Network Planning

What device isn’t network connected?

Network Stack Overview

7 Layers of the OSI Model

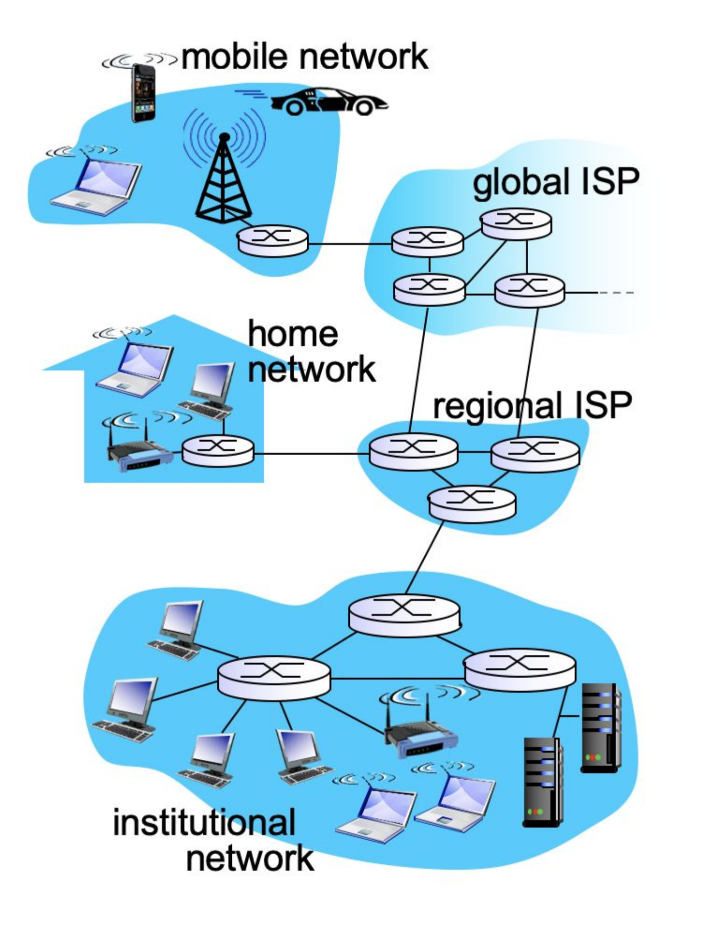
Application	• End User layer • HTTP, FTP, IRC, SSH, DNS
Presentation	• Syntax layer • SSL, SSH, IMAP, FTP, MPEG, JPEG
Session	• Synch & send to port • API's, Sockets, WinSock
Transport	• End-to-end connections • TCP, UDP
Network	• Packets • IP, ICMP, IPSec, IGMP
Data Link	• Frames • Ethernet, PPP, Switch, Bridge
Physical	• Physical structure • Coax, Fiber, Wireless, Hubs, Repeaters

Getting Connected

The Internet is a ‘Network of Networks’:
tons of interconnected ISPs and protocols

Connections can come in many forms

- Wireless
  - WiFi
  - Cellular
  - Satellite
- Wired
  - Ethernet
  - Fiber
  - Direct Attach Copper
  - USB



A closer look at network structure

Network Edge

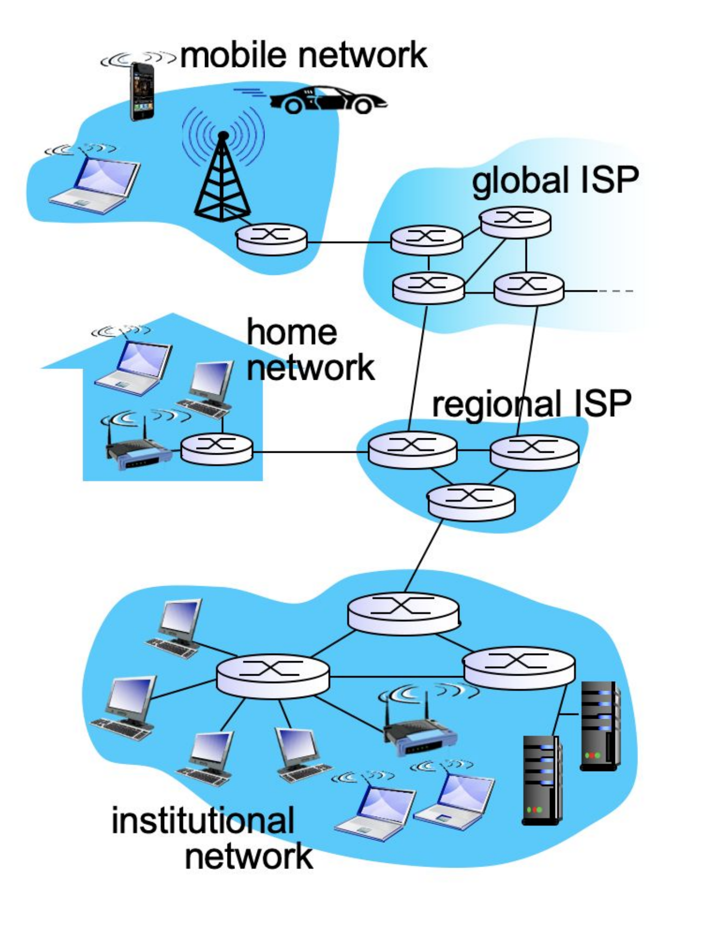
- Hosts - Clients and Servers
  (A device connected to the network)

Access Networks

- Communication links connecting the Network Edge to the Network Core

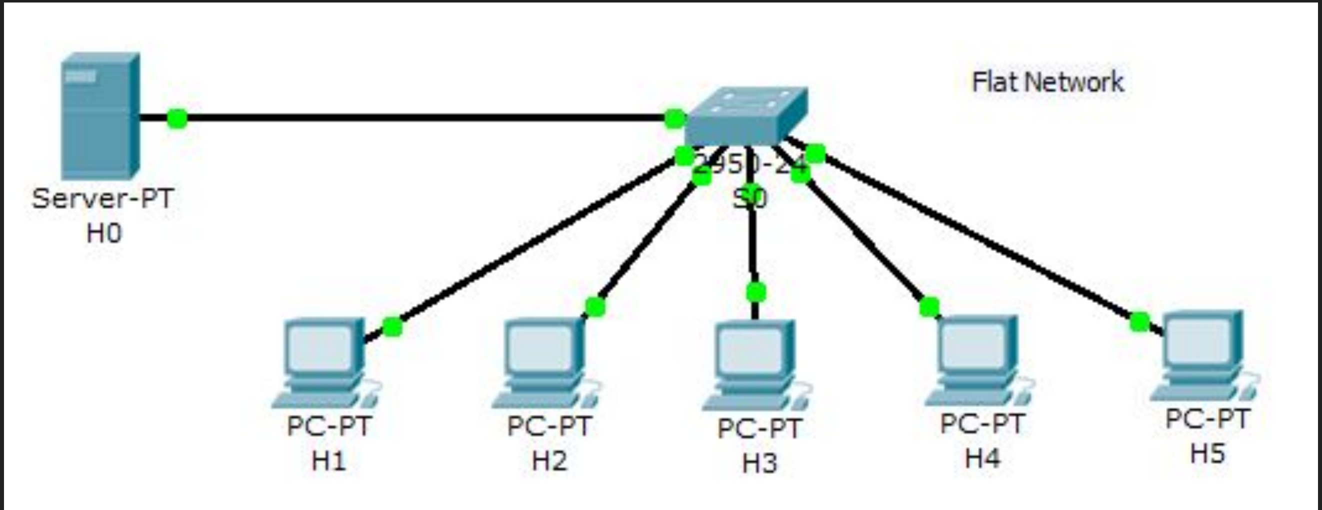
Network Core

- Interconnected Routers
- “Network of Networks”



What topologies exist for network designs?

- Flat Network - not segmented or separated into different broadcast areas by using routers
  - Easy to design and implement
  - Easy to maintain
  - Ideal for small networks



Can’t we just have one big network?

This gets a bit tricky - for small environments it will work just fine however for environments with a lot of hosts congestion and broadcast floods can cause problems

So what options do we have?

- Subnets! - Breaking one network into multiple smaller networks

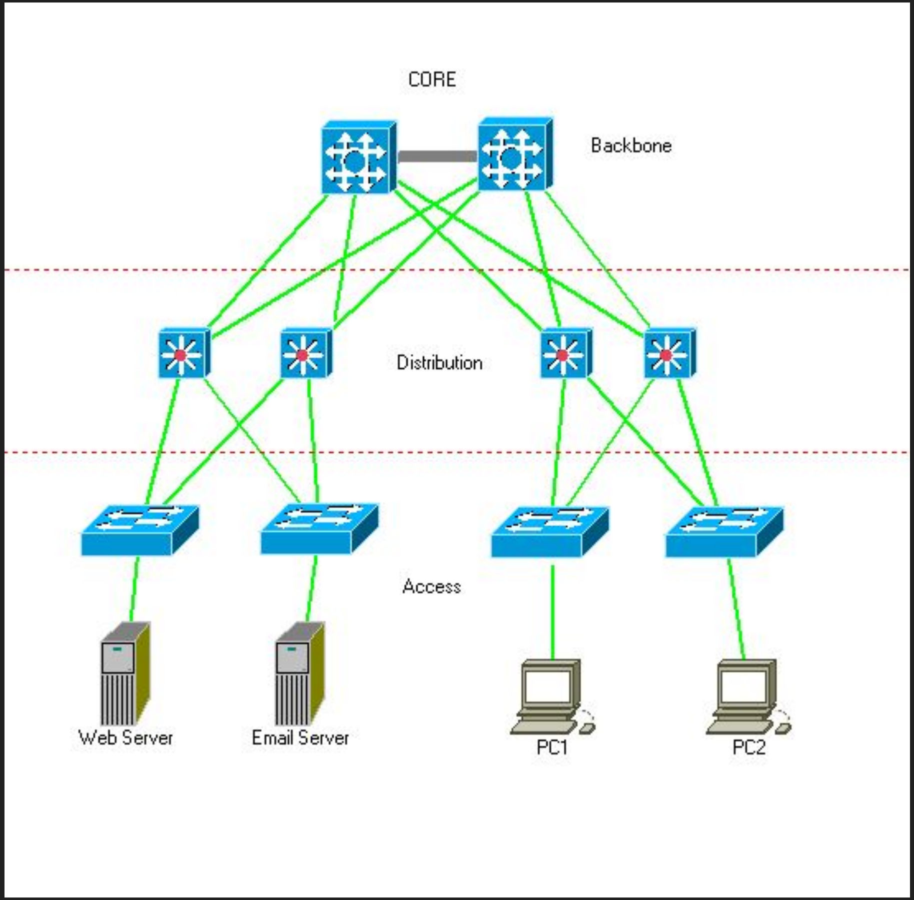
Q: What about security?

- Do all devices need to be able to communicate with each-other?

Can’t we just have one big network?

- Hierarchical Networks
  - The network is broken up into logical segments - usually based on functionality

- A typical hierarchical topology contains the following layers
  - A core layer of high-end routers and switches that are optimized for availability and performance
  - A distribution layer of routers and switches that implement policies
  - An access layer that connects users via lower-end switches and wireless access points



Obtaining an IP Address

What network space is available for us to use?

RFC1918
Determined network ranges for private networks

- 10.0.0.0/8
- 172.16.0.0/20
- 192.168.0.0/16

Lets go over CIDR notation and subnets

Classless Inter-Domain Routing

CIDR Notation: a way to convey network and subnet information in a compact format

Example: 192.168.0.0/24

- IP Range - 192.168.0.0 - 192.168.0.255
- Subnet Mask 255.255.255.0
- Total of 256 addresses (254 usable)

The “/number” determines how many network bits and how many host bits are part of the address. Therefore for a /24 network there are 24 network bits and 8 host bits.

Classless Inter-Domain Routing

How to ‘manually’ convert CIDR to address range and subnet

- i.e. Convert the range 18.125.128.0/17 to an IP range and subnet mask
- Step 1 - Determine the Subnet
  - In CIDR notation, the ‘/’ determines how many bits are for the network, and how many are for the host.
  - In our example that would mean 17 bits for the network
  - We can then represent the “Subnet Mask” as 17 1’s followed by the remainder of 0’s -> 11111111 11111111 10000000 0000000.
  - We then can convert each 8-bit block back to decimal
  - 11111111 -> 255   11111111 -> 255   10000000 -> 128   00000000 -> 0

So, our subnet mask is 255.255.128.0

Classless Inter-Domain Routing

- Step 2 - Convert the address to 8 bit binary
  - 18 -> 00010010 125 -> 01111101 128-> 10000000 0->00000000
  - We have the lower bound of the address given to us, now we need to determine the upper bound
  - To do this we take our lower bound address in binary and compare it with our subnet mask
  - 00010010 01111101 10000000 00000000 <- lower bound
  - 11111111 11111111 10000000 00000000 <- subnet mask
  - We can then set all the bits that are not 1 in the subnet mask to 1 in our lower bound to find the top of the range
  - 00010010 01111101 11111111 1111111 <- this is the upper bound
  - Converting that back to decimal yields 18.125.255.255

Classless Inter-Domain Routing

- Therefore, 18.125.128.0/17 has a subnet mask of 255.255.128.0 and a range of 18.125.128.0 - 18.125.255.255
- Also as there are 17 bits reserved for the network, there are 15 remaining for the hosts. This means we have 2^15 (32768) total addresses in this subnet

Classless Inter-Domain Routing

Subnet address	Netmask	Range of addresses	Useable IPs	Hosts	Divide	Join
192.168.0.0/17	255.255.128.0	192.168.0.0 - 192.168.127.255	192.168.0.1 - 192.168.127.254	32766	Divide	/17 /16
192.168.128.0/18	255.255.192.0	192.168.128.0 - 192.168.191.255	192.168.128.1 - 192.168.191.254	16382	Divide	/18 /17
192.168.192.0/19	255.255.224.0	192.168.192.0 - 192.168.223.255	192.168.192.1 - 192.168.223.254	8190	Divide	/19 /18
192.168.224.0/20	255.255.240.0	192.168.224.0 - 192.168.239.255	192.168.224.1 - 192.168.239.254	4094	Divide	/20 /19
192.168.240.0/21	255.255.248.0	192.168.240.0 - 192.168.247.255	192.168.240.1 - 192.168.247.254	2046	Divide	/21 /20
192.168.248.0/22	255.255.252.0	192.168.248.0 - 192.168.251.255	192.168.248.1 - 192.168.251.254	1022	Divide	/22 /21
192.168.252.0/23	255.255.254.0	192.168.252.0 - 192.168.253.255	192.168.252.1 - 192.168.253.254	510	Divide	/23 /22
192.168.254.0/24	255.255.255.0	192.168.254.0 - 192.168.254.255	192.168.254.1 - 192.168.254.254	254	Divide	/24 /23
192.168.255.0/24	255.255.255.0	192.168.255.0 - 192.168.255.255	192.168.255.1 - 192.168.255.254	254	Divide	/24

How big of a subnet do we need?

Depends on how many devices we need to connect - including:

- Virtual Machines
- Physical Servers
- Management Interfaces
- Switches
- Routers

Usually informed by Capacity Planning and Monitoring

- Often critical to plan for growth - re-configuring a network can be very messy (Q: Why?)

How big of a subnet do we need?

Example: Company ABC is currently rolling out a new system. They have performed capacity planning and determined that they will require

- 250 VMs to serve the application
- 10 VMs per host
- Each host has 3 management interfaces

What is the smallest subnet they would require for this project?

- 250 + (250/10 * 3) = 325
- 2^8 < 325 IPs < 2^9
- 32 - 9 = 23 -> A /23 is the smallest subnet that can handle this project

Obtaining an IP Address

Static IP configuration

- ISP or Network Admin provides configuration
- "Static" - fixed/permanent and not subject to change
- Often cumbersome - imagine needing a static configuration in a Coffee Shop, at home, at work, at school (and they're all different!)

Dynamic IP Configuration (DHCP)

- A Host broadcasts that it needs an IP address to the network
- The request is sent to a DHCP server, if it accepts the request it sends back the IP, Subnet Mask, Default Gateway, and even DNS, NTP, and Print servers for the Host to use for a specific amount of time (Known as a "DHCP Lease")

Connectivity isn’t the only requirement

Imagine if the Internet still used Dial-Up

- What’s the problem? You can still access all of the services you need to.

Throughput/Bandwidth!

If your application is serving a static webpage it requires much less throughput than if it is a video streaming platform

- Back to Bottlenecks again:
  Throughput is only as high as the slowest link in the chain

How do we calculate throughput?

By calculating two parts: current throughput and required throughput

Current throughput

- Determine the capacity of each link between Host A and Host B and determine the bottleneck
    - Ex. Host A --- 10G --- 1G --- 50G --- Host B (Throughput is limited by the 1G link)

What if there are multiple Hosts that share a link?

- Assuming the link is shared equally, divide the link capacity by the number of hosts.
- Using the example above, if there are 20 Hosts also connected to and utilizing the 10G link, its speed would be 512M - thus becoming the bottleneck and limiting throughput

How do we calculate throughput?

Throughput Requirements

- Usually determined similar to other forms of capacity planning
- Use application/system requirements obtained by developer requirements or by using heuristics, multiply that by the amount of concurrent instances

Theoretical Throughput can be determined by taking the amount of hosts and dividing by the shared link speed.

Measuring ‘Actual’ Throughput

- “Speed Test” - speedtest.net, fast.com, speakeasy.net, etc
- iperf3 - performance testing from one host to another
- Link Metrics on gateways/switches

What’s next?

So we have a network with enough IPs and throughput for all of our Hosts, are we done?

- Short answer, no.

Let’s walk through the day in the life of a web request and figure out what we are missing.

What happens when we visit a website?

A day in the life of a web request

The Basics

1. Acquiring an IP address for our computer
2. ***Acquiring the IP address of the server to request data from***
3. Establishing a connection to the server
4. Negotiating a secure connection
5. Requesting content

DNS

The Domain Name System

- Imagine trying to access Reddit, Twitter, <insert website here>, and needing to remember the IP address for each one
- People are terrible at remembering numbers - so we use names instead

DNS handles **resolving** domain names like www.youtube.com to an IP address for you

Q: How does it accomplish this?

DNS

The Domain Name System is

- A distributed database implemented in hierarchy of many name servers
  - Why make it a distributed system?
    - Avoid a single point of failure
    - Sheer traffic volume
    - Distant centralized database would cause high latency
    - Maintenance/Downtime
- A core Internet function, implemented as an application-layer protocol
- Moves complexity to the edge of the network

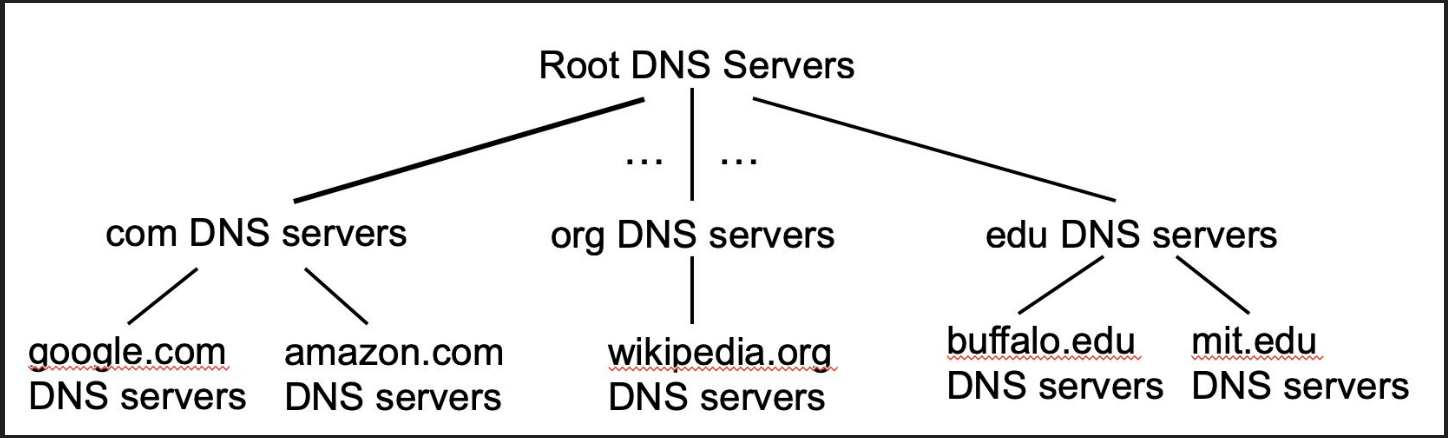
DNS: Types of DNS Servers

Root - responsible for maintaining records for TLD name servers

TLD - responsible for com, org, net, edu, ... and all top-level country domains

Authoritative - organization’s own DNS server(s), providing authoritative hostname to IP mappings for organization’s named hosts

Local - the DNS server that clients are configured to request resolutions from



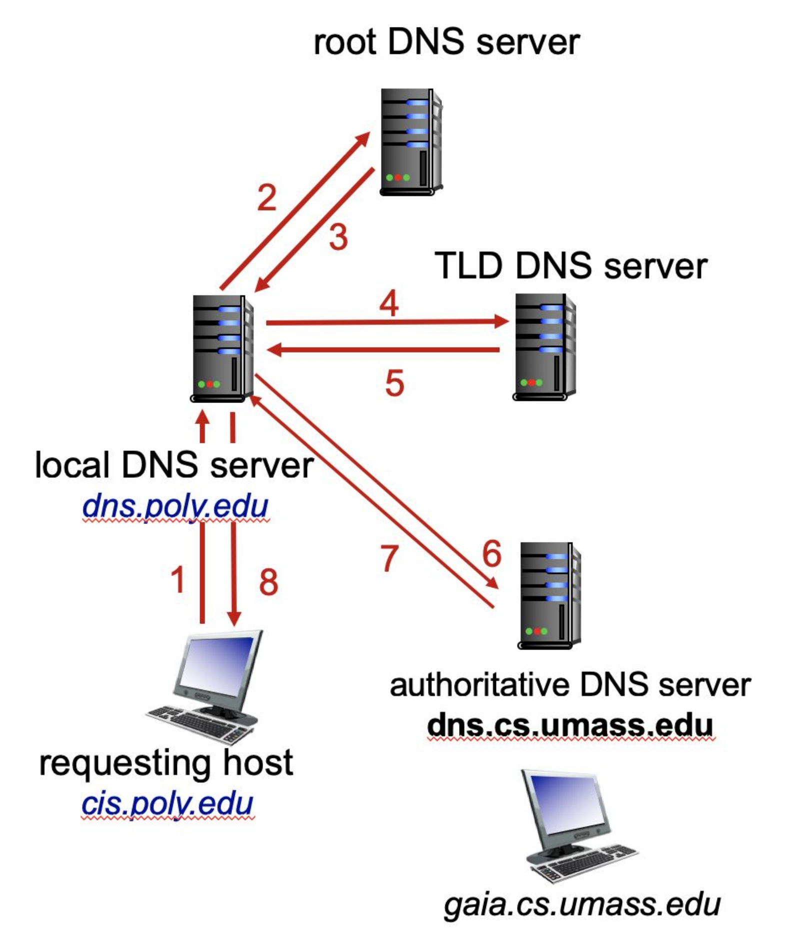
DNS

Example:

Host at cis.poly.edu wants IP address for gaia.cs.umass.edu

Iterated query:

Contacted server replies with name of server to contact, Eg. “I don’t know this name, go ask this server”



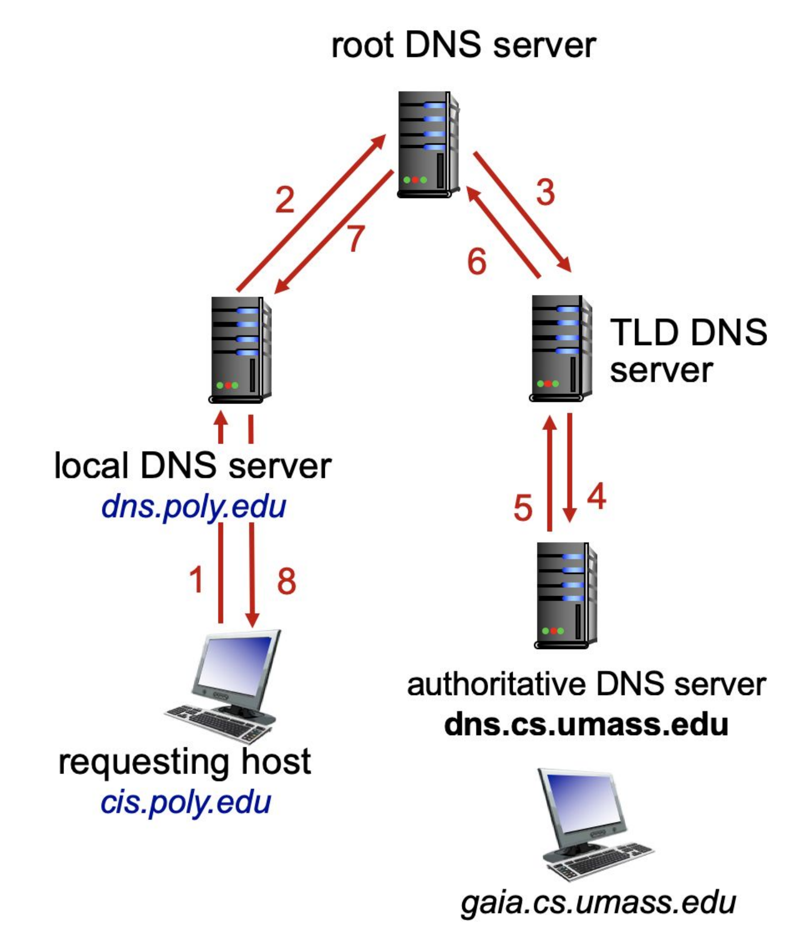
DNS

Example:

Host at cis.poly.edu wants IP address for gaia.cs.umass.edu

Recursive query:

Puts the burden of name resolution on the contacted name server.



DNS Caching

Once (any) name server learns a mapping, it caches the mapping

- Cache entries timeout (disappear) after some requested time (TTL)

Pro's

- Faster Lookups!
- TLD servers typically always cached in local name servers
  - Therefore, root name servers not often visited

Con's

- Cached entries may be out-of-date
  - If name host changes IP address, may not be known Internet-wide until all TTLs expire

DNS Records

DNS records are made up of a Type, Name, Value, and TTL

Types of DNS Records

- **A Name** is a hostname (i.e. buffalo.edu) Value is an IP address (10.128.136.234)
- **CNAME** - Name is an Alias (www.buffalo.edu), Value is the ‘real’ name (buffalo.edu)
- **NS** - Name is a domain (i.e. buffalo.edu) Value is the hostname of the authoritative name server for that domain
- **MX** - Name is a domain (i.e. buffalo.edu) Value is the hostname of the mail server to receive the mail, Priority determines order of servers
- **TXT** - Name is an arbitrary string, as is Content (used for extended uses - DNS validation, SPF for mail, etc)

What happens when we visit a website?

A day in the life of a web request

The Basics

1. Acquiring an IP address for our computer
2. Acquiring the IP address of the server to request data from
3. Establishing a connection to the server
4. ***Negotiating a secure connection***
5. Requesting content

Brief mention of SSL/TLS

We will be spending an entire lecture on this topic, however it is important to mention why they need to be thought about during the planning phase

Quick breakdown of a cert

- Common Name
- Subject Alt Names
- Key Usages
- Certificate Authorities

Can take a while to be issued and can cost lots of money

- Want to ensure we have the DNS names and/or IP addresses locked in and planned before issuing certs as they cannot be edited once issued

What happens when we visit a website?

A day in the life of a web request

The Basics

1. Acquiring an IP address for our computer
2. Acquiring the IP address of the server to request data from
3. Establishing a connection to the server
4. Negotiating a secure connection
5. ***Requesting content***

Time to finally make requests

Once the TCP connection has been established and SSL/TLS has been negotiated, the connection is ready to finally make requests

- The Protocol can be HTTP, QUIC, a Custom Protocol, etc...

Scaling

Now that we have a basic understanding of how a Client requests data from a server, let’s talk Scaling/High Availability.

Q: Where can we scale up? Where can we optimize?

- How do we ensure clients don’t just all go to one server?
- Where can we implement caching?

Questions?

What questions do you have?

Key Ideas

- Day in the life of a web request (What systems/protocols are used?)
- CIDR Notation
- Different network layouts (Flat, Star, Tree)
- How to determine how big of a subnet you need
- How to calculate throughput
- What components need to scale? / How do we scale?
- What if we run out of network space?

References

- Computer Networking: A Top Down Approach. 6th edition Jim Kurose, Keith Ross Addison-Wesley - March 2012
- Visual Subnet Calculator - http://www.davidc.net/sites/default/subnets/subnets.html