Introduction to Electrical Engineering

Introduction to Computer Networks

INTRODUCTION TO COMPUTER NETWORKS

Introduction

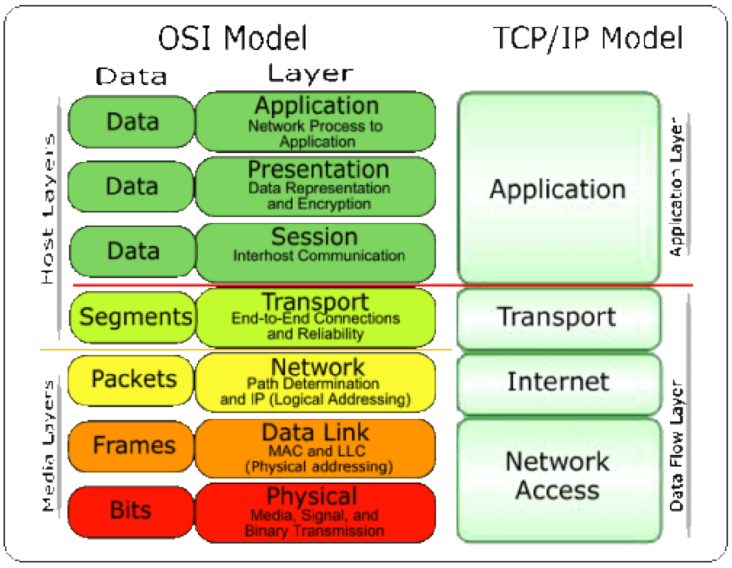
There are many interconnections in the field of networking, as in most technical fields, and it is difficult to find an order of presentation that does not involve endless "forward references" to other topics; this is true even if – as is done here – a largely bottom-up ordering is followed. This chapter is a summary of the bare essentials – LANs, IP and TCP – across the board.

Local Area Networks, or LANs, are the "physical" networks that provide the connection between machines within, say, a home, school or corporation. LANs are, as the name says, "local"; it is the IP, or Internet Protocol, layer that provides an abstraction for connecting multiple LANs into, well, the Internet. Finally, TCP deals with transport and connections and actually sending user data. This chapter also discusses packets and congestion. Firewalls and network address translation are also covered here.

Layers

These three topics – LANs, IP and TCP – are often called layers; they constitute the Link layer, the Internetwork layer, and the Transport layer respectively. Together with the Application layer (the software you use), these form the "four-layer model" for networks. A layer, in this context, corresponds strongly to the idea of a programming interface or library, with the understanding that a given layer communicates directly only with the two layers immediately above and below it. An application hands off a chunk of data to the TCP library, which in turn makes calls to the IP library, which in turn calls the LAN layer for actual delivery. An application does *not* interact directly with the IP and LAN layers at all.

More recently, the Open Systems Interconnection (OSI) model introduced two more rather obscure layers that complete the seven-layer model. The OSI model is a conceptual framework that describes the functions of a networking or telecommunication system. The model uses layers to help give a visual description of what is going on with a particular networking system. This can help network managers narrow down problems (Is it a physical issue or something with the application?), as well as computer programmers (when developing an application, which other layers does it need to work with?). Tech vendors selling new products will often refer to the OSI model to help customers understand which layer their products work with or whether it works "across the stack".



Conceived in the 1970s when computer networking was taking off, two separate models were merged in 1983 and published in 1984 to create the OSI model that most people are familiar with today. Most descriptions of the OSI model go from top to bottom, with the numbers going from Layer 7 down to Layer 1. The layers, and what they represent, are as follows:

Layer 7 – Application: To further our bean dip analogy, the Application Layer is the one at the top - it's what most users see. In the OSI model, this is the layer that is the "closest to the end user". Applications that work at Layer 7 are the ones that users interact with directly. A web browser (Google

Chrome, Firefox, Safari, etc.) or other app - Skype, Outlook, Office - are examples of Layer 7 applications.

<u>Layer 6</u> – Presentation: The Presentation Layer represents the area that is independent of data representation at the application layer - in general, it represents the preparation or translation of application format to network format, or from network formatting to application format. In other words, the layer "presents" data for the application or the network. A good example of this is encryption and decryption of data for secure transmission - this happens at Layer 6.

<u>Layer 5</u> – Session: When two devices, computers or servers need to "speak" with one another, a session needs to be created, and this is done at the Session Layer. Functions at this layer involve setup, coordination (how long should a system wait for a response, for example) and termination between the applications at each end of the session.

<u>Layer 4</u> – Transport: The Transport Layer deals with the coordination of the data transfer between end systems and hosts. How much data to send, at what rate, where it goes, etc. The best known example of the Transport Layer is the Transmission Control Protocol (TCP), which is built on top of the Internet Protocol (IP), commonly known as TCP/IP. TCP and UDP port numbers work at Layer 4, while IP addresses work at Layer 3, the Network Layer.

<u>Layer 3</u> – Network: Here at the Network Layer is where you'll find most of the router functionality that most networking professionals care about and love. In its most basic sense, this layer is responsible for packet forwarding, including routing through different routers. You might know that your Boston computer wants to connect to a server in California, but there are millions of different paths to take. Routers at this layer help do this efficiently.

<u>Layer 2</u> – Data Link: The Data Link Layer provides node-to-node data transfer (between two directly connected nodes), and also handles error correction from the physical layer. Two sublayers exist here as well - the Media Access Control (MAC) layer and the Logical Link Control (LLC) layer. In the networking world, most switches operate at Layer 2.

<u>Layer 1</u> – Physical: At the bottom of our OSI bean dip we have the Physical Layer, which represents the electrical and physical representation of the system. This can include everything from the cable type, radio frequency link (as in an 802.11 wireless systems), as well as the layout of pins, voltages and other physical requirements. When a networking problem occurs, many networking pros go right to the physical layer to check that all of the cables are properly connected and that the power plug hasn't been pulled from the router, switch or computer, for example.

Most people in the IT space will likely need to know about the different layers when they're going for their certifications, much like a civics student needs to learn about the three branches of the U.S. government. After that, you hear about the OSI model when vendors are making pitches about which layer(s) their products work with. In a Quora post asking about the purpose of the OSI model, Vikram Kumar answered this way: "The purpose of the OSI reference model is to guide vendors and developers so the digital communication products and software programs they create will interoperate, and to facilitate clear comparisons among communications tools." While some people may argue that the OSI model is obsolete (due to its theoretical nature and less important than the 4 layers of the TCP/IP model), Kumar says that "it is difficult to read about networking technology today without seeing references to the OSI model and its layers, because the model's structure helps to frame discussions of protocols and contrast various technologies." If you can understand the OSI model and its layers, you can also then understand which protocols and devices can interoperate with each other when new technologies are developed and explained.

Data Rate, Throughput and Bandwidth

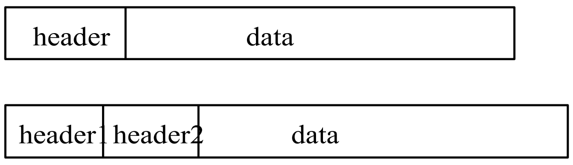
Any one network connection – *eg* at the LAN layer – has a data rate: the rate at which bits are transmitted. In some LANs (*eg* Wi-Fi) the data rate can vary with time. Throughput refers to the overall effective transmission rate, taking into account things like transmission overhead, protocol inefficiencies and perhaps even competing traffic. It is generally measured at a higher network layer than the data rate.

The term bandwidth can be used to refer to either of these, though we here use it mostly as a synonym for data rate. The term comes from radio transmission, where the width of the frequency band available is proportional, all else being equal, to the data rate that can be achieved.

Data rates are generally measured in kilobits per second (kbps) or megabits per second (Mbps); the use of the lower-case "b" here denotes bits. In the context of data rates, a kilobit is 10^3 bits (not 2^{10}) and a megabit is 10^6 bits. Somewhat inconsistently, we follow the tradition of using kB and MB to denote data *volumes* of 2^{10} and 2^{20} bytes respectively, with the upper-case B denoting bytes. The newer abbreviations KiB and MiB would be more precise, but the consequences of confusion are modest.

Packets

Packets are modest-sized buffers of data, transmitted as a unit through some shared set of links. Of necessity, packets need to be prefixed with a header containing delivery information. In the common case known as datagram forwarding, the header contains a destination address; headers in networks using so-called virtual-circuit forwarding contain instead an identifier for the *connection*. Almost all networking today (and for the past 50 years) is packet-based, although we will later look briefly at some "circuit-switched" options for voice telephony.



Single and multiple

The maximum packet size supported by a given LAN (*eg* Ethernet, Token Ring or ATM) is an intrinsic attribute of that LAN. Ethernet allows a maximum of 1500 bytes of data. By comparison, TCP/IP packets originally often held only 512 bytes of data, while early Token Ring packets could contain up to 4 kB of data. While there are proponents of very large packet sizes, larger even than 64 kB, at the other extreme the ATM (Asynchronous Transfer Mode) protocol uses 48 bytes of data per packet, and there are good reasons for believing in modest packet sizes.

Generally each layer adds its own header. Ethernet headers are typically 14 bytes, IP headers 20 bytes, and TCP headers 20 bytes. If a TCP connection sends 512 bytes of data per packet, then the headers amount to 10% of the total, a not-unreasonable overhead. For one common Voice-over-IP option, packets contain 160 bytes of data and 54 bytes of headers, making the header about 25% of the total. Compressing the 160 bytes of audio, however, may bring the data portion down to 20 bytes, meaning that the headers are now 73% of the total.

In datagram-forwarding networks the appropriate header will contain the address of the destination and perhaps other delivery information. Internal nodes of the network called routers or switches will then try to ensure that the packet is delivered to the requested destination.

The concept of packets and packet switching was first introduced by Paul Baran in 1962. Baran's primary concern was with network survivability in the event of node failure; existing centrally switched protocols were vulnerable to central failure. In 1964, Donald Davies independently developed many of the same concepts; it was Davies who coined the term "packet".

Datagram Forwarding

In the datagram-forwarding model of packet delivery, packet headers contain a destination address. It is up to the intervening switches or routers to look at this address and get the packet to the correct destination.

In datagram forwarding this is achieved by providing each switch with a forwarding table of <destination, next_hop> pairs. When a packet arrives, the switch looks up the destination address (presumed globally unique) in its forwarding table and finds the next_hop information: the immediateneighbor address to which – or interface by which – the packet should be forwarded in order to bring it one step closer to its final destination. The next_hop value in a forwarding table is a single entry; each switch is responsible for only one step in the packet's path. However, if all is well, the network of switches will be able to deliver the packet, one hop at a time, to its ultimate destination.

The "destination" entries in the forwarding table do not have to correspond exactly with the packet destination addresses, though in the examples here they do, and they do for Ethernet datagram forwarding. However, for IP routing, the table "destination" entries will correspond to prefixes of IP addresses; this leads to a huge savings in space. The fundamental requirement is that the switch can perform a lookup operation, using its forwarding table and the destination address in the arriving packet, to determine the next hop.

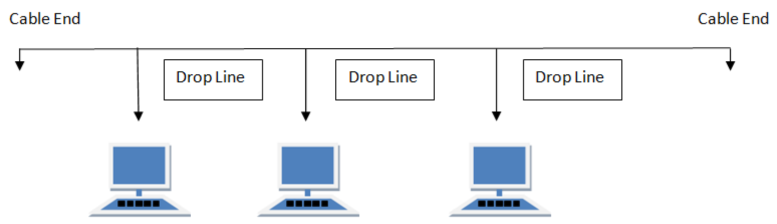
A central feature of datagram forwarding is that each packet is forwarded "in isolation"; the switches involved do not have any awareness of any higher-layer logical connections established between endpoints. This is so to improve robustness of the communication system, gateways are designed to be stateless, forwarding each IP datagram independently of other datagrams. As a result, redundant paths can be exploited to provide robust service in spite of failures of intervening gateways and networks.

Topology

Network Topology is the schematic description of a network arrangement, connecting various nodes (sender and receiver) through lines of connection.

BUS Topology

Bus topology is a network type in which every computer and network device is connected to single cable. When it has exactly two endpoints, then it is called **Linear Bus topology**.



Features of Bus Topology

1. It transmits data only in one direction.
2. Every device is connected to a single cable

Advantages of Bus Topology

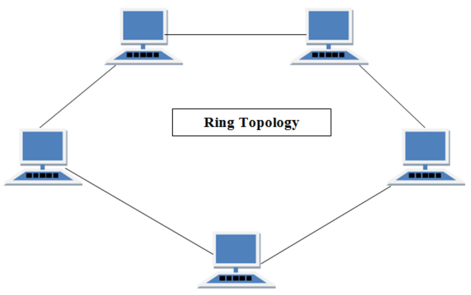
1. It is cost effective.
2. Cable required is least compared to other network topology.
3. Used in small networks.
4. It is easy to understand.
5. Easy to expand joining two cables together.

Disadvantages of Bus Topology

1. Cables fails then whole network fails.
2. If network traffic is heavy or nodes are more the performance of the network decreases.
3. Cable has a limited length.
4. It is slower than the ring topology.

RING Topology

It is called ring topology because it forms a ring as each computer is connected to another computer, with the last one connected to the first. Exactly two neighbours for each device.



Features of Ring Topology

1. A number of repeaters are used for Ring topology with large number of nodes, because if someone wants to send some data to the last node in the ring topology with 100 nodes, then the data will have to pass through 99 nodes to reach the 100th node. Hence to prevent data loss repeaters are used in the network.
2. The transmission is unidirectional, but it can be made bidirectional by having 2 connections between each Network Node, it is called **Dual Ring Topology**.
3. In Dual Ring Topology, two ring networks are formed, and data flow is in opposite direction in them. Also, if one ring fails, the second ring can act as a backup, to keep the network up.
4. Data is transferred in a sequential manner that is bit by bit. Data transmitted, has to pass through each node of the network, till the destination node.

Advantages of Ring Topology

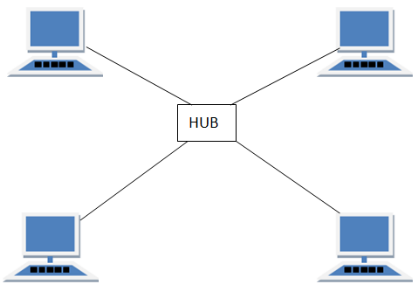
1. Transmitting network is not affected by high traffic or by adding more nodes, as only the nodes having tokens can transmit data.
2. Cheap to install and expand

Disadvantages of Ring Topology

1. Troubleshooting is difficult in ring topology.
2. Adding or deleting the computers disturbs the network activity.
3. Failure of one computer disturbs the whole network.

STAR Topology

In this type of topology all the computers are connected to a single hub through a cable. This hub is the central node and all others nodes are connected to the central node.



Features of Star Topology

1. Every node has its own dedicated connection to the hub.
2. Hub acts as a repeater for data flow.
3. Can be used with twisted pair, Optical Fibre or coaxial cable.

Advantages of Star Topology

1. Fast performance with few nodes and low network traffic.
2. Hub can be upgraded easily.
3. Easy to troubleshoot.

4. Easy to setup and modify.
5. Only that node is affected which has failed, rest of the nodes can work smoothly.

Disadvantages of Star Topology

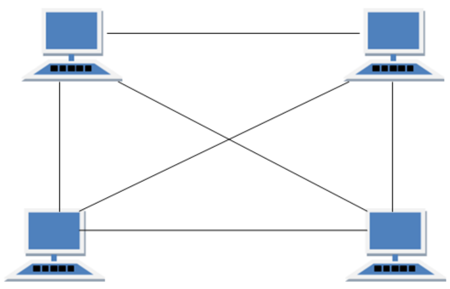
1. Cost of installation is high.
2. Expensive to use.
3. If the hub fails then the whole network is stopped because all the nodes depend on the hub.
4. Performance is based on the hub that is it depends on its capacity

MESH Topology

It is a point-to-point connection to other nodes or devices. All the network nodes are connected to each other. Mesh has n(n-1)/2 physical channels to link n devices.

There are two techniques to transmit data over the Mesh topology, they are :

1. **Routing**: In routing, the nodes have a routing logic, as per the network requirements. Like routing logic to direct the data to reach the destination using the shortest distance. Or, routing logic which has information about the broken links, and it avoids those node etc. We can even have routing logic, to re-configure the failed nodes.
2. **Flooding**: In flooding, the same data is transmitted to all the network nodes, hence no routing logic is required. The network is robust, and the its very unlikely to lose the data. But it leads to unwanted load over the network.



Types of Mesh Topology

1. **Partial Mesh Topology :** In this topology some of the systems are connected in the same fashion as mesh topology but some devices are only connected to two or three devices.
2. **Full Mesh Topology :** Each and every nodes or devices are connected to each other.

Features of Mesh Topology

1. Fully connected.
2. Robust.
3. Not flexible.

Advantages of Mesh Topology

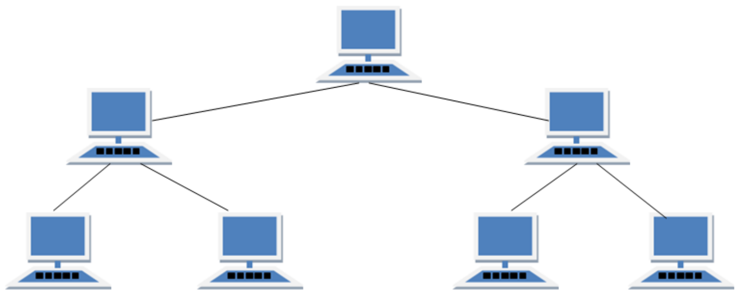
1. Each connection can carry its own data load.
2. It is robust.
3. Fault is diagnosed easily.
4. Provides security and privacy.

Disadvantages of Mesh Topology

1. Installation and configuration is difficult.
2. Cabling cost is more.
3. Bulk wiring is required.

TREE Topology

It has a root node and all other nodes are connected to it forming a hierarchy. It is also called hierarchical topology. It should at least have three levels to the hierarchy.



Features of Tree Topology

1. Ideal if workstations are located in groups.
2. Used in Wide Area Network.

Advantages of Tree Topology

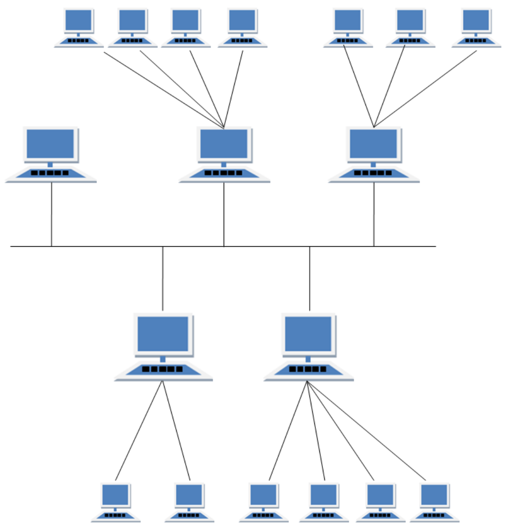
1. Extension of bus and star topologies.
2. Expansion of nodes is possible and easy.
3. Easily managed and maintained.
4. Error detection is easily done.

Disadvantages of Tree Topology

1. Heavily cabled.
2. Costly.
3. If more nodes are added maintenance is difficult.
4. Central hub fails, network fails.

HYBRID Topology

It is two different types of topologies which is a mixture of two or more topologies. For example if in an office in one department ring topology is used and in another star topology is used, connecting these topologies will result in Hybrid Topology (ring topology and star topology).



Features of Hybrid Topology

1. It is a combination of two or topologies
2. Inherits the advantages and disadvantages of the topologies included

Advantages of Hybrid Topology

1. Reliable as Error detecting and trouble shooting is easy.
2. Effective.
3. Scalable as size can be increased easily.
4. Flexible.

Disadvantages of Hybrid Topology

1. Complex in design.
2. Costly.

Traffic Engineering

In some cases the decision between routes might be of material significance – perhaps the one link is slower than the others, or is more congested. We will use the term traffic engineering to refer to any intentional selection of one route over another, or any elevation of the priority of one class of traffic. The route selection can either be directly intentional, through configuration, or can be implicit in the selection or tuning of algorithms that then make these route-selection choices automatically.

With pure datagram forwarding, used at either the LAN or the IP layer, the path taken by a packet is determined solely by its destination, and traffic engineering is limited to the choices made between alternative paths. We have already, however, suggested that datagram forwarding can be extended to take quality-of-service information into account; this may be used to have voice traffic – with its relatively low bandwidth but intolerance for delay – take an entirely different path than bulk file transfers. Alternatively, the network manager may simply assign voice traffic a higher priority, so it does not have to wait in queues behind file-transfer traffic.

The quality-of-service information may be set by the end-user, in which case an ISP may wish to recognize it only for designated users, which in turn means that the ISP will implicitly use the traffic source when making routing decisions. Alternatively, the quality-of-service information may be set by the ISP itself, based on its best guess as to the application; this means that the ISP may be using packet size, port number and other contents as part of the routing decision.

Routing Loops

A potential drawback to datagram forwarding is the possibility of a routing loop: a set of entries in the forwarding tables that cause some packets to circulate endlessly and might easily consume a large majority of the bandwidth. Routing loops typically arise because the creation of the forwarding tables is often “distributed”, and there is no global authority to detect inconsistencies. Even when there is such an authority, temporary routing loops can be created due to notification delays.

Congestion

Switches introduce the possibility of congestion: packets arriving faster than they can be sent out. This can happen with just two interfaces, if the inbound interface has a higher bandwidth than the outbound interface; another common source of congestion is traffic arriving on multiple inputs and all destined for the same output.

Whatever the reason, if packets are arriving for a given outbound interface faster than they can be sent, a queue will form for that interface. Once that queue is full, packets will be dropped. The most common strategy (though not the only one) is to drop any packets that arrive when the queue is full.

The term “congestion” may refer either to the point where the queue is just beginning to build up, or to the point where the queue is full and packets are lost. In their paper, Chiu and Jain refer to the first point as the knee; this is where the slope of the load vs throughput graph flattens. They refer to the second point as the cliff; this is where packet losses may lead to a precipitous decline in throughput. Other authors use the term contention for knee-congestion.

In the Internet, most packet losses are due to congestion. This is not because congestion is especially bad (though it can be, at times), but rather that other types of losses (*eg* due to packet corruption) are insignificant by comparison.

We emphasize that the presence of congestion does *not* mean that a network has a shortage of bandwidth. Bulk-traffic senders (though not real-time senders) attempt to send as fast as possible, and congestion is simply the network's feedback that the maximum transmission rate has been reached.

Congestion *is* a sign of a problem in real-time networks. In these networks losses due to congestion must generally be kept to an absolute minimum; one way to achieve this is to limit the acceptance of new connections unless sufficient resources are available.

LANs and Ethernet

A local-area network, or LAN, is a system consisting of

- physical links that are, ultimately, serial lines
- common interfacing hardware connecting the hosts to the links
- protocols to make everything work together

We will explicitly assume that every LAN node is able to communicate with every other LAN node. Sometimes this will require the cooperation of intermediate nodes acting as switches.

Far and away the most common type of (wired) LAN is Ethernet, originally described in a 1976 paper by Metcalfe and Boggs. Ethernet's popularity is due to low cost more than anything else, though the primary reason Ethernet cost is low is that high demand has led to manufacturing economies of scale.

The original Ethernet had a bandwidth of 10 Mbps (megabits per second; we will use lower-case "b" for bits and upper-case "B" for bytes), though nowadays most Ethernet operates at 100 Mbps and gigabit (1000 Mbps) Ethernet (and faster) is widely used in server rooms. (By comparison, as of this writing (2015) the data transfer rate to a typical faster hard disk is about 1000 Mbps.) Wireless ("Wi-Fi") LANs are gaining popularity, and in many settings have supplanted wired Ethernet to end-users.

The original 10BASE5 Ethernet uses coaxial cable as a shared medium, while the newer Ethernet variants use twisted pair and fiber optic links in conjunction with switches. Over the course of its history, Ethernet data transfer rates have been increased from the original 2.94 megabits per second (Mbit/s) to the latest 400 gigabits per second (Gbit/s). The Ethernet standards comprise several wiring and signaling variants of the OSI physical layer in use with Ethernet.

Systems communicating over Ethernet divide a stream of data into shorter pieces called frames. Each frame contains source and destination addresses, and error-checking data so that damaged frames can be detected and discarded; most often, higher-layer protocols trigger retransmission of lost frames. As per the OSI model, Ethernet provides services up to and including the data link layer.[3] The 48-bit MAC address was adopted by other IEEE 802 networking standards, including IEEE 802.11 Wi-Fi, as well as by FDDI, and EtherType values are also used in Subnetwork Access Protocol (SNAP) headers.

Ethernet is widely used in homes and industry. The Internet Protocol is commonly carried over Ethernet and so it is considered one of the key technologies that make up the Internet.

Ethernet addresses are six bytes long. Each Ethernet card (or network interface) is assigned a (supposedly) unique address at the time of manufacture; this address is burned into the card's ROM and is called the card's physical address or hardware address or MAC (Media Access Control) address. The first three bytes of the physical address have been assigned to the manufacturer; the subsequent three bytes are a serial number assigned by that manufacturer.

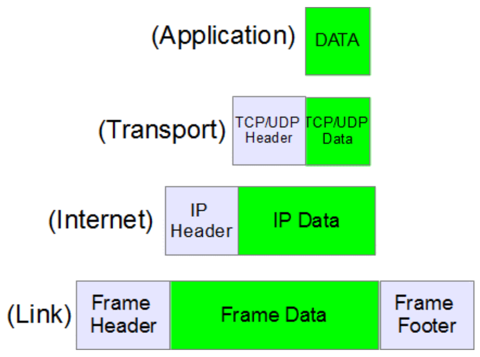
The network interface continually monitors all arriving packets; if it sees any packet containing a destination address that matches its own physical address, it grabs the packet and forwards it to the attached CPU (via a CPU interrupt).

Ethernet also has a designated broadcast address. A host sending to the broadcast address has its packet received by every other host on the network; if a switch receives a broadcast packet on one port, it forwards the packet out every other port. This broadcast mechanism allows host A to contact host B when A does not yet know B's physical address; typical broadcast queries have forms such as "Will the designated server please answer" or (from the ARP protocol) "will the host with the given IP address please tell me your physical address".

Traffic addressed to a particular host – that is, not broadcast – is said to be unicast.

IP - Internet Protocol

The Internet Protocol (IP) is the principal communications protocol in the Internet protocol suite for relaying datagrams across network boundaries. Its routing function enables internetworking, and essentially establishes the Internet. IP has the task of delivering packets from the source host to the destination host solely based on the IP addresses in the packet headers. For this purpose, IP defines packet structures that encapsulate the data to be delivered. It also defines addressing methods that are used to label the datagram with source and destination information.



Historically, IP was the connectionless datagram service in the original Transmission Control Program introduced by Vint Cerf and Bob Kahn in 1974, which was complemented by a connection-oriented service that became the basis for the Transmission Control Protocol (TCP). The Internet protocol suite is therefore often referred to as TCP/IP. The first major version of IP, Internet Protocol Version 4 (IPv4), is the dominant protocol of the Internet. Its successor, Internet Protocol Version 6 (IPv6), has been growing in adoption, reaching almost 25% of all Internet traffic as of October, 2018. IPv6 was a result of several years of experimentation and dialog during which various protocol models were proposed. Its most prominent difference from version 4 is the size of the addresses. While IPv4 uses 32 bits for addressing, yielding c. 4.3 billion (4.3×10^9) addresses, IPv6 uses 128-bit addresses providing ca. 340 undecillion, or 3.4×10^{38} addresses. Although adoption of IPv6 has been slow, as of June 2008, all United States government systems have demonstrated basic infrastructure support for IPv6.

The Internet Protocol is responsible for addressing host interfaces, encapsulating data into datagrams (including fragmentation and reassembly) and routing datagrams from a source host interface to a destination host interface across one or more IP networks. For these purposes, the Internet Protocol defines the format of packets and provides an addressing system.

Each datagram has two components: a header and a payload. The IP header includes source IP address, destination IP address, and other metadata needed to route and deliver the datagram. The payload is the data that is transported. This method of nesting the data payload in a packet with a header is called encapsulation.

IP addressing entails the assignment of IP addresses and associated parameters to host interfaces. The address space is divided into subnetworks, involving the designation of network prefixes. IP routing is performed by all hosts, as well as routers, whose main function is to transport packets across network boundaries. Routers communicate with one another via specially designed routing protocols, either interior gateway protocols or exterior gateway protocols, as needed for the topology of the network.

The design of the Internet protocol suite adheres to the end-to-end principle. Under the end-to-end principle, the network infrastructure is considered inherently unreliable at any single network element or transmission medium and is dynamic in terms of availability of links and nodes. No central monitoring or performance measurement facility exists that tracks or maintains the state of the network. For the benefit of reducing network complexity, the intelligence in the network is purposely located in the end nodes.

As a consequence of this design, the Internet Protocol only provides best-effort delivery and its service is characterized as unreliable. Various error conditions may occur, such as data corruption, packet loss and duplication. Because routing is dynamic, meaning every packet is treated independently, and because the network maintains no state based on the path of prior packets, different packets may be routed to the same destination via different paths, resulting in out-of-order delivery to the receiver. All error conditions in the network must be detected and compensated by the participating end nodes. The upper layer protocols of the Internet protocol suite are responsible for resolving reliability issues. For example, a host may buffer network data to ensure correct ordering before the data is delivered to an application. IPv4 provides safeguards to ensure that the header of an IP packet is error-free. A routing node discards packets that fail a header checksum test. Although the Internet Control Message Protocol (ICMP) provides notification of errors, a routing node is not required to notify either end node of errors. IPv6, by contrast, operates without header checksums, since current link layer technology is assumed to provide sufficient error detection.

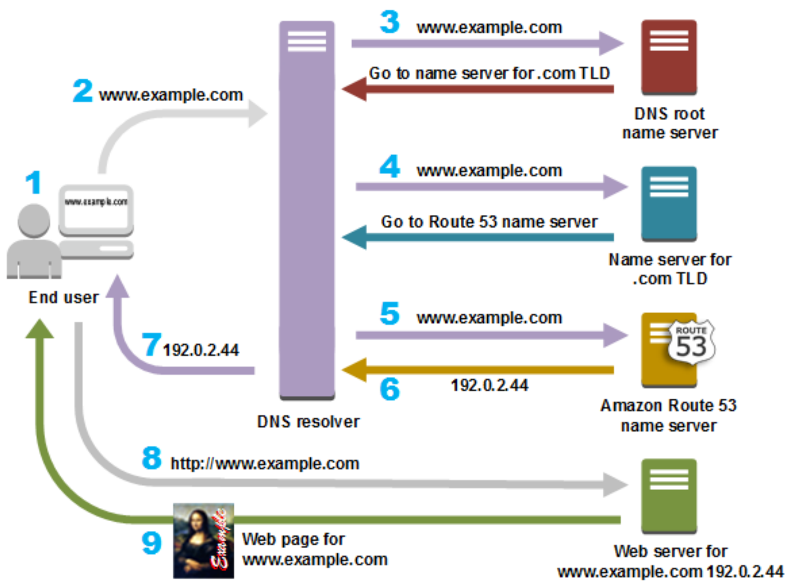
During the design phase of the ARPANET and the early Internet, the security aspects and needs of a public, international network could not be adequately anticipated. Consequently, many Internet protocols exhibited vulnerabilities highlighted by network attacks and later security assessments.

DNS

IP addresses are hard to remember (nearly impossible in IPv6). The domain name system, or DNS, comes to the rescue by creating a way to convert hierarchical text names to IP addresses. Thus, for example, one can type www.luc.edu instead of 147.126.1.230. Virtually all Internet software uses the same basic library calls to convert DNS names to actual addresses.

One thing DNS makes possible is changing a website's IP address while leaving the name alone. This allows moving a site to a new provider, for example, without requiring users to learn anything new. It is also possible to have several different DNS names resolve to the same IP address, and – through some modest trickery – have the http (web) server at that IP address handle the different DNS names as completely different websites.

DNS is hierarchical and distributed. In looking up cs.luc.edu four different DNS servers may be queried: for the so-called "DNS root zone", for edu, for luc.edu and for cs.luc.edu. Searching a hierarchy can be cumbersome, so DNS search results are normally cached locally. If a name is not found in the cache, the lookup may take a couple seconds. The DNS hierarchy need have nothing to do with the IP-address hierarchy.



Transport

The IP layer gets packets from one node to another, but it is not well-suited to transport. First, IP routing is a "best-effort" mechanism, which means packets can and do get lost sometimes. Additionally, data that does arrive can arrive out of order. Finally, IP only supports sending to a specific host; normally, one wants to send to a given application running on that host. Email and web traffic, or two different web sessions, should not be commingled!

The Transport layer is the layer above the IP layer that handles these sorts of issues, often by creating some sort of *connection* abstraction. Far and away the most popular mechanism in the Transport layer is the Transmission Control Protocol, or TCP. TCP extends IP with the following features:

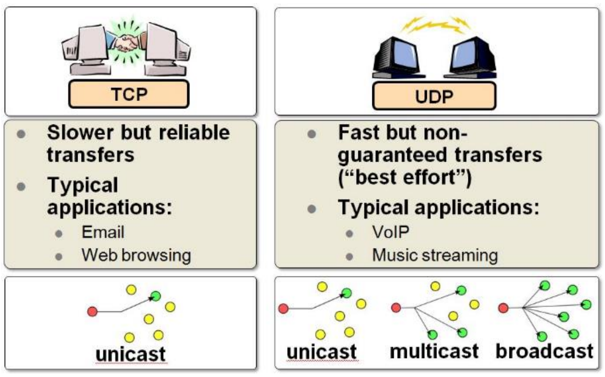
- reliability: TCP numbers each packet, and keeps track of which are lost and retransmits them after a timeout. It holds early-arriving out-of-order packets for delivery at the correct time. Every arriving data packet is acknowledged by the receiver; timeout and retransmission occurs when an acknowledgment packet isn't received by the sender within a given time.
- connection-orientation: Once a TCP connection is made, an application sends data simply by writing to that connection. No further application-level addressing is needed. TCP connections are managed by the operating-system kernel, not by the application.

- stream-orientation: An application using TCP can write 1 byte at a time, or 100 kB at a time; TCP will buffer and/or divide up the data into appropriate sized packets.
- port numbers: these provide a way to specify the receiving application for the data, and also to identify the sending application.
- throughput management: TCP attempts to maximize throughput, while at the same time not contributing unnecessarily to network congestion.

While TCP is ubiquitous, the real-time performance of TCP is not always consistent: if a packet is lost, the receiving TCP host will not turn over anything further to the receiving application until the lost packet has been retransmitted successfully; this is often called head-of-line blocking. This is a serious problem for sound and video applications, which can discretely handle modest losses but which have much more difficulty with sudden large delays. A few lost packets ideally should mean just a few brief voice dropouts (pretty common on cell phones) or flicker/snow on the video screen (or just reuse of the previous frame); both of these are better than pausing completely.

The basic alternative to TCP is known as UDP, for User Datagram Protocol. UDP, like TCP, provides port numbers to support delivery to multiple endpoints within the receiving host, in effect to a specific process on the host. As with TCP, a UDP socket consists of a host-port pair. UDP also includes, like TCP, a checksum over the data. However, UDP omits the other TCP features: there is no connection setup, no lost-packet detection, no automatic timeout/retransmission, and the application must manage its own packetization. This simplicity should not be seen as all negative: the absence of connection setup means data transmission can get started faster, and the absence of lost-packet detection means there is no head-of-line blocking.

The Real-time Transport Protocol, or RTP, sits above UDP and adds some additional support for voice and video applications.



Transport Communications Patterns

The two "classic" traffic patterns for Internet communication are these:

- Interactive or bursty communications such as via ssh or telnet, with long idle times between short bursts
- Bulk file transfers, such as downloading a web page

TCP handles both of these well, although its congestion-management features apply only when a large amount of data is in transit at once. Web browsing is something of a hybrid; over time, there is usually considerable burstiness, but individual pages now often exceed 1 MB.

This century has seen an explosion in streaming video, in lengths from a few minutes to a few hours. Streaming radio stations might be left playing indefinitely. TCP generally works well here, assuming the receiver can get, say, a minute ahead, buffering the video that has been received but not yet viewed. That way, if there is a dip in throughput due to congestion, the receiver has time to recover. Buffering works a little less well for streaming radio, as the listener doesn't want to get too far behind, though ten seconds is reasonable. Fortunately, audio bandwidth is smaller.

Another issue with streaming video is the bandwidth demand. Most streaming-video services attempt to estimate the available throughput, and then *adapt* to that throughput by changing the video resolution.Typically, video streaming operates on a start/stop basis: the sender pauses when the receiver's playback buffer is "full", and resumes when the playback buffer drops below a certain threshold. If the video (or, for that matter, voice audio) is *interactive*, there is much less opportunity for stream buffering. If someone asks a simple question on an Internet telephone call, they generally want an answer more or less immediately; they do not expect to wait for the answer to make it through the other party’s stream buffer. 200 ms of buffering is noticeable. Here we enter the realm of genuine real-time traffic. UDP is often used to avoid head-of-line blocking. Lower bandwidth helps; voice-grade communications traditionally need only 8 kB/sec, less if compression is used. On the other hand, there may be constraints on the *variation* in delivery time (known as *jitter*). Interactive video, with its much higher bandwidth requirements, is more difficult; fortunately, users seem to tolerate the common pauses and freezes.

Within the Transport layer, essentially all network connections involve a client and a server. Often this pattern is repeated at the Application layer as well: the client contacts the server and initiates a login session, or browses some web pages, or watches a movie. Sometimes, however, Application-layer exchanges fit the peer-to-peer model better, in which the two endpoints are more-or-less co-equals. Some examples include

- Internet telephony: there is no benefit in designating the party who place the call as the “client”
- Message passing in a CPU cluster
- So-called peer-to-peer file-sharing, where individuals exchange files with other individuals (and as opposed to “cloud-based” file-sharing in which the “cloud” is the server).

Content-Distribution Networks

Sites with an extremely large volume of content to distribute often turn to a specialized communication pattern called a content-distribution network or CDN. To reduce the amount of long-distance traffic, or to reduce the round-trip time, a site replicates its content at multiple datacenters (also called *Points of Presence* (PoPs), *nodes*, *access points* or *edge servers*). When a user makes a request (*eg* for a web page or a video), the request is routed to the nearest (or approximately nearest) datacenter, and the content is delivered from there.

Large web pages typically contain both *static* content and also individualized *dynamic* content. On a typical Facebook page, for example, the videos and javascript might be considered static, while the individual wall posts might be considered dynamic. The CDN may cache all or most of the static content at each of its edge servers, leaving the dynamic content to come from a centralized server. Alternatively, the dynamic content may be replicated at each CDN edge node as well, though this introduces some real-time coordination issues.

If dynamic content is *not* replicated, the CDN may include private high-speed links between its nodes, allowing for rapid low-congestion delivery to any node. Alternatively, CDN nodes may simply communicate using the public Internet. Finally, the CDN may (or may not) be configured to support fast *interactive* traffic between nodes, *eg* teleconferencing traffic.

Organizations can create their own CDNs, but often turn to specialized CDN providers, who often combine their CDN services with website-hosting services. In principle, all that is needed to create a CDN is a multiplicity of datacenters, each with its own connection to the Internet; private links between datacenters are also common. In practice, many CDN providers also try to build direct connections with the ISPs that serve their customers; the Google Edge Network above does this. This can improve performance and reduce traffic costs.

Finding the edge server that is closest to a given user is a tricky issue. There are three techniques in common use. In the first, the edge servers are all given different IP addresses, and DNS is configured to have users receive the IP address of the closest edge server. In the second, each edge server has the *same* IP address, and specialized (*anycast*) routing is used to route traffic from the user to the closest edge server. Finally, for HTTP applications a centralized server can look up the approximate location of the user, and then redirect the web page to the closest edge server.

Firewalls

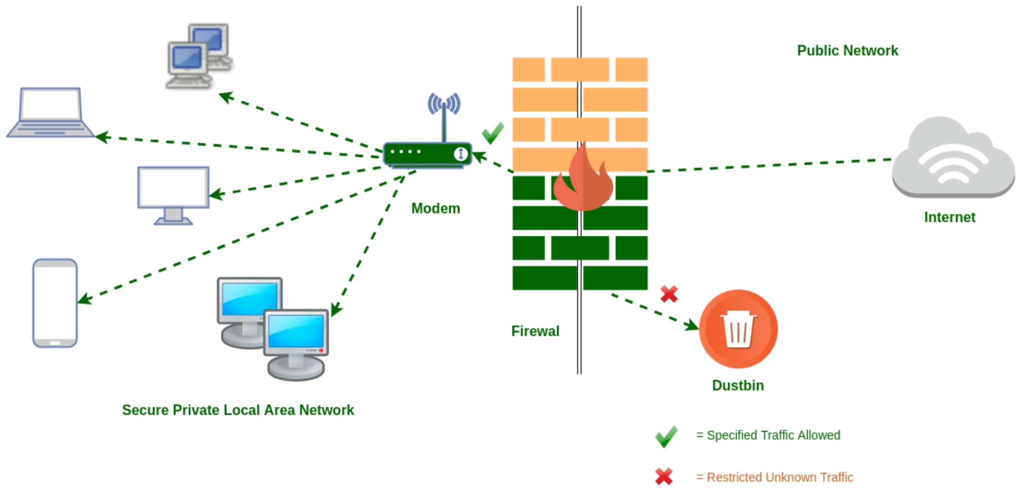
One problem with having a program on your machine listening on an open TCP port is that someone may connect and then, using some flaw in the software on your end, do something malicious to your machine. Damage can range from the unintended downloading of personal data to compromise and takeover of your entire machine, making it a distributor of viruses and worms or a steppingstone in later break-ins of other machines.

A strategy known as buffer overflow has been the basis for a great many total-compromise attacks. The idea is to identify a point in a server program where it fills a memory buffer with network-supplied data without careful length checking; almost any call to the C library function gets(buf) will suffice. The attacker then crafts an oversized input string which, when read by the server and stored in memory, overflows the buffer and overwrites subsequent portions of memory, typically containing the stack-frame pointers. The usual goal is to arrange things so that when the server reaches the end of the currently executing function, control is returned not to the calling function but instead to the attacker's own payload code located within the string.

A firewall is a mechanism to block connections deemed potentially risky, e.g. those originating from outside the site. Generally ordinary workstations do not ever need to accept connections from the Internet; client machines instead *initiate* connections to (better-protected) servers. So blocking incoming connections works reasonably well; when necessary (*eg* for games) certain ports can be selectively unblocked.

The original firewalls were built into routers. Incoming traffic to servers was often blocked unless it was sent to one of a modest number of "open" ports; for non-servers, typically all inbound connections were blocked. This allowed internal machines to operate reasonably safely, though being unable to accept incoming connections is sometimes inconvenient.

Nowadays per-host firewalls – in addition to router-based firewalls – are common: you can configure your workstation not to accept inbound connections to most (or all) ports regardless of whether software on the workstation requests such a connection. Outbound connections can, in many cases, also be prevented.


Public Network
Modem
Internet
Firewal
Dustbin
Secure Private Local Area Network
✓ = Specified Traffic Allowed
✗ = Restricted Unknown Traffic

The internet and the WWW

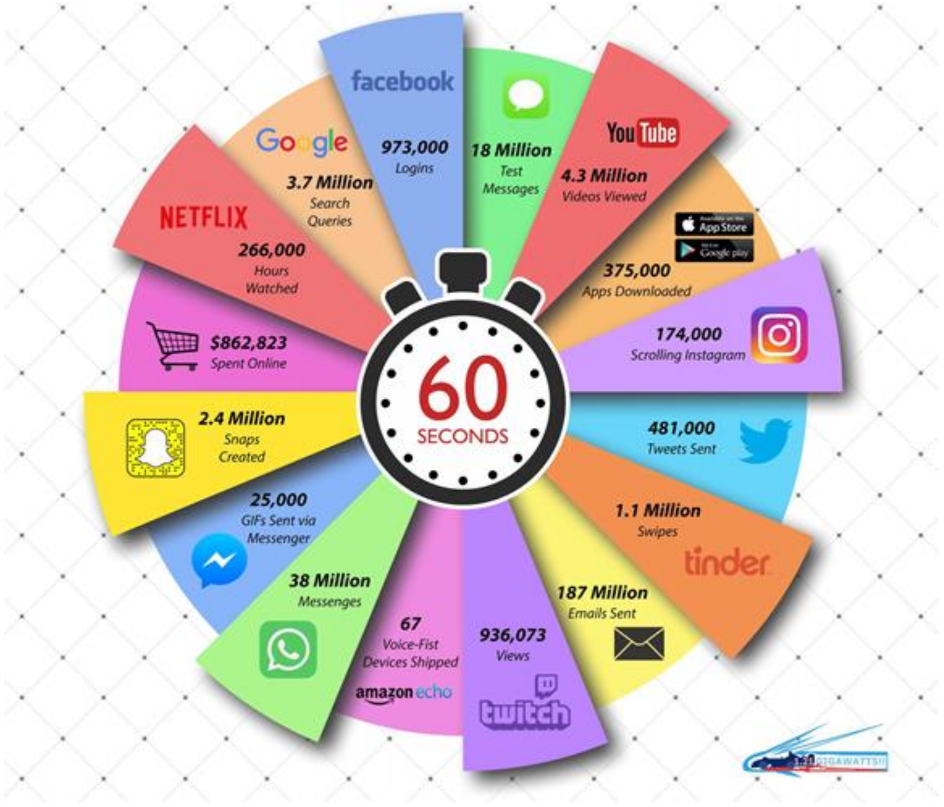
The Internet

The Internet (portmanteau of interconnected network) is the global system of interconnected computer networks that use the Internet protocol suite (TCP/IP) to link devices worldwide. It is a network of networks that consists of private, public, academic, business, and government networks of local to global scope, linked by a broad array of electronic, wireless, and optical networking technologies. The Internet carries a vast range of information resources and services, such as the inter-linked hypertext documents and applications of the World Wide Web (WWW), electronic mail, telephony, and file sharing.

The origins of the Internet date back to research commissioned by the federal government of the United States in the 1960s to build robust, fault-tolerant communication with computer networks. The primary precursor network, the ARPANET, initially served as a backbone for interconnection of regional academic and military networks in the 1980s. The funding of the National Science Foundation Network as a new backbone in the 1980s, as well as private funding for other commercial extensions, led to worldwide participation in the development of new networking technologies, and the merger of many networks. The linking of commercial networks and enterprises by the early 1990s marked the beginning of the transition to the modern Internet, and generated a sustained exponential growth as generations of institutional, personal, and mobile computers were connected to the network. Although the Internet was widely used by academia since the 1980s, commercialization incorporated its services and technologies into virtually every aspect of modern life.

Most traditional communication media, including telephony, radio, television, paper mail and newspapers are reshaped, redefined, or even bypassed by the Internet, giving birth to new services such as email, Internet telephony, Internet television, online music, digital newspapers, and video streaming websites. Newspaper, book, and other print publishing are adapting to website technology, or are reshaped into blogging, web feeds and online news aggregators. The Internet has enabled and accelerated new forms of personal interactions through instant messaging, Internet forums, and social networking. Online shopping has grown exponentially both for major retailers and small businesses and entrepreneurs, as it enables firms to extend their "brick and mortar" presence to serve a larger market or even sell goods and services entirely online. Business-to-business and financial services on the Internet affect supply chains across entire industries.

The Internet has no single centralized governance in either technological implementation or policies for access and usage; each constituent network sets its own policies. The overreaching definitions of the two principal name spaces in the Internet, the Internet Protocol address (IP address) space and the Domain Name System (DNS), are directed by a maintainer organization, the Internet Corporation for Assigned Names and Numbers (ICANN). The technical underpinning and standardization of the core protocols is an activity of the Internet Engineering Task Force (IETF), a non-profit organization of loosely affiliated international participants that anyone may associate with by contributing technical expertise. In November 2006, the Internet was included on USA Today's list of New Seven Wonders.



Average internet usage in 60 seconds

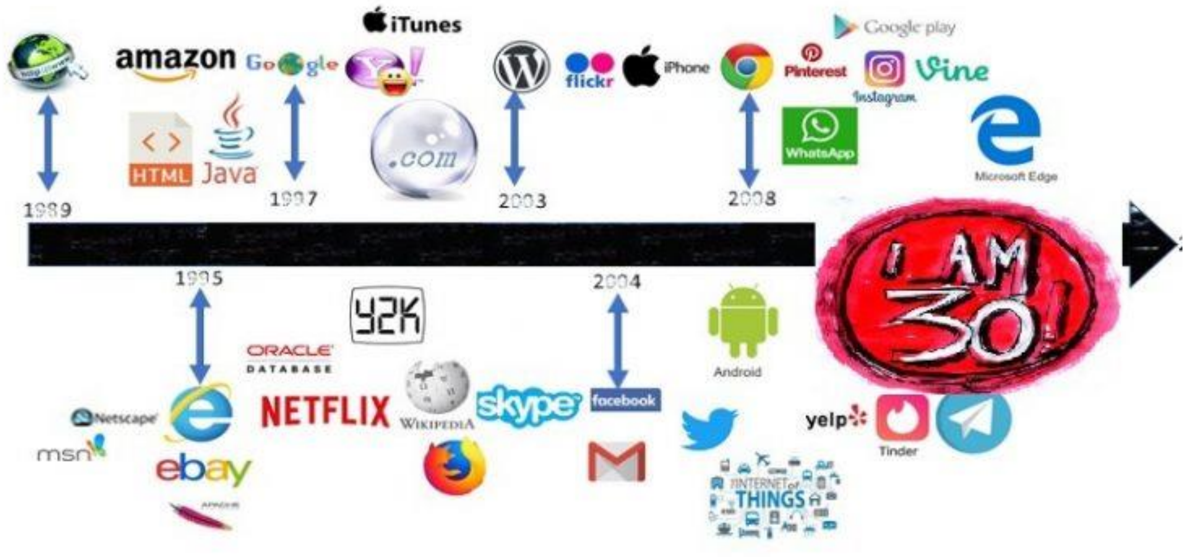
The WWW

The World Wide Web (WWW), commonly known as the Web, is an information system where documents and other web resources are identified by Uniform Resource Locators (URLs, such as https://www.example.com/), which may be interlinked by hypertext, and are accessible over the Internet. The resources of the WWW may be accessed by users by a software application called a web browser.

English scientist Tim Berners-Lee invented the World Wide Web in 1989. He wrote the first web browser in 1990 while employed at CERN near Geneva, Switzerland. The browser was released outside CERN in 1991, first to other research institutions starting in January 1991 and then to the general public in August 1991. The World Wide Web has been central to the development of the Information Age and is the primary tool billions of people use to interact on the Internet.

Web resources may be any type of downloaded media, but web pages are hypertext media that have been formatted in Hypertext Markup Language (HTML). Such formatting allows for embedded hyperlinks that contain URLs and permit users to navigate to other web resources. In addition to text, web pages may contain images, video, audio, and software components that are rendered in the user's web browser as coherent pages of multimedia content.

Multiple web resources with a common theme, a common domain name, or both, make up a website. Websites are stored in computers that are running a program called a web server that responds to requests made over the Internet from web browsers running on a user's computer. Website content can be largely provided by a publisher, or interactively where users contribute content or the content depends upon the users or their actions. Websites may be provided for a myriad of informative, entertainment, commercial, governmental, or non-governmental reasons.



The WWW turns 30 years old!