Introduction to Internet & Web

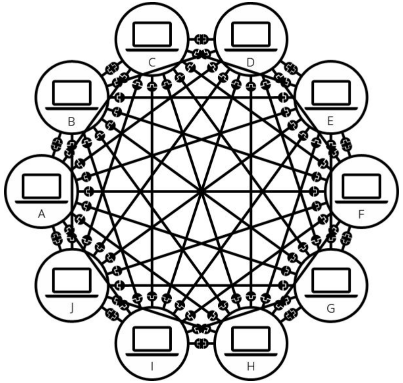
Tuğberk Kocatekin
tugberkkocatekin@arel.edu.tr

Internet

- Backbone of the web
- Technical infrastructure that makes web possible
- A large network of computers which communicate together.
  - Began as a US-army funded research project
- Resources
  - How the internet works in 5 minutes (5 min)
  - How does the internet work? (9 min)

Internet

- When two (or more) computers want to communicate we **link** them
  - Either physically
    - With ethernet cable
  - Or wirelessly
    - Wifi, bluetooth, etc.
- When multiple computers, the communications



IP Address (Internet Protocol Address)

- Unique numerical identifier assigned to a device connected to a computer network.
- Public IP
  - The IP assigned by the ISP (Internet Service Provider).
- Private IP
  - Used within local networks (192.168.1.1, 10.0.0.1)
- Static IP
  - Remains the same over time
- Dynamic IP
  - Changes periodically. Generally everytime you connect, you have a new IP address.

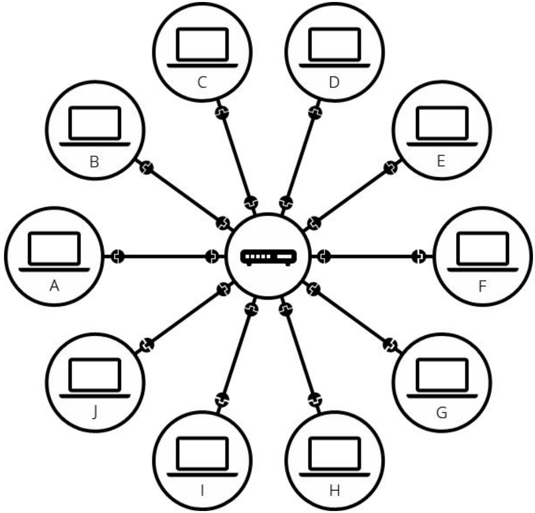
IPv4 and IPv6

- IPv4 (version 4)
  - 32-bit address
  - Four decimal numbers (192.168.1.1)
  - Supports approx 4.3b addresses
- IPv6 (version 6)
  - 128-bit address
  - Hexadecimal format with colons (2001:db8:ff00:42:8329)
  - Supports trillions of addresses
- Why? There are lots of machines now and IPv4 is not enough.

NAT: Network Address Translation

- Method allowing multiple devices on a local network to share a single public IP address.
  - e.g. in your home, all devices have different **private ip** but the same **public ip**.
    - Can learn your public ip via internet
    - Can learn your private ip via shell commands.
- When a device wants to communicate with outside world (website) NAT replaces the private IP with its own public IP before forwarding the request.
- Router keeps track of the request, so when the response comes, it knows which device to send back.

- To solve it, we connect each computer to a special tiny computer: **router**
  - It makes sure that a message from the sender arrives at the destination.
- If A wants to send a message to B
  - It sends the message to the router, and the router forwards the message to B.
  - Can check it with tracert command on Windows.
  - `traceroute` in Linux.
- With a router, instead of connecting every computer to each other (complete graph); we connect every computer to the router.



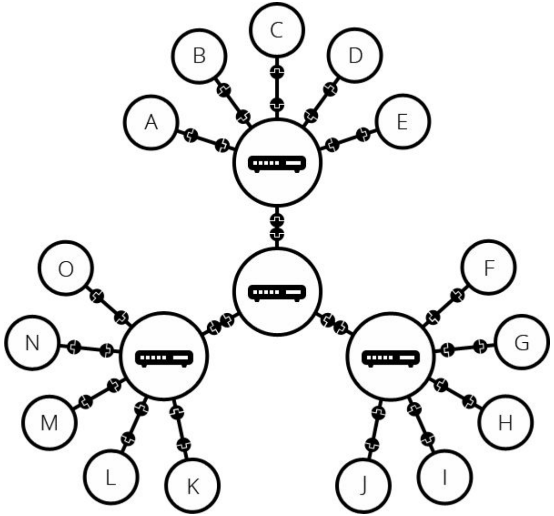
Router

- A networking device that directs data packets between computer networks.
    - When we send a request, that request is sent via **packets**.
    - These packets are addressed to a **destination IP**.
    - Router examines them and acts accordingly.
        - If within network, sends it.
        - If outside, it checks the **routing table**.

- There are different types of routers.
- In summary:
  - Router receives an IP packet.
  - Checks the destination IP
  - Checks the routing table to determine the next hop.
  - Forwards the packet.
- There are dynamic routing stuff such as
  - RIP, OSPF, BGP
  - Those interested in networks should research

A network of networks

- What happens when we want to connect hundreds, thousands, billions of computers?
  - A single router is not enough.
- Remember, a router is like a *tiny* computer.
  - Can we connect routers to each other?

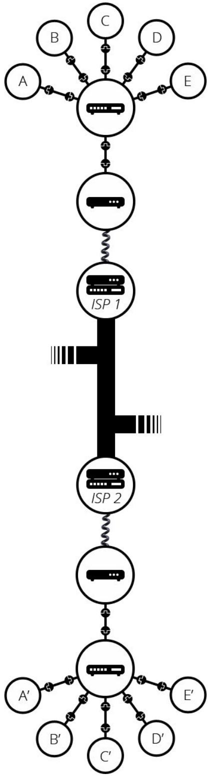


Problem

- How are we going to connect the routers together?
    - Home A does not have a cable connecting itself to Home B.
    - Or maybe does it?
- We already have a connection:
    - Electric cables
    - Phone cables
- Telephone infrastructure connects our house with almost anyone in the world via phone lines
    - If we can connect our network to telephone infrastructure, we can connect everyone!
- To do that, we need a **modem**
    - Modulator, demodulator
- It turns the information from the network into information which can be managed by the telephone infrastructure.

How to send?

- We are all connected via phone line.
- How can my network send information to the other network?
    - We need an Internet Service Provider (ISP)
    - A company with special routers that are all linked together and also can access **other** ISPs routers.
- Therefore, our message is carried through the network of ISP networks to the destination networks.



How to find a computer?

- When you are sending a message, you need to tell **who** are you sending it to.
    - Every computer linked to a network has a unique address.
        - IP Address
- It is hard for us to remember IP addresses.
    - We can remember names though.
    - We can alias IP addresses with a human-readable name called **domain name**.
        - google.com
        - Ping to learn an ip address of a domain name

Internet and the web

- In the internet, we usually use a domain name to access a webpage.
    - internet != web
- Internet is a technical infrastructure which allows computers to be connected.
    - Some of these are **web servers**
- Internet is an **infrastructure**, web is a **service** built on top of it.
- There are also other services such as email, IRC, etc.

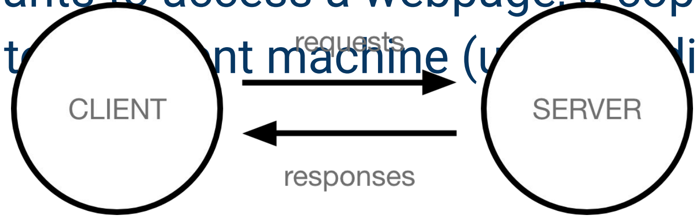
Intranet

- Intranets are **private** networks that are restricted to members of an organization.
    - Used to provide a portal for members to securely access shared resources.
    - e.g. An organization’s intranet might host web pages for sharing department or team information, shared drives to manage documents, portal for business administrations, wikis, etc.

web

Client & Server

- Computers connected to the internet are called **clients** and **servers**.
- Clients are web user’s internet-connected devices.
  - Such as our computers, mobile phones etc.
- Servers are computers that store web pages, sites or apps.
  - When a client wants to access a webpage, a copy of it is downloaded from the server t nt machine (u isplayed in the user’s web browser.



Connection

- It is not that easy though.
  - On the way, we must make some stops.
- Internet connection
  - First, we need to be online to access the web server.
- TCP/IP
  - We need to use a communication protocol to define how data should travel across the internet.
- DNS
  - Domain name system. When we type the address, we should ask someplace the real IP address for that website. Otherwise, it cannot send HTTP messages.
- HTTP
  - Hypertext Transfer Protocol. An application protocol that defines a language for clients and servers to speak to each other.

How?

1. The browser goes to DNS server, finds the IP address of the server where the website lives on.
2. Browser sends an HTTP request message to the server, asking it to send a copy of the website. These messages are all sent across the internet connection using TCP/IP.
3. If the server approves the client request, they send a “200 OK” message, which means “ok”. After that, starts sending the website’s files to browser as a series of small chunks called *data packets*.
4. Browser assembles the small chunks into a complete web page and displays it to you.

Order

- When browsers send requests, HTML files often contain `<link>` elements referencing CSS styles and `<script>`elements referencing Javascript scripts.
- Important to know the order in which those files are **parsed** by the browser while loading.

How?

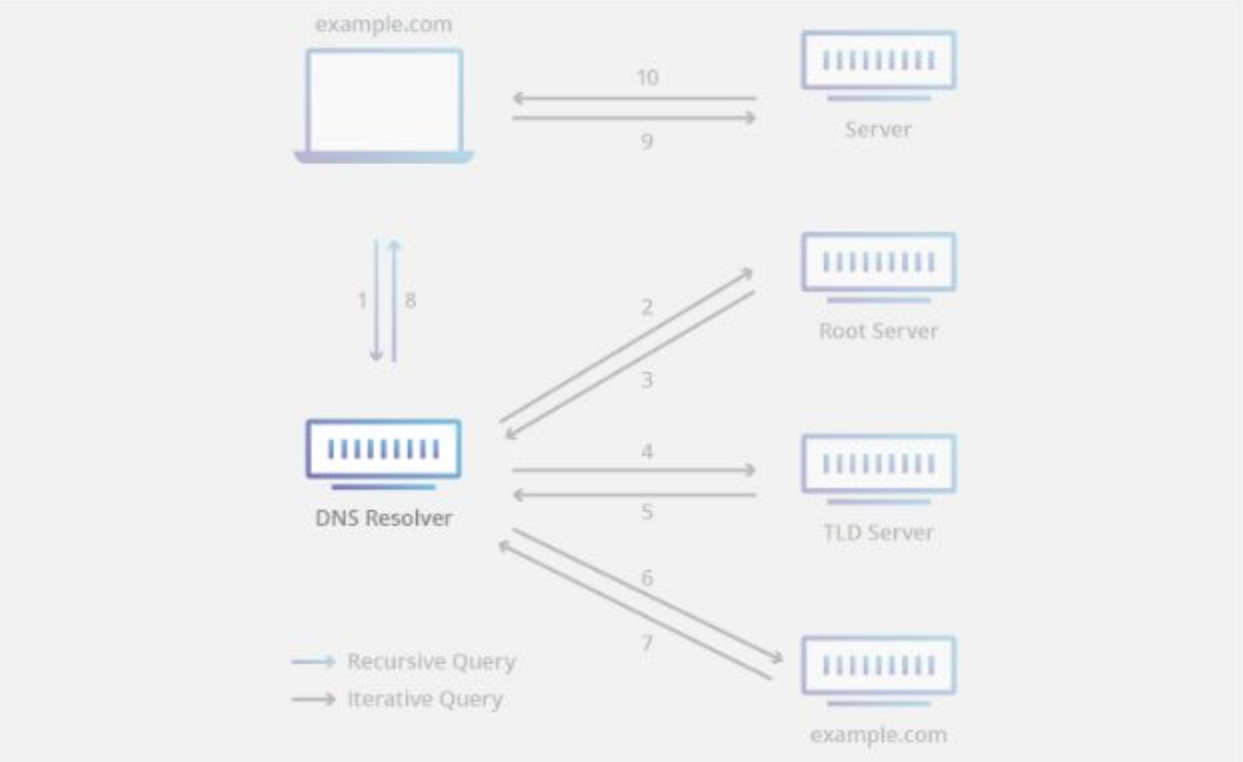
- Browser first parses the HTML file, and recognize link and scripts element references.
- As it parses the HTML, sends requests back to the server for any CSS files it found from link and any JS files found from scripts elements, and then parses CSS and JS.
- Generates an in-memory DOM tree from the parsed HTML, generates an in-memory CSSOM structure from the parsed CSS and compiles and executes the parsed Javascript.
- As it generates those, a visual representation of the page is painted to the screen. User can see the content and begin to interact with it.
  - We will see how these work in detail later on.

DNS

- Hard to remember IP addresses
  - Each IP address represents a unique machine
- DNS system uses special servers that match up a web address you type into your browser to real IP address.

DNS

- All DNS servers fall into one of four categories.
  - Recursive resolver
  - Root nameservers
  - TLD nameservers
  - Authoritative nameservers
- Usually work together


example.com
10
9
Server
1
8
2
3
Root Server
DNS Resolver
4
5
TLD Server
6
7
example.com
Recursive Query
Iterative Query

Recursive Resolver

- First stop is a *recursive resolver* (DNS recursor)
- Acts as a middleman between the client and DNS nameserver.
- Either responds with cached data, or send a request to root nameserver, followed by another request to TLD nameserver and lastly authoritative nameserver.
    - Lastly, gets the IP address and returns it.
- It will cache received information.
    - If someone else requests that, it will give the cached return.

DNS root nameserver

- There are 13 types of DNS root nameservers.
  - These are known to every recursive resolver.
- Accepts a RR query which includes a domain name.
  - Responds by directing RR to a TLD nameserver based on the extension (com, net, org, etc)
- Usually overseen by ICANN.
  - Internet Corporation for Assigned Names and Numbers (A nonprofit)

TLD nameserver

- Maintains information for all domain names with the same extension
  - com, net, org, etc.
- A .com nameserver contains information for every website that ends with *com*.
- TLD nameserver responds by pointing to *authoritative nameserver*.
- Managed by IANA
  - Internet Assigned Numbers Authority
  - A branch of ICANN
- IANA breaks TLD into 2 main groups:
  - Generic top-level domains: Not country specific (com, org, net, etc)
  - Country code top-level domains: Country specific (uk, us, tr, jp, etc)

Authoritative nameserver

- Last part of the DNS lookup
- Contains information specific to the domain name it servers
- Can provide RR with the IP address of the server found in **DNS A record**,
- or if the domain has a **CNAME record** (alias) it will return an alias domain
  - If that is the case, RR will have to perform a whole new DNS lookup to get the DNS A record.

- *A record maps a hostname to one or more IP addresses*
- *CNAME record maps a hostname to another hostname.*

web

Why web?

- Multiplatform
  - Can reach from everywhere.
  - Mobile phone, Linux, etc.
  - If you have a browser, you can use it.
- Attractive market

Overview of modern web tech

- Browsers
  - Software people use to consume the web
  - Firefox, Opera, Chrome, Edge, Safari, etc.
  - These programs can interpret Javascript, HTML and CSS.
- HTTP
  - Hypertext Transfer Protocol
  - A messaging protocol that allows web browsers to communicate with web servers.
- HTML, CSS, JS
  - HTML: A markup language consisting of different elements.
  - CSS: Rule-based language to apply styles to HTML.
  - JS: Programming language to add interactivity to websites: dynamic style, fetching updates, complex 3d graphics, etc.

Front-end (client-side)

- Client and server. Client is the **browser** and the server is the **webserver**.
  - Client can also be an application.
- Client sends **requests** and the server gives back **responses**
- Front-end is mainly about **presentation**
  - How things look.
  - Images, content and structure.
- Written by HTML, CSS and JS.

Back-end (server-side)

- Back-end is about *how things work*.
  - What are you going to present in the front-end?
- Business Logic and Data are on the back-end.
  - We have the database and server-side code here.
- When a user wants to create/read/update/delete something, all these are sent to back-end.
- Languages:
  - Any programming language can be used as back-end.
  - Some well known programming languages and web frameworks are:
    - Python (Flask, Django)
    - PHP (Laravel)
    - Java (Spring, Spring Boot)
    - C# (dotnet)
    - Go
    - …

Mobile development

- Kind of front-end.
  - However, there are additional stuff to take care here.
- Logic is front-end though, because it is for *presentation* again.
- Languages and Frameworks:
  - IOS: Swift
  - Android: Kotlin, Java
  - There are also React Native and Flutter, which can be compiled into native applications for both IOS and Android.

Front-end

- In a broad sense, frontend can be divided into two main parts:
- UI (User interface)
    - CSS is very important here.
    - You need to be able to create an image with CSS (or the native lang. for mobile)
    - For example: https://www.youtube.com/watch?v=P_qjGB6W2To
- Javascript (Interactivity & Logic)
    - DOM manipulation
    - Event handling
    - API communication
    - State management

Javascript

- Originally built as a client side programming language
  - EcmaScript
  - Nothing to do with Java.
- When you have JS, it only works on the person who is viewing the page.
  - Nothing to do with the server.
- Lots of libraries and frameworks for Javascript.
  - JQuery and Coffeescript was very common.
  - Modern: React, Vue (some angular)

Javascript

- Knowing Vanilla JS is the best for starters.
  - You need to understand why we use those libraries.
- Typescript
  - Javascript is not a very good language and no one expected it to be this popular.
  - Web runs on JS.
  - However, for large projects it is not good enough (weakly-typed etc)
  - Typescript is a strongly typed programming language which compiles to Javascript.
    - Created by Microsoft.
    - VSCode was written in it.

Backend

- Any programming language with a web framework can be used in the back-end.
    - Now, even Javascript (a client-side language) can be used on the backend thanks to NodeJS.
- We do interactions with file system and database.
- Input validation, sanitization
- Database interactions
- Security etc.
- All are done on backend.
- Ideally, we don’t do **any** business logic in the frontend. It is just responsible for sending/receiving data.