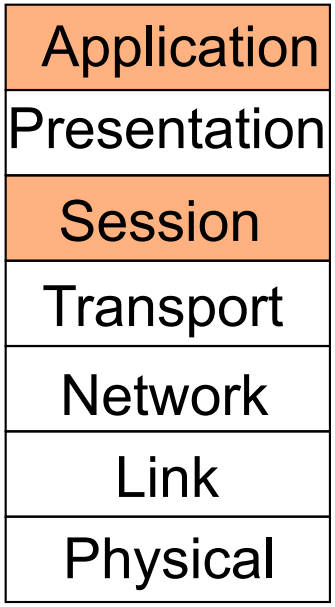
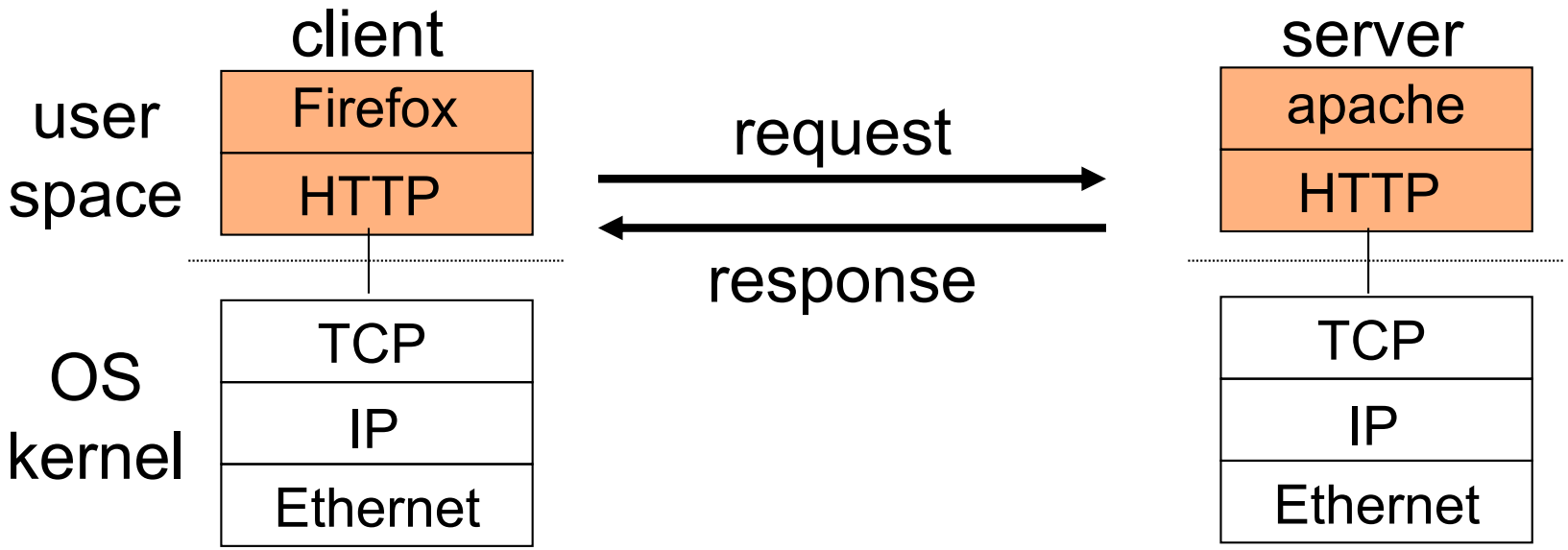
The application layer

- Topics
  - DNS
  - CDNs
  - HTTP, HTTP/1.1, HTTP/2



Application layer on top of other abstractions



What is the applicaton?

What is the application layer protocol?

Application layer protocol

- Support application
- E.g.,
  - HTTP allows transfer of objects. Used by mobile apps, Web, and many other applications
  - DASH allows transfer of video streams. Used by several video streaming applications
- Protocols can also be used for support of application
  - E.g., Domain Name Service or DNS

Domain Name Service*

*Some slides borrowed from D. Choffnes (NEU)

Internet Names and Addresses

- Addresses, e.g. 129.49.2.176
  - Computer usable labels for machines
  - Conform to structure of the network
- Names, e.g. www.stonybrook.edu
  - Human usable labels for machines
  - Conform to organizational structure
- How do you map from one to the other?
  - Domain Name System (DNS)

History

- Before DNS, all mappings were in *hosts.txt*
  - */etc/hosts* on Linux
  - *C:\Windows\System32\drivers\etc\hosts* on Windows
- Centralized, manual system
  - Changes were submitted to SRI via email
  - Machines periodically FTP new copies of *hosts.txt*
  - Administrators could pick names at their discretion

Towards DNS

- Eventually, the *hosts.txt* system fell apart
  - Not scalable, SRI couldn't handle the load
  - Hard to enforce uniqueness of names
    - e.g MIT
      - Massachusetts Institute of Technology?
      - Melbourne Institute of Technology?
  - Many machines had inaccurate copies of *hosts.txt*
- Thus, DNS was born

What are some of the desired properties of a DNS?

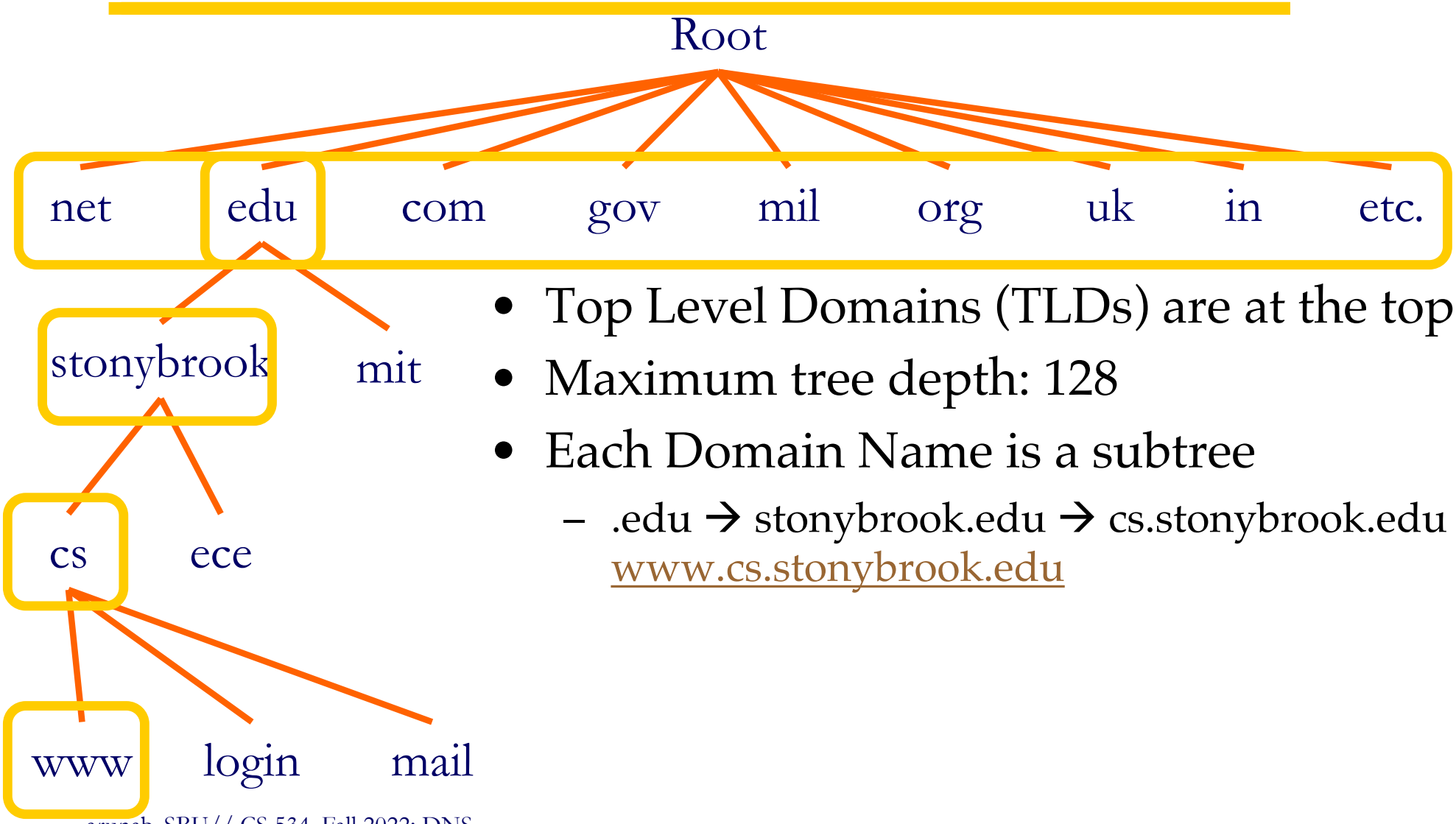
- Scalable
- Fault Tolerant
- Low latency
- Universally accessible

How can we achieve these properties?

- Hierarchical (Scalability)
- Geographically distributed (Universal accessibility)
- Several replicas (Fault tolerance)
- Anycast (Low latency)

DNS Structure

The key design choice: Hierarchy



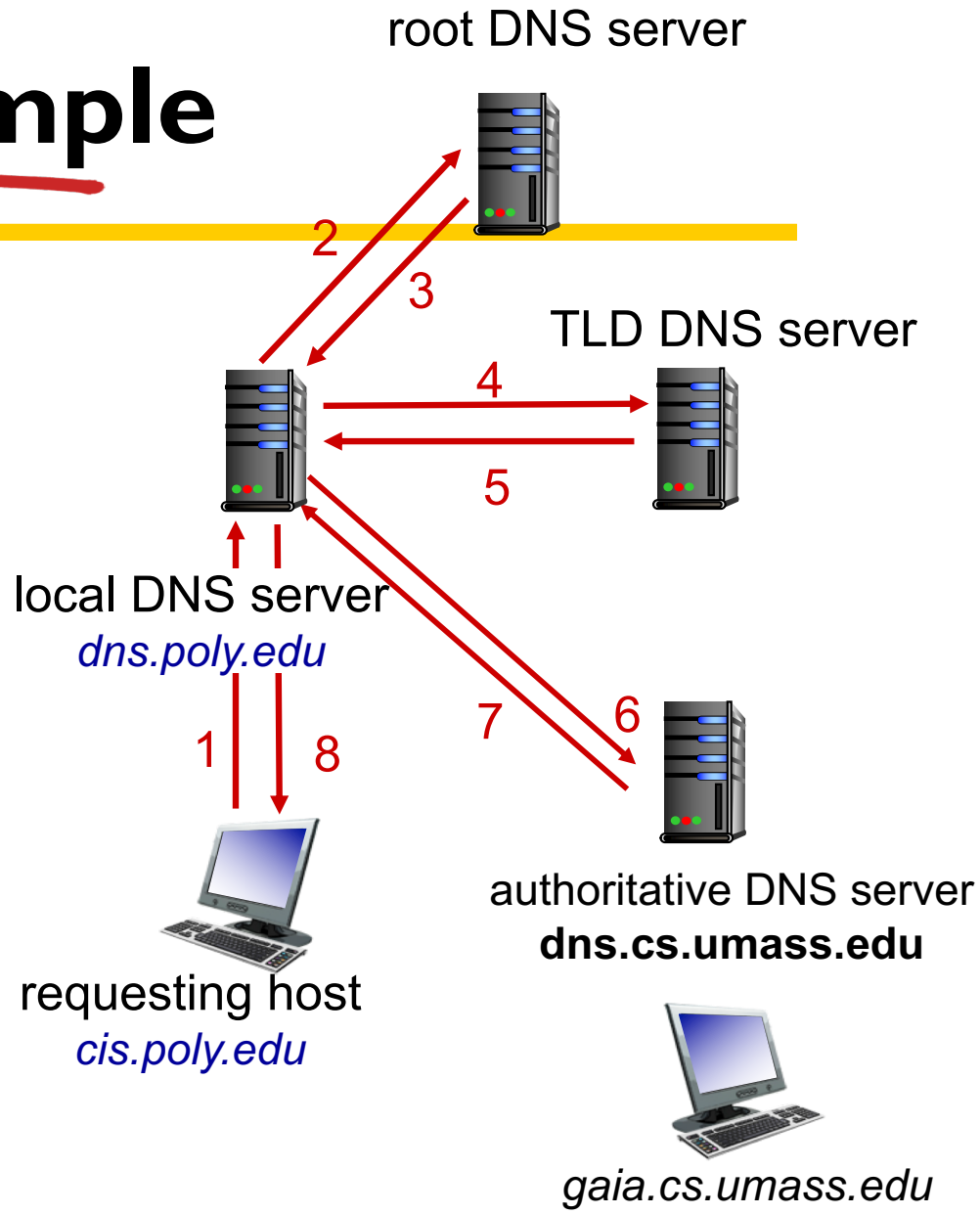
- Top Level Domains (TLDs) are at the top
- Maximum tree depth: 128
- Each Domain Name is a subtree
  - .edu → stonybrook.edu → cs.stonybrook.edu → www.cs.stonybrook.edu

DNS name resolution example

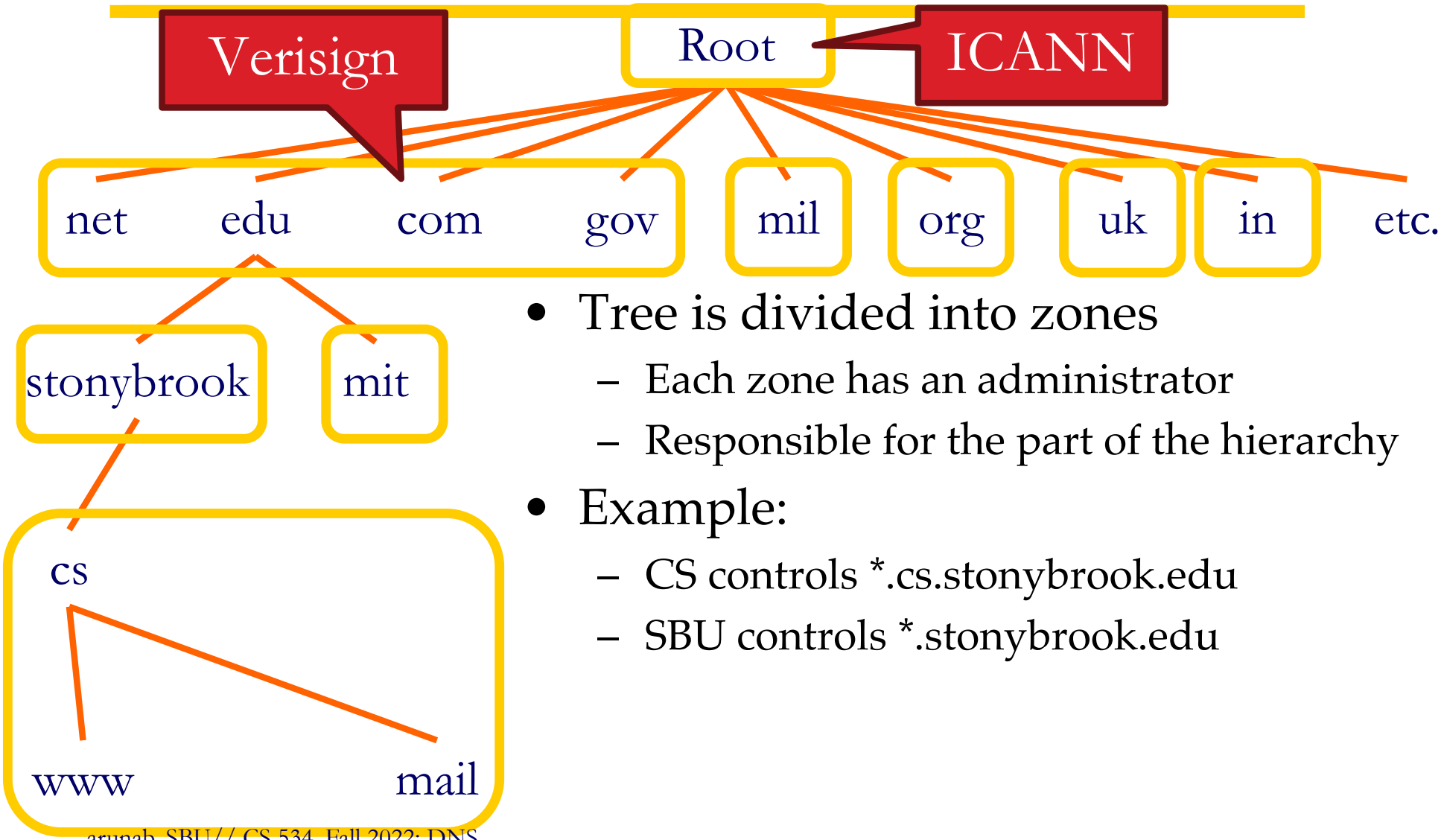
- host at cis.poly.edu wants IP address for gaia.cs.umass.edu

iterated query:

- contacted server replies with name of server to contact
- "I don't know this name, but ask this server"



Administration



- Tree is divided into zones
  - Each zone has an administrator
  - Responsible for the part of the hierarchy
- Example:
  - CS controls *.cs.stonybrook.edu
  - SBU controls *.stonybrook.edu

Basic Domain Name Resolution

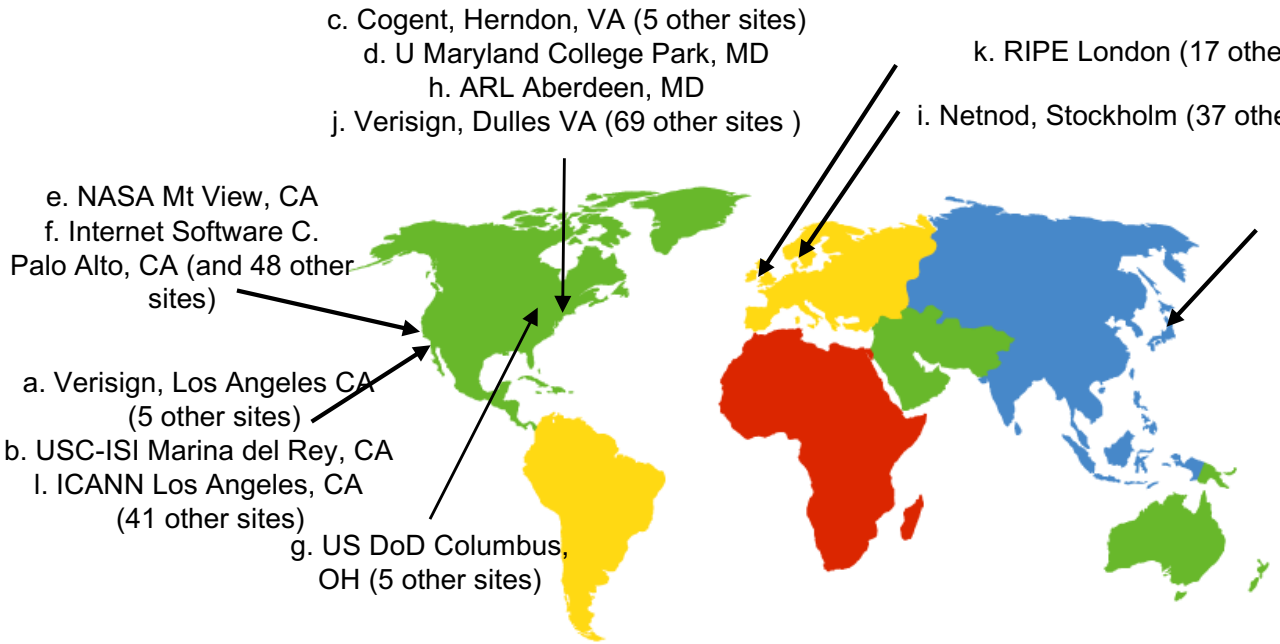
- Every host knows a local DNS server/resolver (How?)
  - Sends all queries to the local DNS server
- If the local DNS can answer the query, then you're done
- Otherwise, go down the hierarchy and search for the **authoritative name server**

Some terminology

- Top Level Domain
- Name server/DNS server
- Authoritative name server
- DNS resolver/Local DNS server
- DNS request
- DNS response

DNS: root name servers

- contacted by local name server that can not resolve name
- 13 root name servers worldwide.



13 logical root name "servers" worldwide

- *each "server" replicated many times*

TLD, authoritative servers

top-level domain (TLD) servers:

- responsible for com, org, net, edu, aero, jobs, museums, and all top-level country domains, e.g.: uk, fr, ca, jp
- Network Solutions maintains servers for .com TLD
- Educause for .edu TLD

authoritative DNS servers:

- organization's own DNS server(s), providing authoritative hostname to IP mappings for organization's named hosts
- can be maintained by organization or service provider

Local DNS name server

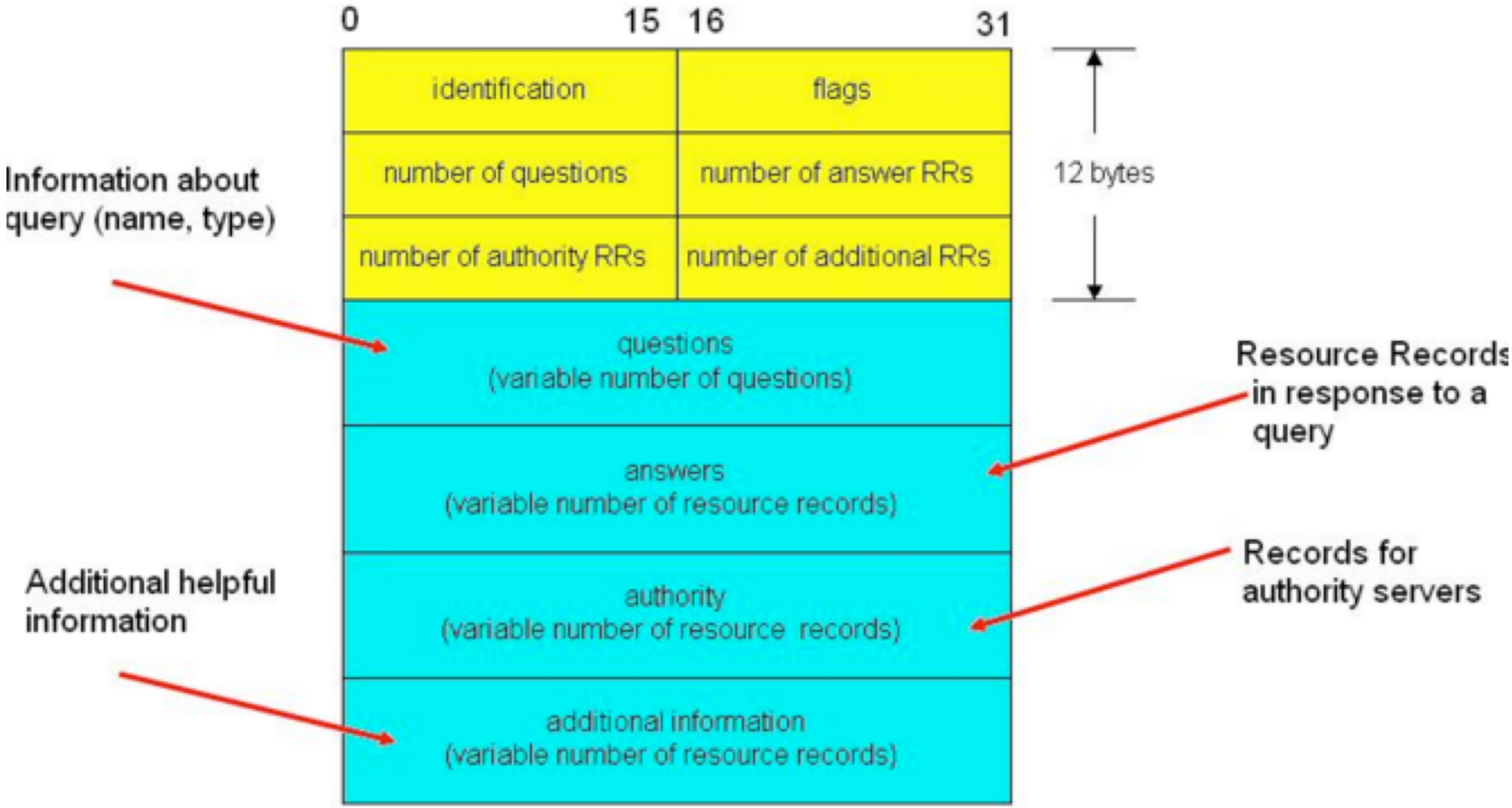
- does not strictly belong to hierarchy
- each Internet Service Provider (AT&T, Comcast, university, companies) have one
  - also called "default name server"
- when host makes DNS query, query is sent to its local DNS server
  - has local cache of recent name-to-address translation pairs (but may be out of date!)
  - acts as proxy, forwards query into hierarchy

DNS Caching

- Performing all these queries take time
  - And all this before the actual communication takes place
  - E.g., 1-second latency before starting Web download
- Caching can substantially reduce overhead
  - The top-level servers very rarely change
  - Popular sites (e.g., www.cnn.com) visited often
  - Local DNS server often has the information cached
- How DNS caching works
  - DNS servers cache responses to queries
  - Responses include a "time to live" (TTL) field
  - Server deletes the cached entry after TTL expires

DNS structure.

DNS packet format



DNS Resource Records

- Resource record has four fields: (name, value, type, TTL)
  - There may be multiple records returned for one query (Why?)

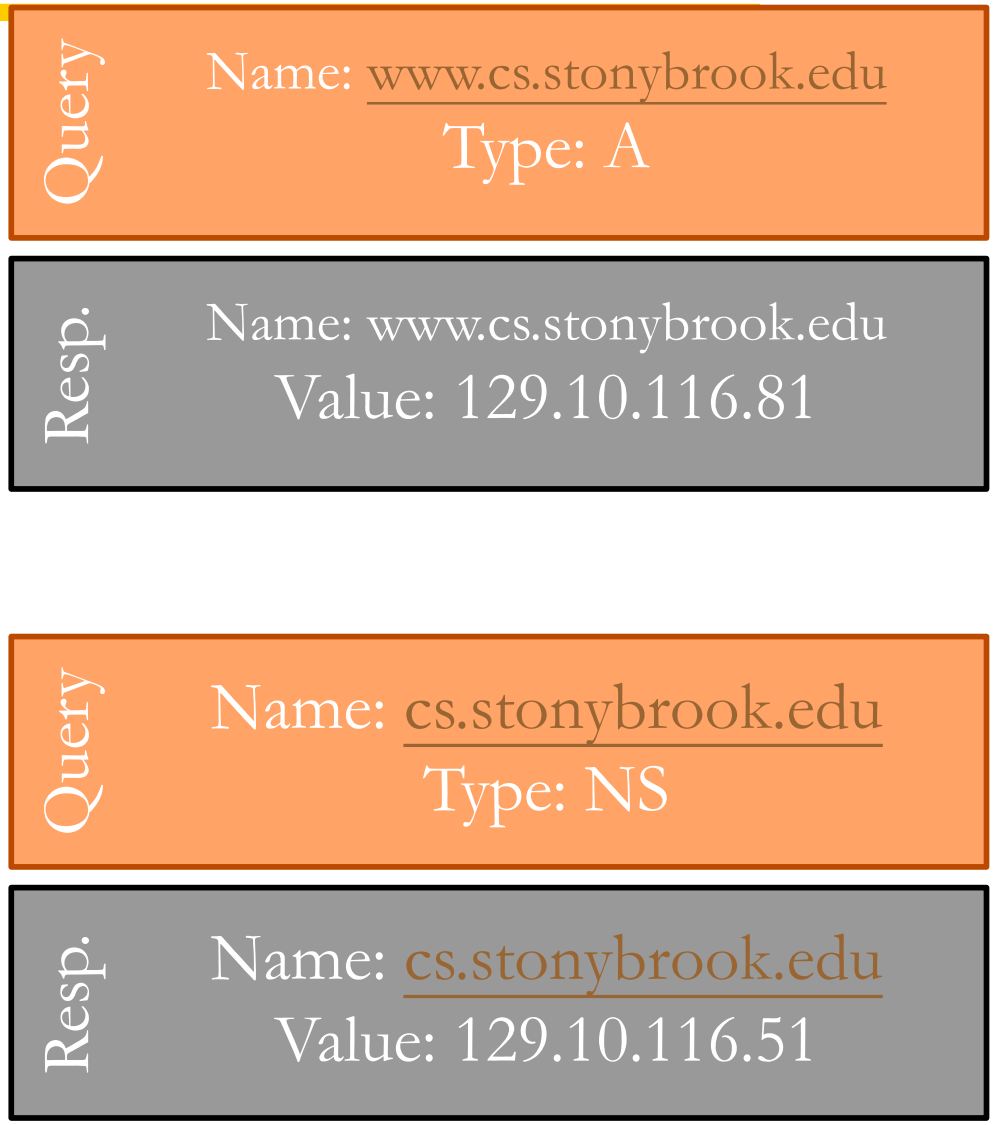
- Fields depends on the type of query and response.
  - Name: ID
  - TTL: time to live.
  - Type: MX Record (mail server), NS record (name server), CName record (canonical name), A record (IPv4)
  - Value: Address

- Use the "dig" command to get DNS records

DNS Types

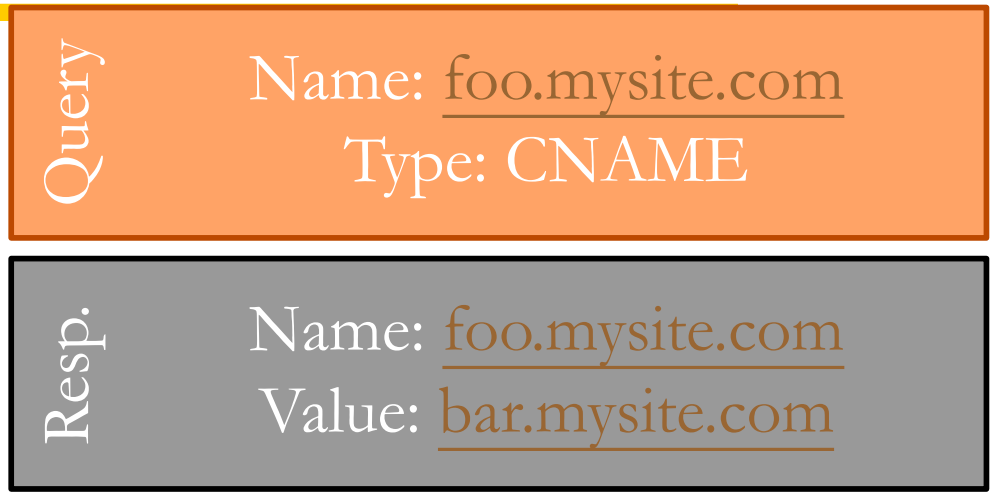
- Type = A / AAAA
  - Name = domain name
  - Value = IP address
  - A is IPv4, AAAA is IPv6

- Type = NS
  - Name = partial domain
  - Value = name of DNS server for this domain
  - "Go send your query to this other server"



DNS Types, Continued

- Type = CNAME
  - Name = hostname
  - Value = canonical hostname
  - Useful for aliasing
  - CDNs use this



- Type = MX
  - Name = domain in email address
  - Value = canonical name of mail server

A record versus NS record

foo.com.	IN	NS	ns1.bar.com.
foo.com.	IN	A	192.168.100.1

A Record = "The host called foo.com lives at address 192.168.100.1"
NS Record = "If you want to know about hosts in the foo.com zone, ask the name server ns1.bar.com"

How to use DNS in practice.

DNS Bootstrapping

- Need to know IP addresses of root servers before we can make any queries
- Addresses for 13 root servers ([a-m].root-servers.net) https://www.iana.org/domains/root/servers

Who is my local DNS server?

- Need to know the local DNS server. (/etc/resolv.conf)
  - Who is your local DNS server at home?
  - At school?
  - Public DNS?

- What if I want to create my own domain?
  - Pay someone to add your DNS entry
  - E.g., Amazon Router 53

Importance of DNS

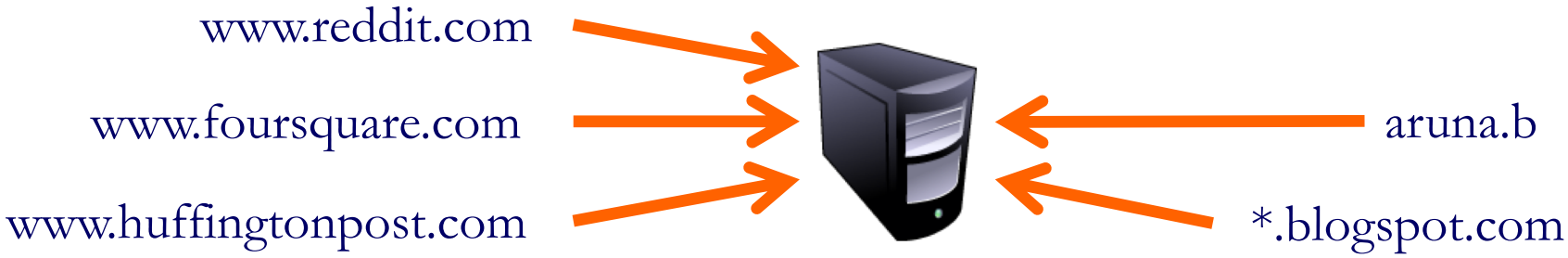
DNS as Indirection Service

Changing the IPs of machines becomes trivial

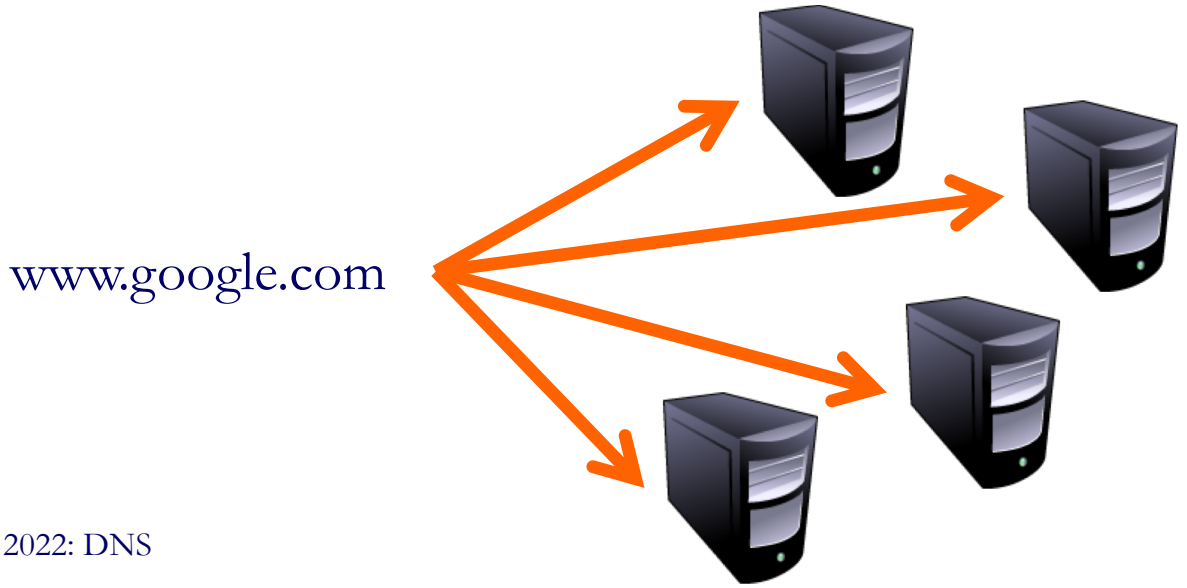
- e.g. you want to move your web server to a new host
- Just change the DNS record!
- What will you have to do if you used hosts.txt?

Aliasing and Load Balancing

- One machine can have many aliases



- One domain can map to multiple machines (basis of anycast)



DNS security

- DNS is the root of trust for the web
  - When a user types www.bankofamerica.com, they expect to be taken to their bank's website
  - What if the DNS record is compromised?
- Hacking your mail server
  - If an attacker hacks the MX record of your mail server, they can read all your mails (Lenova hack)
- DDos attacks.
  - Doesn't always work, especially against the top level domains.
  - A 2002 attack on the TLD records went virtually unnoticed.

DNSSec

- A new protocol that is designed for DNS security
- You will learn all about DNSSec in your first homework.