COSC 360
Web Programming

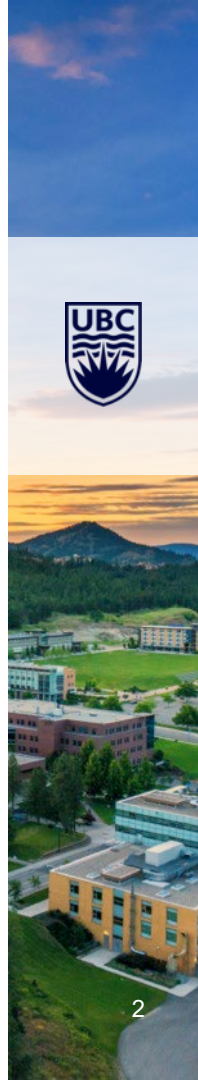
Introduction to Web Development

I respectfully acknowledge that UBC Okanagan is situated on the traditional, unceded and ancestral territory of the Syilx Okanagan Nation

Objectives

At the end of this lecture, students should be able to describe:

- Web development in general
- Fundamental concepts that form the foundation of the Internet
- Hardware and software that support the Internet
- The fundamental protocols that make the web possible
- How the domain name system works
- HTTP
- How browsers and servers work to exchange and interpret HTML



Intro to web development

* A Complicated Ecosystem
* Definitions and History
* The Client-Server Model
* Where is the Internet?

Intro to web development

* **A Complicated Ecosystem**
* Definitions and History
* The Client-Server Model
* Where is the Internet?

A Complicated Ecosystem

Web development is a series of related platforms

The 2 teal platforms represent topics typically understood to be web development

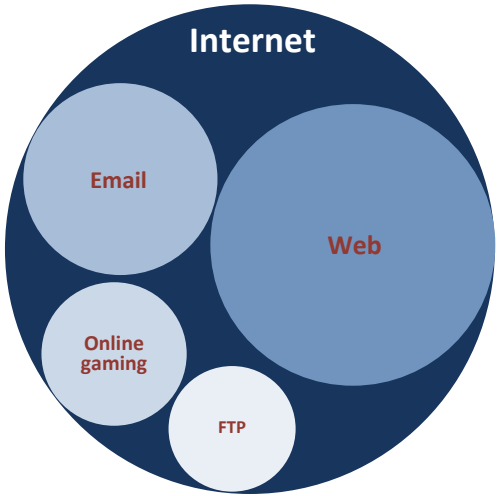
Intro to web development

* A Complicated Ecosystem
* Definitions and History
* The Client-Server Model
* Where is the Internet?

Internet = Web?

The answer is no.

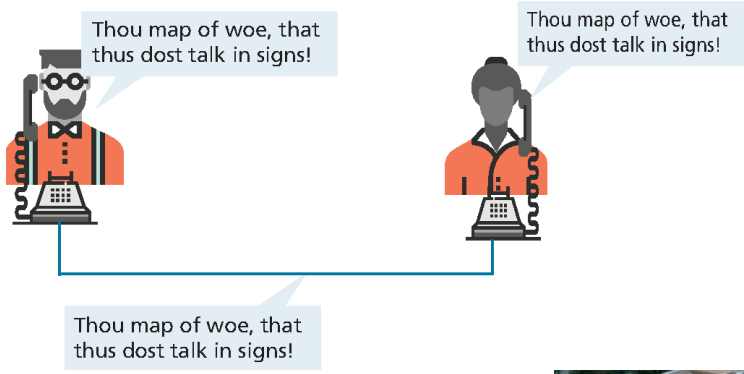
- The World-Wide Web (WWW or simply the Web) is certainly what most people think of when they see the word “internet.”
- But the WWW is only a subset of the Internet.

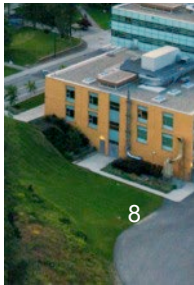


Communication Definitions

Circuit switching in early networking

- In the past, telephone calls were routed through operators who **physically connected** the caller and the receiver by connecting a wire to a switchboard to complete a circuit
- Inefficient use of bandwidth
- Difficult to scale

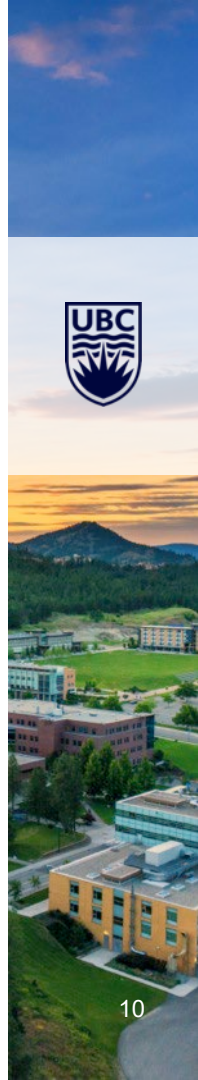




Circuit Switching

Circuit Switching Weaknesses

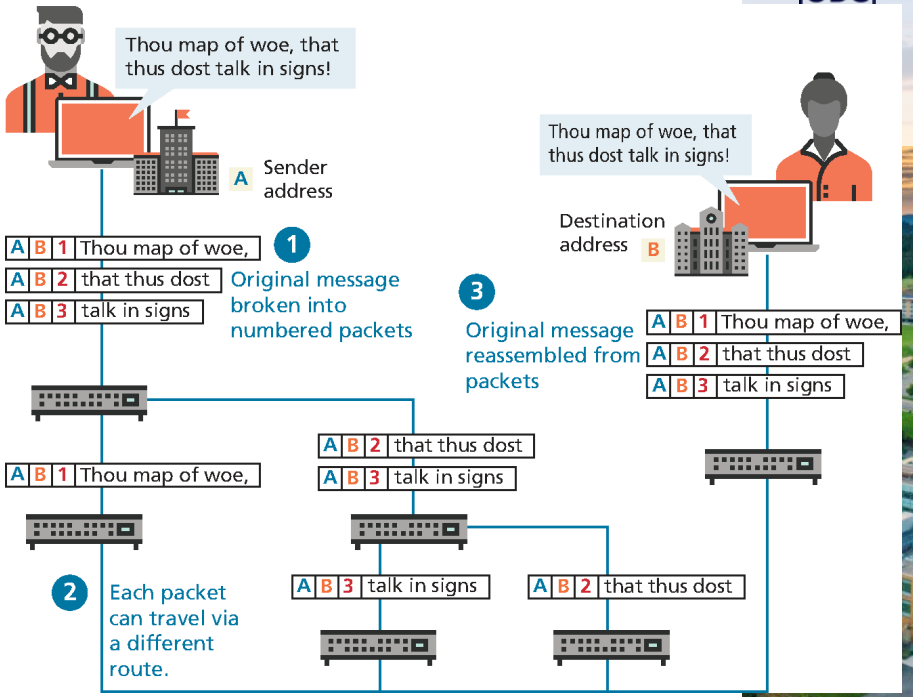
- You must establish a link and maintain a dedicated circuit for the duration of the call
- Difficult to have multiple conversations simultaneously
- Wastes bandwidth since even the silences are transmitted
    - Unused capacity in the network is not being used efficiently



Packet Switching

The beginning of the Internet

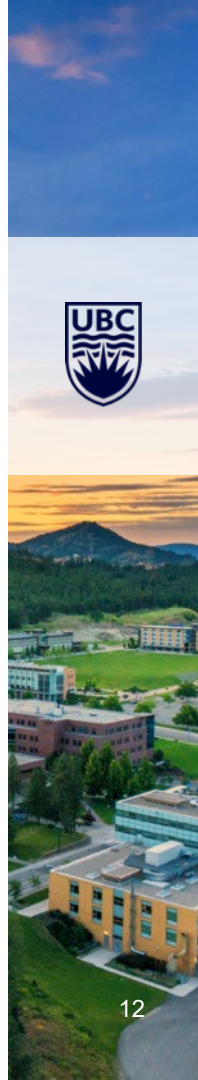
- The research network ARPANET was created in the 1960s
  - ARPANET did not use circuit switching
  - Used **packet switching**
- A packet-switched network does not require a continuous connection
- It splits the messages into smaller chunks called **packets** and routes them to the appropriate place based on the destination address



Packet Switching

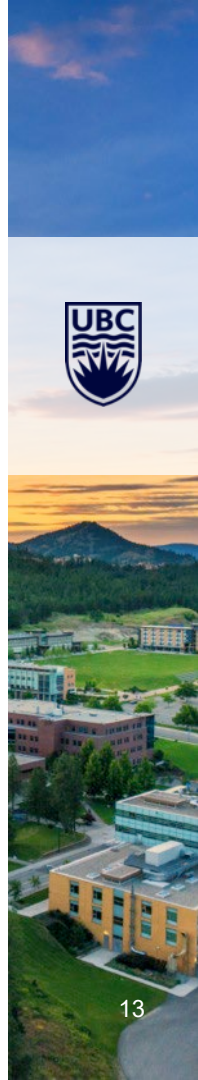
While **packet switching** may seem a more complicated and inefficient approach than **circuit switching**, it is:

- more robust (it is not reliant on a single pathway that may fail) and
- more efficient use of network resources (since a circuit can communicate with data from multiple connections)



Sir Tim Berners-Lee

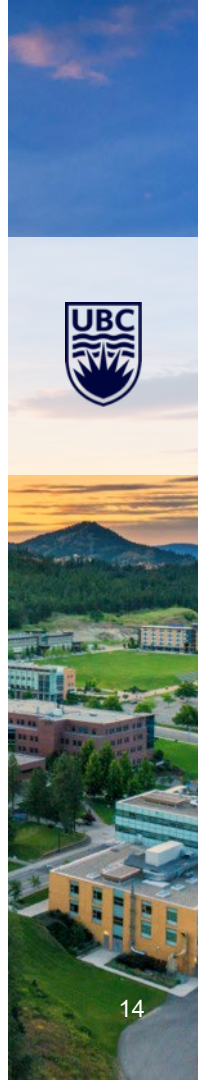
The invention of the WWW is usually attributed to the British Tim Berners-Lee, who, along with the Belgian Robert Cailliau, published a proposal in 1990 for a hypertext system while both were working at CERN (the European Organization for Nuclear Research) in Switzerland.



Core Features of the Web

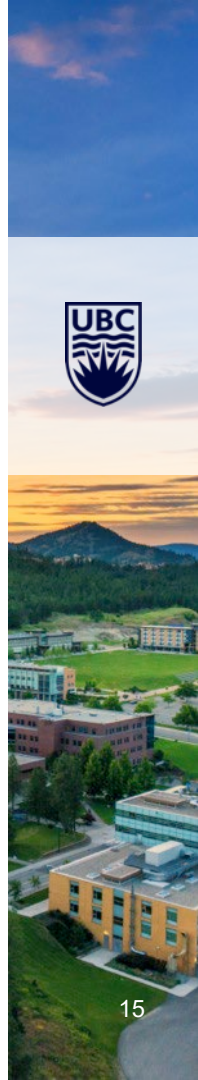
Sr. Tim Berners-Lee developed the main features of the web we know today in 1992:

1. A URL to uniquely identify a resource on the WWW
2. The HTTP protocol to describe how requests and responses operate
3. A software program (later called web server software) that can respond to HTTP requests
4. HTML to publish documents
5. A program (later called a browser) to make HTTP requests from URLs and that can display the HTML it receives



TCP/IP

- To promote the growth and unification of the disparate networks a suite of protocols was invented to unify the networks together
- A **protocol** is the name given to a formal set of publicly available rules that manage data exchange between 2 points
  - Communications protocols allow any two computers to talk to one another, so long as they implement the protocol
- By 1981, new networks built in the US began to adopt the **TCP/IP** (**Transmission Control Protocol / Internet Protocol**) communication model, while older networks were transitioned over to it
  - The TCP/IP defines how devices should transmit data between them and enables communication over networks and large distances

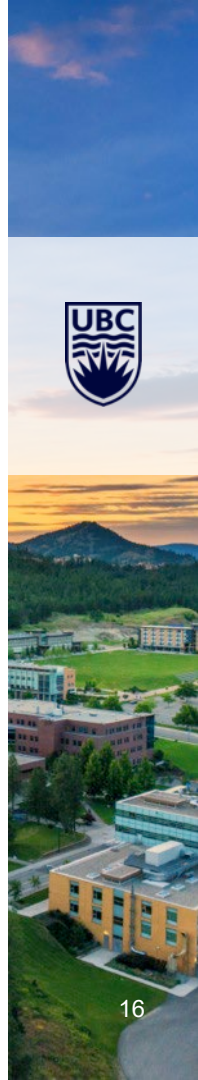


W3C

The World Wide Web Consortium

Also in late 1994, Berners-Lee helped found the **World Wide Web Consortium** (**W3C**), which would soon become the international standards organization that would oversee the growth of the web.

This growth was very much facilitated by the decision of CERN to not patent the work and ideas done by its employee and instead left the web protocols and code-base royalty free.



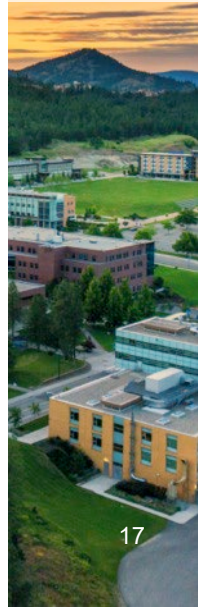
Static Web Sites

In the earliest days of the web, a **webmaster** would publish web pages, and periodically update them.

In those early days, the skills needed to create a web site were pretty basic:

- Knowledge of the HTML markup language and perhaps familiarity with editing and creating images

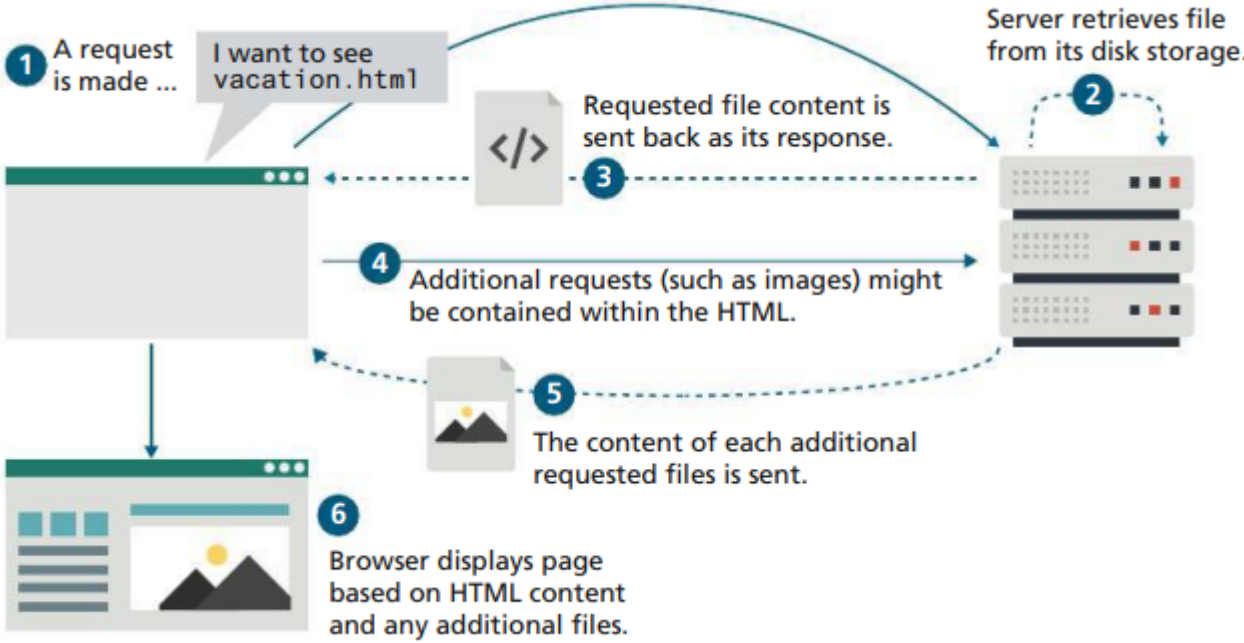
Commonly referred to as a **static web site**, in that it consists only of HTML pages that look identical for all users at all times.



Definitions and History

Static Websites versus

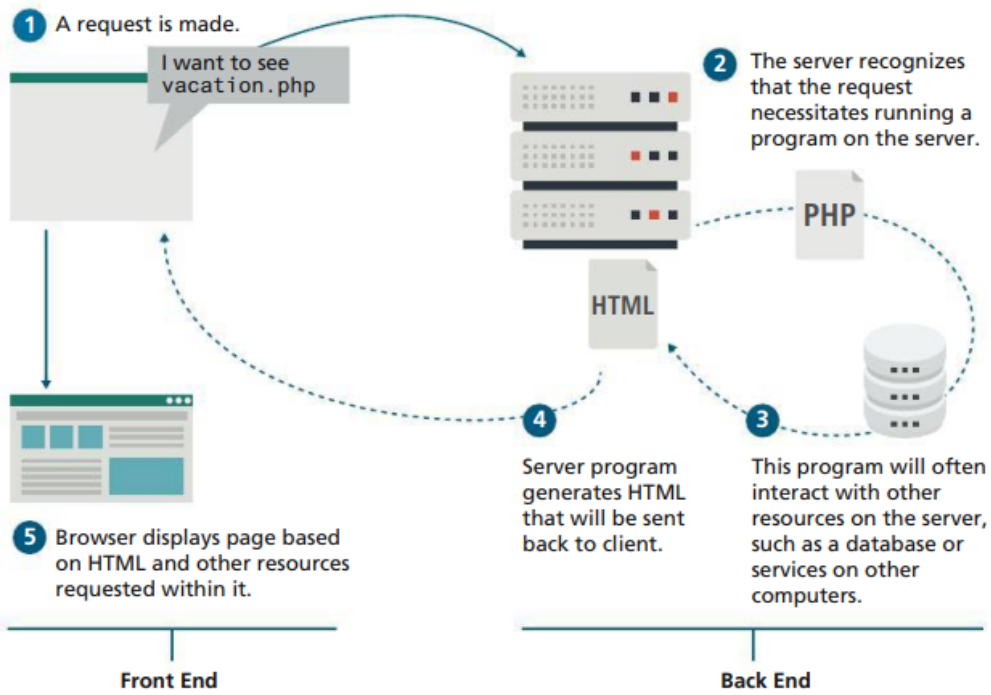
Dynamic Websites



Static website (first generation)

Definitions and History

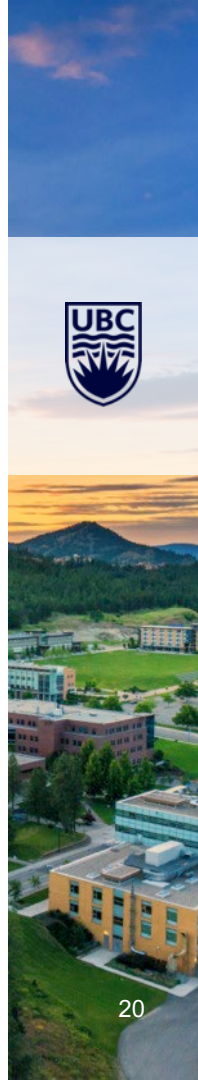
Later, programs running on web servers let websites generate content dynamically. This type of website is called a **dynamic server-side website**



Dynamic Server-Side website

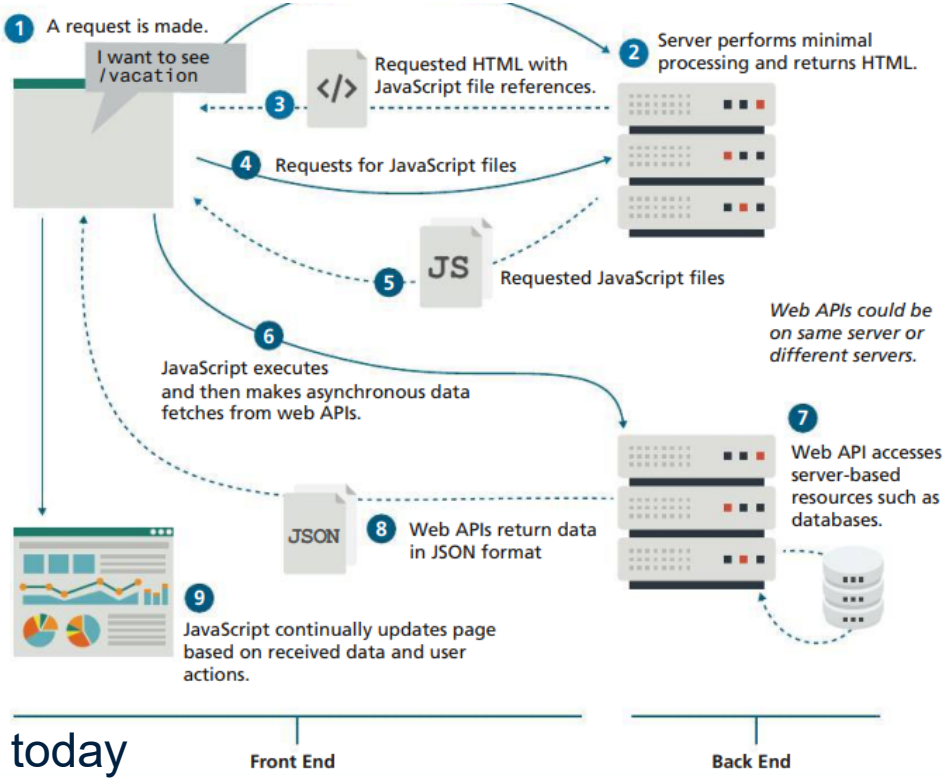
Web 2.0 and Beyond

- In the mid 2000s, a new buzz-word entered the computer lexicon: **web 2.0**
- This term had two meanings, one for users and one for developers
- For the users, Web 2.0 referred to an interactive experience where users could contribute *and* consume web content, thus creating a more user-driven web experience (social media platforms, reviews on e-commerce platforms)
- For software developers, Web 2.0 also referred to a change in the paradigm of how dynamic web sites are created
- Programming logic, which previously existed only on the server, began to migrate to the browser
- This required learning JavaScript, a programming language that runs in the browser, as well as mastering the programming techniques involved in asynchronous communication



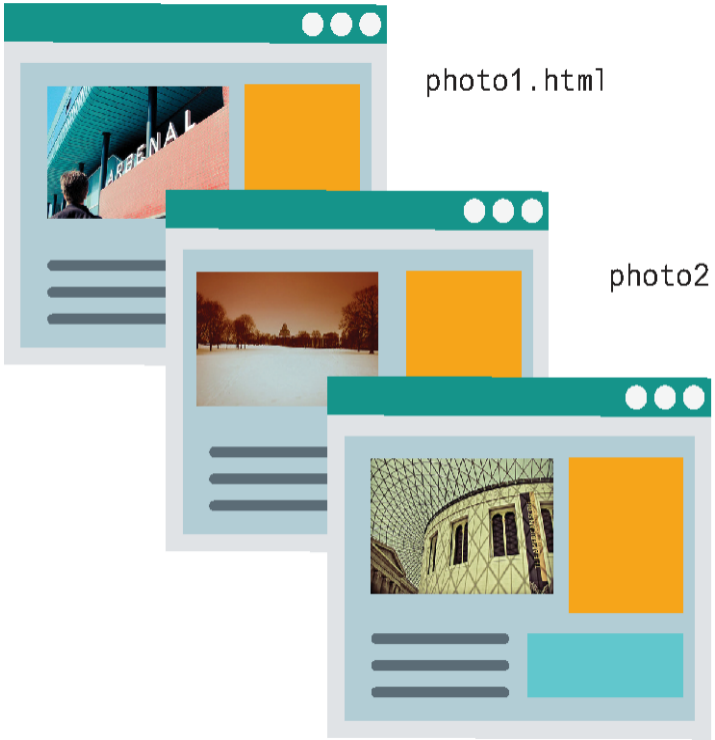
Definitions and History

Static Websites versus Dynamic Websites



Dynamic websites today

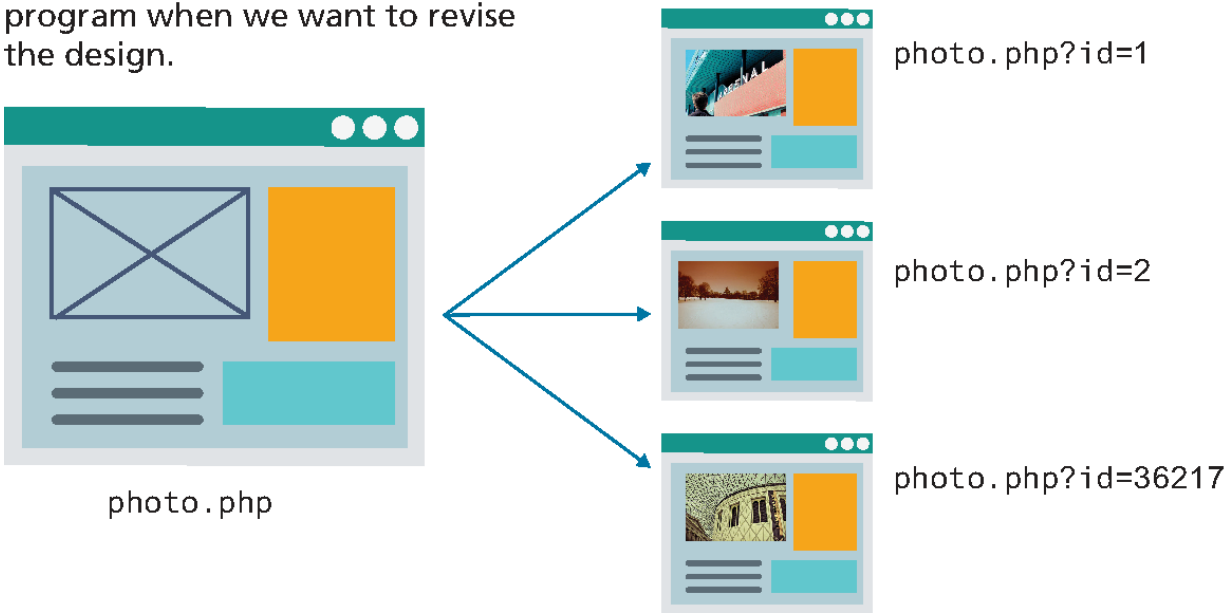
Why are Programs Needed?



Having separate HTML pages for each photo would be impractical and very time-consuming to maintain or update the design.

Why are Programs Needed?

Instead, using a single program, we can display any of our photos. We will only need to update a single program when we want to revise the design.



The Client-Server Model

Client machines are the desktops, laptops, smart phones, and tablets you see everywhere.

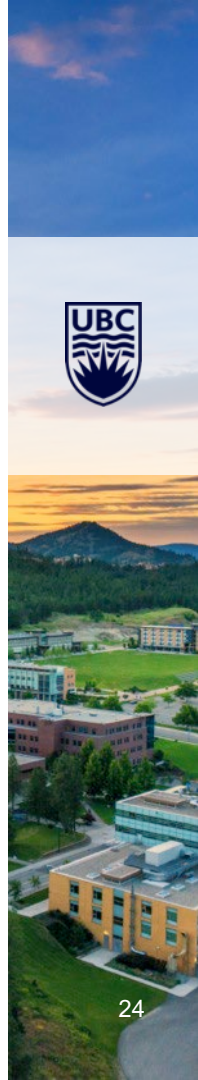
Broad range of specifications regarding

- operating system,
- processing speed,
- screen size,
- available memory, and
- storage

Server machines hosts web applications, stores user and program data, and performs security authorization tasks

Powerful machines to handle high traffic and bandwidth.

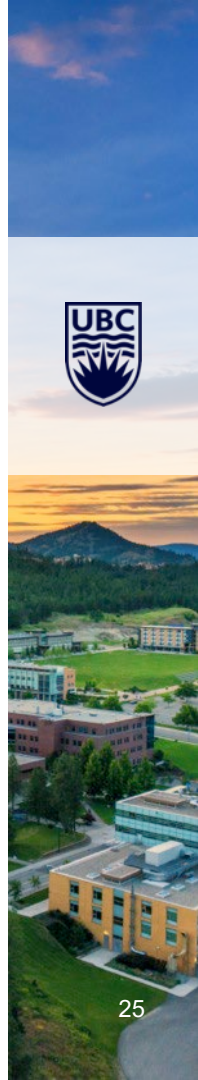
The essential characteristic of a server is that it is listening for requests, and upon getting one, responds with a message.



The Client-Server Model

Server Types

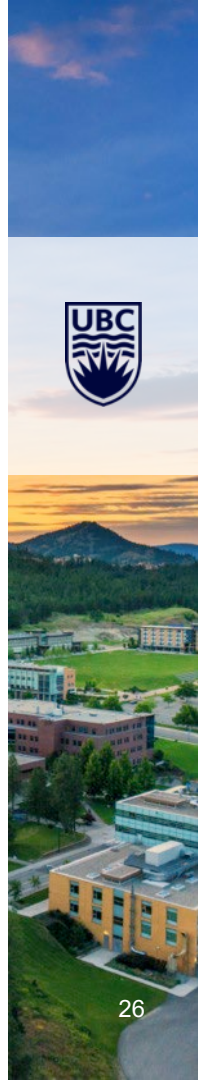
- Web servers
- Application servers
- Database servers
- Mail servers
- Media servers
- Authentication servers



Real World Server Installations

Not one server, but a cluster of multiple machines working together.

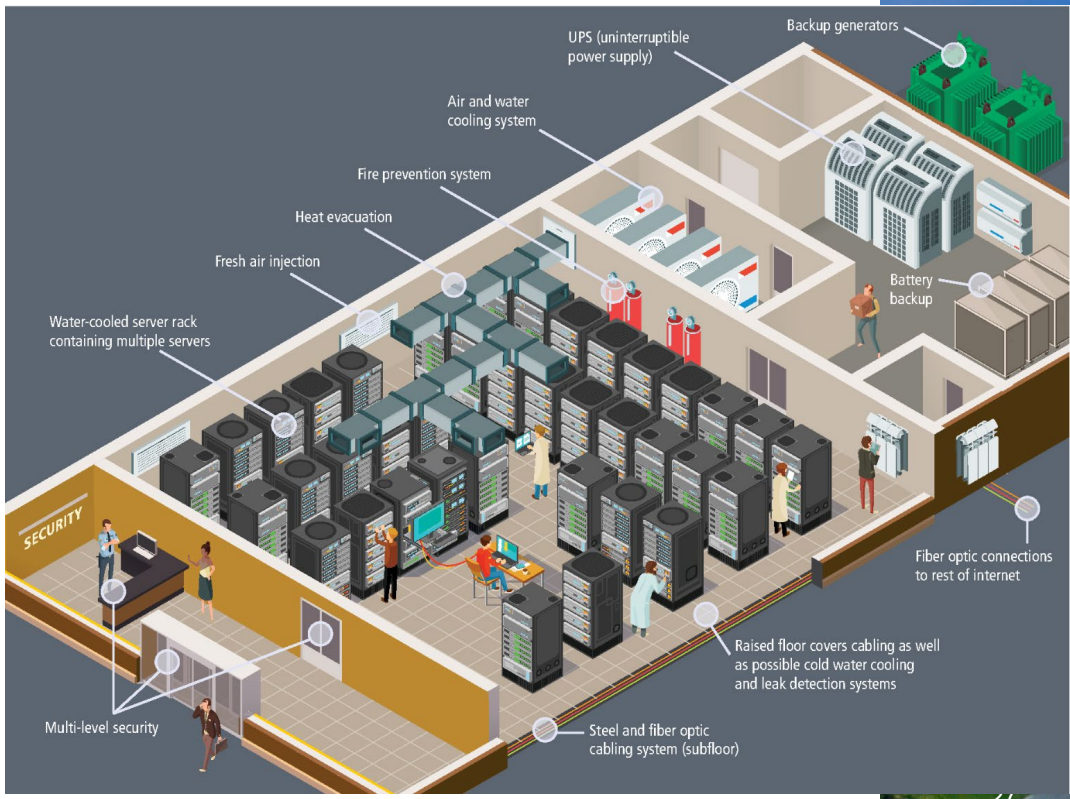
- Server Farm
- Load Balancers
- Failover Redundancy
- Server Racks
- Data Centers
- Cloud Services



Hypothetical data center

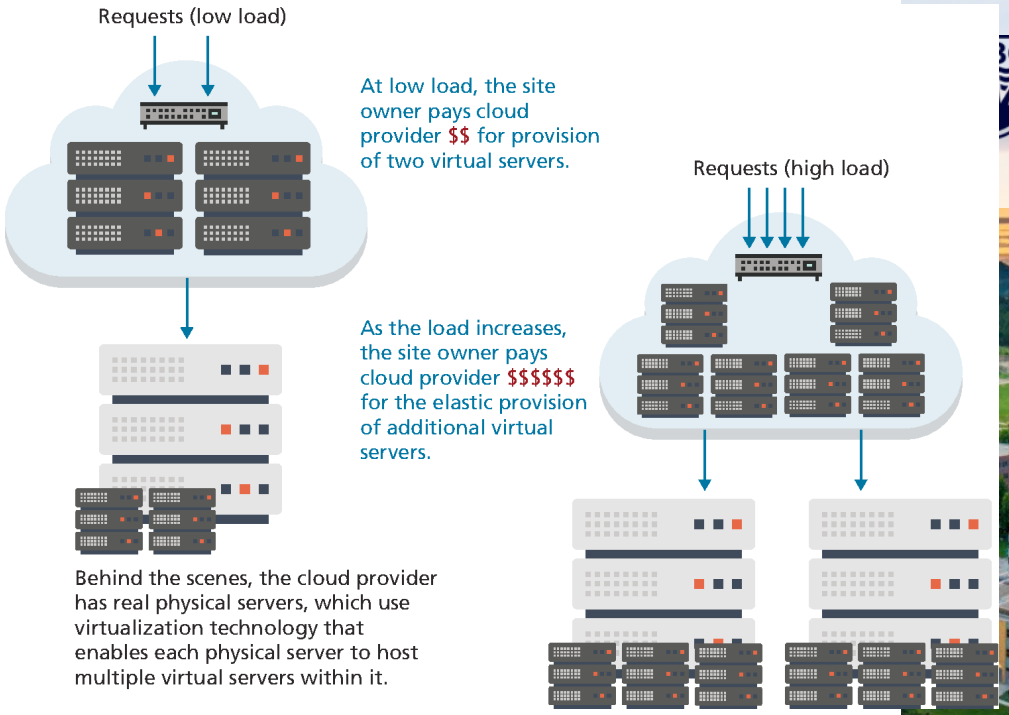
Many additional considerations can be handled at a data center including:

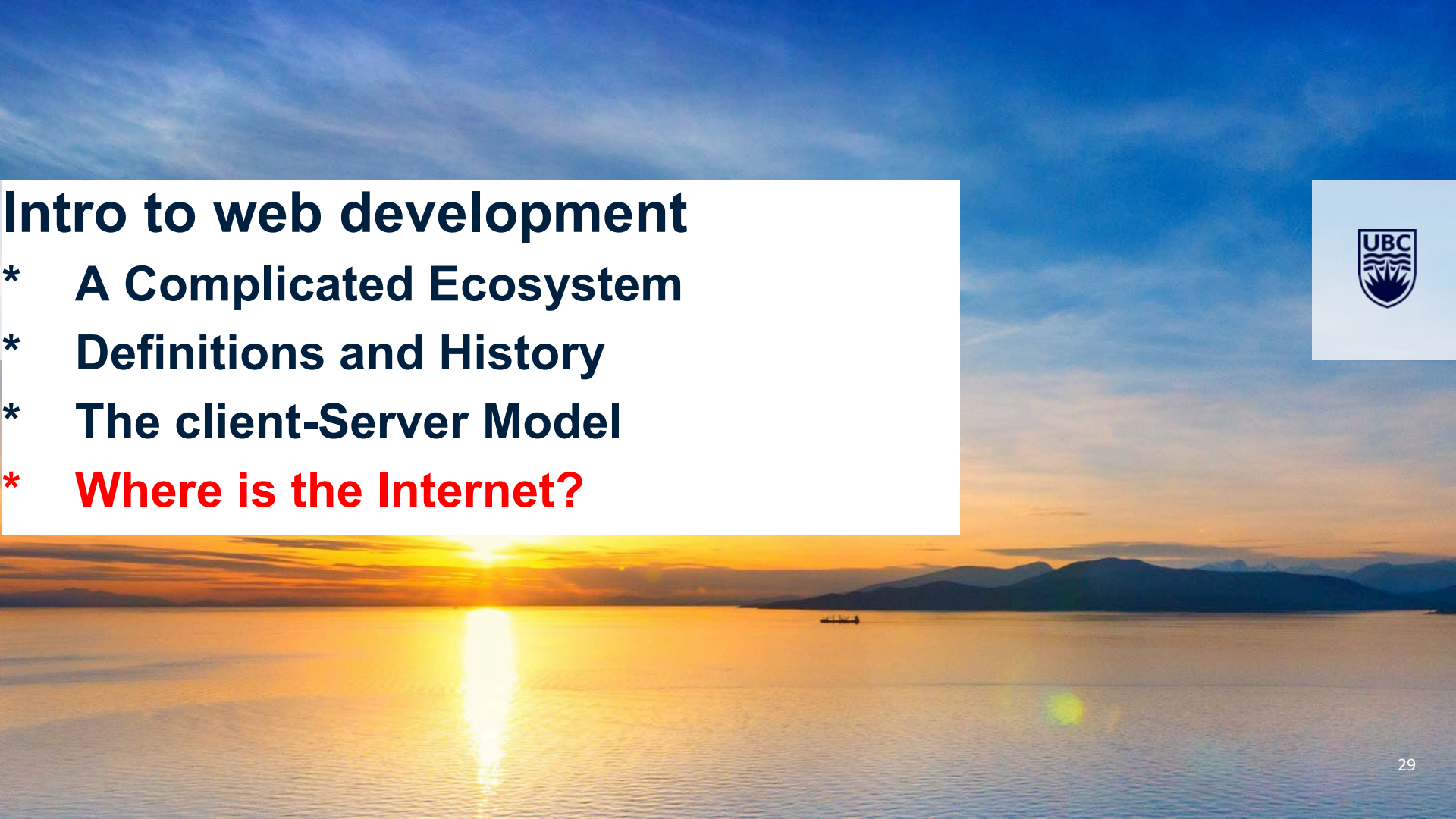
- Fire suppression,
- Biometric security,
- Failover data,
- Redundant power,
- and more!



Cloud Servers

Instead of spending too much or spending too little to handle peak loads, cloud providers offer **elastic provisioning** of virtual servers, which scales costs and hardware to the demand

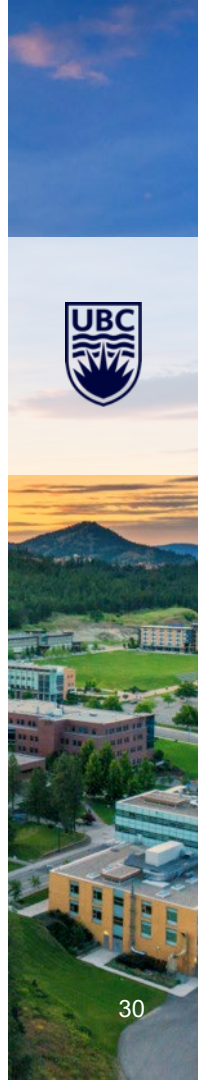



Intro to web development
* A Complicated Ecosystem
* Definitions and History
* The client-Server Model
* Where is the Internet?
UBC

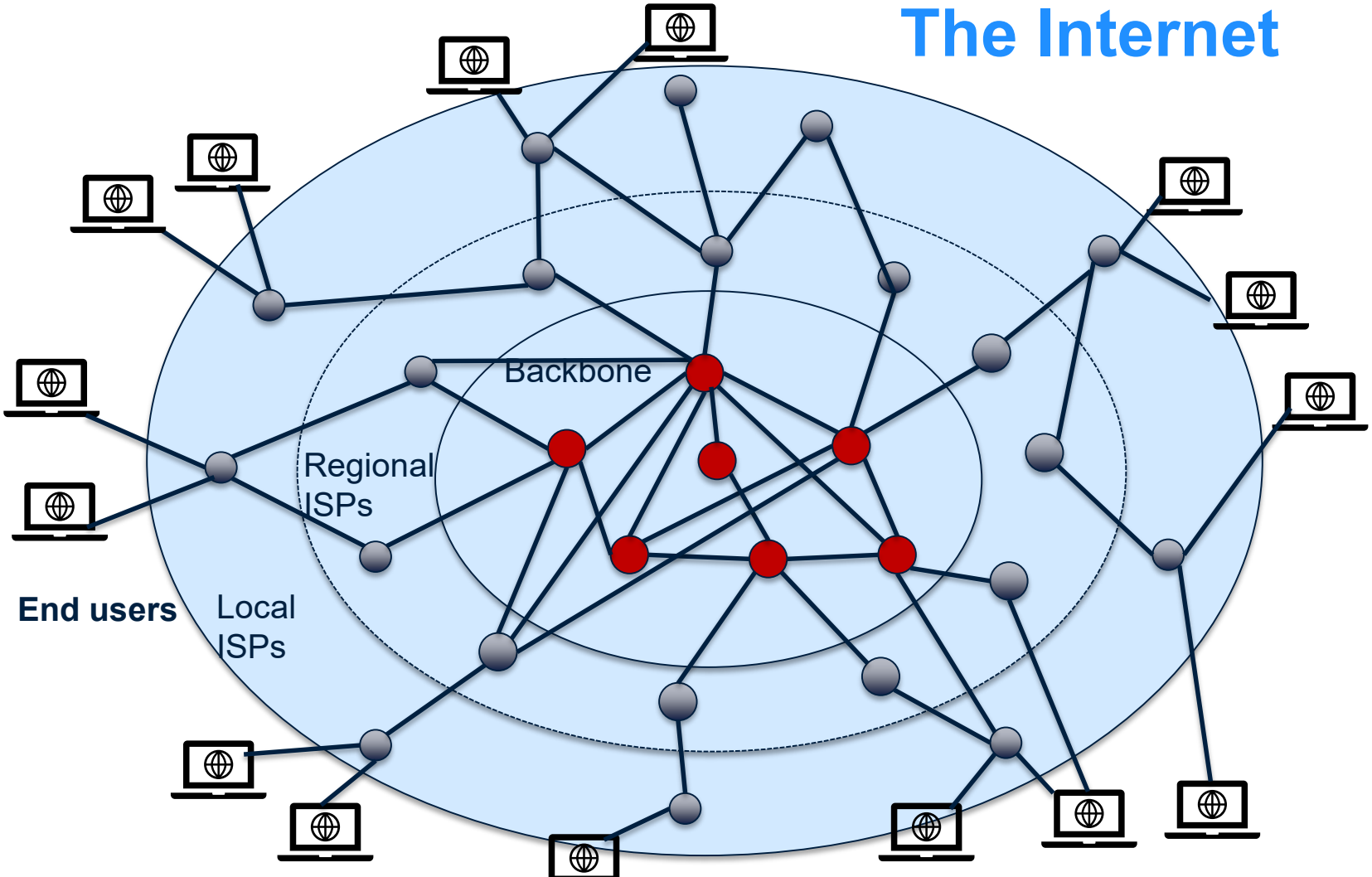
Where is the Internet?

It is quite common for the Internet to be visually represented as a cloud.

Actually implemented via millions of miles of copper wires and fiber optic cables connecting millions of server computers and other networked devices!



The Internet

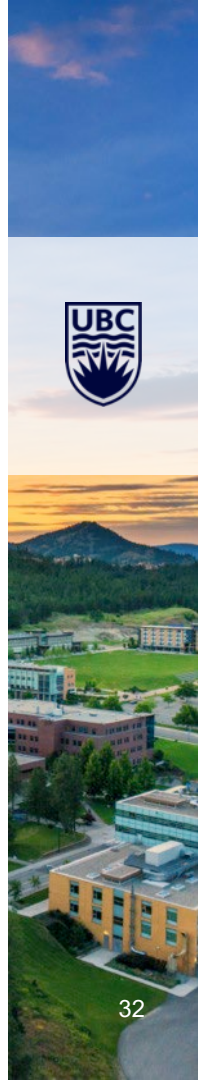


From the Computer to Outside the Home

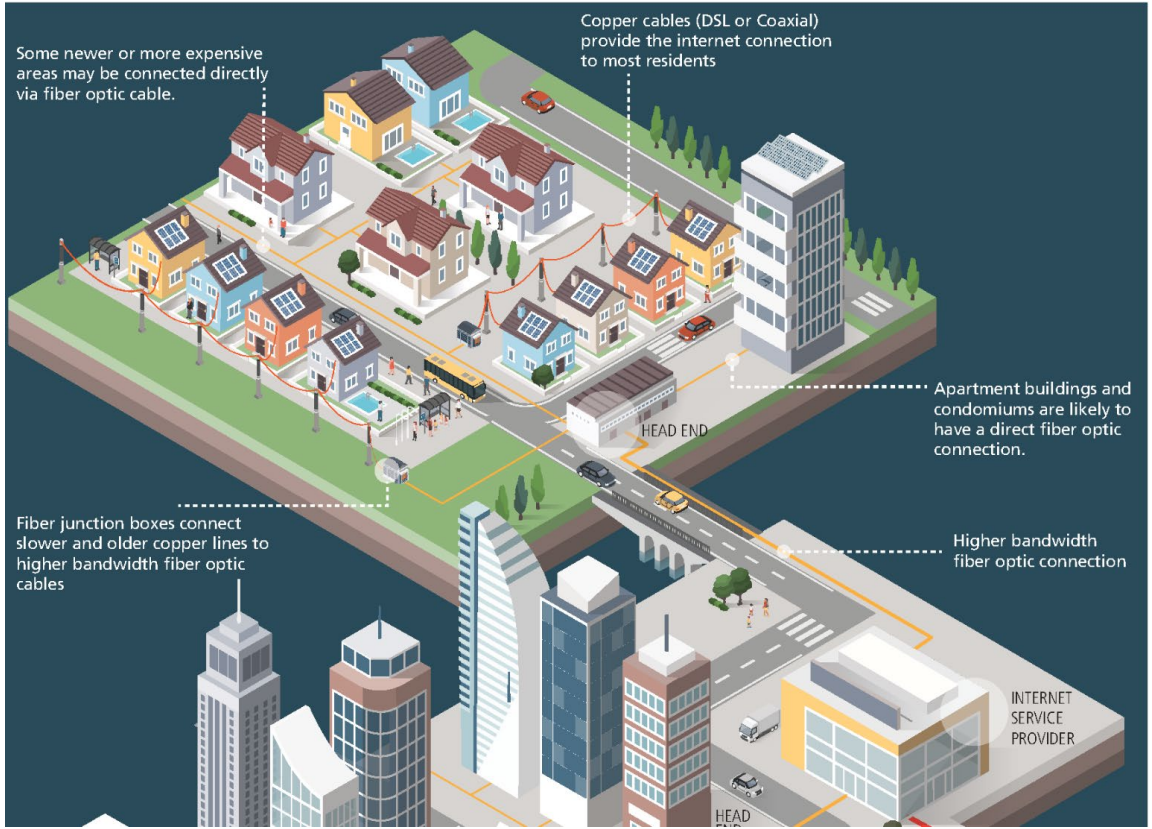
The **broadband modem** is a bridge between the network hardware outside the house and the network hardware inside the house. These devices are often supplied by the ISP.

The **wireless router** is perhaps the most visible manifestation of the Internet in one's home. At its simplest, a router is a hardware device that forwards data packets from one network to another network.

A detailed discussion of all the networking hardware involved in making the Internet work is far beyond the scope of this course.

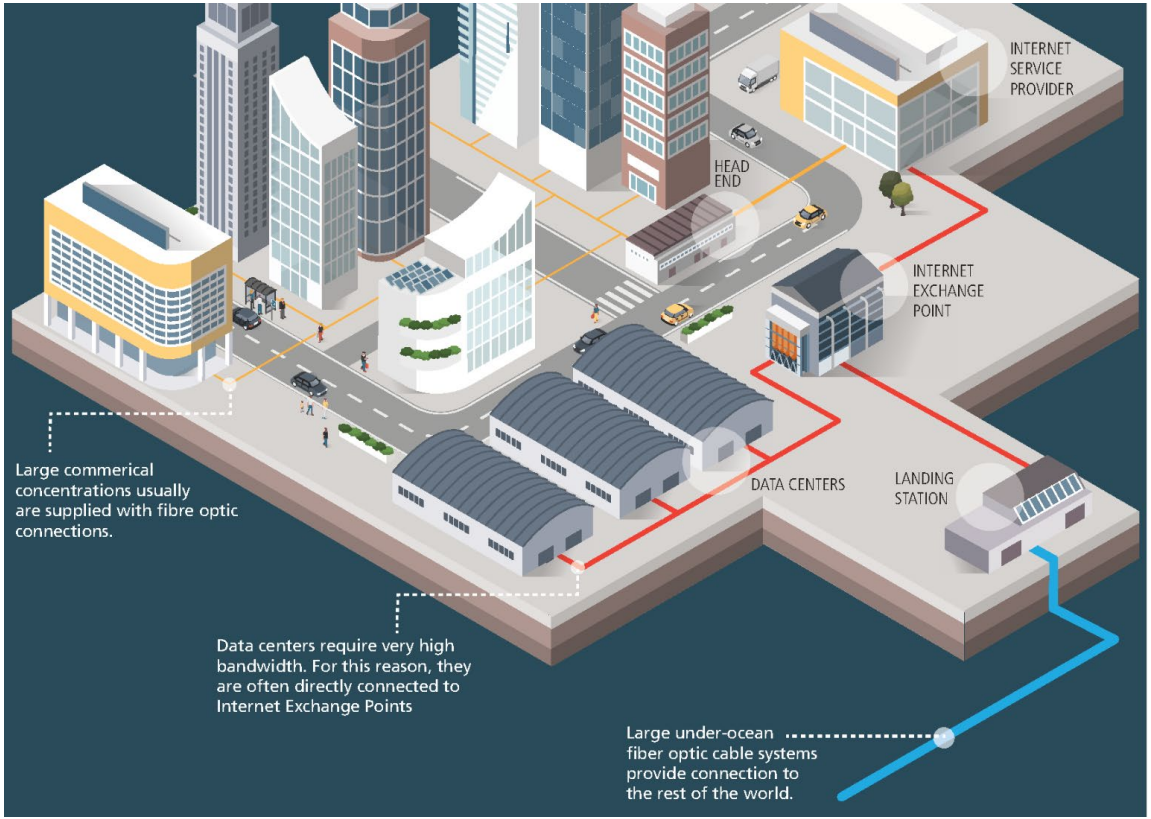


From the Home to the Ocean's Edge



A typical home Internet configuration and the beginnings of its connection to the outside world

From the Home to the Ocean's Edge



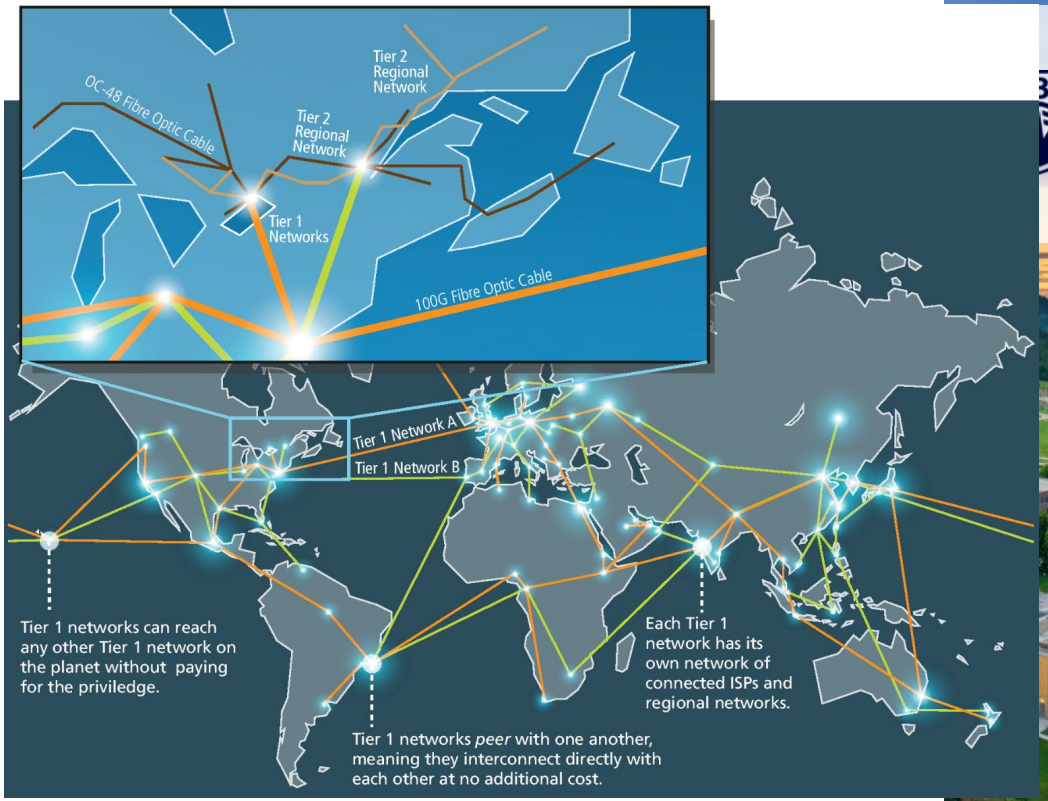
A typical home Internet configuration and the beginnings of its connection to the outside world

How the Internet Is Organized Today

When someone talks about the "Internet Backbone" they are talking about Tier 1 networks.

Tier 1 Networks make use of very fast fiber optic cable.

Regional networks have traditionally used less speedy infrastructure, though this is rapidly changing as prices of optical hardware decreases.



How the web works

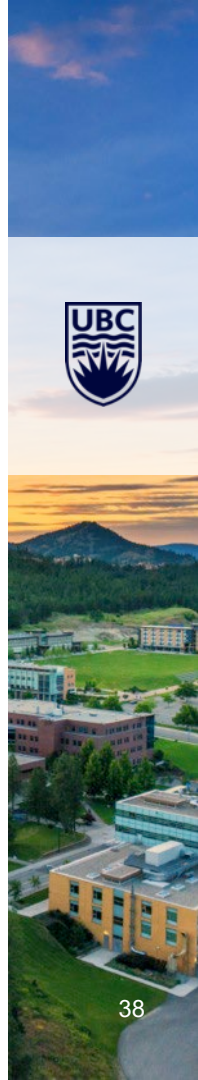
* Internet Protocols
* Domain Name System
* Uniform Resource Locators
* Hypertext Transfer Protocol
* Web Browsers
* Web Servers

How the web works

* **Internet Protocols**
* Domain Name System
* Uniform Resource Locators
* Hypertext Transfer Protocol
* Web Browsers
* Web Servers

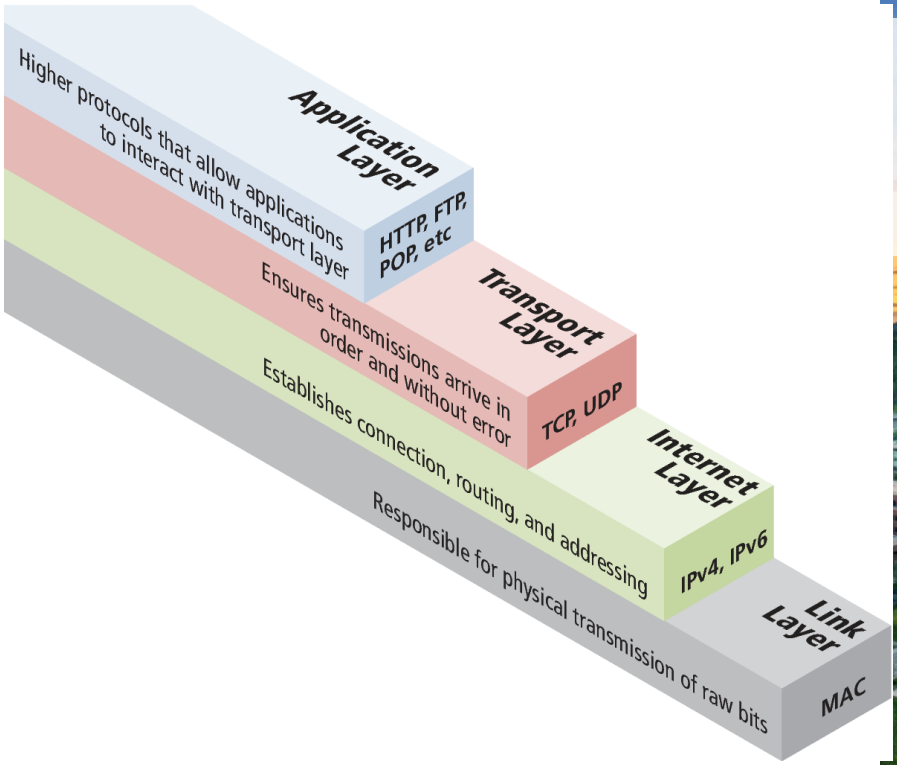
What's a Protocol?

- A **protocol** is a set of rules that partners in communication use when they communicate
- Transmission Control Protocol/Internet Protocol (TCP/IP) have been implemented in every operating system, and make fast web development possible
- Web developers need general awareness of what the suite of Internet protocols does



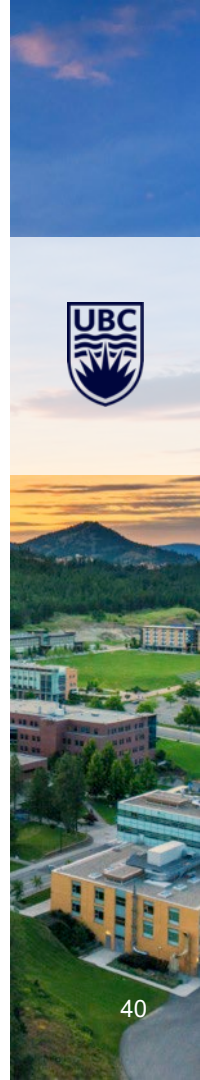
A Layered Architecture

- The TCP/IP Internet protocols were originally abstracted as a four-layer stack
- Later abstractions subdivide it further into five or seven layers
- Since we focus on the top layer, we will use the earliest and simplest four-layer network model



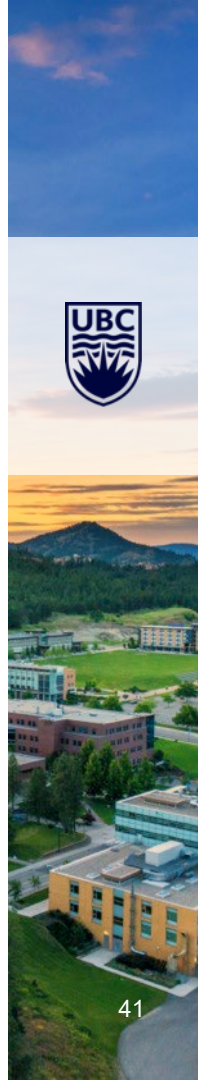
Link Layer

- The **link layer** is the lowest layer, responsible for
  - Physical transmission of data across media (both wired and wireless) and
  - Establishing logical links
- It handles issues like packet creation, transmission, reception and error detection, collisions, line sharing and more



Internet Layer

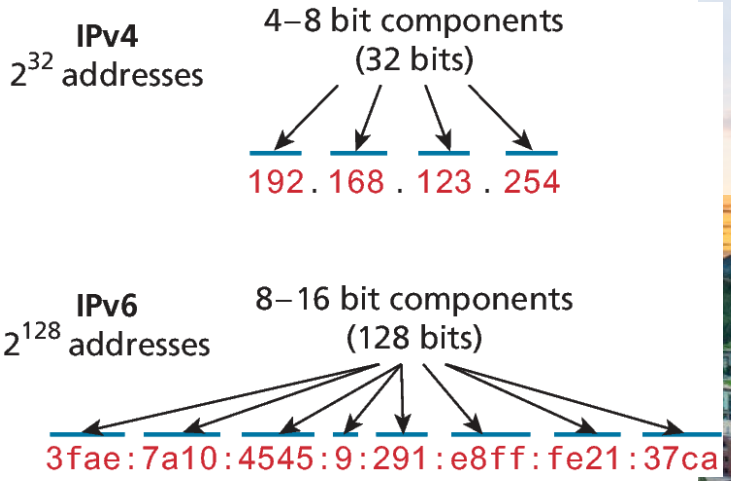
- The **internet layer** (sometimes also called the IP Layer) routes packets between communication partners across networks
- The Internet uses the **Internet Protocol** (**IP**) addresses to identify destinations on the Internet
- IP addresses are numeric codes that uniquely identify destinations on the Internet
- Every device connected to the Internet has an **IP address**
- Two types of IP addresses: IPv4 and IPv6



IP Addresses

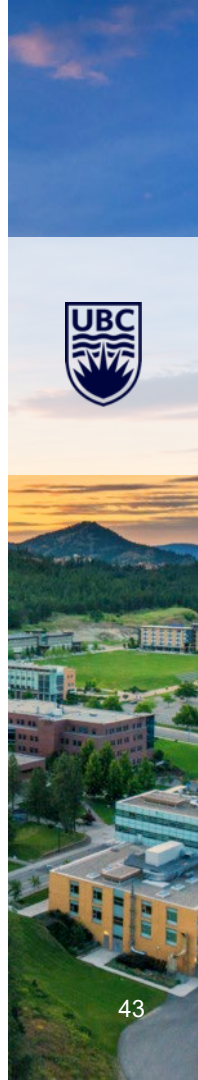
There are two types of IP addresses: IPv4 and IPv6.

- In **IPv4**, four 8-bit integers separated by . encode the address
- **IPv6** uses eight 16-bit integers and has over a billion times the number in IPv4

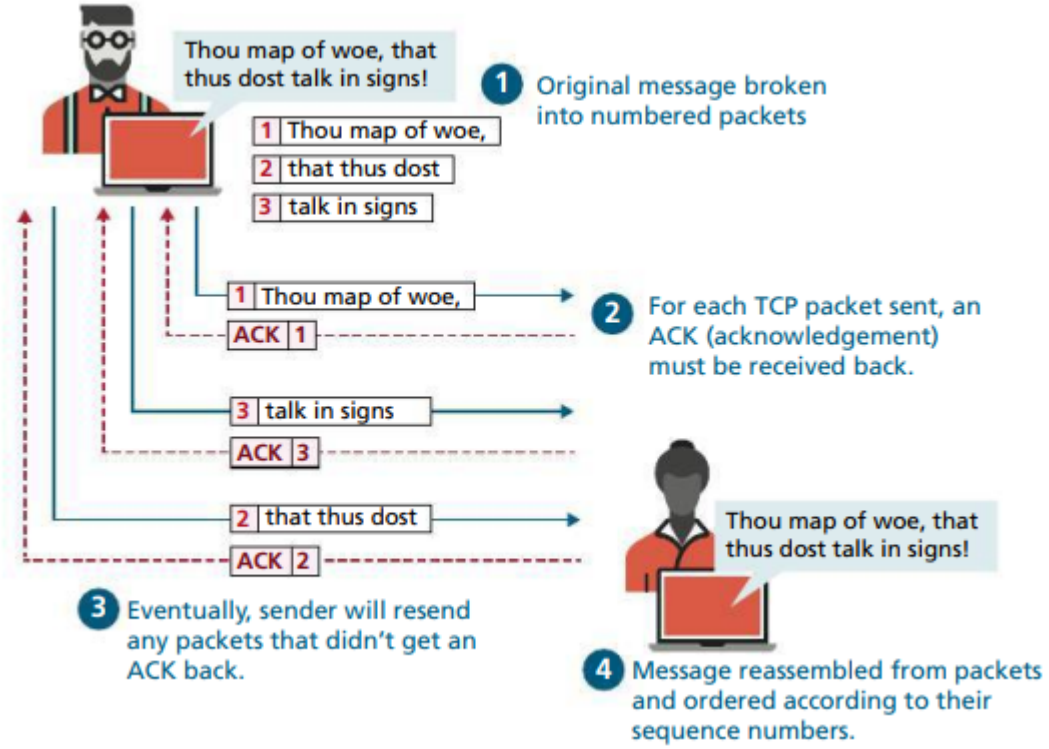


Transport Layer

- The transport layer ensures transmissions arrive in order and without error
- First, the data is broken into packets formatted according to the Transmission Control Protocol (TCP)
  - Each data packet has a header that includes a sequence number, so the receiver can put the original message back in order
  - Each packet acknowledges its successful arrival back to the sender
  - In the event of a lost packet (since no acknowledgement arrived for that packet, the packet will be retransmitted
- This means you have a guarantee that messages sent will arrive and will be in order



Internet Protocols in Transport Layer

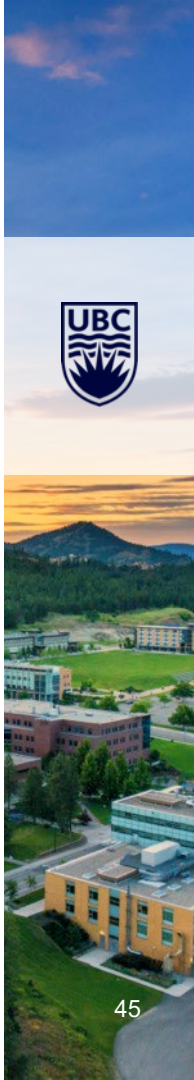


Application Layer

Application layer protocols implement process-to-process communication and are at a higher level of abstraction in comparison to the low-level packet and IP address protocols in the layers below it.

There are **many application layer** protocols. Web developers should be aware of :

- **HTTP.** The Hypertext Transfer Protocol is used for web communication
- **SSH.** The Secure Shell Protocol allows remote command-line connections to servers
- **FTP**. The File Transfer Protocol is used for transferring files between computers
- **POP/IMAP/SMTP**. Email-related protocols for transferring and storing email
  - Post Office Protocol, Internet Message Access Protocol, Simple Mail Transfer Protocol
- **DNS**. The Domain Name System protocol used for resolving domain names to IP addresses



How the web works

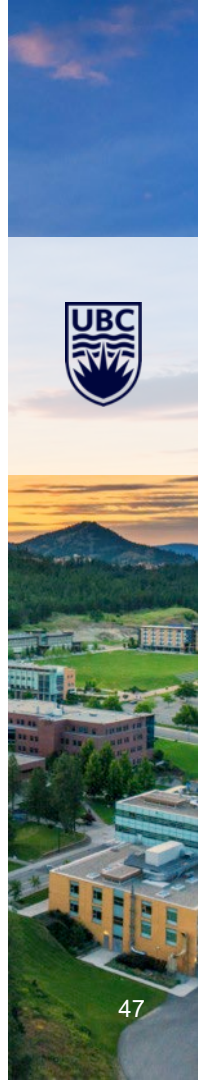
* Internet Protocols
* Domain Name System
* Uniform Resource Locators
* Hypertext Transfer Protocol
* Web Browsers
* Web Servers

Domain Name System

Why do we need it?

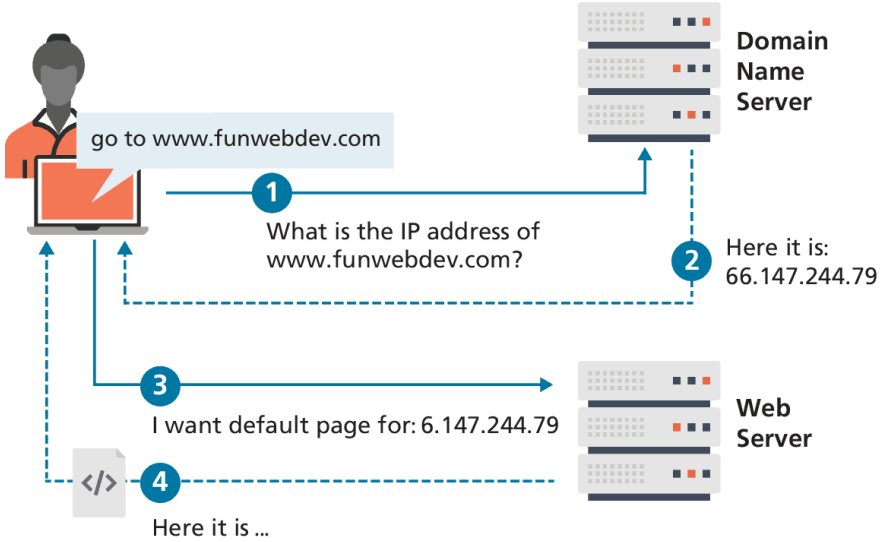
As elegant as IP addresses may be, human beings do not enjoy having to recall long strings of numbers. Instead of IP addresses, we use the **Domain Name System** (**DNS**)

Imagine having to type IP addresses to visit Google, Facebook etc!



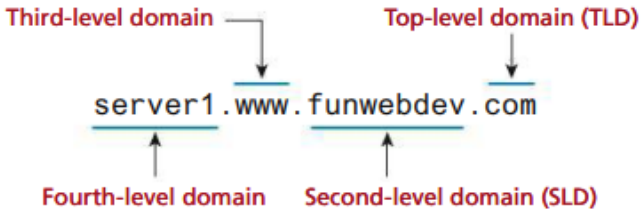
DNS Overview

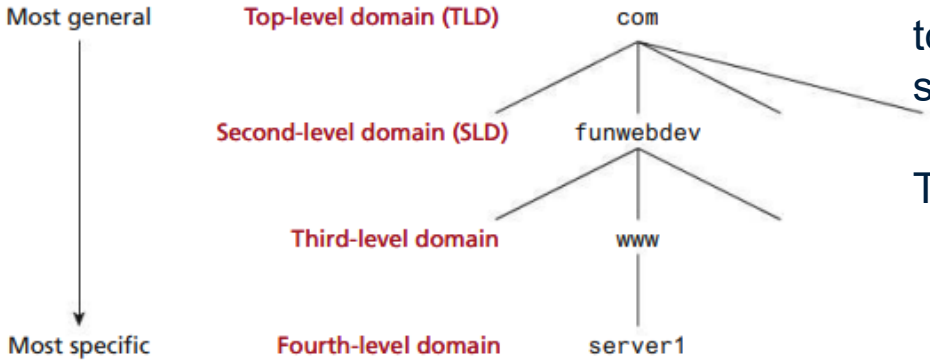
- The DNS system maps resolves domain names to IP addresses.
- By separating the domain name of a server from its IP address, a site can move to a different host without changing its name.
- Since the entire request-response cycle can take less than a second, it is easy to forget that DNS requests are happening at all.
- The actual process is more complex (more on that later)



Simplified form of the Domain Name System (DNS)

Domain Name System

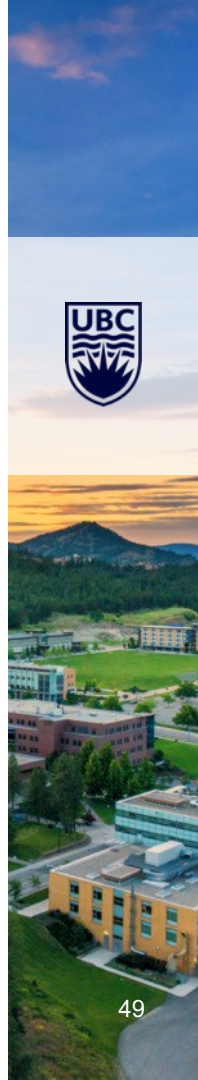




A domain name can be broken down into several parts, which describe a hierarchy.

All domain names have at least a top-level domain (TLD) name and a second-level domain (SLD) name.

TLDs include .us, .ca, .uk, and .au



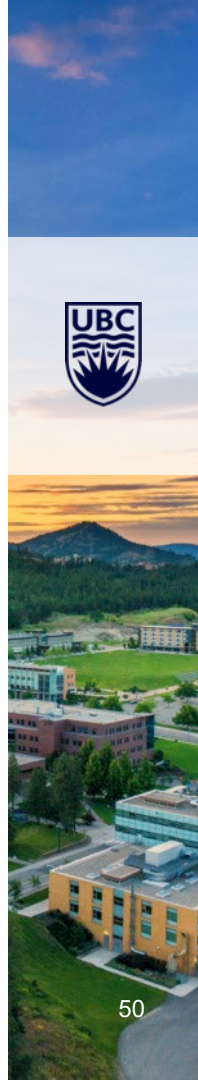
Name Registration

How are domain names assigned?

Special organizations or companies called **domain name registrars** manage the registration of domain names.

These domain name registrars are given permission to do so by the appropriate generic top-level domain (gTLD) registry and/or a country code top-level domain (ccTLD) registry.

The nonprofit Internet Corporation for Assigned Names and Numbers (ICANN)—oversees the management of top-level domains, accredits registrars, and coordinates other aspects of DNS.



Types of TLDs

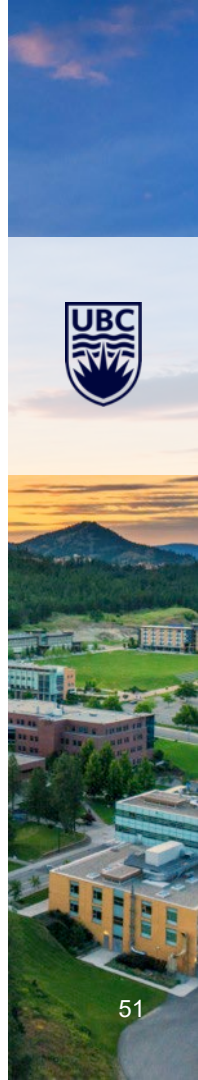
Generic top-level domains (gTLD)

- Unrestricted. TLDs include .com, .net, .org, and .info
- Sponsored. TLDs including .gov, .mil, .edu, and others
- New TLDs; .art, .io, .jobs, etc

Country code top-level domain (ccTLD)

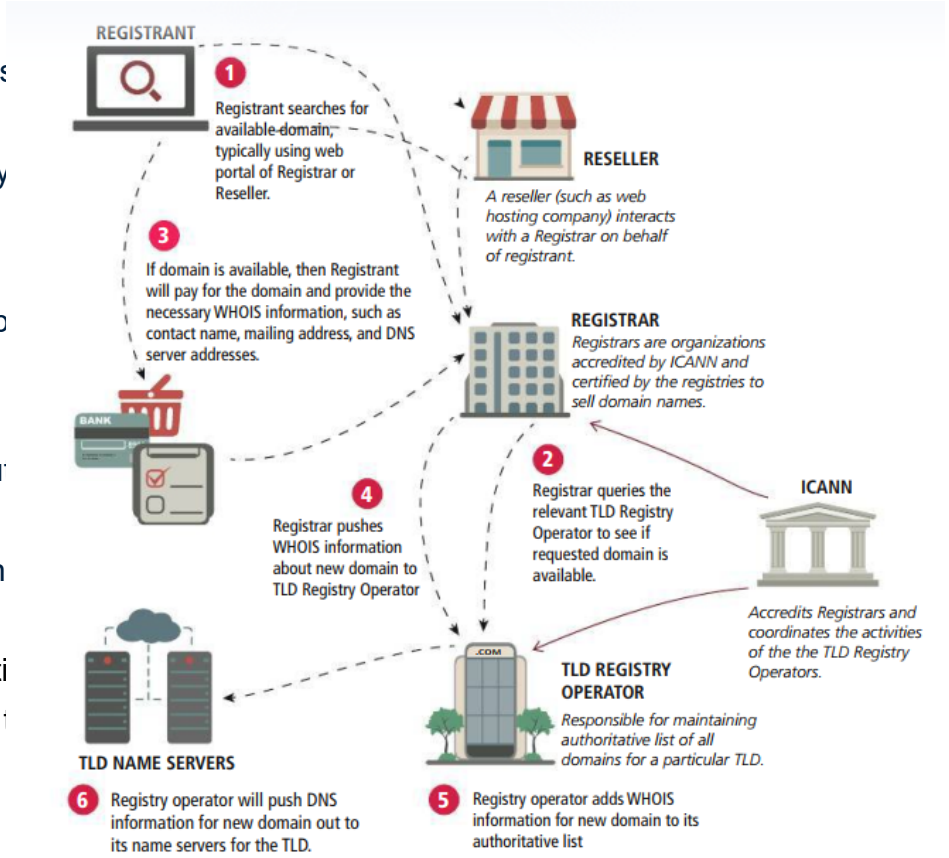
- TLDs include .us , .ca , .uk , and .au.
- Internationalized Domain Names - allows domains to use non-ascii characters

Since June 2012, ICANN invited companies to launch new TLDs in order to provide more choice. Since then over 1000 new TLD have been created including .art, .cash, .cool, .jobs, .tax and so on



Domain name registration process

1. Registrant searches for domain, typically us
   web portal of Registrar or Reseller.
2. Registrar queries the relevant TLD Registry
   Operator to see if requested domain is
   available.
3. If domain is available, then Registrant will p
   for the domain and provide the necessary
   WHOIS information
4. Registrar pushes WHOIS information abou
   domain to TLD Registry Operator
5. Registry operator adds WHOIS information
   new domain to its authoritative list
6. Registry operator will push DNS informati
   for new domain out to its name servers for
   TLD.

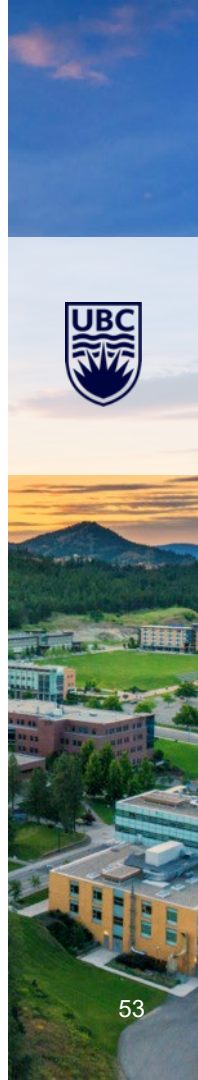


DNS Address Resolution

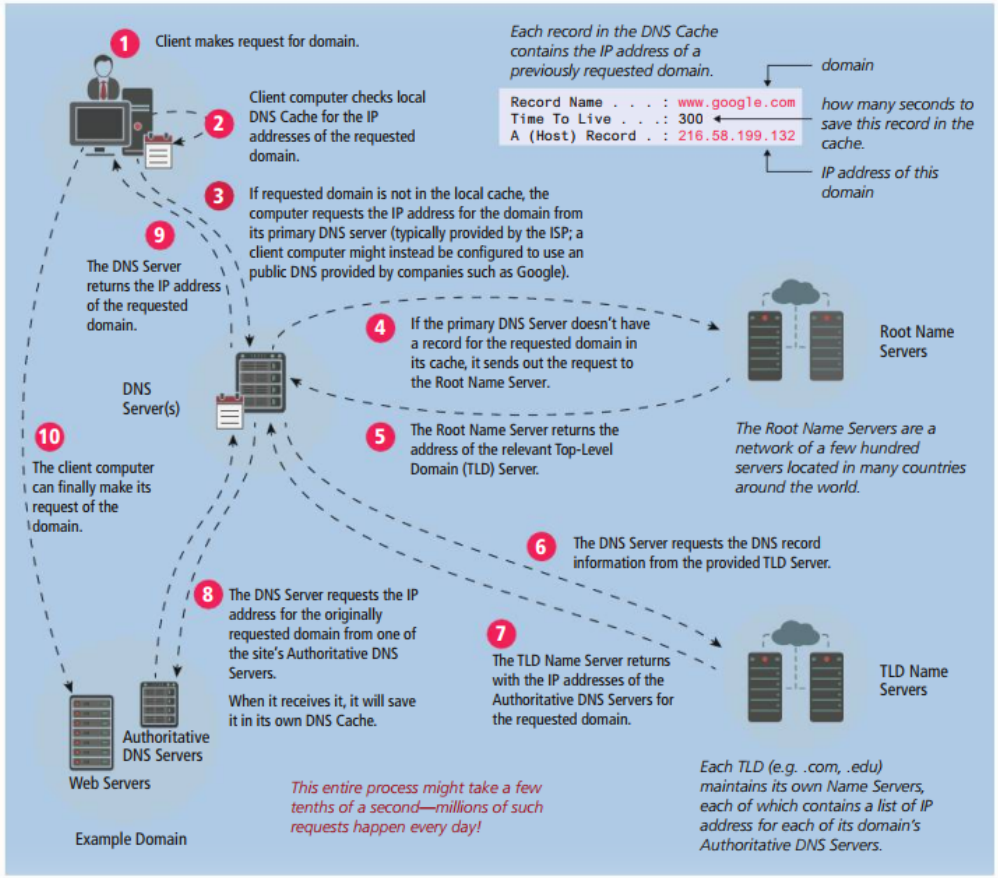
While domain names are certainly an easier way for users to reference a web site, eventually, your browser needs to know the IP address of the web site in order to request any resources from it.

The Domain Name System provides a mechanism for software to discover this numeric IP address.

This process is referred to here as **address resolution**.



Domain Name System



How the web works

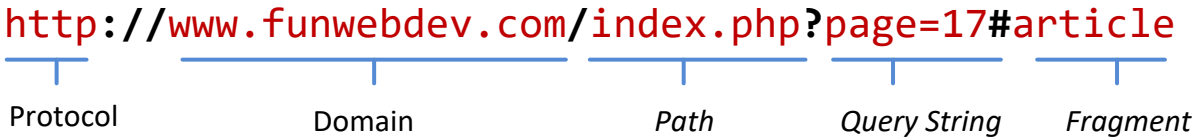
* Internet Protocols
* Domain Name System
* Uniform Resource Locators
* Hypertext Transfer Protocol
* Web Browsers
* Web Servers

URL Components

Uniform Resource Locators (URL) allow clients to request particular resources (files) from the server.

URL's consist of two required components:

1. the protocol used to connect and
2. the domain (or IP address) to connect to



`ftp://example.com/abc.txt`

→ sends out an FTP request on port 21

`http://example.com/abc.txt`

→ transmits an HTTP request on port 80

Uniform Resource Locators

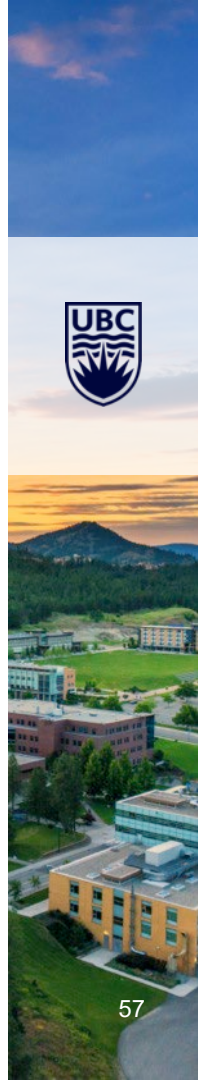
Domain

- The domain identifies the server from which we are requesting resources
- Since the DNS system is case insensitive, this part of the URL is case insensitive
- Alternatively, an IP address can be used for the domain

Port

- The optional port attribute allows us to specify connections to ports other than the defaults
- Port is a type of connection point used by the TCP/IP protocol and the connecting computer
- Add a colon after the domain, then specify an integer port number

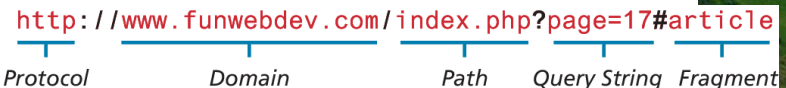
http://funwebdev.com:8080/



Uniform Resource Locators

The **path** is a familiar concept to anyone who has ever used a computer file system

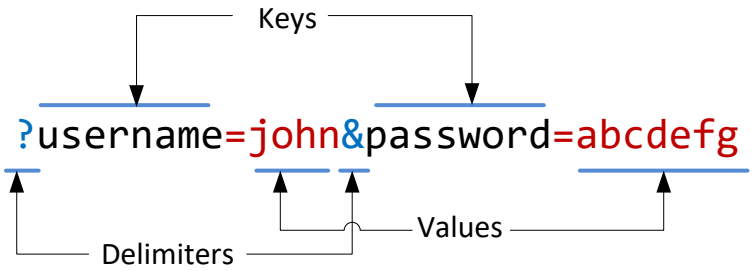
- The root of a web server corresponds to a folder somewhere on that server
  - On many Linux servers that path is /var/www/html/
  - On Windows IIS (Internet Information Services) machines it is often /inetpub/wwwroot/
- The path is **optional**
  - However, when requesting a folder or the top-level page of a domain, the web server will decide which file to send you
  - On Apache servers, it is generally index.html or index.php. On windows servers: Default.html or Default.aspx



Query String

Query strings will be covered in depth when we learn more about HTML forms and server-side programming.

Way of passing information such as user form input from the client to the server. In URL's they are encoded as key-value pairs delimited by “&” symbols and preceded by the “?” symbol.

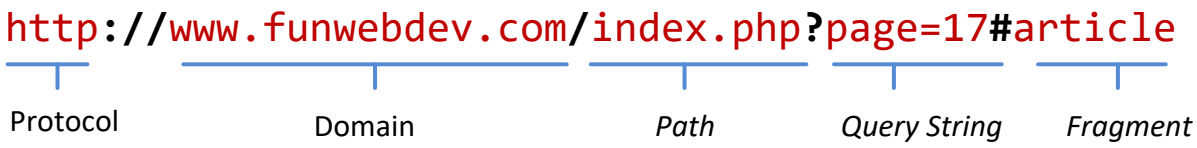


The components for a query string encoding a username and password

Uniform Resource Locators - Fragment

A way of requesting a portion of a page.

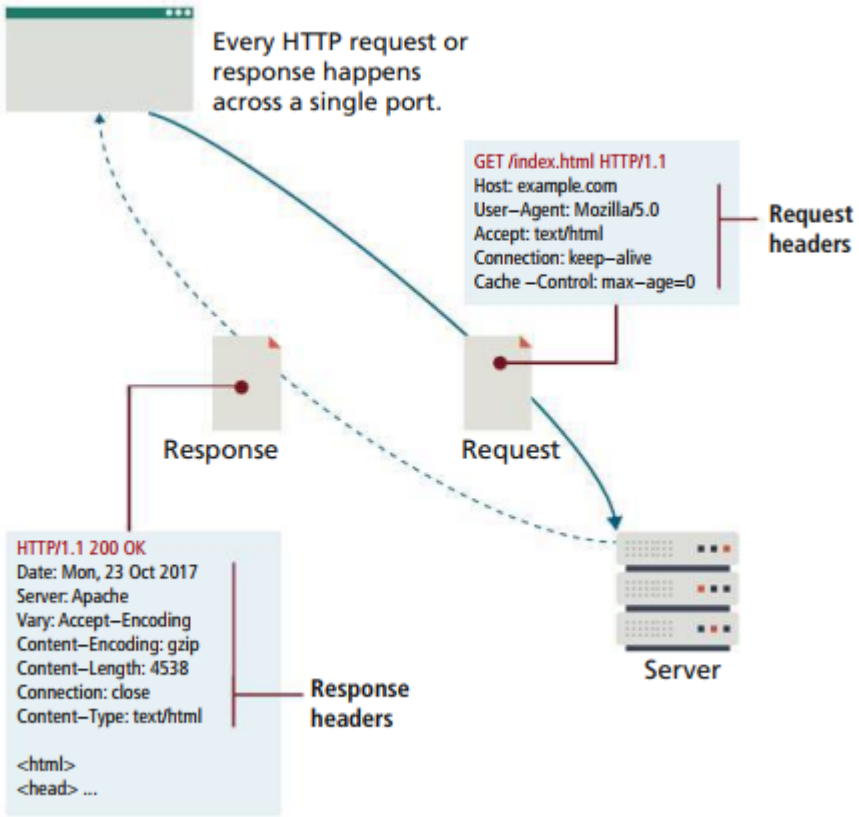
- Browsers will see the fragment in the URL, seek out the tag anchor in the HTML, and scroll the website to it.
- “back to top” links are a common use of fragments



How the web works

* Internet Protocols
* Domain Name System
* Uniform Resource Locators
* Hypertext Transfer Protocol
* Web Browsers
* Web Servers

Hypertext Transfer Protocol (HTTP)



World Wide Web's protocol

HTTP establishes a TCP connection on port 80 (by default)

Server waits for the request, and then responds with a response code, headers and an optional message (which can include files)

Request headers include data about the client machine

Response headers have information about the server answering the request and the data being sent

HTTP Request Methods

The most common requests are the GET and POST request

```
<form method="POST" action="FormProcess.php">
```


```
<a href="somePage.html">Hyperlink</a>
```

GET requests should NOT be used for state-changing actions (such as modifying data).

POST method is normally used to transmit data to the server using an HTML form

In a POST request, data is transmitted through the header of the request, and as such is not subject to length limitations like with GET

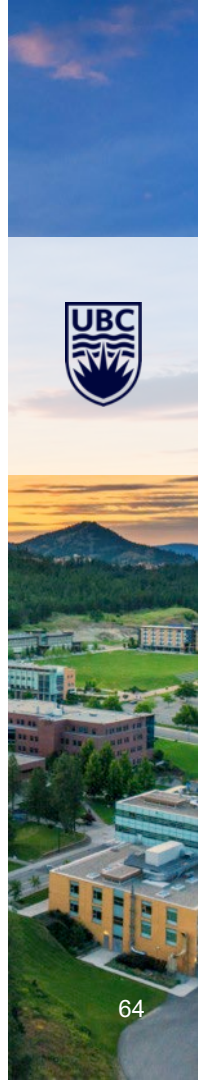
In the GET request, one is asking for a resource located at a specified URL to be retrieved

When you click on a link, type in a URL in browser or click on a bookmark, you are making a GET request

Hypertext Transfer Protocol – Response Codes

- They are integer values returned by the server as part of the response header
- Describe the state of the request: whether it was successful, had errors, requires permission, etc.
  - 2## codes are for successful responses,
  - 3## are for redirection-related responses,
  - 4## codes are **client** errors,
  - 5## codes are **server** errors.

200: OK
301: Moved Permanently
304: Not Modified
307: Temporary redirect
400: Bad Request
401: Unauthorized
404: Not found
414: Request URI too long
500: Internal server error

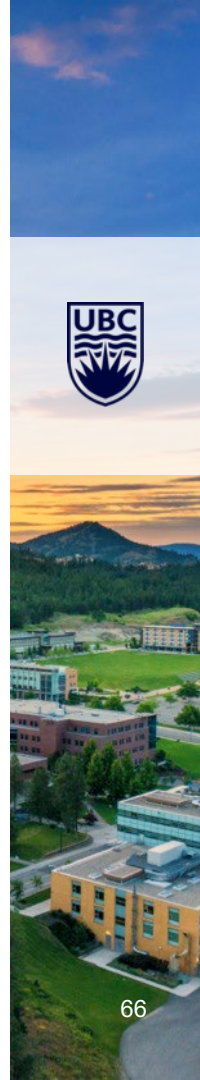


How the web works

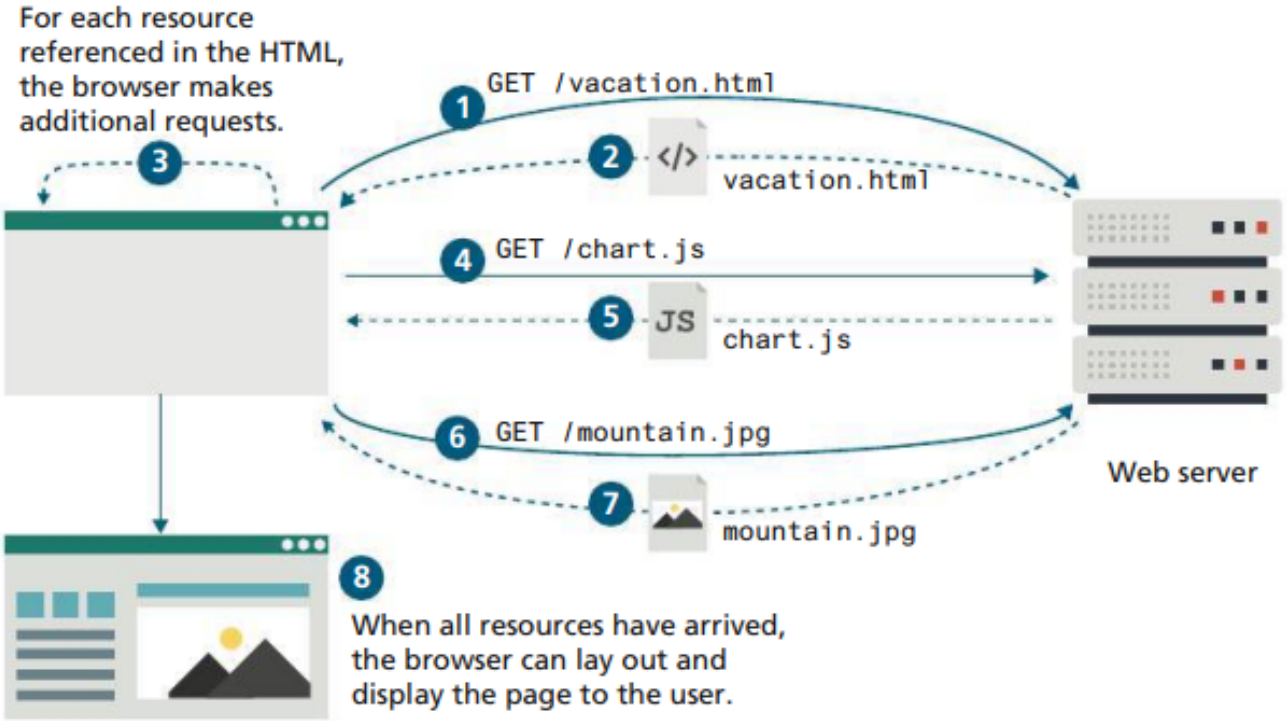
* Internet Protocols
* Domain Name System
* Uniform Resource Locators
* Hypertext Transfer Protocol
* **Web Browsers**
* Web Servers

Web Requests

- While we as web users might be tempted to think of an entire page being returned in a single HTTP response, this is not what happens
- Seeing a single web page is facilitated by the client's browser which requests the initial HTML page, then parses the returned HTML to find all the resources referenced from within it, like images, style sheets and scripts
- Only when all the files have been retrieved is the page fully loaded for the user
- A single web page can reference dozens of files and requires many HTTP requests and responses

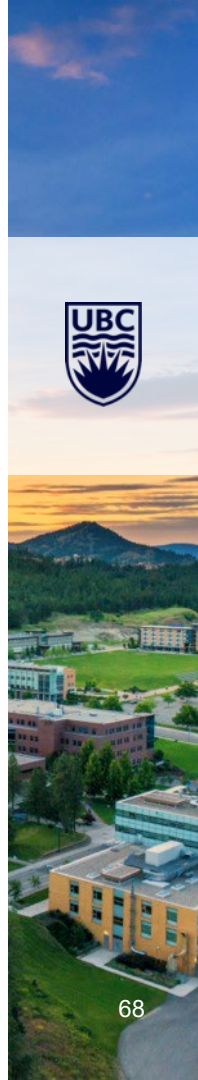


Browser parsing HTML and making subsequent requests



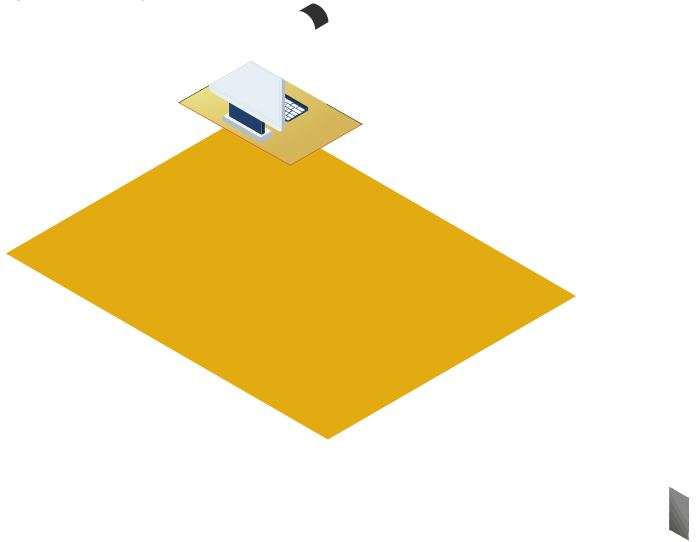
Web Browsers - Rendering

- Interpreting the entire HTML markup together with the image and other assets into a grid of pixels for display within the browser window is called **rendering** the webpage
- Implemented differently for each browser (Firefox, Chrome, Safari, Explorer, and Opera)



Web Browsers – Browser Caching

- Once a webpage has been downloaded from the server, it's possible that the user, a short time later, wants to see the same web page and refreshes the browser or re-requests the URL
- Although some content might have changed, the majority of the referenced files are likely to be unchanged (fresh) so they needn't be redownloaded

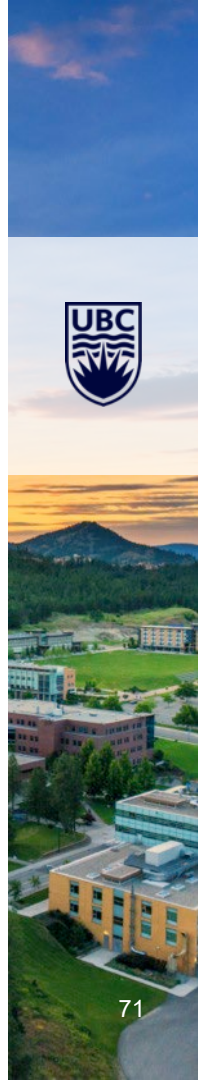


How the web works

* Internet Protocols
* Domain Name System
* Uniform Resource Locators
* Hypertext Transfer Protocol
* Web Browsers
* Web Servers

Web Servers

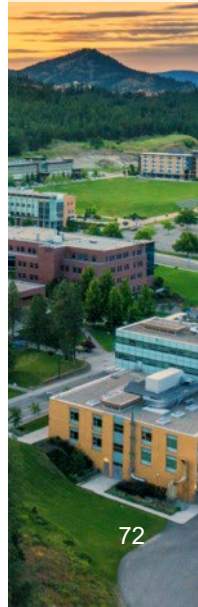
- A **web server** is, at a fundamental level, nothing more than a computer that responds to HTTP requests
- Real-world web servers are often more powerful than your own desktop computer
- Webservers must choose an **application stack** to run a website. This application stack will include an
  - operating system,
  - web server software,
  - a database,
  - and a scripting language for dynamic requests



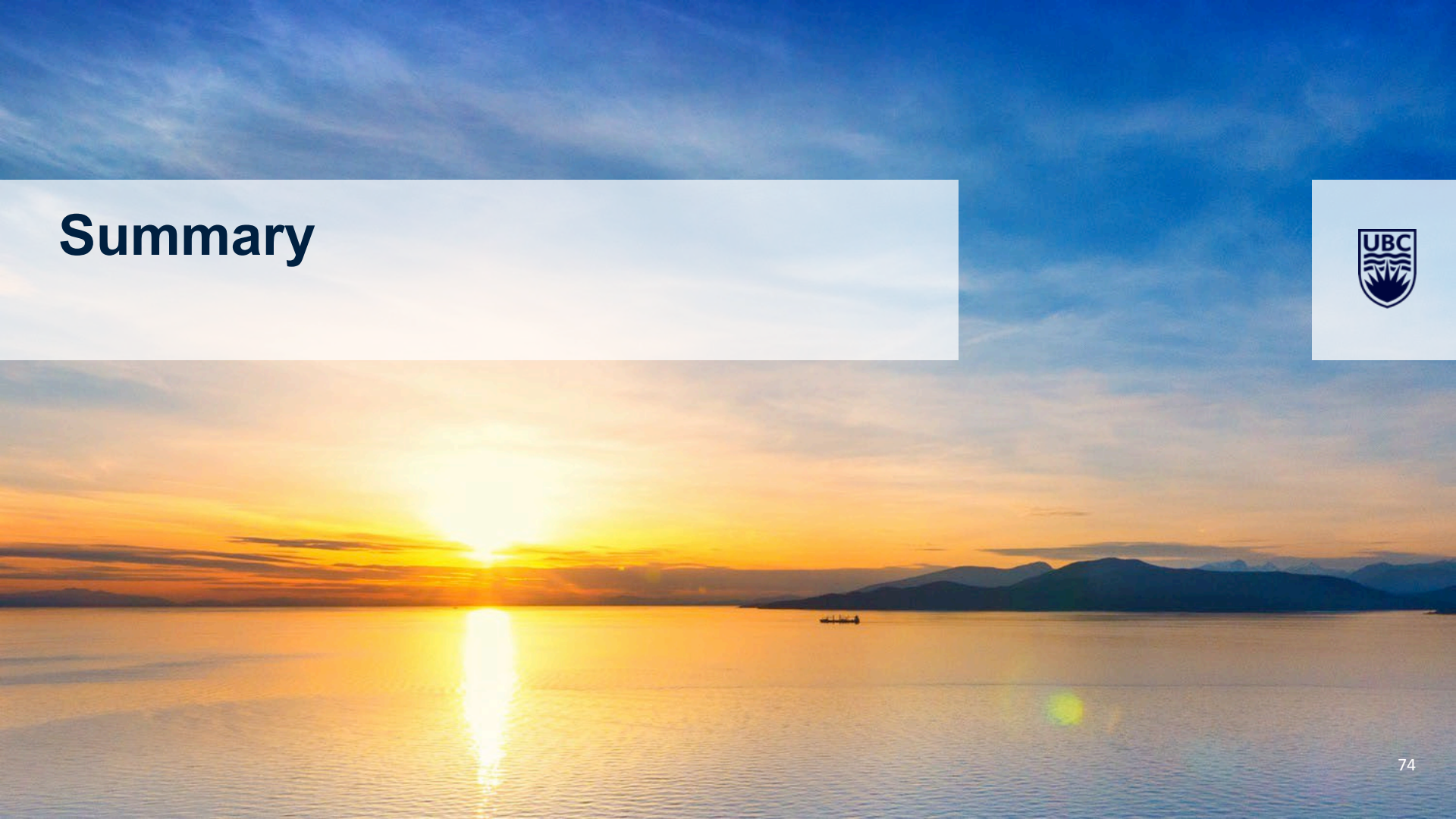
MERN Software Stack

We will be using the **MERN software stack**, which refers to the

- **M**ongoDB database,
- **E**xpress web application framework for Node,
- **R**eact framework,
- **N**ode.js runtime environment

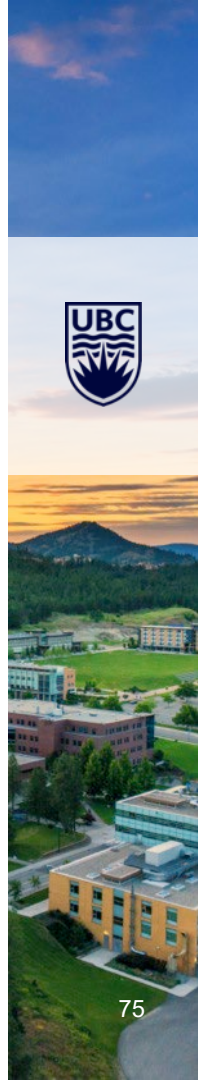


Summary



Summary

- Web development in general
- Fundamental concepts that form the foundation of the Internet
- Hardware and software that support the Internet
- The fundamental protocols that make the web possible
- How the domain name system works
- HTTP
- How browsers and servers work to exchange and interpret HTML



Reference

- Fundamentals of Web Development By Randy Connolly and Ricardo Hoar, 3rd Edition, ISBN: 0134481267