it not use NAKs, and Selective Repeat because the receiver holds the out of order segments until the missing ones arrive. The two major differences between sliding window of TCP and that of data link layer are (i) window of TCP is byte-oriented whereas window of data link layer is frame oriented, (ii) TCP window is of variable size whereas data link layer window is of fixed size. The size of the window at one end is determined by the lesser of the two values: *receiver window (rwnd)* and *congestion window (cwnd)*. The window can be opened or closed by the receiver but should not be shrunk.

TCP Error Control is achieved by Checksum, Acknowledgement, and Retransmission.

5.3 Congestion Control in TCP

Congestion in a network may occur if the load on the network is greater than the number of packets it can handle i.e. number of packets sent to the network is greater than the capacity of the network. Congestion occurs because routers and switches have queue buffers that hold packets before and after processing.

- **Open-Loop Congestion Control -** Policies are applied to prevent congestion before it happens. In these mechanisms, congestion control is handled by either the source or the destination. It includes retransmission policy, window policy, acknowledgement policy, discarding policy, and admission policy.

- **Closed-Loop Congestion Control -** Policies are applied to alleviate congestion after it happens. Mechanisms that are used are Backpressure, Choke Packet, Implicit Signaling, Explicit Signaling, Backward Signaling, and Forward Signaling.

Congestion Window In addition to the receiver, the network is a second entity that determines the size of the sender's window. $Actual\ Window\ size = minimum(rwnd, cwnd)$

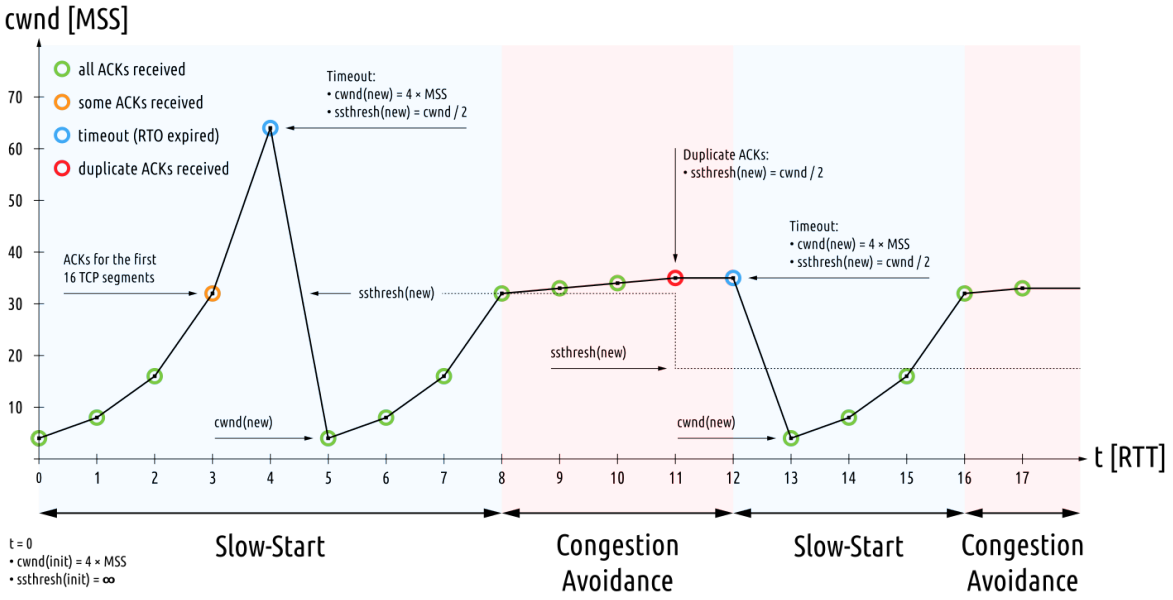
Congestion Policy TCP's policy for handling congestion is based on three phases:

(a) **Slow-Start Phase: Exponential Increase** is based on the idea that the size of congestion window (cwnd) starts with one maximum segment size (MSS). The size of window increases by one MSS each time acknowledgement is received. That means, sender initially starts with a very slow rate of transmission, but increases exponentially to reach a threshold. Since the phase cannot continue indefinitely, the sender keeps track of *ssthresh*. When the window size (in bytes) reaches the threshold, slow start phase ends and next phase begins.

(b) **Congestion Avoidance: Additive Increase** When the slow start threshold is reached, it undergoes additive increase instead of exponential one. Each time the whole window of segments is acknowledged, the size of congestion window is increased by 1. This is done until congestion is detected.

(c) **Congestion Detection: Multiplicative Decrease** If congestion occurs, the congestion window size must be decreased. Here, the size of the threshold is dropped to one-half of the window size i.e. multiplicative decrease.

- If RTO (retransmission time-out) occurs, there's a high possibility of congestion. Here, TCP reacts strongly: sets threshold value to one-half of current window size, sets *cwnd* to the size of one segment, and starts a new slow start phase.
- If three ACKs are received, there's a weaker possibility of congestion. A segment may have been dropped, but some segments after that may have arrived safely since three ACKs are received. This is called fast transmission and fast recovery. Here, TCP has a weaker reaction: sets threshold value to one-half of current window size, sets *cwnd* to the value of threshold, and starts the congestion avoidance phase.



5.4 Quality of Service

Four types of characterstics are attributed to flow: Reliability, Delay, Jitter(variation in delay for packets belonging to same flow), and Bandwidth.

5.4.1 Techniques to Improve QoS

There are four common methods:

- **Scheduling -** There are three methods:

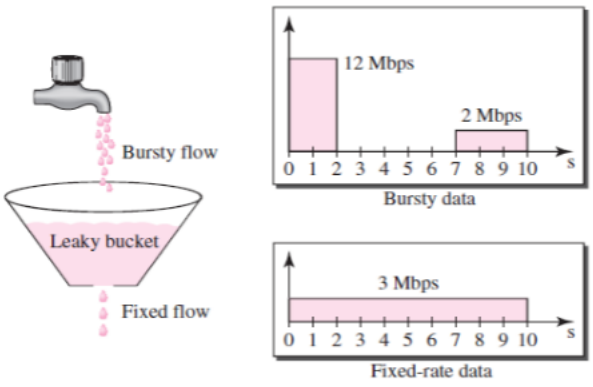
  (a) FIFO Queuing, where packets wait in the buffer until the node is ready to process them. If average packet arrival rate is higher than average processing rate, the queue will fill up and new packets will be discarded.

  (b) Priority Queuing, where packets are first assigned to a priority class. Each priority class has its own queue. The packets in the highest priority queue are processed first. Drawback of priority queuing is neglecting of low priority queue which might lead to starvation.

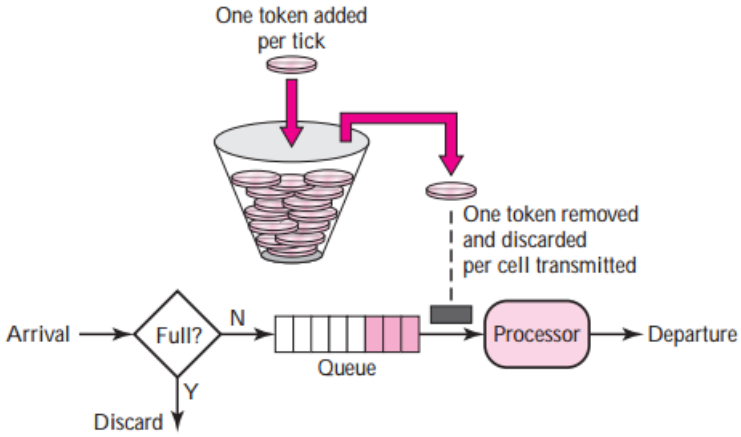
  (c) Weighted Fair Queuing, where packets are assigned to different queues and classes, and the queues are weighted based on the priority. The system processes packets in each queue in a round-robin fashion with the number of packets selected from each queue is based on the corresponding weight. For example, if the weights are 3, 2, and 1, three packets are processed from first queue, two from second queue, and one from the third queue.

- **Traffic Shaping -** Used to control the amount and the rate of traffic sent to the network. There are two techniques:

  (a) *Leaky Bucket* - If a bucket has a small hole at the bottom, the water leaks from the bucket at a constant rate, as long as there is water in the bucket. The input rate can vary, but the output rate remains constant. Similarly, leaky bucket can smooth out bursty traffic. Bursty chunks are stored in the bucket and sent out at an average rate. It may drop the packets if the bucket is full.



  (b) *Token Bucket* - The algorithm allows idle hosts to accumulate credit for future in the form of tokens. For each tick of the clock, the system sends n tokens to the bucket. The system removes one token for each cell (or byte) of data sent and the host can send bursty data as long as the bucket is not empty. Thus, the token bucket allows bursty traffic at a regulated maximum rate.



Note Leaky bucket and token bucket can be combined. Leaky bucket is applied after token bucket as the rate of leaky bucket needs to be higher than the rate of tokens dropped in the bucket.

- **Resource Reservation -** Flow of data needs resources such as a buffer, bandwidth, CPU time, and so on. QoS is improved heavily if the resources are reserved beforehand.

- **Admission Control -** Mechanism used by router, or a switch, to accept or reject a flow based on predefined parameters called flow specifications. Before a router accepts a flow for processing, it checks the flow specifications to see if its capacity and its previous commitments to other flows can handle the new flow.

Chapter 6

Application Layer

The application layer is responsible for providing services to the user, including electronic mail, file access and transfer, access to system resources, surfing the world wide web, and network management.

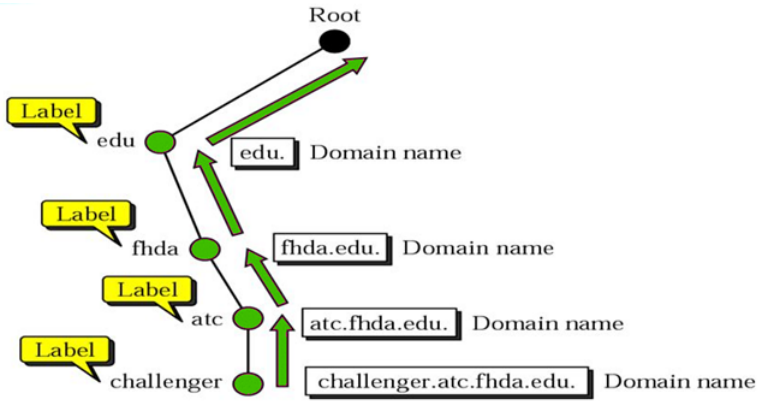
6.1 Domain Name System

Name Space maps each address to a unique name since addresses are unique. It can be organised in two ways: Flat Name Space and Hierarchical Name Space.

6.1.1 Domain Name Space

To have a hierarchical name space, the names are defined in an inverted tree structure with the root at the top. The tree can have only 128 levels.

- **Label**, which is a string with maximum of 63 characters. The root label is a null string. DNS requires children of a node to have different labels, which guarantees uniqueness of the domain names.
- **Domain Name**, where a full domain name is a sequence of labels separated by dots. The domain names are always read from the node upto the root. If a domain name ends in null string, it's called *Fully Qualified Domain Name*, else it's called a *Partially Qualified Domain Name*.
- **Domain** is a subtree of the domain name space. The name of the domain is the domain name of the node at the top of the subtree.



6.1.2 Distribution of Name Space

It is inefficient and unreliable to store the name space on a single system. DNS servers are used to divide the name space. DNS allows domains to be divided and further divided into subdomains. Each server can be responsible (authoritative) for either a large or a small domain. In other words, hierarchy of servers is made the same way as the hierarchy of names. DNS in the internet can be classified into General Domains, Country Domains, and Inverse Domains.

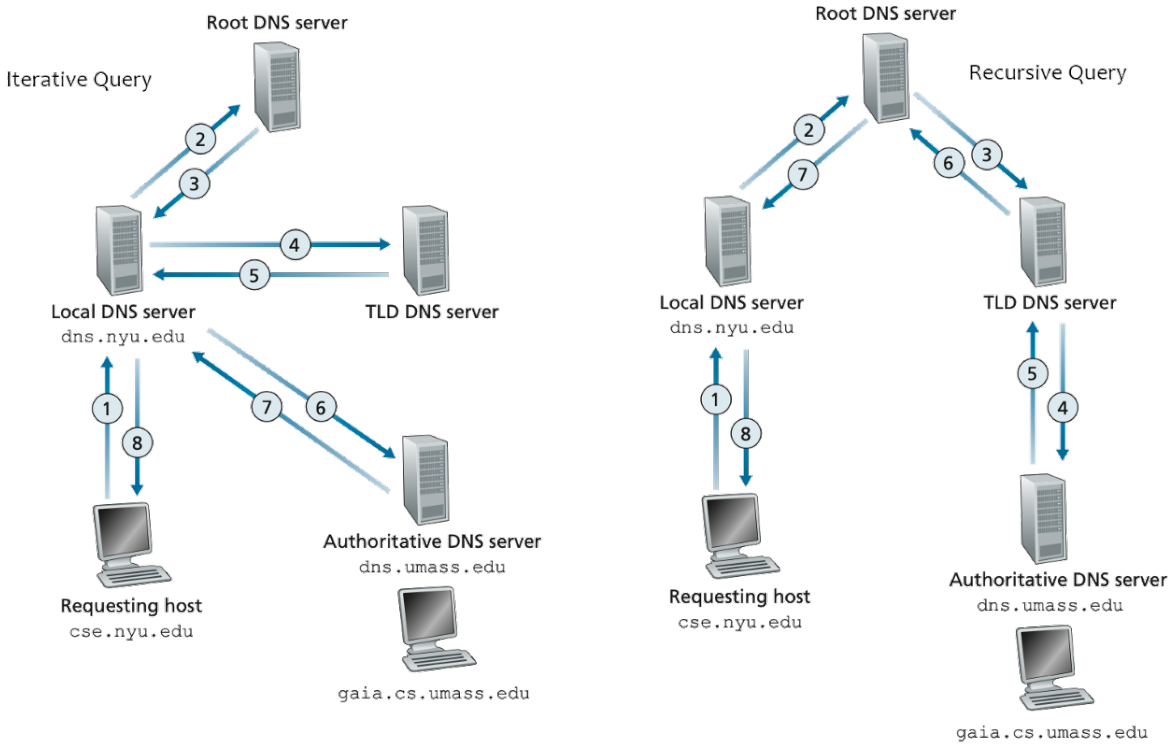
6.1.3 Name-Space Resolution

Resolver - DNS is designed as client/server application. A host that needs to map an address to a name to an address calls a DNS client called a resolver. The resolver accesses the closest DNS server with a mapping request. The server provides the information if it contains the required one, or else requests other servers. The result is delivered to the client after it's checked for a real resolution or an error.

Mapping Addresses to Names , where a client can send an IP address to a server to be mapped to a domain name, called a PTR query, where DNS uses the inverse domain.

(a) Recursive Resolution

(b) Iterative Resolution



6.2 Electronic Mail

6.2.1 User Agent

It's the first component of an electronic mail system. It provides service to the user to make the process of sending and receiving messages easier, which include composing messages, reading messages, replying

to messages, forwarding messages, and handling mailboxes.

Email Address - To deliver mail, mail handling system must use an addressing system with unique addresses. The address consists of two parts:

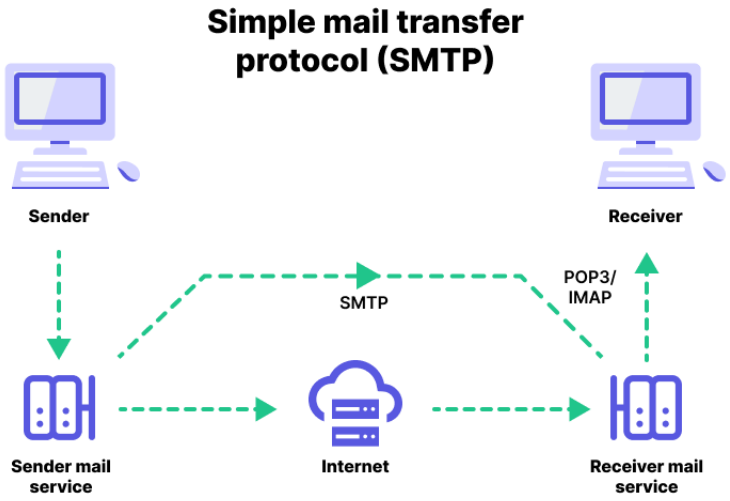
(a) Local Part defines the name of a special file, called user mailbox, where all the mail received for a user is stored for retrieval by the message access agent.

(b) Domain Name - An organisation usually selects one or more hosts to send and receive email, called mail servers or exchangers. The domain name assigned to each mail exchanger either comes from the DNS database or is a logical name.

MIME or Multipurpose Internet Mail Extensions, is a supplementary protocol that allows non-ASCII data to be sent through e-mail. MIME transforms non-ASCII data at the sender site to NVT (Network Virtual Terminal) ASCII data and delivers them to the client MTA to be sent through the internet. The message at receiving side is transformed back to the original data.

6.2.2 Message Transfer Agent: SMTP

To send mail, a system must have the client MTA, and to receive mail, a system must have a server MTA. The formal protocol that defines MTA client and server in the Internet is called Simple Mail Transfer Protocol. SMTP is a *push* protocol i.e. pushes messages from the client to the server.



6.2.3 Message Access Agent: POP and IMAP

A *pull* protocol is needed i.e. the client must pull messages from the server. The direction of bulk data is from server to client and uses a message access agent. Currently, two message access protocols are available:

(a) **POP3**, or Post Office Protocol. Client POP3 is installed on the recipient system, and server POP3 software is installed on the mail server. Mail access starts with the client when user needs to download email from the mailbox on mail server. The client opens a connection to the server on TCP port 110. It sends its username and password to access the mailbox. The user can then list and retrieve mail messages one by one. POP3 has two modes: (i) Delete mode where the mail is deleted from mailbox after each retrieval, and (ii) Keep mode where the mail remains in the mailbox after retrieval. As for drawbacks, POP3 does not allow user to organize mail on the server, and does not allow user to partially check the contents of the mail before downloading.

(b) **IMAP4**, or Internet Mail Access Protocol provides the following extra functions than POP3:

- User can check email header prior to downloading.
- User can search the contents of email for specific string of characters prior to downloading.
- User can partially download email. It's useful if bandwidth is limited and email contains multimedia with high bandwidth requirements.
- User can create, delete, or rename mailboxes on the mail server.
- User can create a hierarchy of mailboxes in a folder for email storage.

Web-Based Mail Websites provide mail service to users who access the site. Mail transfer from sender to the mail server is done through HTTP. The transfer of message from sending mail server to receiving mail server is still through SMTP. When receiver needs to retrieve emails, the website sends the email from web server to receiver's browser in HTML format after authenticating the user.

6.3 File Transfer Protocol (FTP)

FTP is the standard mechanism provided by TCP/IP for copying a file from one host to another. There are issues in file transfer like systems using different file name conventions, different ways to represent text and data, and different directory structures. FTP establishes two connections between the hosts, one is used for data transfer, and other for control information. The control connection uses very simple rules of communication where only a line of command or response needs to be transferred. The data connection needs more complex rules for variety of data types transferred. However, the difference in complexity is at FTP level, not TCP. Both the connections are identical for TCP. FTP uses services of TCP. The well-known port 21 is used for control connection and the well-known port 20 is for data connection.

6.3.1 Basic Model of FTP

The client has three components: user interface, client control process, and client data transfer process. The server has two components: server control process, and server data transfer process. The control connection is made between control processes. The data connection is made between data transfer processes. The control connection remains connected during the entire interactive FTP session. The data connection is open and closed for each file transfer.

Communication over Data Connection occurs over the data connection under the control of the commands sent over the control connection. However, file transfer over FTP means one of the three things:

- File is to be copied from the server to the client. This is called *retrieving aft/e* and is done under supervision of RETR command.
- File is to copied from the client to the server. This is called *storing aft/e* and is done under the supervision of STOR command.
- A list of directory or file names is to be sent from server to the client. This is done under the supervision of LIST command.

The heterogeneity of data problem is resolved by defining three attributes of communication: file type, data structure, and transmission mode.