Networking and the Internet

Dr. Tuğberk Kocatekin
Istanbul Arel University

Network

- We need to share information and resources between computers.

- Network is a set of computers connected to each other using network nodes in order to satisfy that need.

Classifications

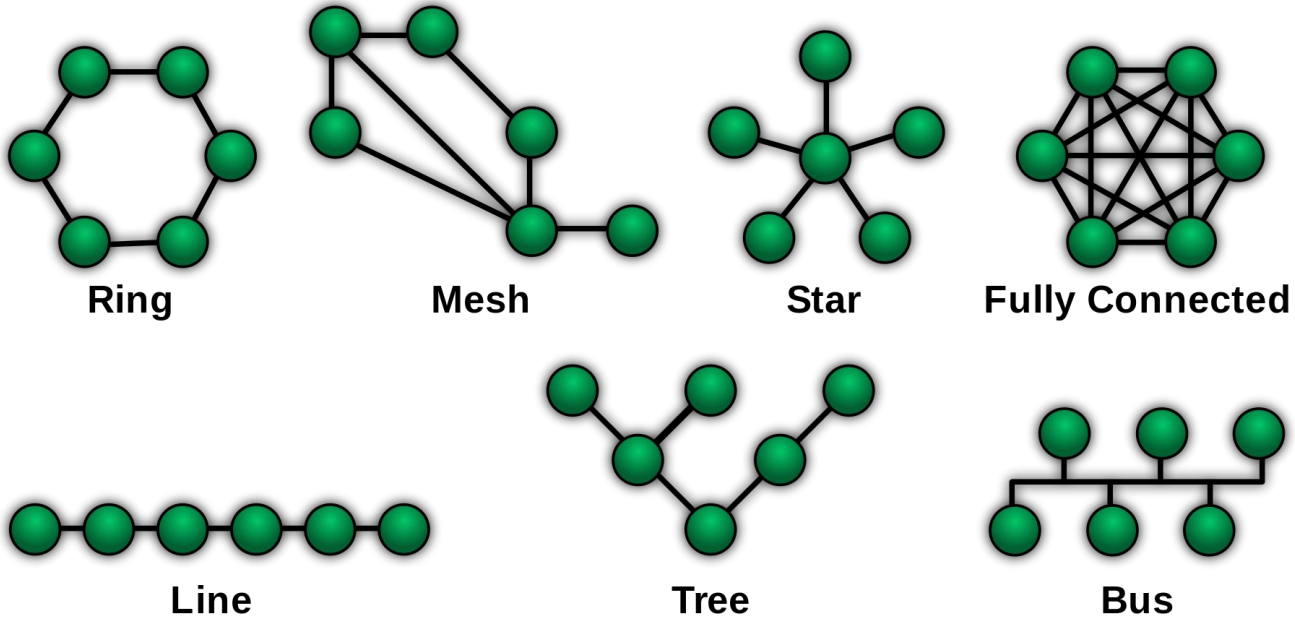
- PAN: Personal Area Network
  - Short-range comm. Less than a few meters (between a smartphone and bluetooth speakers)
- LAN: Local Area Network
  - Collection of computers in a single building. (Computers in a campus)
- MAN: Metropolitan Area Network
  - Network of intermediate size. (Local community)
- WAN: Wide Area Network
  - Connects machines over a greater distance. Between cities or countries.

Open vs Closed (proprietary) Networks

- Open networks are based on designs which are in the public domain
- Closed networks are those which are controlled by a particular entity

- Internet is an open system. Communication is satisfied by an open collection of standards called TCP/IP protocol suite. Anyone is free to use them.

Network Topologies



Network Topologies

- Two popular topologies: **Bus & Star**
- **Bus** was popularized under a set of standard known as Ethernet.
- **Star** topology evolved from the old times. A central computer serving many users.
- Star topology (configuration) is popular in **wireless communication**. There is a central **access point (AP)** which serves as a focal point.
- Sometimes bus and star topologies may look similar but the important distinction is not the physical appearance.
- **Are the machines communicating directly with each other over a single bus or indirectly through an central machine?**

Protocols

- In order for a network to function reliably, there must be certain rules. These rules are called protocols.

- Let’s say we have a network and computers want to communicate with each other. If there is no one to orchestrate, how this will happen?
    - Computers may try to send things at the same time and maybe packages will be lost.

CSMA/CD (Carrier Sense, Multiple Access with Collision Detection)

- Each message is broadcasted to all machines on the bus.
- Each machine monitors all the messages but keeps only those who are addressed to itself.
- In order to transmit a message, the machine waits when the bus is silent. At that time, starts transmission but carries on monitoring. If anyone else tries to send a message, they both understand that there is a collision and pause for a random period before re-transmission.

CSMA/CD (Carrier Sense, Multiple Access with Collision Detection)

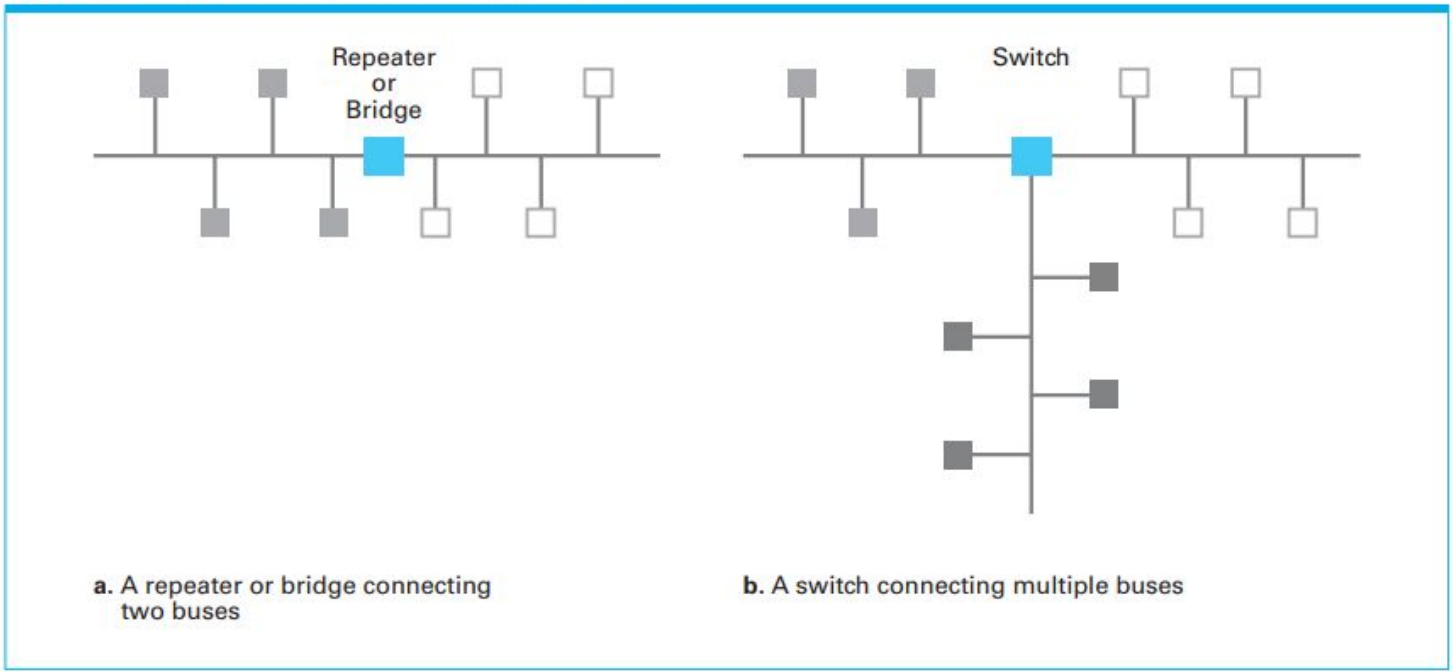
- Not compatible with **wireless star networks**.
  - Why?
    - Because there is no way to see whether the communication line is busy or not. There is no bus.(Star)
    - Maybe the signals are blocked because of the environment.
    - Transmission may collide and cancel each other.
- That is why wireless networks try to **avoid** collisions rather than **detect** them.
  - CSMA/CA: Carrier Sense, Multiple Access with Collision Avoidance
  - This is standardized by IEEE (IEEE 802.11) and named WiFi.
  - Collisions still will occur, they will just retransmit.

Combining networks

- Sometimes we need to connect multiple networks to form an extended network.
  - **Repeater:** Passes signals back and forth between two buses without considering the meaning of all messages.
  - **Bridge:** A more complex repeater. Bridge too connects two buses but it looks at the destination address in each message and only forwards it when it is destined for the computer at the other side. Therefore, two messages in the same side of the bridge can exchange messages w/o interfering with communication on the other side.
  - **Switch:** Bridge between multiple buses.

Combining networks

Figure 4.4 Building a large bus network from smaller ones



a. A repeater or bridge connecting two buses

b. A switch connecting multiple buses

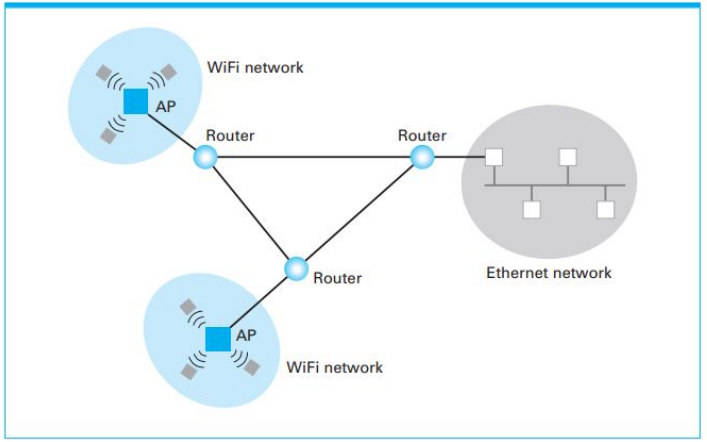
Combining networks

- What happens when networks you want to connect have incompatible characteristics?
  - WiFi vs. Ethernet
- In this case, networks must be connected in such a way where you build a network of networks (**internet**) where the original networks maintain their individuality and continue to function as they are.
  - Internet here does not refer to “The Internet”.
- The connection between networks to form an internet is handled by devices known as **routers**, which are special purpose computers used for forwarding messages.

Routers

- Routers provides links between networks while allowing each network to maintain its unique internal characteristics.
- The name is router because it *routes*.
  - How?
- Forwards messages based on an addressing system where all devices are assigned unique addresses.
- When a machine wants to send a message to a distant network, it attaches the address so that the router knows the destination.

Figure 4.5 Routers connecting two WiFi networks and an Ethernet network to form an internet



Routers

- Each router has a **forwarding table**. This table contains the knowledge about the direction in which messages should be sent depending on their destination addr.
    - There are several tables in networking such as mac table, arp table, etc. These all have their specific mission. When the router receives the package, by checking the ARP table, it can understand which machine is the destination. But how will the router reach that machine? By using which way?

- **Gateway** is the point where a network is linked to an internet. In many cases the gateway is the router, but sometimes access points are also called as gateways in WiFi networks.

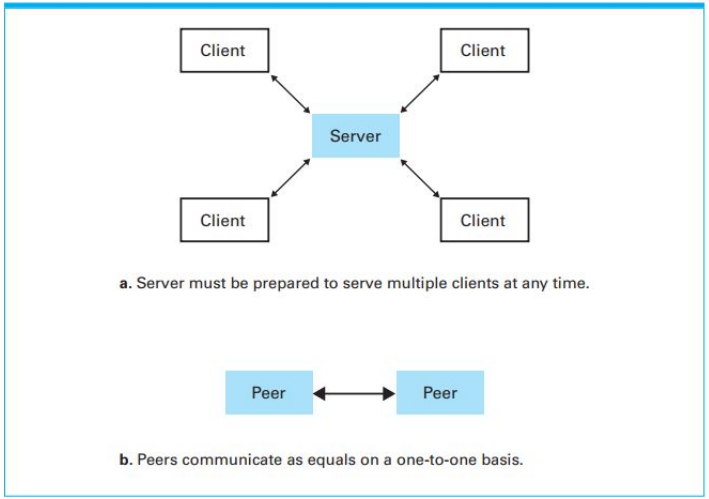
Interprocess communication

- Activities or processes need to communicate with each other to coordinate their actions. This communication between processes are called interprocess communication.
- Popular model is the **client/server** model.
  - **Client** makes requests of other processes
  - **Server** satisfies the requests made by clients. (response)
- Examples:
  - In an office environment there is a printer in the middle any users use the same printer. That printer represents a *print server* and users are *clients*.
  - Again in a similar environment, instead of users storing the same file they can have a file server, residing independently storing files. *Clients* (users) can request access to those files in the *file server*.

Peer-to-peer (P2P)

- Client/server model is not the only model.
- C/S model involves one process (server) providing a service to multiple (client).
- P2P involve processes that provide service to and receive service from each other. *E.g. messaging applications.*
- Also popular for distributing files: *Bittorrent, Napster, Kazaa, etc.*

Figure 4.6 The client/server model compared to the peer-to-peer model



Peer-to-peer (P2P)

- One peer can receive a file while providing that file to other peers (bittorrent).

- The collection of peers participating in such a distribution is called **swarm**.

Client/Server model vs P2P

- P2P distributes the service task over many peers rather than concentrating it at one server.
- In C/S model a single server provides service to multiple clients, whereas in P2P they provide services to each other.
- In C/S model a server must execute continuously to prepare clients at any time, in P2P executes on a temporary basis (not continuous)

Distributed systems

- Distributed systems consists of software units that execute as processes on different computers.
- Cluster computing
  - You use several machines together to provide a computation or service comparable to a much larger machine.
  - Cost of these several machines combined can be cheaper than a supercomputer with higher reliability and maintenance cost.
  - Used for **high-availability**. Because it is possible that a cluster can fail but all of them breaking down is very unlikely.
- Grid computing
  - Similar to cluster computing but more loosely coupled. Can involve specialized software. E.g. You can use your own computer for specific purposes when it is idle (like participating in a cancer research etc).
- Cloud computing
  - Latest trend. Pool of computers in a network can be allocated for use by clients as needed. Very useful, you don't use your own resources and resources are allocated more wisely. A single strong computer can be virtualized into multiple computers and can be used by multiple clients.
  - Google Drive, Dropbox, OneDrive.
  - AWS, EC2.

The Internet

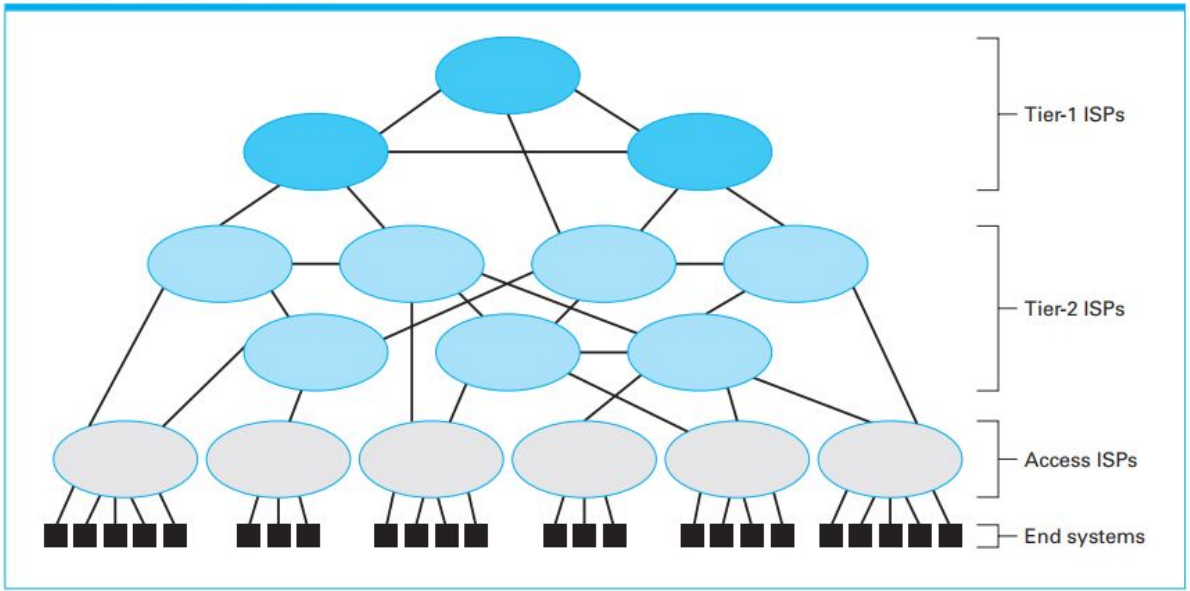
- Originated from a research project sponsored by US government. Aim was to connect multiple networks together.
- These networks are constructed and maintained by organizations called **ISP(Internet Service Providers)**.
- This operation of ISP can be classified in an hierarchy.
  - Tier-1 ISP: Consist of very high-speed, high-capacity, international WAN's. (AT&T, Orange)
  - Tier-2 ISP: More regional in scope and less potent in their abilities. Operated by companies in the communications business. (Vodafone, Turk Telekom International)

Tier-1 and Tier-2

- These are essentially networks of routers that provide the communication infrastructure. They are the core of the Internet.
- Access to this core is usually provided by an intermediary called **access** or **tier-3 ISP**.
- Access ISP is an independent internet, which is also sometimes referred to as **intranet**.
    - Corporations (cable, telephone companies; universities etc) which provide Internet access to individuals within their organizations.
- Devices that individual users connect to the access ISP's are known as **end systems** or **hosts**.
    - Laptops, PC's, mobile phones, game consoles, etc.

Internet composition

Figure 4.7 Internet composition



Last mile problem

- Although modern networks can be designed to carry digital data, it still uses the old analog network infrastructure to carry data.
- Issues arising from these analog linkages are often referred to as **last mile problem**.
- Main arteries are modernized with high-speed digital tech such as fiber optics, but it is very costly to replace existing copper telephone lines and coaxial cables.
- Information originated on the Internet very far away can spend the larger portion of the trip on high-speed digital connections but the “last mile” is an old system. (e.g. DSL modems are a smart solution for this problem)

IP addresses

- Internet wide addressing system.
- Each computer in the system is assigned a unique identifying address.
- Each IPv4 address is a pattern of 32-bits. (There is also IPv6).
- Usually represented in *dot-decimal notation*. Consisting of four decimal numbers separated by dots. 127.0.0.1
- Each part represents a group of 8 bits (octet) of the address. Each ranges from 0-256.

Domain

- Hard to keep numbers in memory. There is an alternative addressing system which is based on the concept of domain.
- Domain can be a university, club, company or government agency.
  - edu, k12, gov, com, net, etc..
    - These are called top-level domains (TLDs)
  - tr, uk, in, etc.
    - Country-code TLDs
- Google.com
  - Google is the domain name
  - admin.google.com
    - subdomain!

DNS (Domain Name Servers)

- We memorize names but everything is transferred by IP addresses.
- Website URL’s are converted to IP addresses.
- This conversion is performed by name servers, directories that provide address translation services to clients.
- These name servers are called **DNS**.
- The process of using DNS to perform this translation is called **DNS lookup**.

Internet Applications

- There were separate protocols in the old times for certain tasks:
  - NNTP (Network News Transfer Protocol): A newsreader application contacted servers using this protocol.
  - FTP (File Transfer Protocol): Application for listing and copying files across the network.
  - Telnet: Application for accessing another computer from a distance
- These are still available but not used as frequently.
- SSH and HTTP are widely used.
  - Secure Shell
  - Hyper Text Transfer Protocol

Email

- SMTP (Simple Mail Transfer Protocol)
  - Designed for transferring text messages encoded with ASCII. Later MIME (Multipurpose Internet Mail Extensions) are developed to convert non-ascii data to SMTP compatible form.
- POP3 & IMAP
  - **Post office protocol version 3**: User downloads messages to her own computer and read, store as desired. Done on user's local machine.
  - **Internet Mail Access Protocol**: Allows user to store and manipulate messages on the same machine.

World Wide Web (www)

- Created by Tim Berners-Lee. Combined Internet technology with the concept of linked-documents: **hypertext**.

- WWW started as a hypertext document format for embedding **hyperlinks** to other documents, a protocol for transferring *hypertext* across the network.

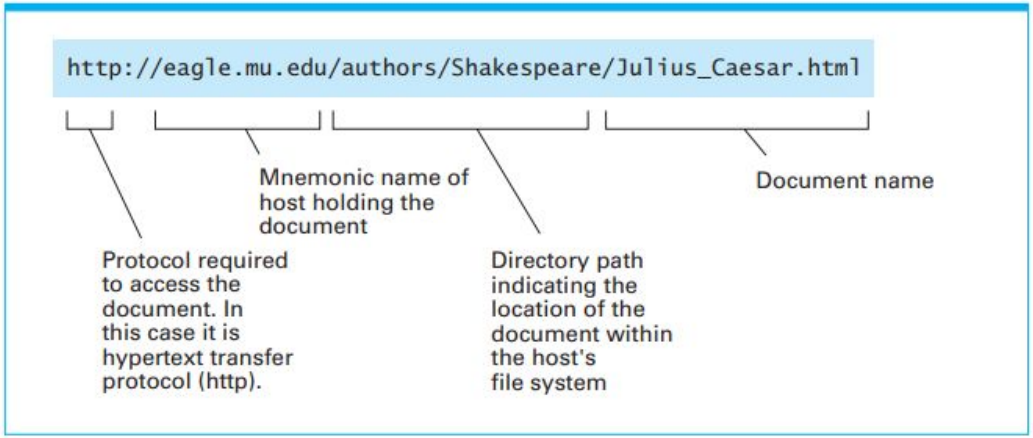
Implementation

- Software that allow users to access these *hypertext* on the Internet fall into two categories:
  - Browsers and webservers
- A browser is an application residing in the users computer. It has the ability of interpreting several languages such as HTML, CSS and JS.
  - Firefox, Safari, Google Chrome, etc.
- A webserver is an application storing the documents which the user wants to access.
  - There is a lot more about webservers which is not in the scope of this lecture.
- Hypertext documents are generally transferred between browsers and webserved using a protocol called HTTP.

URL (Uniform Resource Locator)

- URL has the information needed for browser to contact the proper server and request the *desired* document.

Figure 4.8 A typical URL



HTML (Hypertext Markup Language)

- It is a text file with *tags*. These tags describe how the document should appear on a display screen. How the links, images and lists are going to be used.
- In a webpage, you can right click and see the source of the HTML.
- XML (eXtensible Markup Language)
  - HTML is a part of XML.
- XML was designed to store and transport data. However, since the tags are reused, different languages emerged such as Json, YAML, etc.

HTML & XML

- In HTML there are pre-defined tags.
- In XML you can generate as many tags as you want.
- Browsers generate output from HTML files.

```
<catalog>
        <book>MyBook
        </book>
</catalog>
```

```
<FORM>
        <input type=text>
</form>
```

XML vs Json

```
{
    "sessionStart": "16-03-18-12-33-09",
    "sessionEnd": "16-03-18-12-33-12",
    "mapName": "TestMap",
    "logSections": [{
        "sector": {
            "x": 2.0,
            "y": -1.0,
            "z": 0.0
        },
        "logLines": [{
            "time": 37.84491729736328,
            "state": 0,
            "action": 1,
            "playerPosition": {
                "x": 24.560218811035158,
                "y": -8.940696716308594e-8,
                "z": 3.3498525619506838
            },
            "cameraRotation": {
                "x": 0.24549755454063416,
                "y": 0.017123013734817506,
                "z": 0.031348951160907748,
                "w": -0.9687389135360718
            },
    ...
```

```
<?xml version="1.0" encoding="UTF-8" ?>
<root>
    <sessionStart>16-03-18-12-33-09</sessionStart>
    <sessionEnd>16-03-18-12-33-12</sessionEnd>
    <mapName>TestMap</mapName>
    <logSections>
        <sector>
            <x>2</x>
            <y>-1</y>
            <z>0</z>
        </sector>
        <logLines>
            <time>37.84491729736328</time>
            <state>0</state>
            <action>1</action>
            <playerPosition>
                <x>24.560218811035156</x>
                <y>-8.940696716308594e-8</y>
                <z>3.3498525619506836</z>
            </playerPosition>
            <cameraRotation>
                <x>0.24549755454063416</x>
                <y>0.017123013734817505</y>
                <z>0.031348951160907745</z>
                <w>-0.9687389135360718</w>
            </cameraRotation>
    ...
```

Client side vs Server Side

- Applications can either run on client side or server side.
  - Client side: Javascript apps
  - Server side: PHP, ASP.NET, etc.
- Example:
  - Let's think of a webpage and we want to login. That is a form in an HTML page. You fill out the information and send that information to a remote computer, to a *server*. Up until that point, everything happens in client side. When you send the information, the information is processed in the *server side*. For that, you need a *webserver* (nginx, apache) and a programming language to control webservers and database servers, etc.

Internet Protocols

- In the Internet, transferring information between parties is accomplished by using a hierarchy of software units.
- Example:
  - Let's say you want to send a gift to your friend in Ankara. You buy your gift and write the *address* of your friend. You give it to a shipping company.
  - That company checks the *address* and puts similar shipments in a container.
  - Container are put onto a truck and goes to Ankara.
  - Truck and the container is unloaded.
  - Cargo checks the *address* and takes it to your friend.

Internet Protocols (cont.)

- There are three-levels.
  - User level (You and your friend)
  - Shipment company
  - Truck company
- Each level has its own operations. They all have origin and destination.
- Internet is similar to this. There are several layers and instead of businesses, we talk about software routines.
  - Application Layer
  - Transport Layer
  - Network Layer
  - Link Layer

Internet Protocols (cont.)

- Message is originated in this layer.
- It goes down to **transport** and **network** layers.
- There, message is prepared for transmission and sent to **link** layer.
- Message is sent and the other party receives it at link layer.
- On the other side, same things happen in reverse.

Application Layer

- Application layer uses the transport layer just as we use a shipping company.
- Similar to us writing *addresses* to our shipments, application layer should provide an address. For that reason, app layer can use services of name servers to translate addresses to IP addresses.

Transport Layer

- Accepts messages from application layer.
- Ensures that messages are properly formatted for transmission. For this;
  - Divides long messages into small segments
  - Adds sequence numbers to small segments so that they can be reassembled in the destination.
  - Creates **packets**.
- After doing those, it sends these packets to **network layer**.

Network Layer

- Packets are treated as individual, unrelated messages until they reach the *transport layer* in the destination.
  - Therefore related messages can use different routes while going to the destination.
- Network layer decides which direction a packet should be sent. It maintains the router's *forwarding table* and use that to determine the direction. After that, the packet is handed to *link layer* for the actual transmission.

Link layer

- Has the responsibility of transferring the packet. It deals with the communication details particular to the network.
  - For ethernet, link layer applies CSMA/CD.
  - For WiFi, it applies CSMA/CA.
- When the packet is sent, it is sent to the *link layer* of the destination. There, it gives the packet to its *network layer* and there it checks the *forwarding table* to see the final destination of the packet. If that computer is not the final destination, it is sent back to *link layer* and forwarded to the next destination. In this manner, each packet hops from machine to machine on its way to final destination.

Completing the process

- Remember that the package hops. That is determined at the *network layer*. When the package is at the final destination, instead of sending the packet back to *link layer;* the network layer will send the package up to the *transport layer*.
- Transport layer extracts the message segments and reconstructs the original message according to the *sequence numbers* that were provided. After assembling the message, it is handed to the *application layer*.

Port numbers

- When the transport layer hands the re-constructed message to *application layer* it uses unique **port numbers**. That port number is assigned to the message's address, so that the transport layer knows which port to use.

- Users are generally not concerned with port numbers because common applications generally have universally accepted port numbers. There are 65535 port numbers where 0-1023 are well-known; 1024-49151 are registered and 49152-65535 can be used dynamically by applications.
  - Web: 80
  - SMTP: 25
  - FTP:21

Summary

- Communication over the Internet uses 4 layers of software.
  - **Application layer:** Deals with messages from the application's point of view.
  - **Transport layer:** Converts messages into segments that are compatible with Internet and reassembles them before delivering into the application layer (at destination).
  - **Network layer:** Directs segments through the Internet.
  - **Link layer:** Handles the actual transmission of the segments.

TCP/IP

- OSI model is a standard.
  - 7 levels instead of 4.
  - Defacto standard protocol for the Internet is the 4-layer model, not the OSI. It is generally used as an educational tool.
- TCP/IP protocol suite is a collection of protocol standards used by the Internet to implement the 4-level communication hierarchy. There are two protocols in this collection:
  - TCP: Transmission Control Protocol
  - IP: Internet Protocol

TCP vs UDP

- **TCP** defines a version of the **transport layer**. There is also **UDP** (User Datagram Protocol).
  - You can choose one of this protocols just as you choose a shipping company
  - They both do the same thing but they do it differently.
- TCP is connection-oriented.
  - Before sending a message, transport layer at the sender sends a message to the destination transport layer that a message is about to be sent. Then, waits for an *acknowledgement* (ACK) before sending the message. It establishes a connection before sending a message.
- UDP is connection-less.
  - UDP does not establish any connection and just sends a message to the given address and forgets about it. Maybe the destination is not even online, it doesn't care.

TCP vs UDP

- TCP is reliable.
  - Origin and the destination work together by using *acknowledgments* and *packet retransmissions* to assure all segments are transferred to the destination.
- UDP is not reliable.
  - There is no acknowledgement or retransmissions.
- TCP has **flow control** and **congestion control** in order to control the traffic. The origin can slow down and adjust the transmission rate so the destination and the network is not overwhelmed.

Why use UDP?

- Compared to TCP, UDP looks very weak. However, UDP can be efficient and faster. It depends on what we use it for.
- Where packet loss is not important and/or re-transmission is meaningless, go for UDP.
  - Video streaming, VoIP.
  - Online gaming
  - DNS

IP

- Just as TCP is for transport layer, IP is for *network layer*.
- This layer is for **forwarding** and **routing**; which involves updating the *forwarding table* to reflect changes.
- Much of the IP standard deals with protocols used for communication among neighboring layers as the routing information.
    - Let's say the router is broke down or there is congestion. Traffic should be routed around this blockage. However, the packet should not be forwarded infinitely. That is each time network layer prepares a packet, it appends a value called **hop count**. Each time the layer forwards a packet, hop count decrement by 1. The hop count is 64.

Security

- When you are connected to a network, you are open for attacks and vulnerabilities.
- There are multiple ways that a computer system / network can be attacked. These can be done by using *malicious software* (**malware**). These can run on the host itself or can attack from a distance.

Malware

- Virus
  - Software that infects a computer by inserting itself into programs which are already in the computer. When that software runs, virus also runs.
- Worm
  - Autonomous program which can replicate itself in the computer and through the network.
- Trojan horse
  - Program that enters the computer disguised as a desirable program, game or some utility package. Generally installed by the user willingly by mistaking it as a useful program.
- Spyware
  - Collects information about activities at the computer which it’s executed and generally gives back information to the owner of the spyware.

DoS (Denial of Service)

- Attacker sends a large number of messages over a brief period of time in order to overload the target computer so that it is no longer available.

- These can also be done by employing **botnet**s. Some malware on computers make it possible for attackers to control them in order to use for DoS attacks.

MITM (Man in the middle attack)

- When an adversary comes between the communication and reads / changes the messages.
- This can be done in several ways:
  - Eavesdropping: Listening in to the communication. (sniffing, snooping)
  - Spoofing: Acting as if you are someone else.

Protection

- **Firewalls:** Applications installed to filter the traffic passing through a point in the network.
  - A firewall can be installed at the gateway to filter messages going in and out.
  - They can be configured in order to block certain incoming/outgoing messages.
- **Antivirus Software**
  - Detect and remove known viruses or malwares.