Domain Name System (DNS)

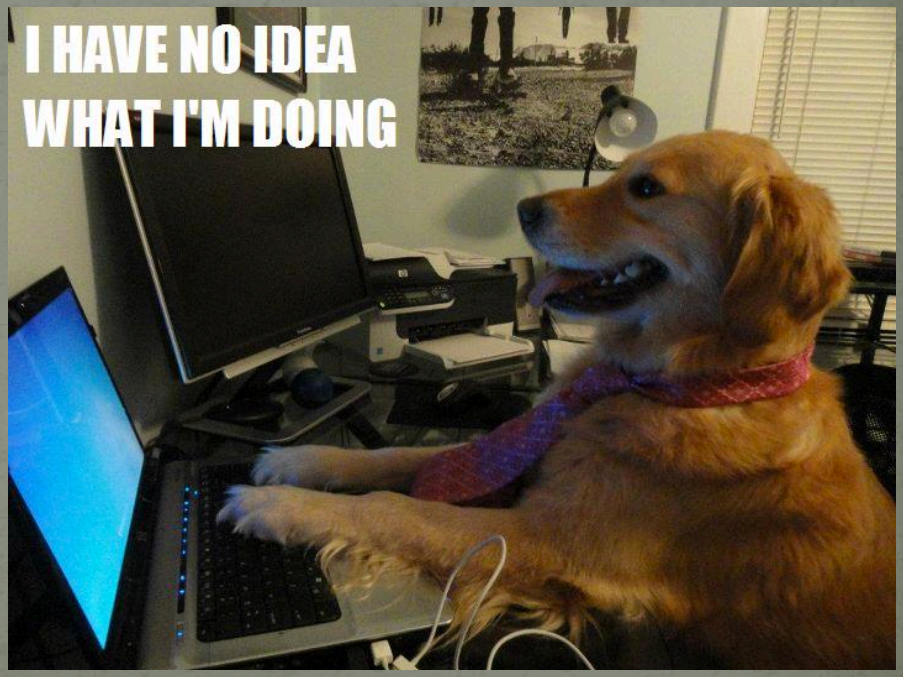
CSE 461 Section

Addressing So Far

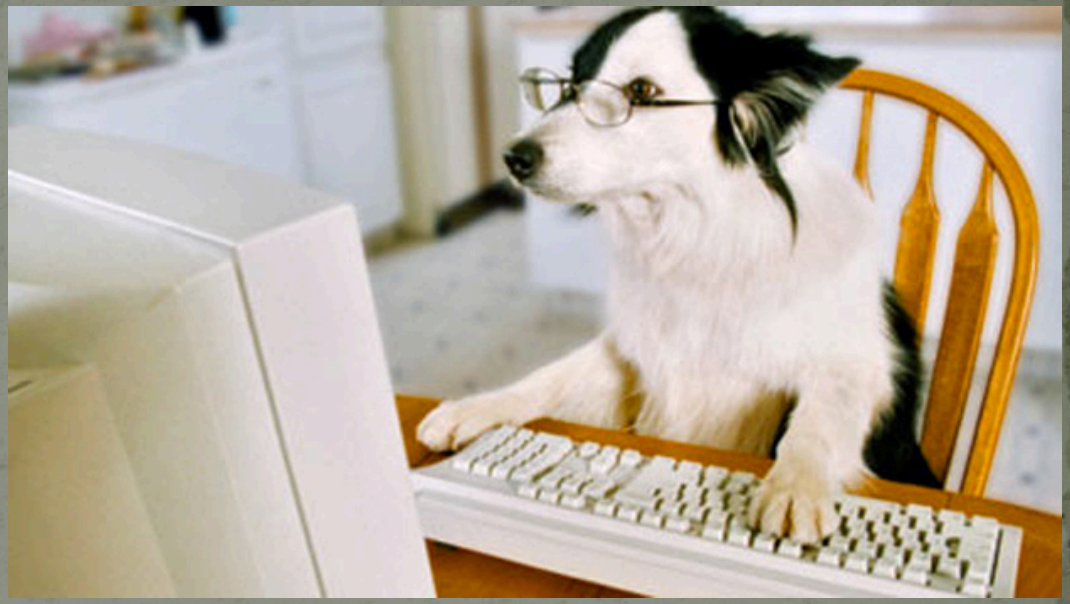
- Port numbers for applications
- MAC addresses for hardware
- IP addresses for a way to send data in a smart, routable way

Problems with MACs/IPs/Ports

- Humans are bad at remembering strings of numbers
- We need a human-friendly naming system!



Requirements for Human-Readable Naming System



- What do we need?
  - As short as possible
  - Easy to memorize (i.e., not arbitrary)
  - Unique
  - Customizable
  - Hierarchical
  - Reflect organizational structure
  - A way to quickly translate to and from the existing, computer-friendly addressing systems
  - Ideally, we'd like to address specific resources as well

Domain Names

- Human-readable "domain names" map to IP addresses (names < 254 characters)
- A human can type www.google.com into their browser, and the browser will (somehow) know to go to 173.194.33.179
- But how might this be done?
  - Some sort of hash (not really practical)
  - A file of all of the mappings
  - Separate servers to provide the mappings

Hierarchical DNS Servers

- Systems keep a small cache of mappings they know
- When a domain name is used that isn't in the cache, the system queries a name server
- Simple UDP communication on port 53
- Database is distributed
- Hierarchical namespace: it's name servers all the way down

DNS Protocol

- Series of Question/Response messages

DNS Message Format
DNS header (fixed length)
Question entries (variable length)
Answer resource records (variable length)
Authority resource records (variable length)
Additional resource records(variable length)

DNS Protocol – Question Entries

- Questions contain 3 fields:

Question Name	Question Type	Question Class

- Name:
  - What resource we are querying for: `0x6google0x3com0x0`
- `Type:`
  - Can specify what we are trying to resolve for: mail, IPv4, ns…
- `Class:`
  - Usually set to internet class, capable of being others

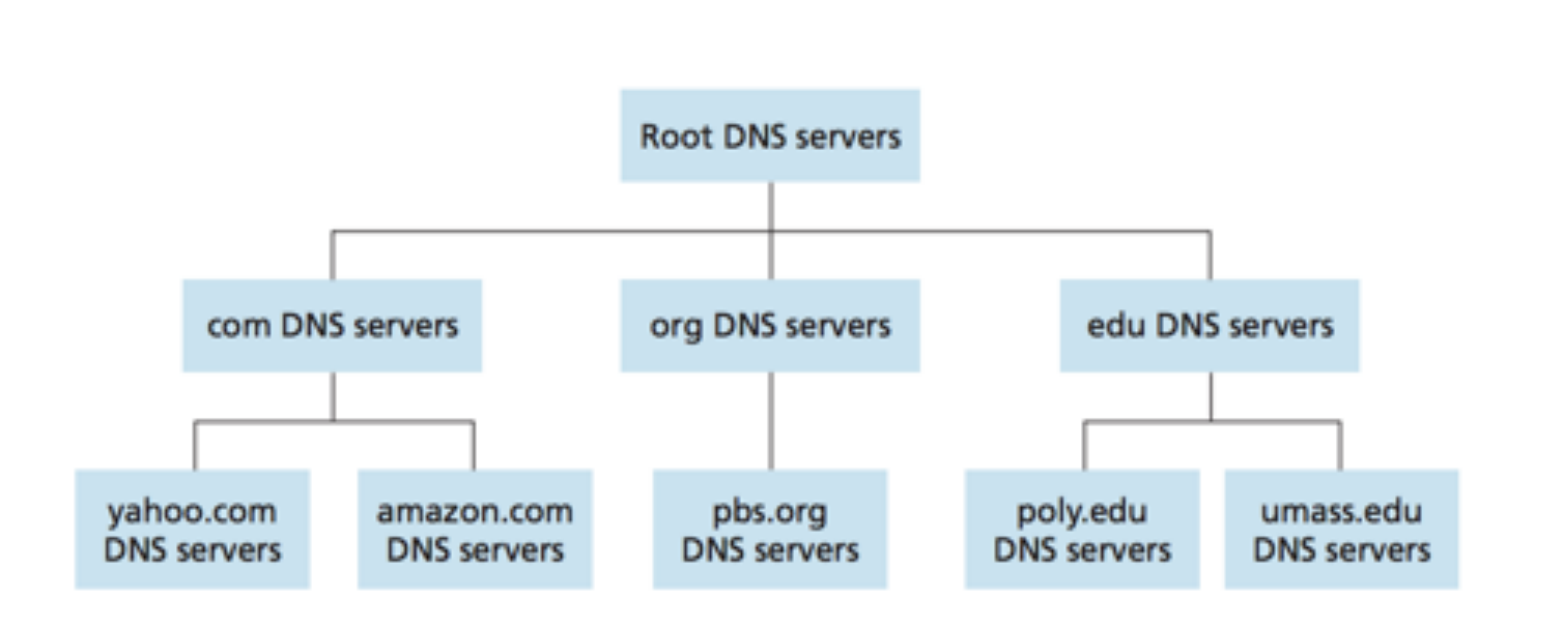
DNS Protocol – Answer Resource Records

- Resource Record:

Name	Type	Class	TTL	Data Length	Data

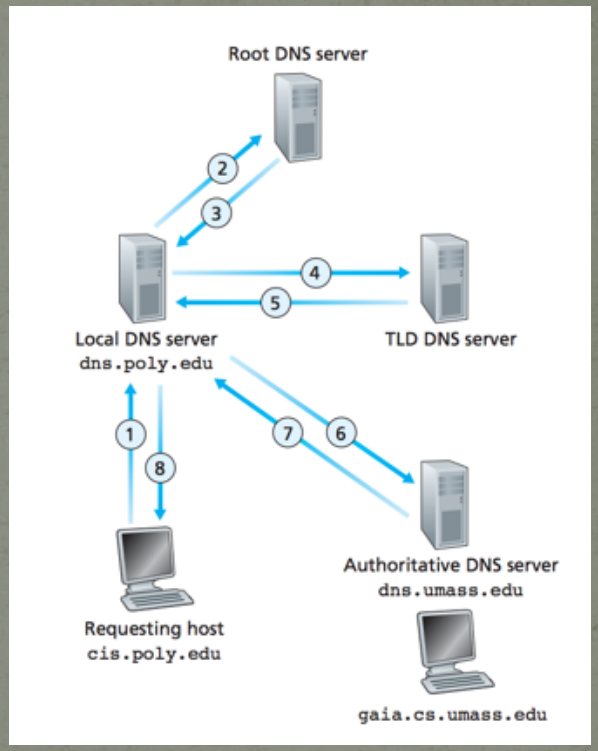
- Name/Type/Class same as before
- Time-To-Live:
  - Lease time this record will be valid to cache for
- Data:
  - Whatever the Type specifies for the data

Domain Hierarchy



Resolving a Domain Name

- If I type sports.huskies.com, what happens?
  - Check /etc/hosts
  - Check DNS cache
  - Check local DNS server
  - Go down hierarchy and ask:
    - Ask . DNS root server
    - Ask .com TLD (Top Level
    - Domain) server
    - Ask huskies.com's NS
    - Send HTTP request to the IP address obtained



Local DNS Server

- “A local DNS server does not strictly belong to the hierarchy of servers but is nevertheless central to the DNS architecture. Each ISP—such as a university, an academic department, an employee’s company, or a residential ISP—has a local DNS server (also called a default name server). ”

Multiple IP Addresses and Aliasing

- DNS servers can return different IP address results for the same domain name
- Why is this useful?
- Also, multiple domain names can map to one IP address
- Why is this useful?

Attacks and Other Fun

- What are some ways this system can break?
  - DoS attacks on DNS server
    - Done before, in 2002 and 2007
    - Not much impact due to filtering and caching
  - Return incorrect IP address to a DNS request
  - Could even return the IP of our own server!
    - Commonly done by ISPs
  - Compromise root servers

DNS Usages – Load Balancing

- Take advantage of multiple IP aliasing to round robin deliver services different IP addresses
- Linux queries IP of first record returned from DNS
- IP address returned does not guarantee that host is available