

# What is the Internet

Internet: networked computers

Each computer has an address - like mail or deliveries

- local traffic goes to a local hub
- local hub sends stuff for outside to a higher hub
- until a hub has the destination "inside"
- sends to the right lower hub

There is a lot of redundancy, not just one path

Internet was a US Defense Dept exercise, to survive interruptions. (Like nukes)

# Internet Delivery

Internet Protocol **(IP) addresses** are not for humans

Domain Name System (**DNS**) is how to turn a name into an IP address

Humans use names like `www.google.com`

Each "dot" separates a level in reverse

- ".com" knows all the domains inside it
- "google.com" knows all the domains inside it

Three is common, but more depth is supported

# Traffic Protocols

DNS lookups are one kind of traffic over the Internet

There are many others. Examples:

- Email
- Database
- Many online games
- Web

# Internet is down!

"The internet is down" - Probably not

Could be

- their local internet routing is down
- some local web issue is down
- a provider used by many (example: Amazon) is broken, taking a lot of web sites down

But it's very rare for any notable portion of the Internet itself to be down.

# **Don't be that person**

But don't tell be that person

The one that tells them they are wrong

We all know what they mean

# What is a server?

This is really hard <https://jvns.ca/blog/2019/12/26/whats-a-server/>

A server can be

- a program that responds to requests
- the machine that runs that program
- a virtual machine running on a physical machine

A server can run a server running a server

Generally, for this course a server is the program

# Web Request Response

For the Web

- A client (usually a browser) makes a **request**
- A server gets the request and gives a **response**

Each request gets one response

Only with special preparations and moving outside basic web can you get anything else

## **Bad cases on the web**

- A stock-ticker app that is told when stocks change
- A weather app that is told when weather changes

Both of these worked very poorly on the early web.

You couldn't be told of changes, so you had to ask repeatedly and frequently.

That is a lot of pointless traffic.



# **Web was for linking scientific papers**

- Text
- linking back and forth
- readable on different platforms

Not WYSIWYG

- H - Hyper
- T - Text
- T - Transfer
- P - Protocol

# **Web provided unique benefits**

- Common port (80 for HTTP, 443 for HTTPS)
  - Meant once you got through a firewall, you had access to everything
- Not tied to a particular appearance
- Tolerant of bugs/typos
- Human readable
- Searchable

# Web was searchable

- A program (crawler, spider, bot) reads a page
- Makes a list of all the links on that page
  - Adds any new links to list of pages to crawl
- Reads the text of the page and save info (index)
- Repeats with next link on list

Users go to site with index, enter search terms

- Program gets search terms
- Program uses index to get matches
- User sees list of matching links

# **Web is stateless**

Each request is considered by itself

- without respect to previous requests

Can go straight to any link

- without passing through others

# What about requiring login?

Isn't login stateful? (not stateless)

Yes and no. The *protocol* doesn't enforce that.

- Request comes in
- Based on info IN REQUEST, server decides:
  - send you elsewhere (redirect)
  - show you alternate content (login screen)
  - show you the requested material (content)

Request must contain the info to let server decide

There is no state in the **handling of** the request

# Browser Rendering

Not every web client is a browser

- Ex: a spider is a web client and not a browser

Browsers decide what to do with content

- Often this means **rendering** an HTML page

But it could be:

- Displaying an image
- Playing a sound file
- Showing a PDF
- Saving a file

# What is a URL?

A Uniform Resource Locator (**URL**)

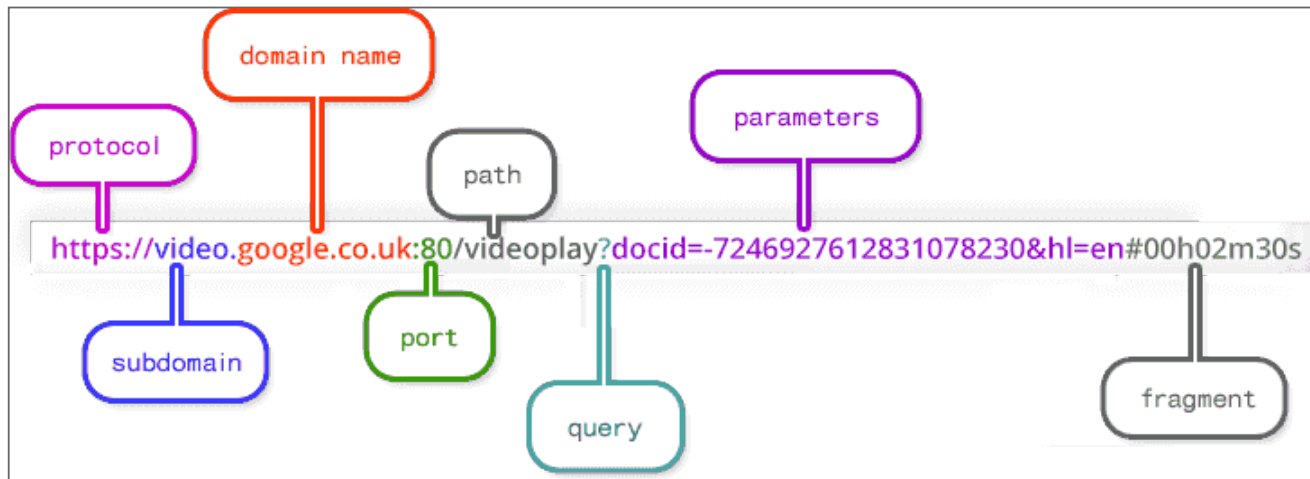
- Where something on the internet can be found

**<http://catalog.northeastern.edu/graduate/engineering/multidisciplinary-systems-msis/>**

Often called a **Web Address**

- But URLs are not limited to the web

# Parts of a URL



From <https://doepud.co.uk/blog/anatomy-of-a-url>



# Fully Qualified

A URL with all the parts is known as "Fully-Qualified"

Without all the parts, it might just be a path

That path might be "absolute" or "relative"

# Absolute Path

**Absolute Paths** are different paths on the same server

- Absolute Paths always begin with `/`

Absolute path is taken from some **"root"** of the server

- This is NOT the "root" of the file system

The **document root** is how the web server treats requests for the "root"

```

```

# Relative Path

A **Relative Path** is based on path of **current page**

- Relative Paths do NOT begin with /
- ``
- ``
- ``

# Paths

Two pages:

- `http://example.com/foo/index.html`
- `http://example.com/bar/images/index.html`

What is different when the urls below are loaded?

- ``
- ``
- ``
- ``
- ``

# Paths - Answered

`http://example.com/foo/index.html` VS `http://example.com/bar/images/index.html`

- ``
  - `/foo/cat.png` VS `/bar/images/cat.png`
- ``
  - `/foo/images/cat.png` VS `/bar/images/images/cat.png`
- ``
  - Both `/images/cat.png` (absolute)

# Paths - More Answers

`http://example.com/foo/index.html` VS `http://example.com/bar/images/index.html`

- ``
  - `..` means "go up one directory"
    - Can't go earlier than root/document root
  - `/images/cat.png` VS `/bar/images/cat.png`
- ``
  - `/foo/cat.png` VS `/bar/images/cat.png`

# Summary - Part 1

- Internet vs Web
- Internet routing
- DNS/Domain names/subdomains
- Web is request/response
- Web is stateless
- Searching isn't built in
- Searching is easy because stateless

## **Summary - Part 2**

- URLs can be fully qualified or not
- A path can be absolute or relative
- Paths in URLs taken from document root
- Browsers render a page after getting the data
- Not all clients are browsers
- Not all data is rendered