



The World Wide Web

The Short Version...

Client Server Model

The **client-server model** describes how a **server** provides resources and services to one or more **clients**. Examples of servers include web servers, mail servers, print servers and file servers. Each of these servers provide resources to **client** devices, such as desktop computers, laptops, tablets, and smartphones.

The essential and often unspoken ingredient to this model is **connectivity**, the ability to have both client and server to be **networked** so they can talk to each other.

Physically we **can connect systems with wires or radio waves**. We use several systems to connect and identify networked systems. To start we have an address 'cooked' into each piece of networking hardware called a MAC address that is supposed to make it unique.

The main transport system of the internet and all networking in general is **TCP/IP** Transfer Connect Protocol is just that. IP stands for Internet Protocol and from it we get IP addresses which are how we actually publish the WWW across the internet.

IP Address

We are in a transition period right now between IPv4 and IPv6. That said we are going to only concern ourselves with IPv4 examples here because they are easier to read. This is also not a computer networking class so we are taking the approach of learning just enough to be dangerous.

An IPv4 address looks like this

000.000.000.000 placeholders of 0-255 (this is shorthand for binary so stop asking questions)

The truth is there are so many places where ranges of numbers are used for different things that we are going to talk about specifics. Webservers all have IP addresses.

All devices that are on a network have IP addresses. We have 2 ranges designed for home networks. You are on 1 of them. 192.168.0.0 or 10.10.0.0 these are designed to 'fail' if their data is out on the internet. Your router has been given an IP address from your ISP.

Port numbers

There is one more piece to wrap our heads around. TCP/IP gets stuff to the right device, how does my website get to my web browser and my email get to my email program etc. Port numbers Listened to by software and there are standard ones and you can make custom ones yourself.

HTTP 80 Web

HTTPS 443 Web (secure)

Remember the instructions for the Python server defaults to 8000

You would add :8000 to the end of the IP address. <http://10.10.0.200:8000>

FYI This is how firewalls and routers work. Port forwarding is how you can run a server from your home.

DNS Domain Name System

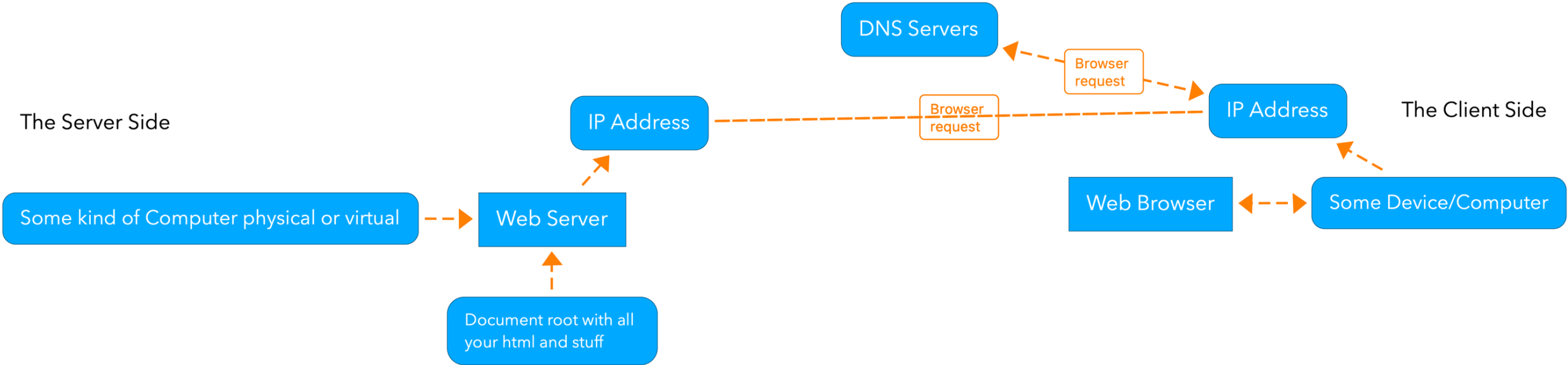
Clients and Servers and IP numbers and Ports! Why can't I just type <http://google.com>
You can, thanks to DNS which is a worldwide chain of dedicated servers that do nothing but translate names into numbers. When you register a domain name you are essentially renting it and the cost goes to maintain the network of servers.

We call them nameservers for short and you are probably getting yours from your ISP but there are others like <https://developers.google.com/speed/public-dns>

A very important thing about domain names is how they are structured. **We read them left to right.**

moonbase.mokturtle.net (which might actually be a real website) is read
.net the top level domain like .com or .gov that acts like a category
mokturtle.net the domain I own and can point it to my webserver
moonbase.mokturtle.net is a subdomain there is little or no limit how many subdomains I can have.

The World Wide Web



This is a very simplified example. The dashed lines imply computer network traffic.

The Secure Web HTTPS

This uses an SSL Certificate. Public and private key encryption and some form of authoritative certificate to vouch for your authenticity. You can pay for the privilege of having it all magically done for you. You can be underground and make your own and only people who know you will trust you and go to your site.

The best solution for a non-commercial site is to use certbot <https://certbot.eff.org/>

Certificates for websites use what's called a chain of trust.

The purpose is to keep someone from jumping between the client and server and to establish the identity of the server to the client.

<https://www.ssl.com/faqs/what-is-a-chain-of-trust/>

Home network

There are a few things you can do if you want to add names to IP numbers locally on your computer or home network.

Hosts file: It is possible to make your computer(s) respond to names locally using a hosts file.
<https://www.howtogeek.com/howto/27350/beginner-geek-how-to-edit-your-hosts-file/>

Your router may have this ability as well. You will need to login and look.

Zeroconfig is a system you may already have and not know. It's called mDNS as a protocol but Apple calls it Bonjour, Linux calls it Avahi
Windows doesn't fully support it.

This article will at least get you thinking about it.

<https://learn.adafruit.com/bonjour-zeroconf-networking-for-windows-and-linux>