Networking Primer

Sources:
https://www.ece.uvic.ca/~itraore/elec567-13/notes/dist-03-4.pdf

https://www.geeksforgeeks.org/basics-computer-networking

https://en.wikipedia.org/wiki/Computer_network

This short set of notes is designed to provide a basic overview of computer networking. We start with some definitions and brief history, then move on to structures, services, protocols, etc. We will finish with some hands-on work using python.

Introduction

What is a computer network? A computer network is a collection of computers, switches, routers and other communication equipment that enables computer users to share information in an easy and convenient manner. Networks support services such as email, file transfer, streaming, online chat, etc. Networks also make it easy to do things like share printers, scanners, and other needed equipment.

A network can be defined as a group of computers and other devices connected in some way so as to be able to exchange data.

- Each of the devices on the network can be thought of as a node; each node has a unique address.
- Addresses are numeric quantities that are easy for computers to work with, but not for humans to remember. Example: 204.160.241.98
- Some networks also provide names that humans can more easily remember than numbers.

Example: www.javasoft.com, corresponds to the above numeric address.

Abbreviated History

The first networks were built for the U.S. military in the late 1950's. It was based on the Bell 101 modem, the first commercial modem. It allowed digital data to be transmitted over telephone lines at a rate of 110 bits per second. Later, in 1963 J.C.R. Licklider wrote a memorandum discussing the concept of a computer network intended to allow general communications between and among computer users. Soon afterwards Western Electric integrated computer control into the telephone network. The idea of data packets, routing, communication etc. were developed by Paul Buran and Donald Davies throughout the 1960's; their contributions were fundamental to the area of computer networking.

In 1969 the first examples of wide area networks were created by linking computers at UCLA, Stanford, University of California at Santa Barbara, and the University of Utah with a 50kbit line. This system became known as ARAPANET. In 1971 the first wireless wide area network came online to link the University of Hawaii to computers located on some of the other Hawaiian

Islands. This network became known as ALOHAnet or the ALOHA system or just ALOHA. Terminals were connected to the system using serial telephone lines and RS-232 protocol at a speed of 9600 bits/s.

In 1973 Robert Metcalfe at XEROX wrote a memo that described Ethernet, a networking system based on the ideas pioneered by ALOHA. Later in 1974, Carl Sunshine wrote the Transmission Control Protocol (TCP) specification and coined the word *Internet*, shorthand for internetworking. In 1976, a token passing network, was used to share storage devices. In 1977 GTE developed the first long distance fiber network.

In 1980, Ethernet was upgraded from 2.94 Mbit to 10bit. Much later, in 1995 it was increased from 10Mbit to 100Mbit; in 1998 it went to 1Gbit, and later in 2018 400 Gbit.

Concepts and Definitions

Internet Protocol address (IP address)

An IP address is a unique numerical identifier that is assigned to every device on a network. Each node in the IP world is identified by a unique 32-bit number called an IP address. IP addresses are used to identify devices and enable communication between them. An IP address identifies a machine that is connected to the internet. IP addresses appear as a series of four 8-bit numbers (bytes) separated by dots (**dotted quads**). With 8 bits representation, each number ranges from 0 to 255 but the notation is converted to decimal. So an IP address originally consisted of 4 groups of numbers separated by periods.

Example: 136.102.233.49

This is (IPv4), but as the internet use expands with more and more devices, a larger address space is needed with a lot more addresses, so IPv6 using 8 bytes was introduced.

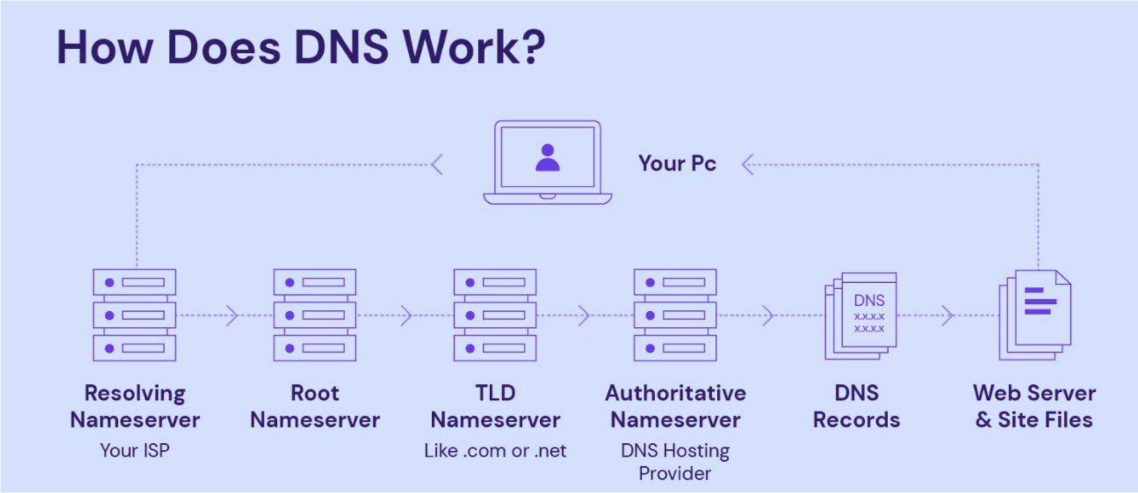
Domain Name System (DNS)

IP addresses are unique for each device. The DNS system provides a way to match IP addresses with text/mnemonic addresses so that users can more easily work with addresses. It is much easier to remember something like www.jenny.com than 867.530.9.000 when referring someone’s web page address. DNS servers act as the internet's phone book and translate the text versions of the address to the numeric IP addresses. The DNS hierarchy consists of servers that collectively form a massive global database. The DNS hierarchy relies on 13 distributed Root Zone servers globally, contrary to a single point depiction. Appointed by ICANN, entities like Verisign and NASA operate these servers. They play a crucial role in resolving domain names to IP addresses. The geographical dispersion and redundancy of these servers enhance the resilience and efficiency of the DNS system. A DNS request from your computer to resolve a specific URL to an IP address (of the server which serves the files of that URL), will typically query the local cache of your computer, and if not found (because it was not recently used) the request would move on to your internet service provider's DNS server, and if not found it would then move to regional DNS server following the hierarchy up until a resolution is found or until the top of the hierarchy is reached. At the top of the DNS hierarchy, if a resolution exists it will be known at that level and if not then the URL is not valid (i.e. not in use or invalid). Typically recent resolutions get cached at the lower levels so that frequent requests to a specific URL get resolved faster.

Here is how this service works:

1. **DNS Query:** When a user types a domain name into their browser, their computer sends a DNS query to a DNS server to find the corresponding IP address.

2. **DNS Resolution:** The DNS server (or a series of servers) then uses a hierarchical system to locate the IP address. This involves querying root servers, top-level domain servers, and potentially authoritative servers. Essentially, the DNS hierarchy of servers is queried from the local servers to the top level servers until one returns the IP address that serves the domain name of the query.



3. **IP Address Retrieval:** Once the IP address is found, it's returned to the user's computer.

4. **Website Loading:** The computer can then use the IP address to connect to the website.

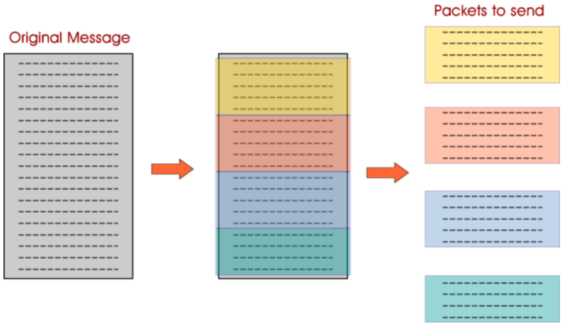
Ports and Data Transmisison

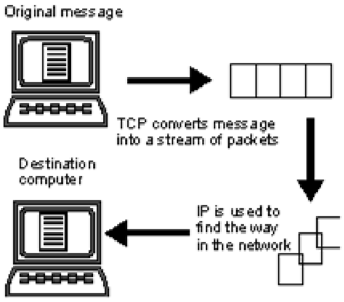
An IP port is a number that identifies a specific application running on a host machine. Common port numbers are below:

Application	Port numbers
HTTP	80
FTP, SFTP	20, 21
SMTP (email)	25
POP3 (email)	110

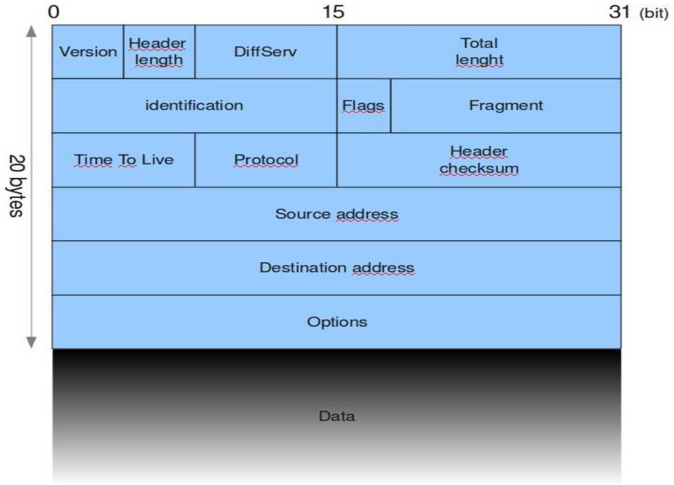
When programming network applications, often people request a "socket". Socket to socket communication is supported by many languages, C/C++ and Python being two of them. A socket is a combination of IP address and port number. So using a socket a program can send information to a port number on a specific computer at some location in the internet.

Data is transmitted throughout the internet using data packets. A data file is broken into pieces which are then individually packaged with routing information into **data packets** which are transmitted individually to the destination and then recompose the file.




Original message
TCP converts message into a stream of packets
Destination computer
IP is used to find the way in the network

IP version 4 packet



A **data packet** is a data structure that has a few parts to it. The first part (the blue area in the picture above) is a **header** section that holds address information of the source and destination computers and packet numbers so that the data can be reassembled in the correct order when it reaches the destination computer. It also contains information regarding the type of data (text or binary), the length of the data carried. It may contain other fields specific to network operations such as the Time-To-Live (TTL) field which specifies the maximum number of network hops which the packet is allowed to perform before getting to its destination, otherwise it is considered lost or obsolete and is to be destroyed. The **data section** holds the "payload", i.e. the data itself. Finally there is a **trailer section** which may contain error detection information such as data check sums, parity bits, or the like.

As an example, if a file is going to be transferred from a source computer to a destination computer that are separated by some distance, it will be broken down into several separate packets, each with a header section that contains address information and a packet number, and then a data section that contains part of the file. The receiving computer will collect the packets and reassemble the file based on the packet numbers. If a packet is missing the receiving computer can request it to be resent (it will notice a gap in the packets it received). The transmitting computer will resend the lost packet, and the file can be reassembled without loss of data.

Network Types and Their Technologies

The most common network types are Wide Area Networks (WAN) and Local Area Networks (LAN). WANs cover large areas like cities, countries, etc. LANs cover small areas, like a building or a group of buildings. Because the transmission distances are different for the two types of networks, they rely on different technologies.

WANs rely on a few different types of technologies. Asynchronous Transfer Mode (ATM) and/or Integrated Services Digital Network (ISDN) are connection based systems, meaning that a connection must be established between the two machines. These have been superseded by Internet Protocol (IP) methods which is a connectionless method. While ATM has many good points, it is less scalable than IP. ISDN is an older technology used for voice over data on telephone lines.

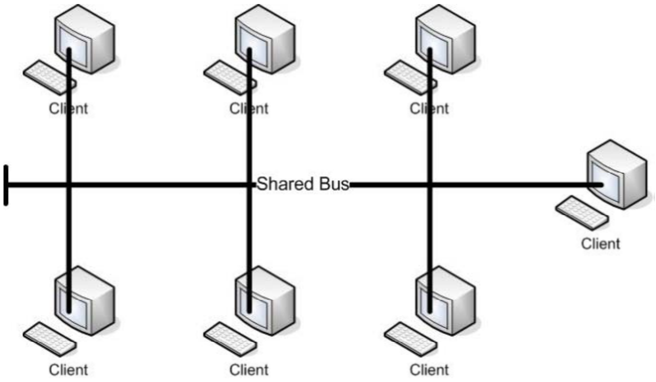
LANs rely on Ethernet, Token Ring, FDDI, etc. Ethernet is a technology that is based on bursting data on a transfer medium. That is, if a computer needs to transfer data, if the bus is idle it starts transmitting, if a data collision occurs because some other machine starts transmitting at the same time, both machines wait a random amount of time and begin transmitting again. This strategy works well for situations in which the data line is not busy. If data traffic is heavy, collisions will become excessive and the data transfer throughput will suffer.

Token Rings work on the principle of whichever machine has the token has the right of way to transmit data. The token is passed in an orderly manner from machine to machine. If a machine has no data to transmit it immediately passes the token to the next machine. In this way collisions are avoided. For systems in which several machines have to transmit data on a regular basis, this system is much more efficient than ethernet. As long as the transmittal time is kept reasonable, this system is effective. FDDI (Fiber Distributed Data Interface) is similar, but it utilizes two rings running in opposite directions.

Network Topologies

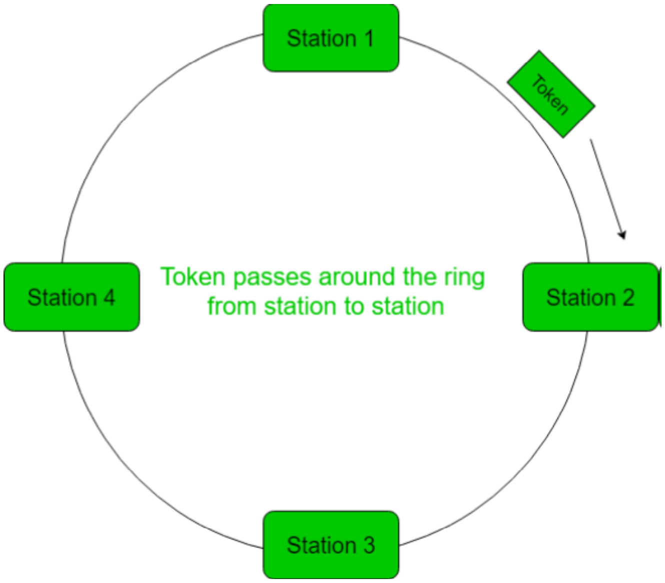
Ethernet LAN (image from https://www.researchgate.net/figure/Example-of-the-original-Ethernet-Bus-structure_fig4_31598273)

Below is a diagram of the original ethernet system with a network topology known as **Bus Topology**. Each of the clients share a data transmission bus. When a client wants to use the bus, they wait till the bus is idle and they start transmitting. If more than one machine attempts to transmit at the same time a collision occurs. Each machine involved in the collision will wait a random time and then try to transmit.



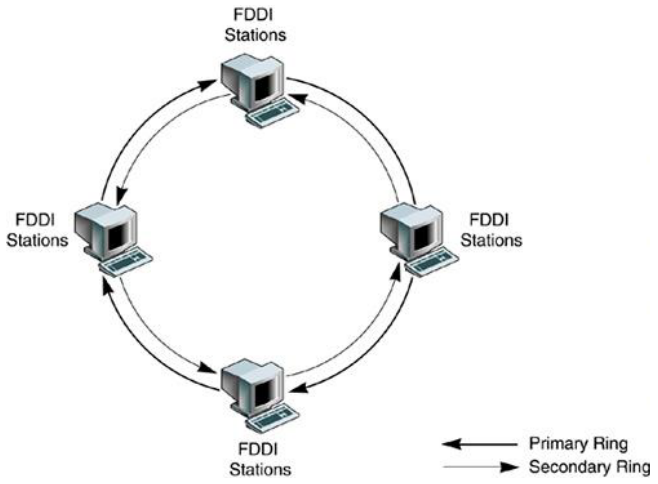
- All devices are connected to a single central cable (the "bus").
- Data travels in both directions, but only one device can transmit at a time.
- **Pros:** Simple, inexpensive.
- **Cons:** If the main cable fails, the whole network goes down.

The **Token Ring** topology is an alternative network structure where all the network devices are arranged in a ring and transmit/receive data tokens. A token carries data and a destination identifier; stations pass on a token if is destined for another station or consume it if it is destined for that station.



Token Ring LAN (https://www.geeksforgeeks.org/efficiency-of-token-ring/)

FDDI LAN (source - https://www.oreilly.com/library/view/networking-concepts-and/0131482076/0131482076_ch05lev1sec2.html)



The **Star Topology** uses a central node to which all other nodes are connected and all traffic goes through that central node.

- All devices are connected to a central hub or switch.
- The hub acts as a repeater or controller.
- **Pros:** Easy to manage and expand; one device failure doesn't affect others.
- **Cons:** Hub failure disables the whole network.

The **Mesh Topology** uses a central node to which all other nodes are connected and all traffic goes through that central node

- Every device is connected to every other device.
- Can be **full mesh** (all devices interconnected) or **partial mesh**.
- **Pros:** Highly reliable and fault-tolerant.
- **Cons:** Expensive and complex to set up.

In the **Tree Topology** (a.k.a. Hierarchical) the nodes are arranged in a tree structure where information is passed from parents to children or from a child to its parent.

- A variation of star and bus topologies in a tree-like structure.
- Central "root" node connected to one or more "branch" nodes.
- **Pros:** Scalable and easy to manage in segments.
- **Cons:** If the root node fails, large sections may go down.

The **Hybrid Topology** is a combination of the above.

- A combination of two or more different topologies (e.g., star-bus, star-ring).
- **Pros:** Flexible and scalable.
- **Cons:** Complex design and potentially higher cost.

Interconnection of Network Devices and Networks.

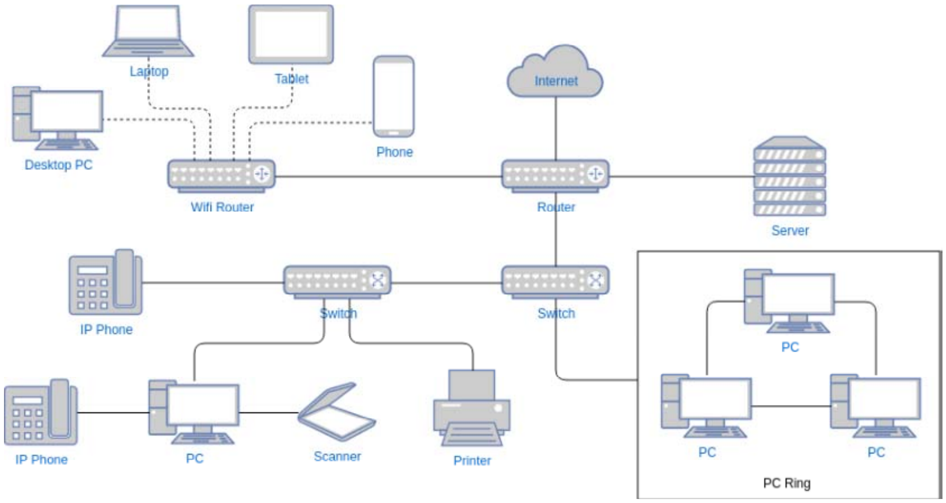
Networks are assembled from a collection of different hardware and cabling technologies.

LANs are commonly comprised of ethernet cables such as Cat5, and switches. These are essentially groups of twisted pairs of copper wires, i.e. telephone wire connected by a switch that forwards data packets based on their MAC addresses (each device has a 48 bit MAC address, assigned when the device is manufactured. It acts as that devices identifier/address).

WANs are a collection of devices and networks connected by high speed digital lines. Devices such as bridges, routers, gateways, etc., manage the translation of data from 1 protocol to another.

Connection Devices

- Bridge – Connects two LANs that use the same protocol.
- Router – Connects different types of networks using different protocols.
- B-router or Bridge/Router – Combines the functionality of bridges and routers in one device.
- Gateway – A device that connects two different systems using direct and systematic translation between protocols.



The diagram above provides an example of the connectivity possible in computer networks. The switches are providing local connections to like devices, while the routers are providing the local groups connectivity to devices/networks in different locations. The routers are making sure that information can be exchanged between the various protocols.

Protocols

The job of a protocol is to define the rules that govern the communications between computers and devices connected to the network. The protocols rules govern addressing, routing, error detection and recovery, etc.

Examples of Protocols

- HTTP – Hyper Text Transfer Protocol, regulates communication between web browsers and servers

- TCP/IP – Transfer Control Protocol/Internet Protocol is the foundational protocol suite of the internet, enabling reliable communication. TCP Ensures data is delivered reliably and in order and IP routes data packets to their destination based on IP addresses.
- SMTP – Simple Mail Transfer Protocol is used to send email. The SM TP protocol works with other protocols like POP3 and IMAP for email retrieval.
- FTP – File Transfer Protocol is used for transferring files between computers. Includes commands for uploading, downloading, and managing files on a remote server.
- DNS – Domain Name System translates human-friendly domain names into IP addresses. Ensures seamless navigation on the internet.

OSI and TCP/IP and IP Models for Network Communication

Source: https://www.splunk.com/en_us/blog/learn/osi-model.html

The foundational model for network communication is the OSI model. These layer's purpose is to make the communication from computer to computer a seamless and error free exercise, even though the system itself is full of machines and data lines in which there many issues, including synchronization of machines/timing issues, transmission errors, etc. It has 7 layers, each with a specific function. The layers are:

Layer 7. **Application** – The application layer is the closest layer to the end user. It receives information from the end user and sends results back to the user. Despite its name, Layer 7 is not where client applications live. This layer provides the protocols that allow software/apps to transmit data, including:

a. HTTP and HTTPS
b. FTP
c. POP & SMTP
d. DNS
e. Telnet
f. DHCP
g. SNMP

Layer 6. **Presentation** – The presentation layer ensures the data is prepared in a usable form for the application layer (receiving side) or for the network layer (sending side). Layer 6 is responsible for:

a. Data translation
b. Encryption & decryption
c. Compression
d. Other data preparation items

Layer 5. **Session** – The session layer creates and maintains the sessions (connections) that two systems need in order to speak to each other. It also creates checkpoints to ensure and synchronize data transfer.

a. When sessions are created and opened
b. How long sessions remain open to successfully exchange data

c. When to close sessions
d. And more

Layer 4. **Transport** – The transport layer uses transmission protocols including Transmission Control Protocol (TCP) and User Datagram Protocol (UDP), to manage network traffic between systems to ensure correct data transfers. Layer 4 also handles flow control and error control, regulates transmission speed and requests retransmissions if needed.

a. On the sending side, it takes data from the session layer and breaks it into segments for the network layer.
b. On the receiving side, it reassembles the segments from the network layer and passes them on to the session layer.

Layer 3. **Network** – The network layer decides which physical path the data will take. It's responsible for breaking up transport layer segments into smaller network packets for transmission and for reassembling those packets on the receiving system. This session routes packets to their destination, mostly by using IP addressing. Layer 3 processing is generally bypassed when the sending and receiving systems are on the same network.

Layer 2. **Data link** – The data link layer defines the format of data on the network. Like the network layer, the data link layer enables data transfer between two directly connected nodes or systems on the same network. Layer 2 also corrects errors that may have occurred at the physical layer (layer 1). It uses media access control (MAC) processing for flow control and multiplexing between two systems. It also uses logical link control (LLC) to provide flow control and error control.

Layer 1. **Physical** – The physical layer converts and transmits raw bit stream data (1s and 0s) over the physical medium. Layer 1 concerns the physical and electrical connections the system uses. The physical layer also discusses network components such as hubs, repeaters, modems, network adaptors, etc. It includes:

a. Wireless frequency links, like Wi-Fi and wireless network connections
b. Network cabling
c. Light-speed transmission, such as fiber-optic cabling
d. The physical specifications for data transmission, including voltages and pin layouts

The TCP/IP Model is similar but only has 4 layers.

Layer 1. **Network layer** – Provides the same functionality as the physical, the data link and network layers in the OSI model.

a. Mapping between IP addresses and network physical addresses.
b. Encapsulation of IP datagrams, e.g packets, in format understandable by the network.

Layer 2. **Internet layer**

a. Lies at the heart of TCP/IP.
b. Based on the Internet Protocol (IP), which provides the frame for transmitting data from place A to place B.

Layer 3. **Transport layer**

a. Based on two main protocols: TCP (Transmission Control Protocol) and UDP (User Datagram protocol)

Layer 4. **Application layer**

a. Combines the functions of the OSI application, presentation, and session layers.

b. Protocols involved in this layer: HTTP, FTP, SMTP etc.

Both the OSI and TCP/IP protocols are used widely in the internet today.

Network to Network Connections

LANs and WANs are connected by various transmission lines and hardware to form the internet. The protocols that manage much of the data transmission and connectivity are IP and TCP.

Internet Protocol (IP)

IP is a connectionless oriented protocol. It does not make a connection request before sending data. Its main purpose is to:

- Transform data into packets for transmission. Reassemble the packets into data upon receiving.
- Route the packets through successive networks from source machine/network to destination machine/network.

Packets are not guaranteed to be delivered (datagram protocol) and there is no error detection.

Structure of a Packet

The fields at the beginning of the packet, called the frame header, define the IP protocol's functionality and limitations.

- 32 bits are allocated for encoding source and destination addresses (32 bits for each of these address fields).
- The remainder of the header (16 bits) encodes various information such as the total packet length in bytes.
- Hence an IP packet can be a maximum of 64Kb long.

Transmission Control Protocol (TCP)

TCP is a connection oriented protocol. It guarantees safe delivery of data. Its functionality includes error detection, making sure that packets are received in order, etc. Before data is sent TCP makes sure that the computers establish a connection using a combination of signals and acknowledges.

- TCP provides support for sending and receiving arbitrary amounts of data as one big stream of byte data (IP is limited to 64Kb).
- TCP does so by breaking up the data stream into separate IP packets.
- Packets are numbered, and reassembled on arrival, using sequence and sequence acknowledge numbers.
- TCP also improves the capability of IP by specifying port numbers. There are 65,536 different TCP ports (sockets) through which every TCP/IP machine can talk.

The TCP data packet supports this functionality by including source and destination ports, packet sequence number, and acknowledgement number.

Other important protocols include UDP (User Datagram Protocol) a connectionless protocol. UDP sends data packets (datagrams) independently, without any handshake or connection management. This allows for faster data transmission but comes at the cost of reliability, as there's no guarantee of delivery, order, or duplication.

Internet Application Protocols

Several protocols have been created that run on top of TCP/IP in order to provide various services to users. These include:

- FTP and SFTP (File Transfer Protocol and Secure FTP) allows the transfer of collection of files between two machines connected to the Internet.
- Telnet (Terminal Protocol) allows a user to connect to a remote host in terminal mode.
- NNTP (Network News Transfer Protocol) allows the constitution of communication groups (newsgroups) organized around specific topics.
- SMTP (Simple Mail Transfer Protocol) defines a basic service for electronic mail.
- SNMP (Simple Network Management Protocol) allows the management of the network.