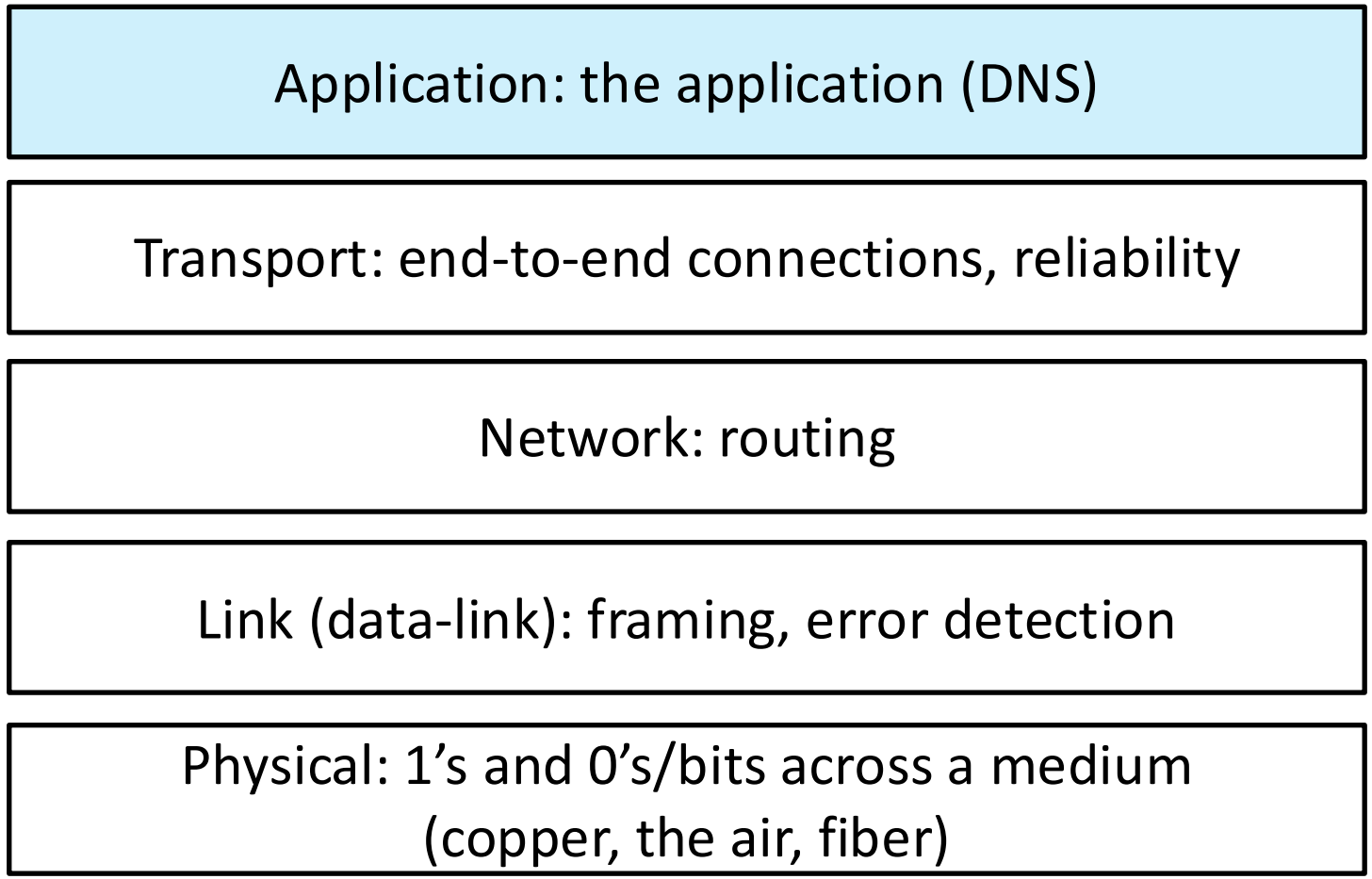
CS 43: Computer Networks

07: Naming and DNS

September 23, 2026

Slides adapted from Kurose & Ross, Vasanta Chaganti, Kevin Webb

Where we are


Application: the application (DNS)
Transport: end-to-end connections, reliability
Network: routing
Link (data-link): framing, error detection
Physical: 1's and 0's/bits across a medium
(copper, the air, fiber)

This week’s question:

How does DNS achieve consistency, availability, and high performance? Which of these does it prioritize?

Uses of DNS

Hostname to IP address translation

- Reverse lookup: IP address to hostname translation

Host name aliasing: other DNS names for a host

- Alias hostnames point to canonical hostname

Email: look up domain's mail server by domain name

What transport protocol does DNS use?

A. UDP
B. TCP
C. I don’t know

What transport protocol does DNS use?

A. **UDP**
B. TCP
C. I don’t know

DNS: a distributed, hierarchical database

Root DNS Servers

Over 1700 replicated root servers with 13 IP address ran by 12 organizations coordinated by ICANN/IANA

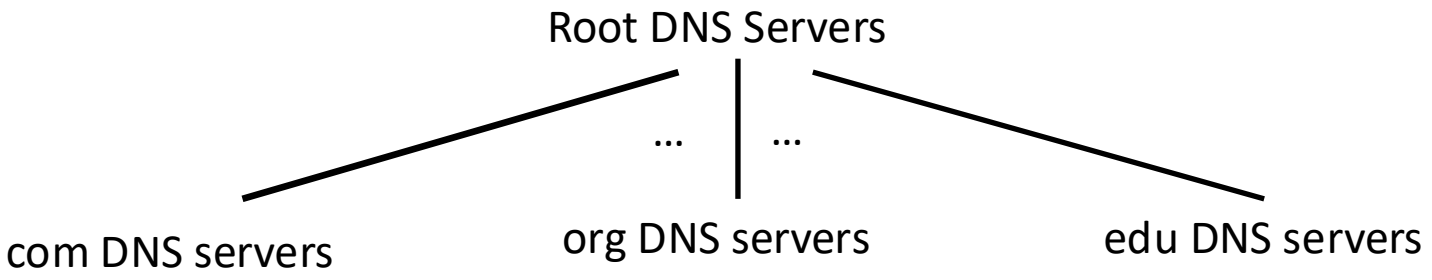
DNS: a distributed, hierarchical database

Root DNS Servers

Over 1700 replicated root servers with 13 IP address ran by 12 organizations coordinated by ICANN/IANA

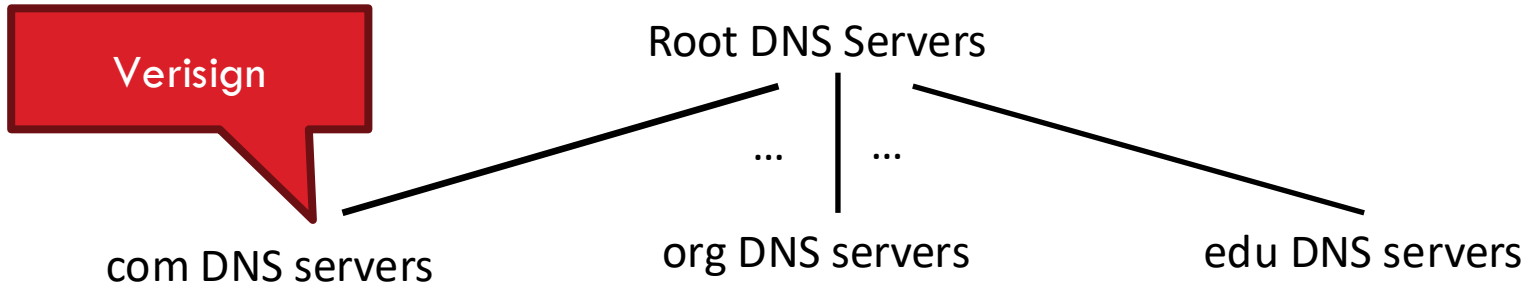
Try: You can see all the root servers with dig query: dig . ns

DNS: a distributed, hierarchical database



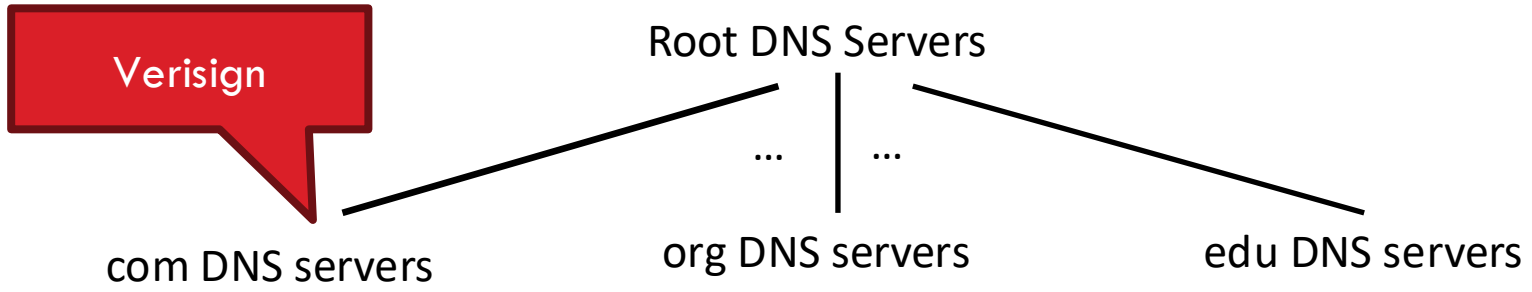
Top-level domain servers: Also assigned by ICANN/IANA and run by companies

DNS: a distributed, hierarchical database



Top-level domain servers: Also assigned by ICANN/IANA and run by companies

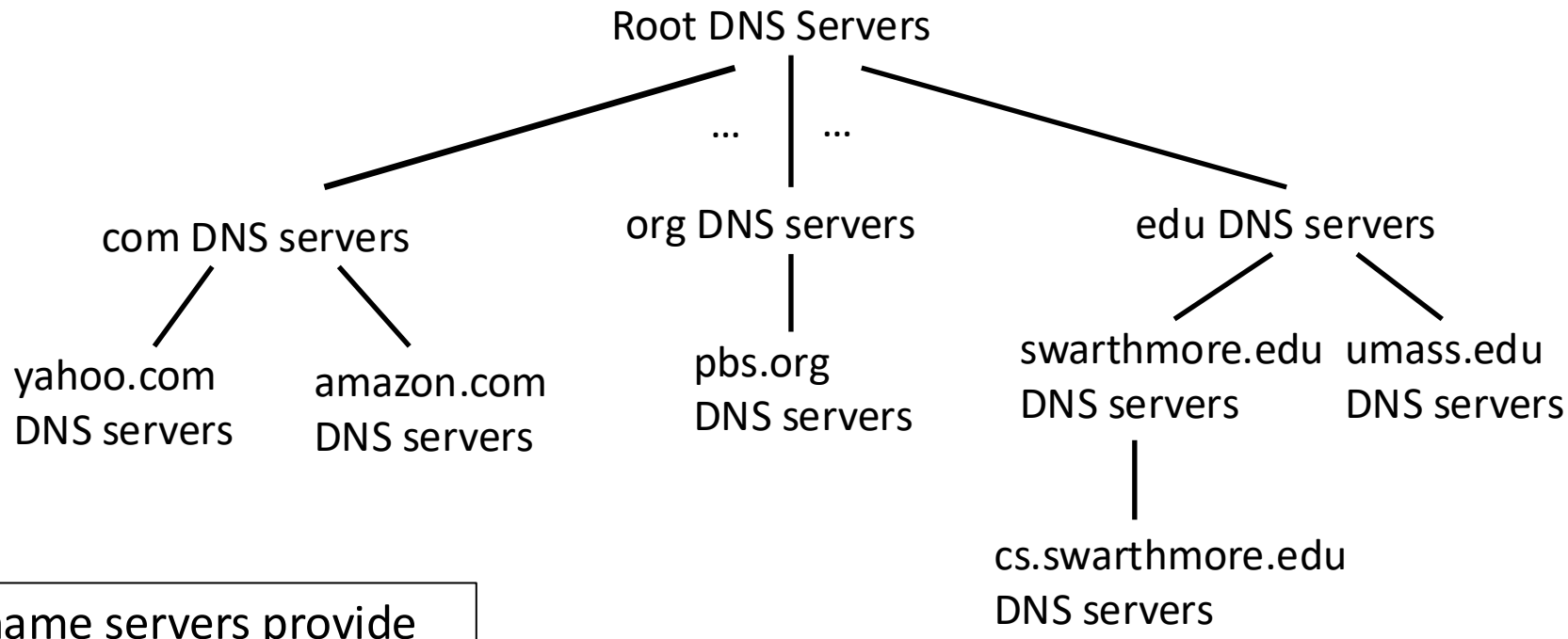
DNS: a distributed, hierarchical database



Root servers have IP address mappings for all the TLD servers in zone files

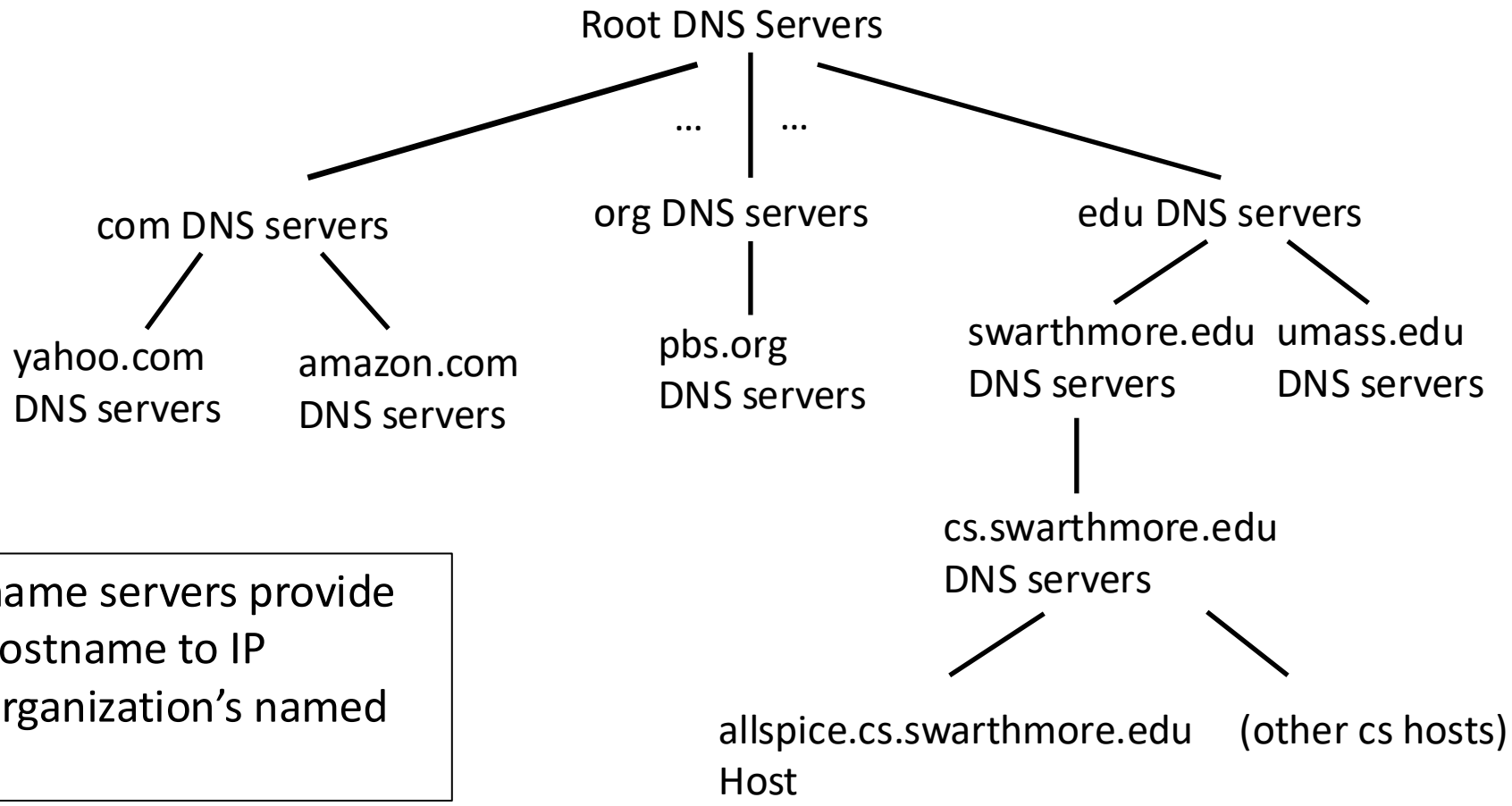
Top-level domain servers: Also assigned by ICANN/IANA and run by companies

DNS: a distributed, hierarchical database



Authoritative name servers provide authoritative hostname to IP mappings for organization's named hosts

DNS: a distributed, hierarchical database



Authoritative name servers provide authoritative hostname to IP mappings for organization's named hosts

- allspice.cs.swarthmore.edu.

Nameless root,
Usually implied.

Why do we structure DNS like this? Which of these helps the most? Drawbacks?

A. It divides up responsibility among parties.

B. It improves performance of the system.

C. It reduces the size of the state that a server needs to store.

D. Some other reason.

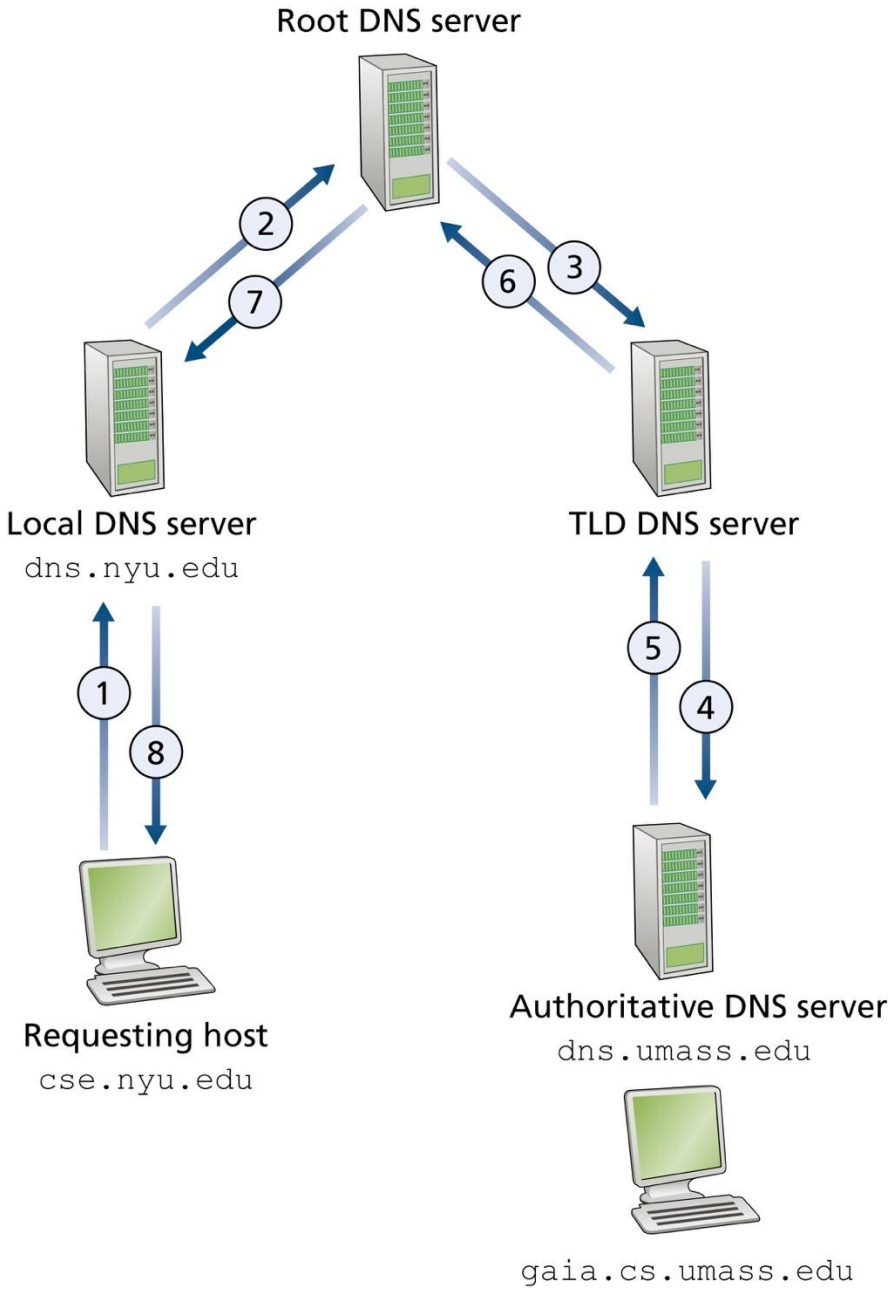
Why do we structure DNS like this? Which of these helps the most? Drawbacks?

A. It divides up responsibility among parties.

B. It improves performance of the system overall but individual end hosts (assuming no caching) have a look-up overhead of traversing the hierarchy .

C. It reduces the size of the state that a server needs to store.

D. Some other reason.



Are these DNS queries recursive or iterative?

A. Iterative
B. Recursive

Example: iterative query using dig()

dig . ns

dig +norec demo.cs.swarthmore.edu @a.root-servers.net

dig +norec demo.cs.swarthmore.edu @a.edu-servers.net

dig +norec demo.cs.swarthmore.edu @ibext.its.swarthmore.edu

demo.cs.swarthmore.edu. 259200 IN A 130.58.68.26

Example: iterative query using dig()

```
dig . ns

dig +norec demo.cs.swarthmore.edu @a.root-servers.net

dig +norec demo.cs.swarthmore.edu @a.edu-servers.net
```

This doesn't work anymore…It seems ITS is blocking DNS queries sent directly to root or TLD servers (outbound to port 53). This will be changed for Lab 3 when you'll have to write a program that does it!

How many answers
Time to live in seconds
How many additional records?

```
$ dig @a.root-servers.net www.freebsd.org +norecurse
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 57494
;; QUERY: 1, ANSWER: 0, AUTHORITY: 2, ADDITIONAL: 2

;; QUESTION SECTION:
;www.freebsd.org.       IN  A

;; AUTHORITY SECTION:
org.            172800  IN  NS  b0.org.afilias-nst.org.
org.            172800  IN  NS  d0.org.afilias-nst.org.

;; ADDITIONAL SECTION:
b0.org.afilias-nst.org.    172800  IN  A   199.19.54.1
d0.org.afilias-nst.org.    172800  IN  A   199.19.57.1
```

How many answers
Time to live in seconds
How many additional records?

```
$ dig @a.root-servers.net www.freebsd.org +norecurse
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 57494
;; QUERY: 1, ANSWER: 0, AUTHORITY: 2, ADDITIONAL: 2

;; QUESTION SECTION:
;www.freebsd.org.        IN  A

;; AUTHORITY SECTION:
org.            172800  IN  NS  b0.org.afilias-nst.org.
org.            172800  IN  NS  d0.org.afilias-nst.org.

;; ADDITIONAL SECTION:
b0.org.afilias-nst.org.    172800  IN  A   199.19.54.1
d0.org.afilias-nst.org.    172800  IN  A   199.19.57.1
```

Glue records

How many answers?
How many additional records?

(authoritative for org.)

```
$ dig @199.19.54.1 www.freebsd.org +norecurse
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 39912
;; QUERY: 1, ANSWER: 0, AUTHORITY: 3, ADDITIONAL: 0

;; QUESTION SECTION:
;www.freebsd.org.      IN  A

;; AUTHORITY SECTION:
freebsd.org.     86400  IN  NS  ns1.isc-sns.net.
freebsd.org.     86400  IN  NS  ns2.isc-sns.com.
freebsd.org.     86400  IN  NS  ns3.isc-sns.info.
```

How many answers?
How many additional records?

(authoritative for org.)

```
$ dig @199.19.54.1 www.freebsd.org +norecurse
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 39912
;; QUERY: 1, ANSWER: 0, AUTHORITY: 3, ADDITIONAL: 0

;; QUESTION SECTION:
;www.freebsd.org.       IN  A

;; AUTHORITY SECTION:
freebsd.org.    86400   IN  NS  ns1.isc-sns.net.
freebsd.org.    86400   IN  NS  ns2.isc-sns.com.
freebsd.org.    86400   IN  NS  ns3.isc-sns.info.
```

(authoritative for freebsd.org.)

```
$ dig @ns1.isc-sns.net www.freebsd.org +norecurse
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 17037
;; QUERY: 1, ANSWER: 1, AUTHORITY: 3, ADDITIONAL: 3

;; QUESTION SECTION:
;www.freebsd.org.        IN  A

;; ANSWER SECTION:
www.freebsd.org. 3600   IN  A   69.147.83.33

;; AUTHORITY SECTION:
freebsd.org.     3600   IN  NS  ns2.isc-sns.com.
freebsd.org.     3600   IN  NS  ns1.isc-sns.net.
freebsd.org.     3600   IN  NS  ns3.isc-sns.info.

;; ADDITIONAL SECTION:
ns1.isc-sns.net. 3600   IN  A   72.52.71.1
ns2.isc-sns.com. 3600   IN  A   38.103.2.1
ns3.isc-sns.info.   3600   IN  A   63.243.194.1
```

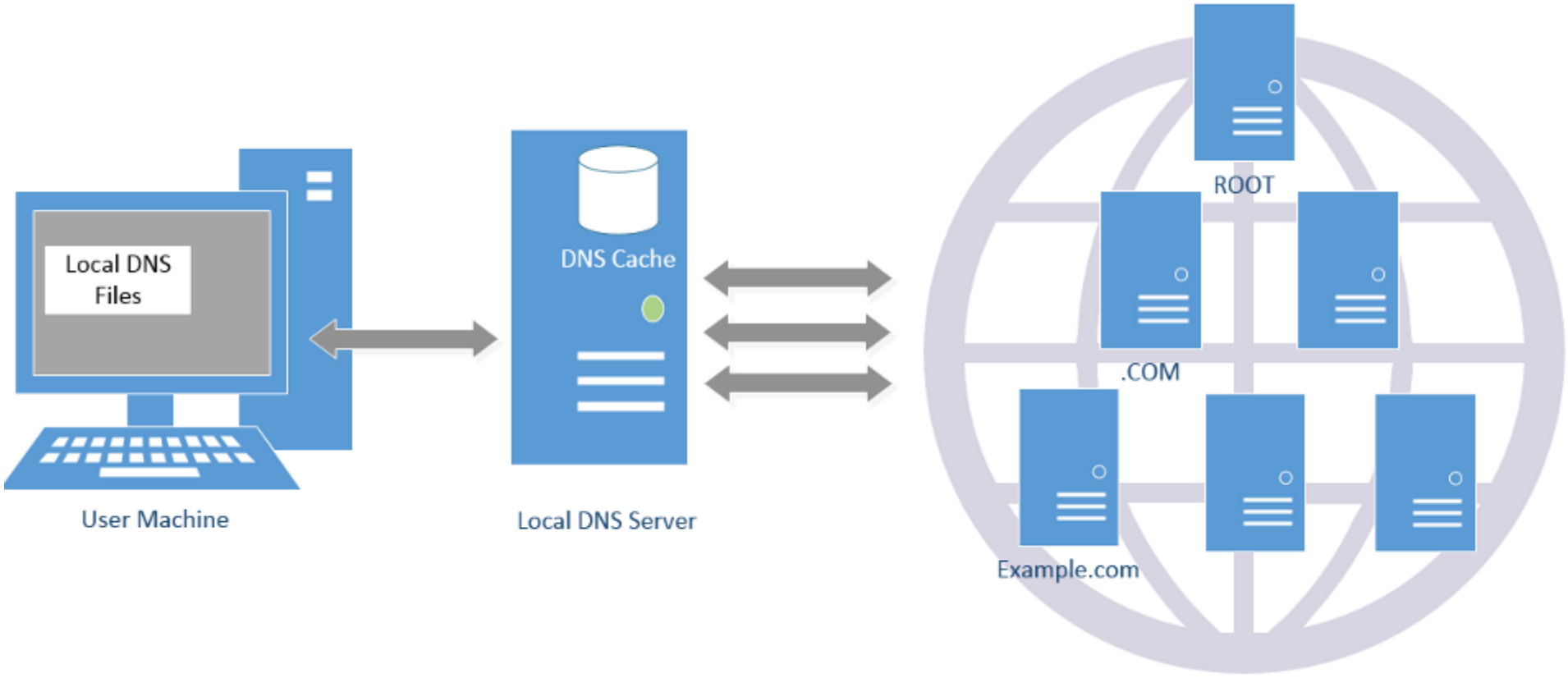
How many answers?
How many authoritative records?
How many additional records?

(authoritative for freebsd.org.)

```
$ dig @ns1.isc-sns.net www.freebsd.org +norecurse
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 17037
;; QUERY: 1, ANSWER: 1, AUTHORITY: 3, ADDITIONAL: 3

;; QUESTION SECTION:
;www.freebsd.org.        IN  A

;; ANSWER SECTION:
www.freebsd.org. 3600   IN  A   69.147.83.33

;; AUTHORITY SECTION:
freebsd.org.     3600   IN  NS  ns2.isc-sns.com.
freebsd.org.     3600   IN  NS  ns1.isc-sns.net.
freebsd.org.     3600   IN  NS  ns3.isc-sns.info.

;; ADDITIONAL SECTION:
ns1.isc-sns.net. 3600   IN  A   72.52.71.1
ns2.isc-sns.com. 3600   IN  A   38.103.2.1
ns3.isc-sns.info.   3600   IN  A   63.243.194.1
```

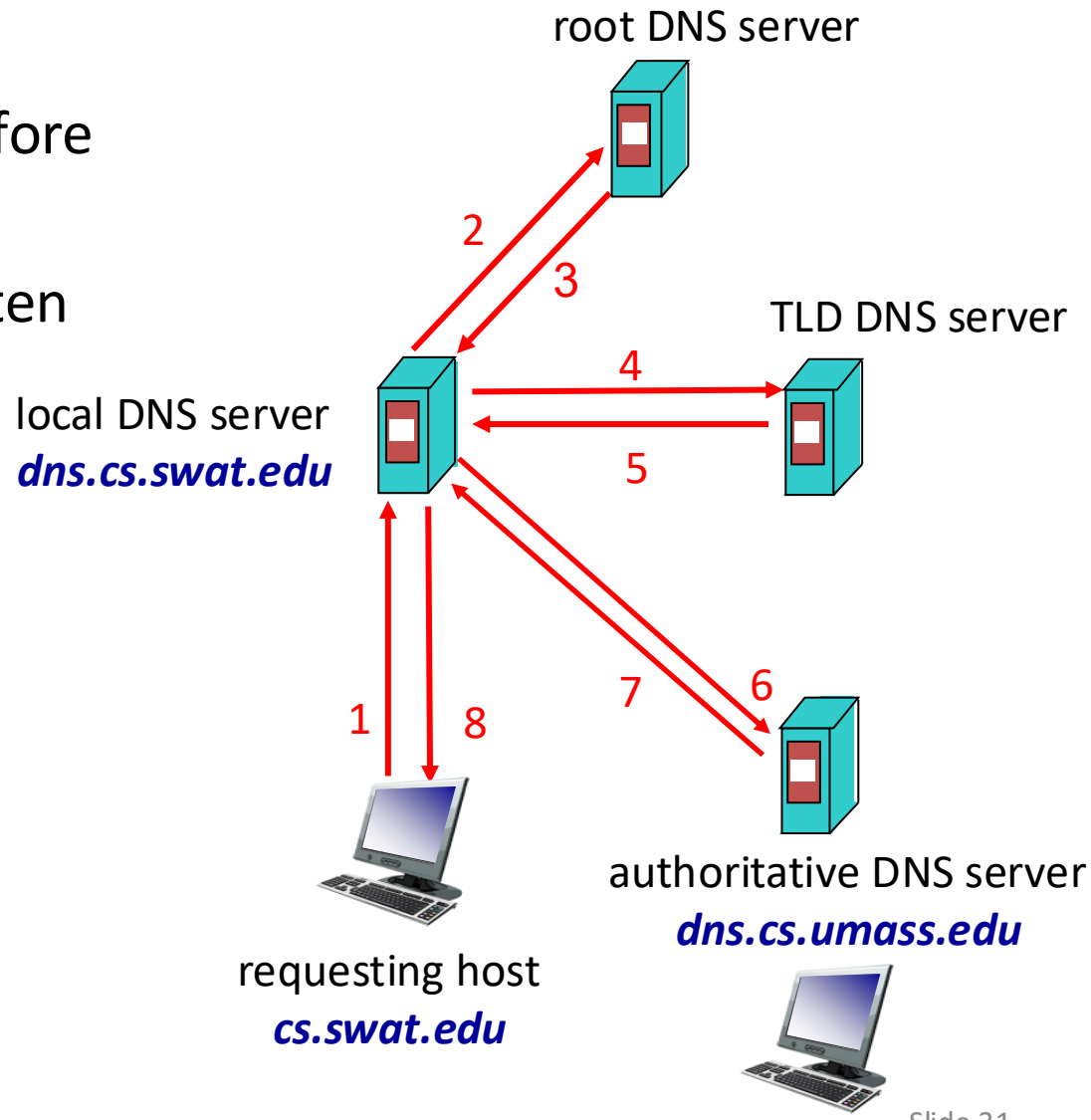
How many answers?
How many authoritative records?
How many additional records?

DNS Query Process and Cache



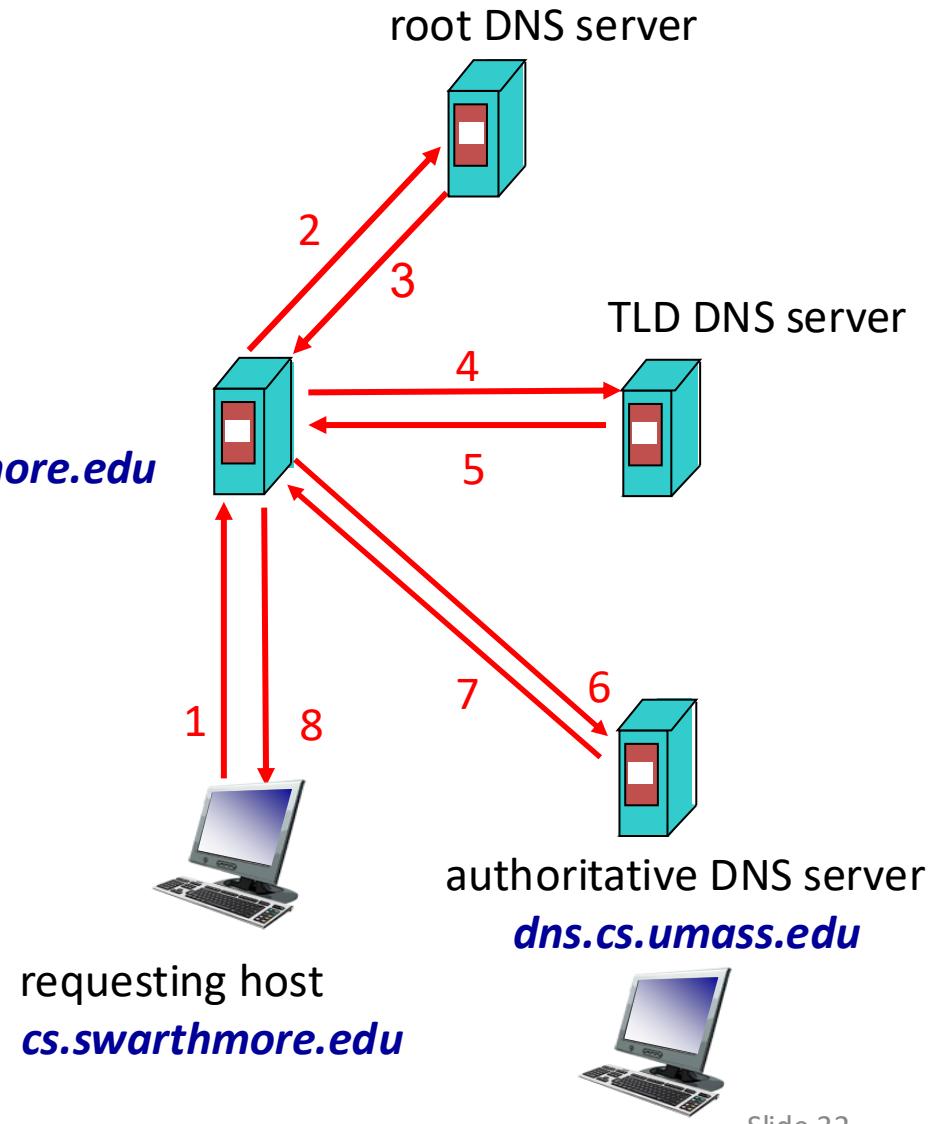
DNS Caching

- Why cache?
  - apprx. 1 sec latency before starting a download
  - Popular sites visited often
- Where to cache?
  - Local DNS server
  - Browser



DNS Caching

- When to cache?
  - learn a mapping? cache!
  - any name server can cache
- For how long?
  - until Time To Live (expires)
- What to cache?
  - TLD servers cached – almost never change
  - Root name servers usually, not visited legitimately



Caching

- Once (any) name server learns a mapping, it caches mapping
  - cache entries timeout (disappear) after some time (TTL: time to live)
  - TLD servers typically cached in local name servers.
  - Root name servers not often (legitimately) visited
- (+) Subsequent requests need not burden DNS
- (-) Cached entries may be out-of-date (best effort!)
  - If host's name or IP address changes, it may not be known Internet-wide until all TTLs expire

The TTL value should be…

A. Short, to make sure that changes are accurately reflected

B. Long, to avoid re-queries of higher-level DNS servers

C. Something else

The TTL value should be…

A. Short, to make sure that changes are accurately reflected

B. Long, to avoid re-queries of higher-level DNS servers

C. Something else

Can connect how caching works to the tradeoffs in the CAP theorem

DNS as Indirection Service

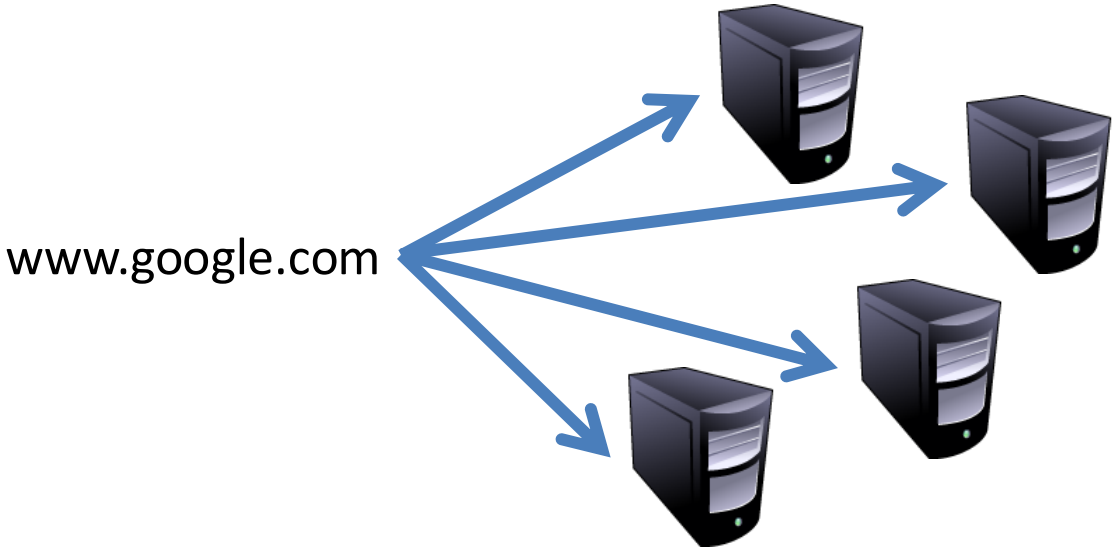
- DNS gives us very powerful capabilities
  - Not only easier for humans to reference machines!

- 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!

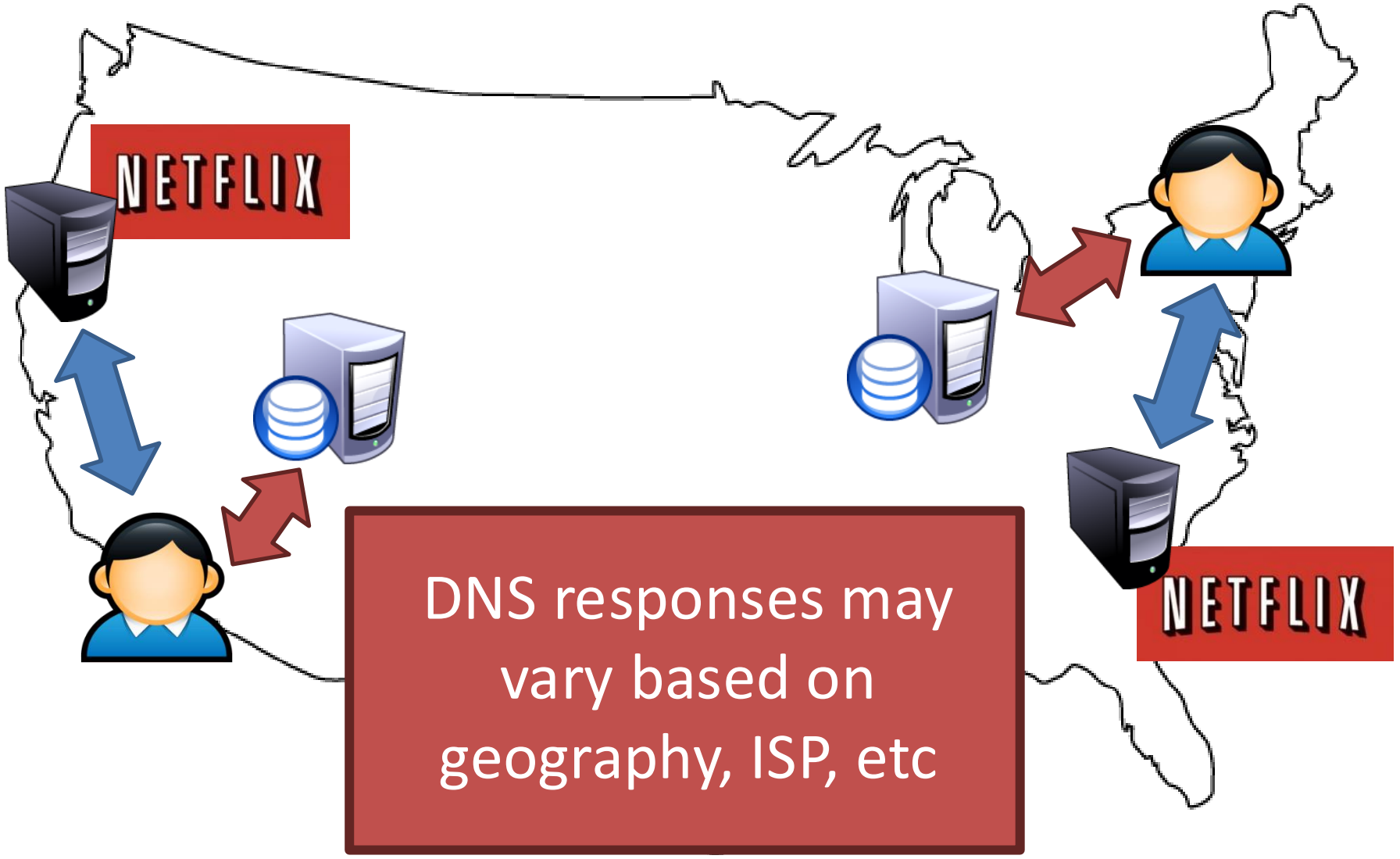
Aliasing and Load Balancing

- One machine can have many aliases

  - One domain can map to multiple machines



Content Delivery Networks



Inserting (or changing) records

Example: new startup "Network Utopia"

- Step 1: Register networkuptopia.com at DNS registrar
  - provide names, IP addresses of authoritative name server (primary and secondary)

Inserting (or changing) records

Example: new startup “Network Utopia”

- Step 2: Registrar inserts two RRs into .com TLD server
  - (networkutopia.com, dns1.networkutopia.com, NS)
  - (dns1.networkutopia.com, 212.212.212.1, A)

Inserting (or changing) records

Example: new startup "Network Utopia"

- Step 3: Set up authoritative server at that name/address
  - Create records for the services:

Inserting (or changing) records

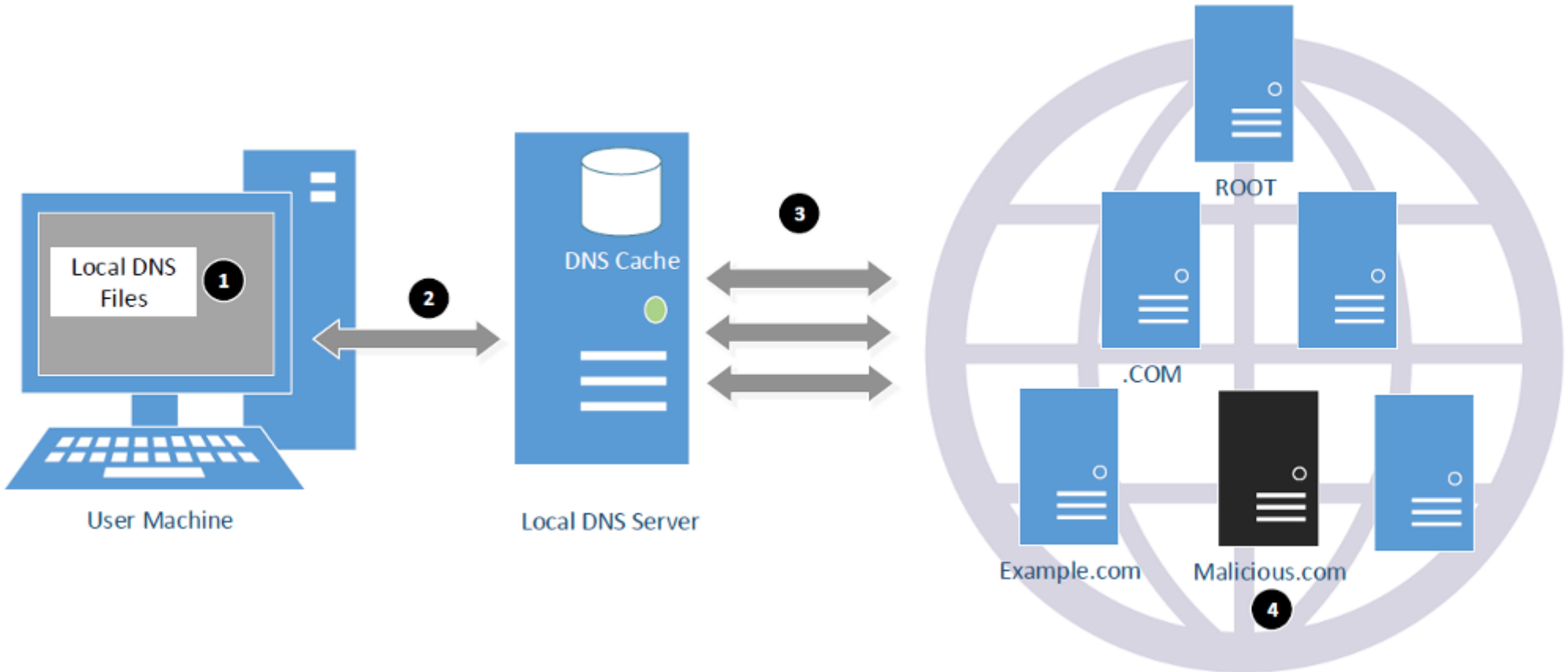
Example: new startup "Network Utopia"

- Step 3: Set up authoritative server at that name/address
  - Create records for the services:
    - type A record for www.networkuptopia.com
    - type MX record for @networkutopia.com email

Inserting (or changing) records

- Example: new startup "Network Utopia"
- Register networkuptopia.com at DNS registrar
  - provide names, IP addresses of authoritative name server (primary and secondary)
  - registrar inserts two RRs into .com TLD server
    - (networkutopia.com, dns1.networkutopia.com, NS)
    - (dns1.networkutopia.com, 212.212.212.1, A)
- Set up authoritative server at that name/address
  - Create records for the services:
    - type A record for www.networkuptopia.com
    - type MX record for @networkutopia.com email

Attack Surface Overview



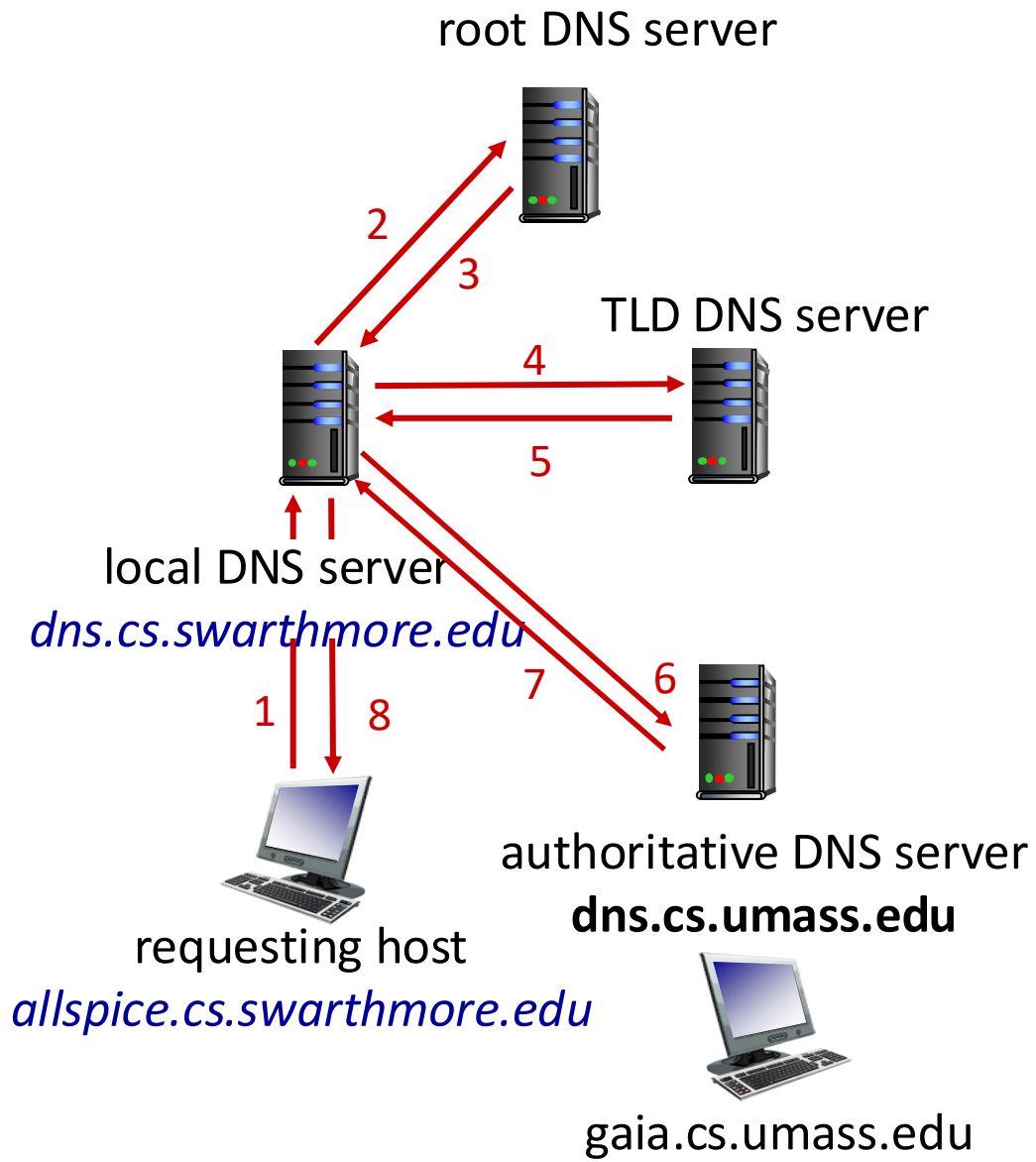
DNS security

DNS Vulnerabilities:

- No authentication
- Connectionless transport layer protocol (UDP)

DNS Attacks:

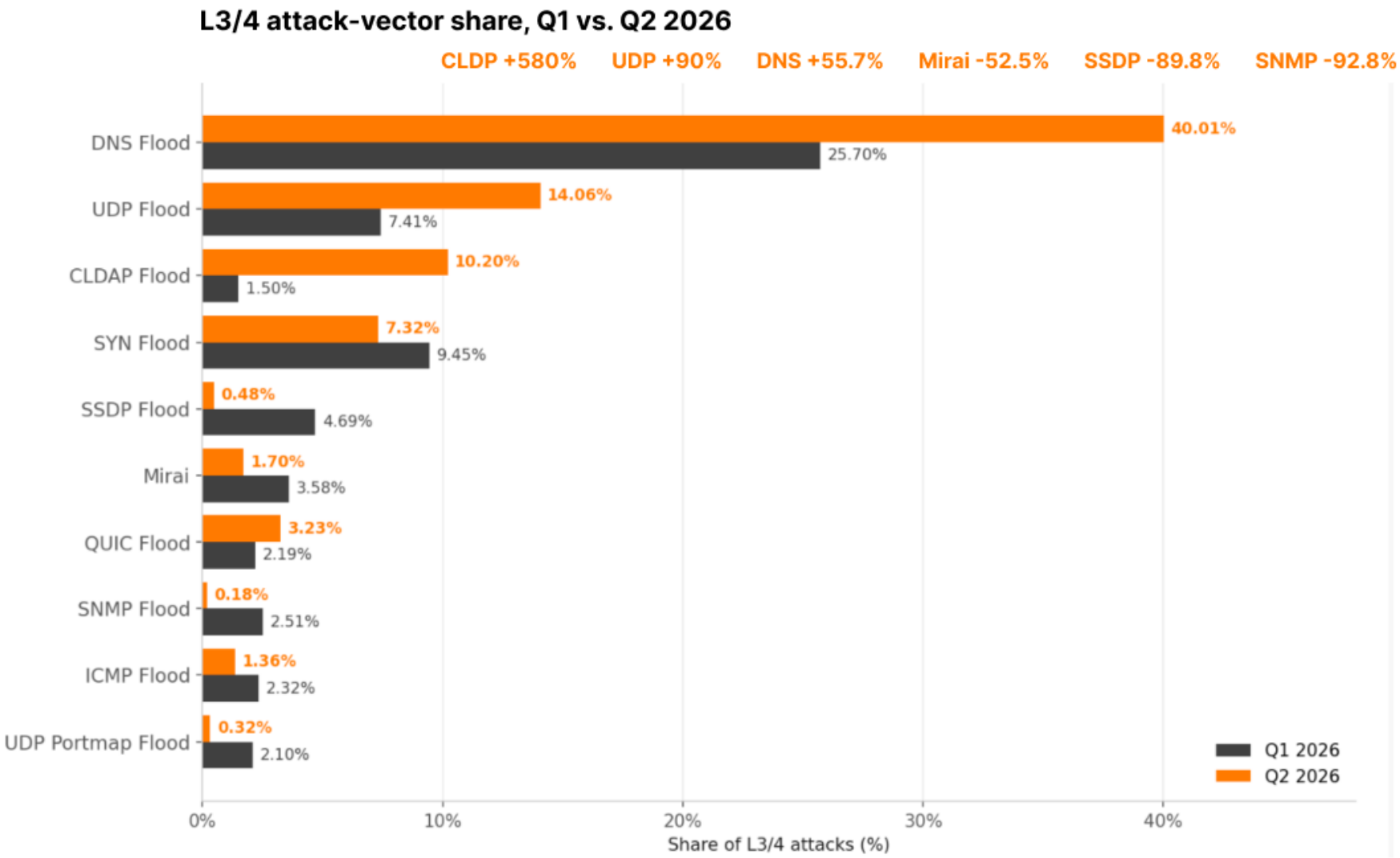
- Amplification Attack
- Cache Poisoning
- Man-in-the-middle
- DNS Redirection
- DDoS
- DNS Injection



Denial Of Service

- Flood DNS servers with requests until they fail
- October 2002: massive DDoS against the root name servers
  - What was the effect?
  - … users didn't even notice
  - Root zone file is cached almost everywhere
- More targeted attacks can be effective
  - Local DNS server → cannot access DNS
  - Authoritative server → cannot access domain

In 2026, Cloudflare noted most DDoS attacks used DNS.

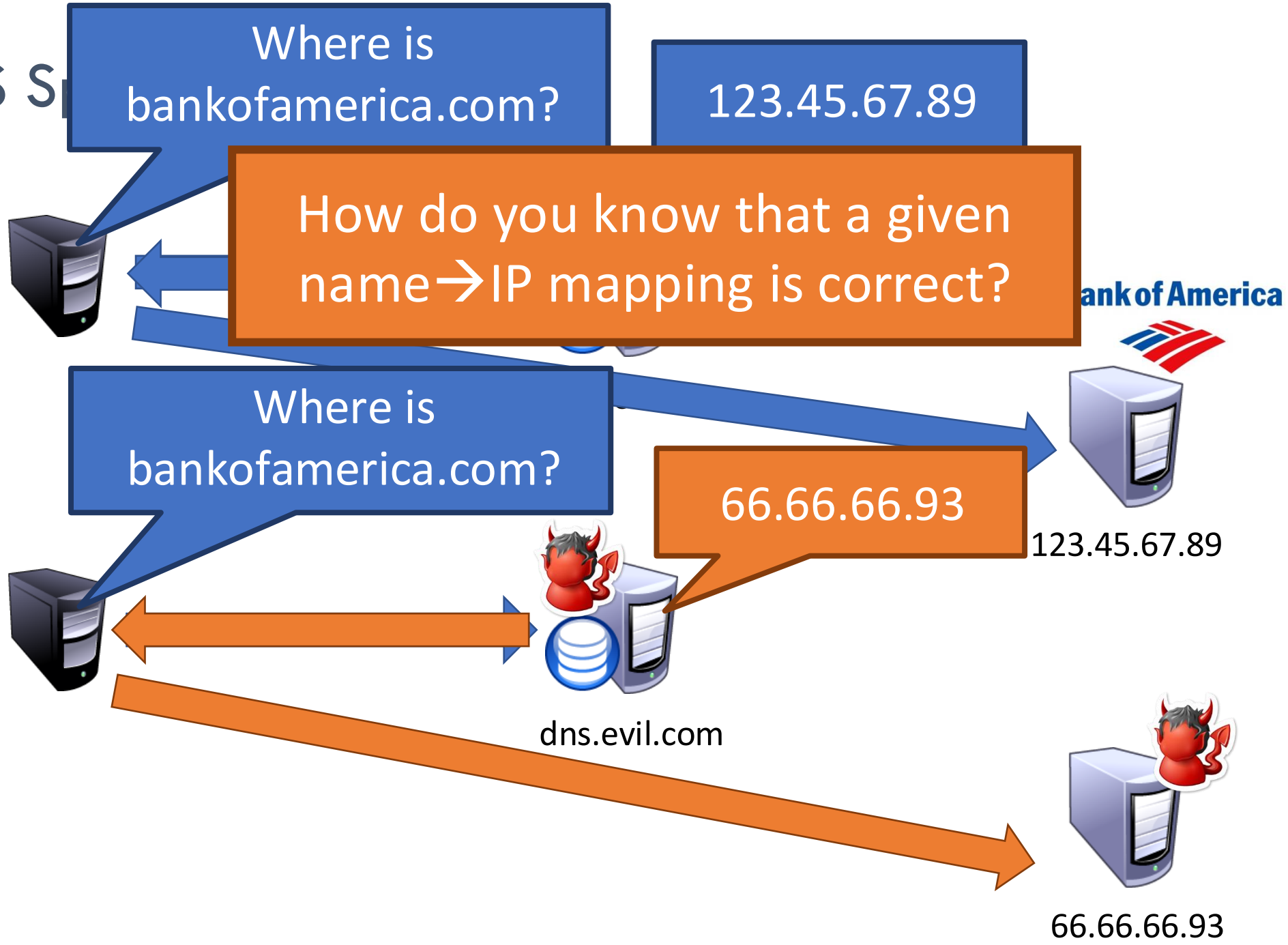


DNS Hijacking

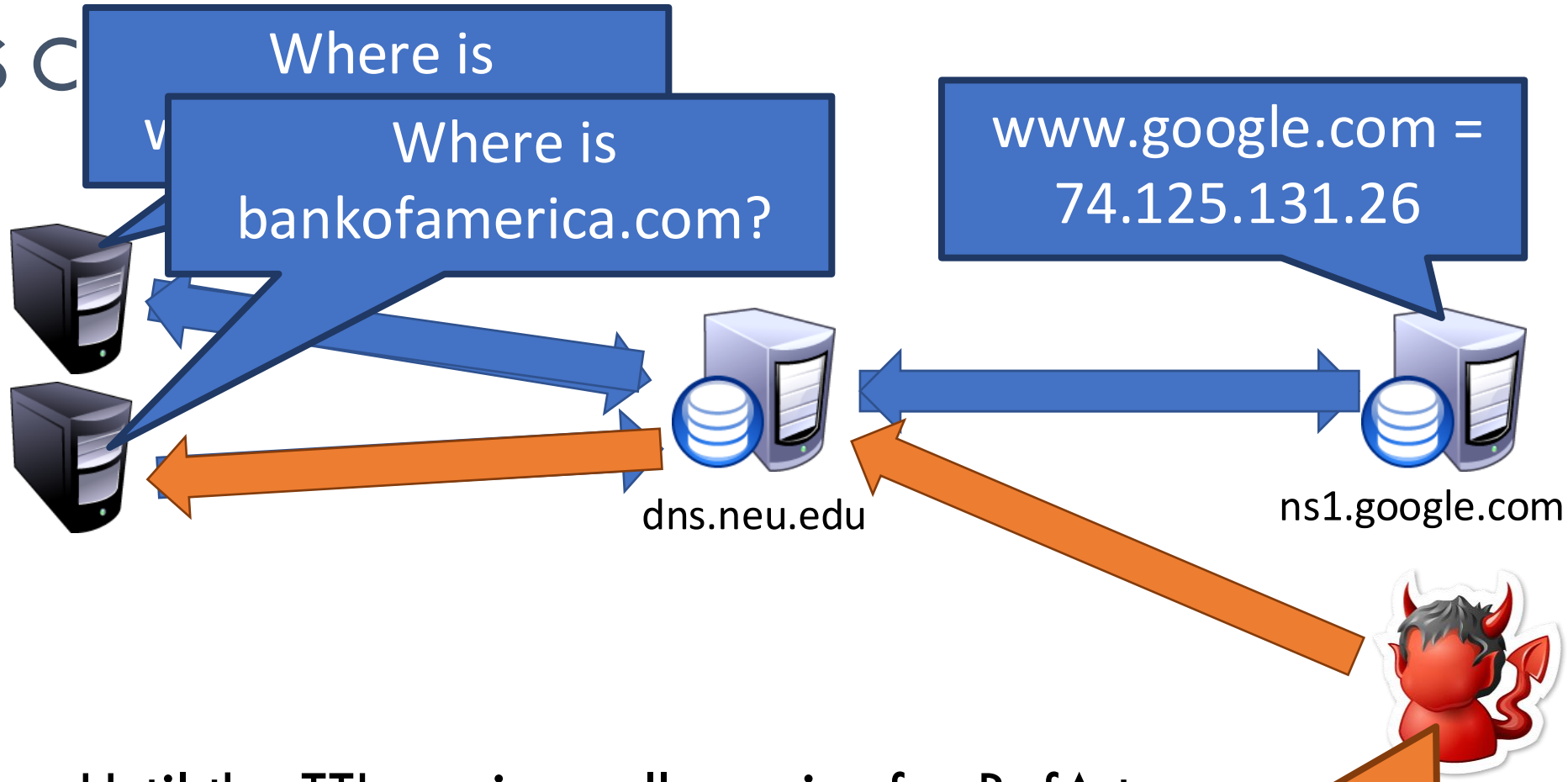
- Infect their OS or browser with a virus/trojan
  - e.g. Many trojans change entries in /etc/hosts
  - *.bankofamerica.com → evilbank.com
- Man-in-the-middle

  - Response Spoofing
    - Eavesdrop on requests
    - Outrace the servers response

DNS S

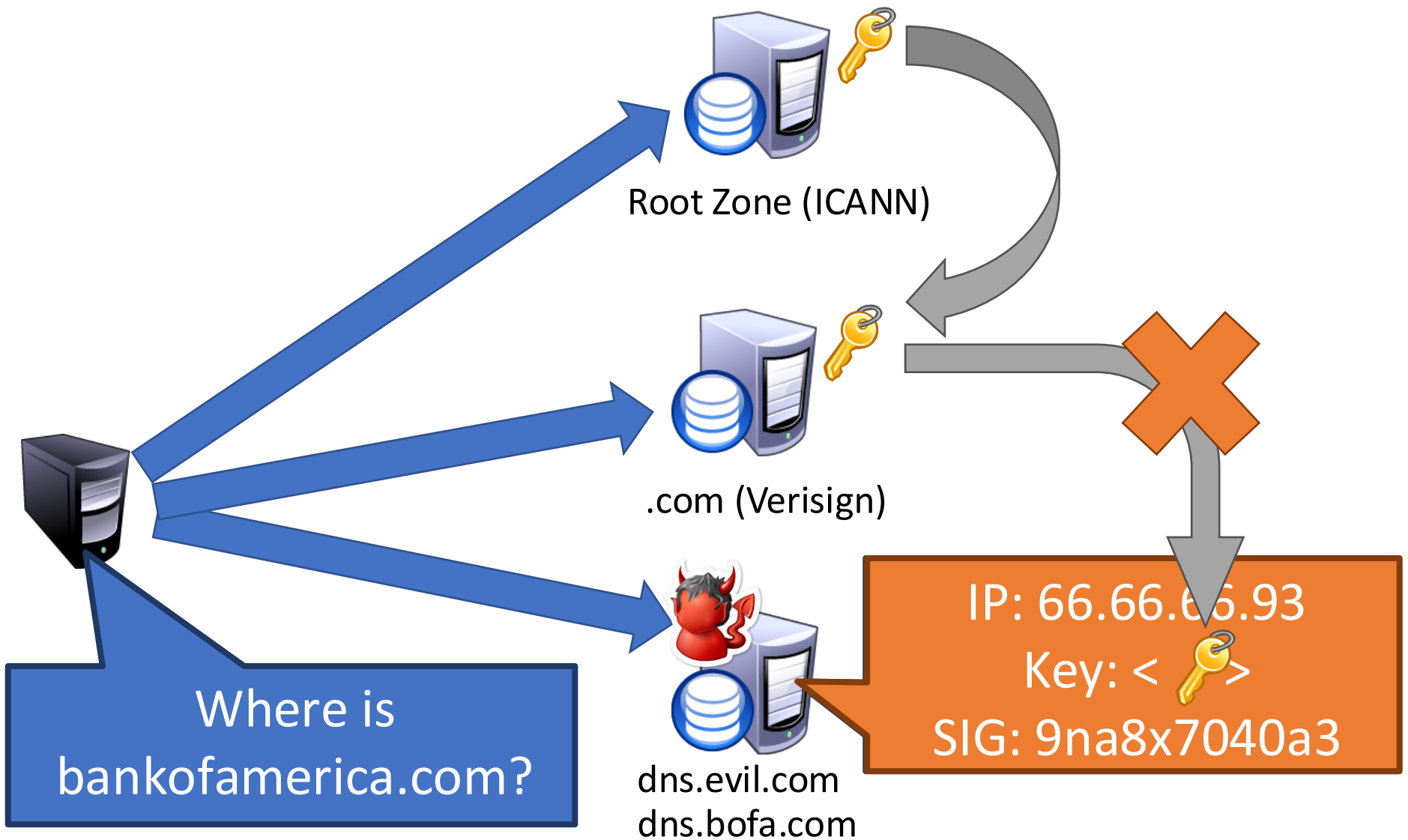

Where is bankofamerica.com?
123.45.67.89
How do you know that a given name→IP mapping is correct?
ank of America
Where is bankofamerica.com?
66.66.66.93
123.45.67.89
dns.evil.com
66.66.66.93

DNS C



- Until the TTL expires, all queries for BofA to dns.neu.edu will return poisoned re
- Much worse than spoofing/man-in-
  - Whole ISPs can be impacted!

DNSSEC Hierarchy of Trust



Solution: DNSSEC

- Cryptographically sign critical resource records
  - Resolver can verify the cryptographic signature
- Two new resource types
  - Type = DNSKEY
    - Name = Zone domain name
    - Value = Public key for the zone
  - Type = RRSIG
    - Name = (type, name) tuple, i.e. the query itself
    - Value = Cryptographic signature of the query results

Creates a hierarchy of trust between zones

Prevents hijacking and spoofing

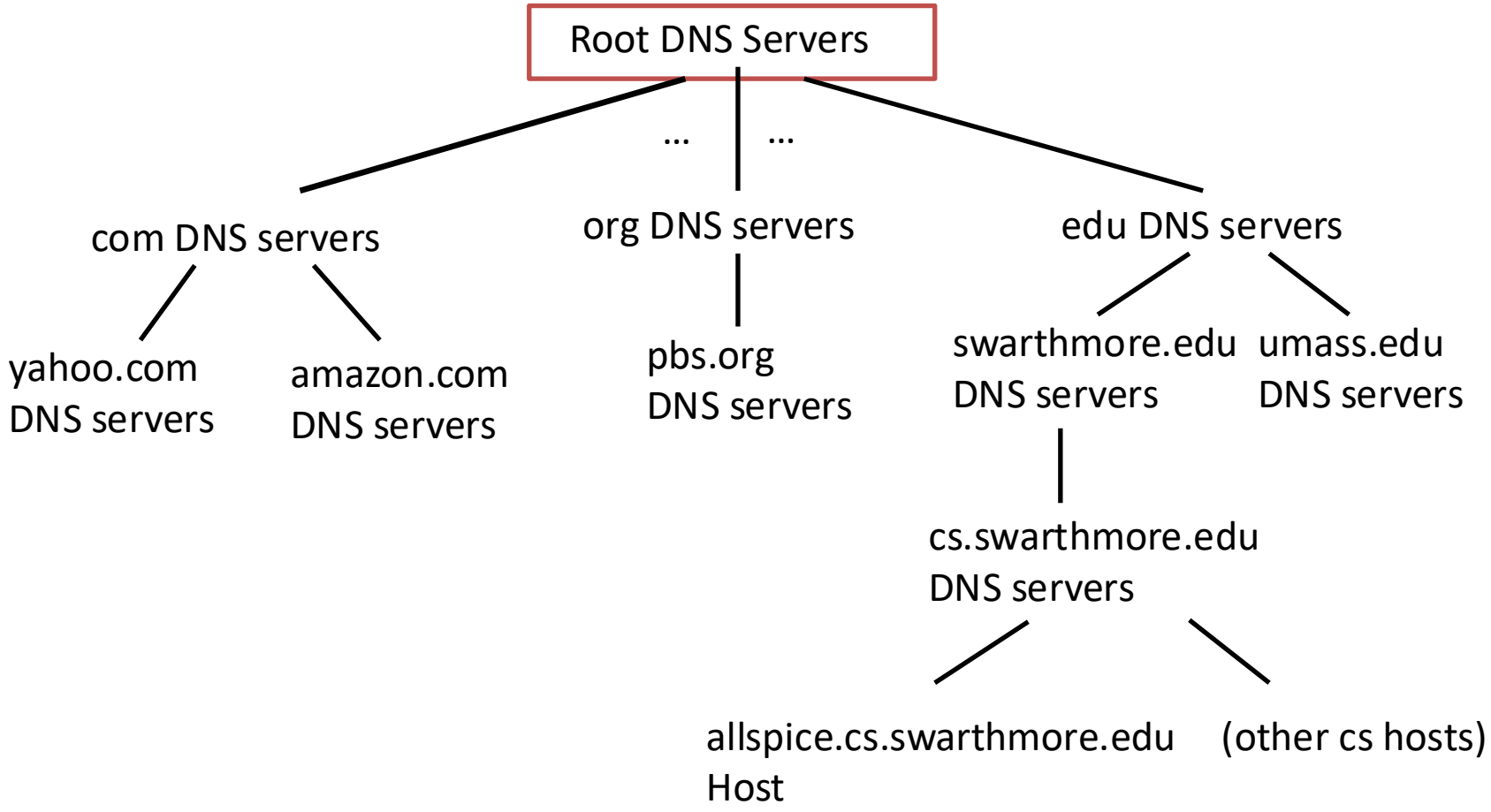
Summary

- DNS maps human readable names to IP addresses
- DNS arranged into a hierarchy
  - Scalability / distributed responsibility
  - Autonomous control of local name servers
- Caching crucial for performance
- DNS vulnerable to many kinds of attacks

Reference Material (useful slides but not shown in class)

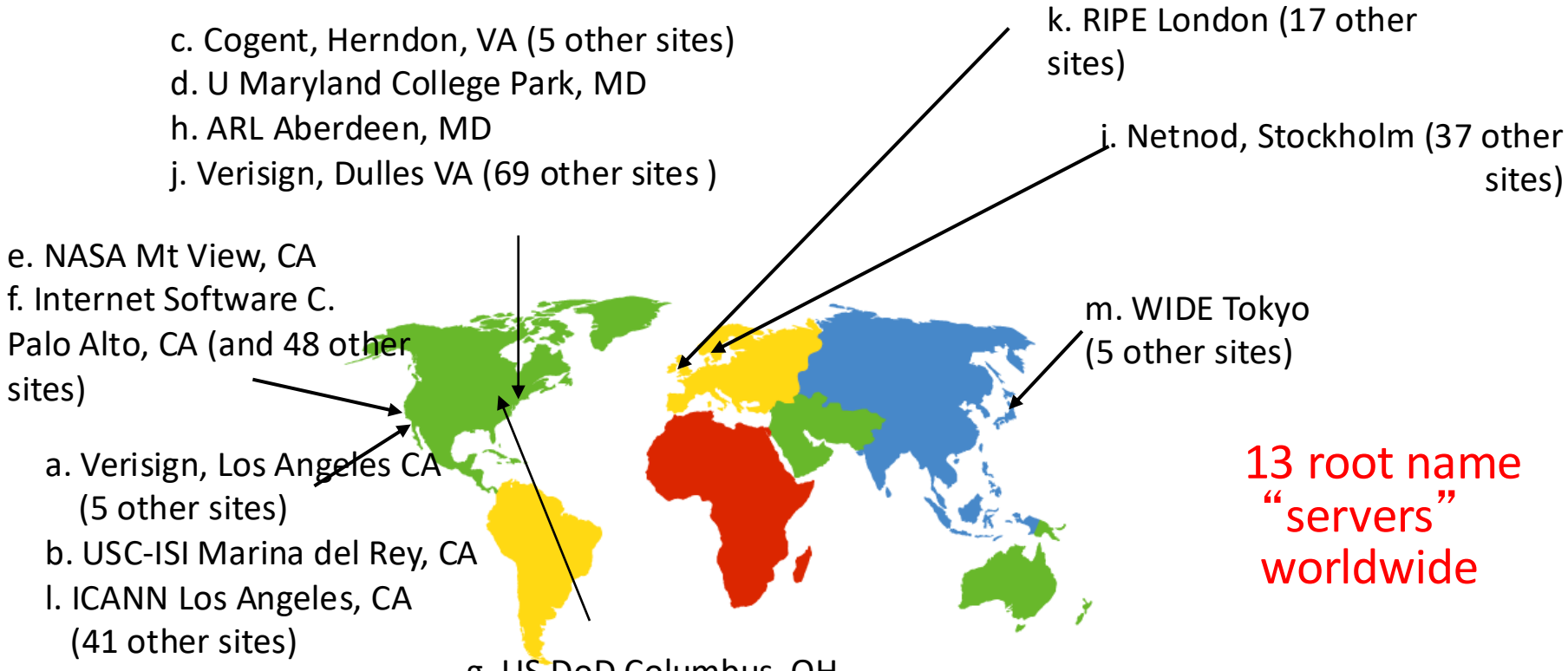
Reference Material (Not reviewed in class)

DNS: a distributed, hierarchical database



DNS: Root Name Servers

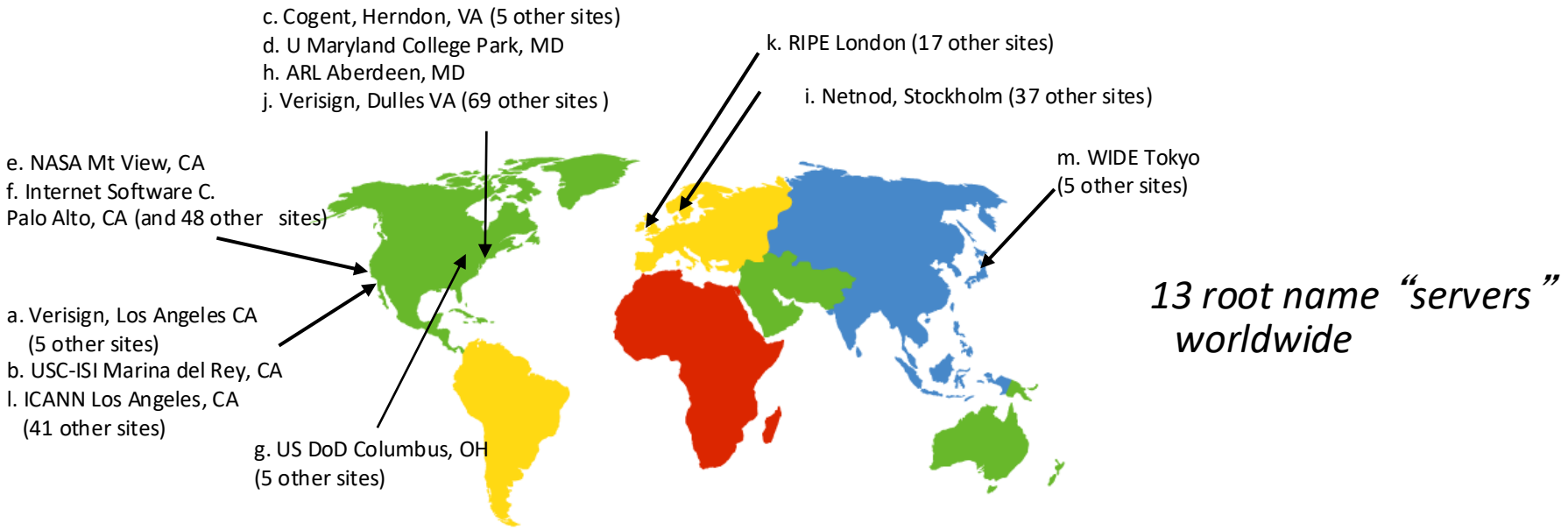
- Root name server:
  - Knows how to find top-level domains (.com, .edu, .gov, etc.)
  - How often does the location of a TLD change?



13 root name "servers" worldwide

DNS: Root Name Servers

- Root name server:
  - Knows how to find top-level domains (.com, .edu, .gov, etc.)
  - How often does the location of a TLD change?
  - approx. 1700 total root servers
  - Significant amount of traffic is not legitimate



13 root name "servers" worldwide

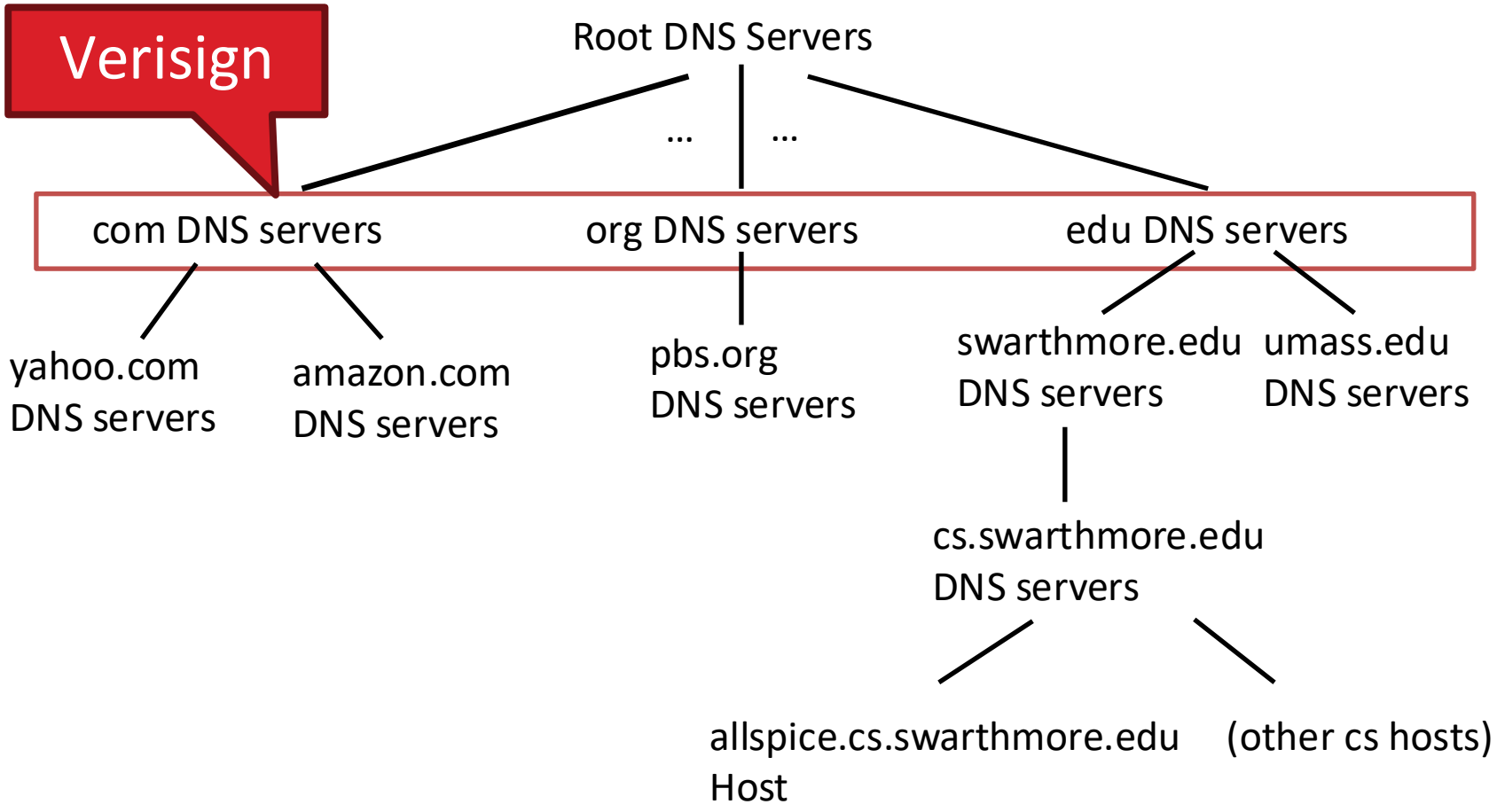
Root Name Servers

- Responsible for the Root Zone File

```
com.        172800 IN   NS  a.gtld-servers.net.
com.        172800 IN   NS  b.gtld-servers.net.
com.        172800 IN   NS  c.gtld-servers.net.
```

  - In practice, most systems cache this information
  - Lists the TLDs and who controls them
  - ~272KB in size

Top Level Domain (TLD) servers

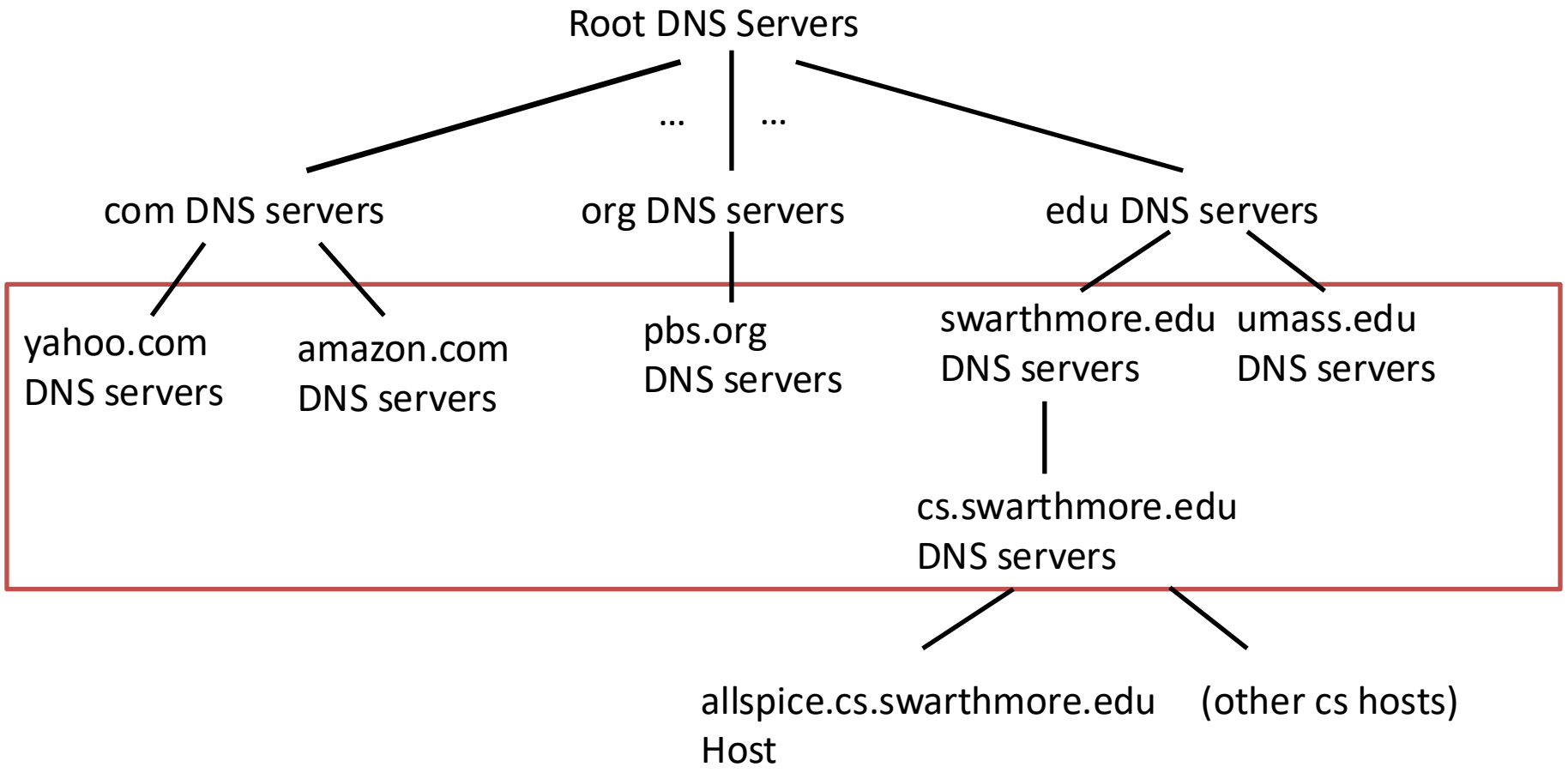


Top Level Domain (TLD) servers

- who maintains the servers?:
  - Verisign: .com, .net
  - Educause: .edu (Verisign backend)
  - local governments or companies

- Responsible for:
  - com, org, net, edu, gov, aero, jobs, museums,
  - all top-level country domains, e.g.: uk, fr, de, ca, jp, etc

Authoritative Servers



Authoritative Servers

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, easily changing entries
- Often, but not always, acts as organization's local name server (for responding to look-ups)

Resolution Process

- End host wants to look up a name, who should it contact?
  - It could traverse the hierarchy, starting at a root
  - More efficient for ISP to provide a local server

- ISP's local server for handling queries not necessarily a part of the pictured hierarchy

Local DNS Name Server

- Each ISP (residential ISP, company, university) has (at least) one
  - also called "default name server"

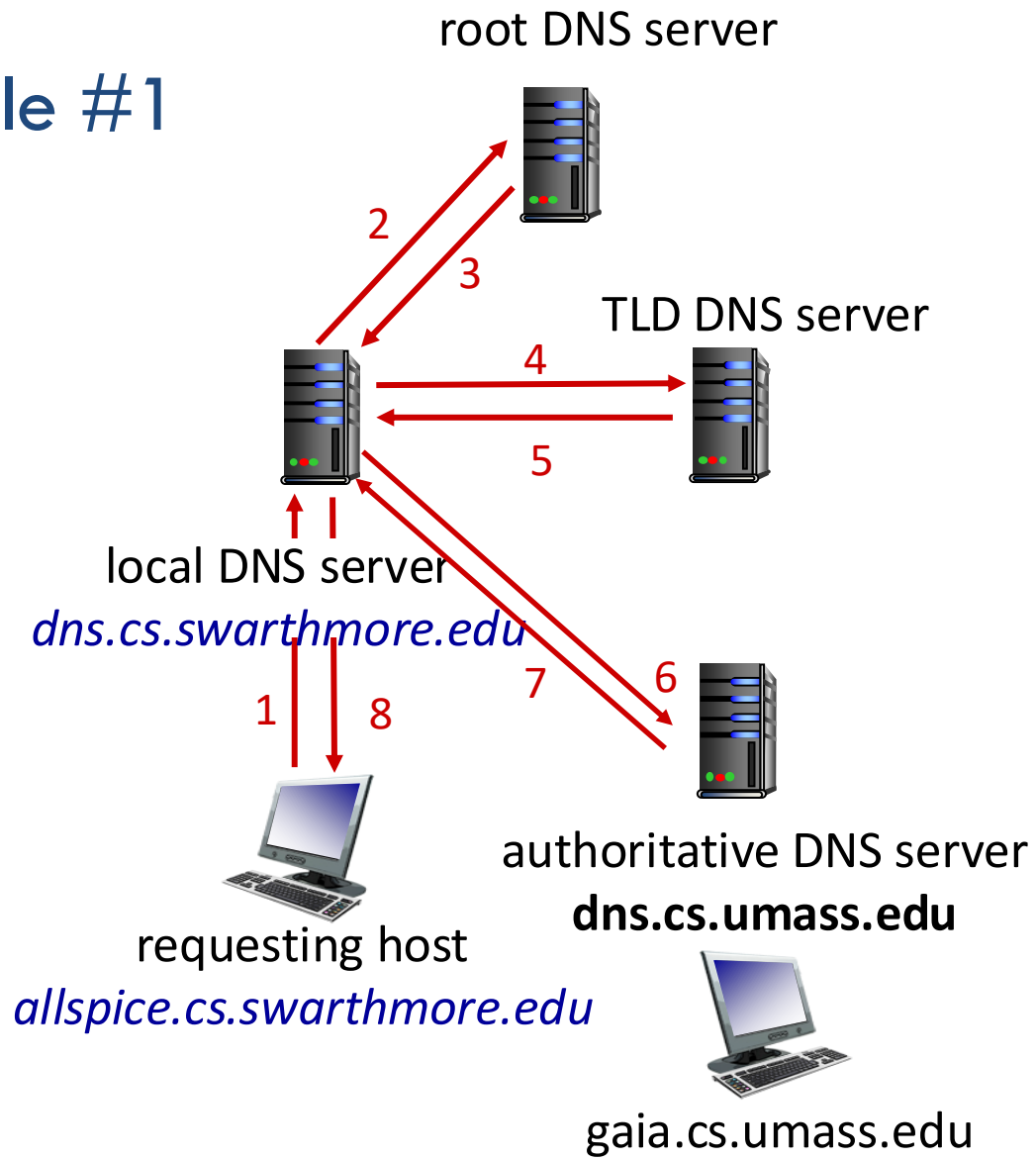
- When host makes DNS query, query is sent to its local DNS server
  - has local cache of recent name-to-address translation pairs (but may be out of date!)
  - acts as proxy, forwards query into hierarchy

DNS name resolution example #1

- allspice wants IP address for gaia.cs.umass.edu

iterative query:

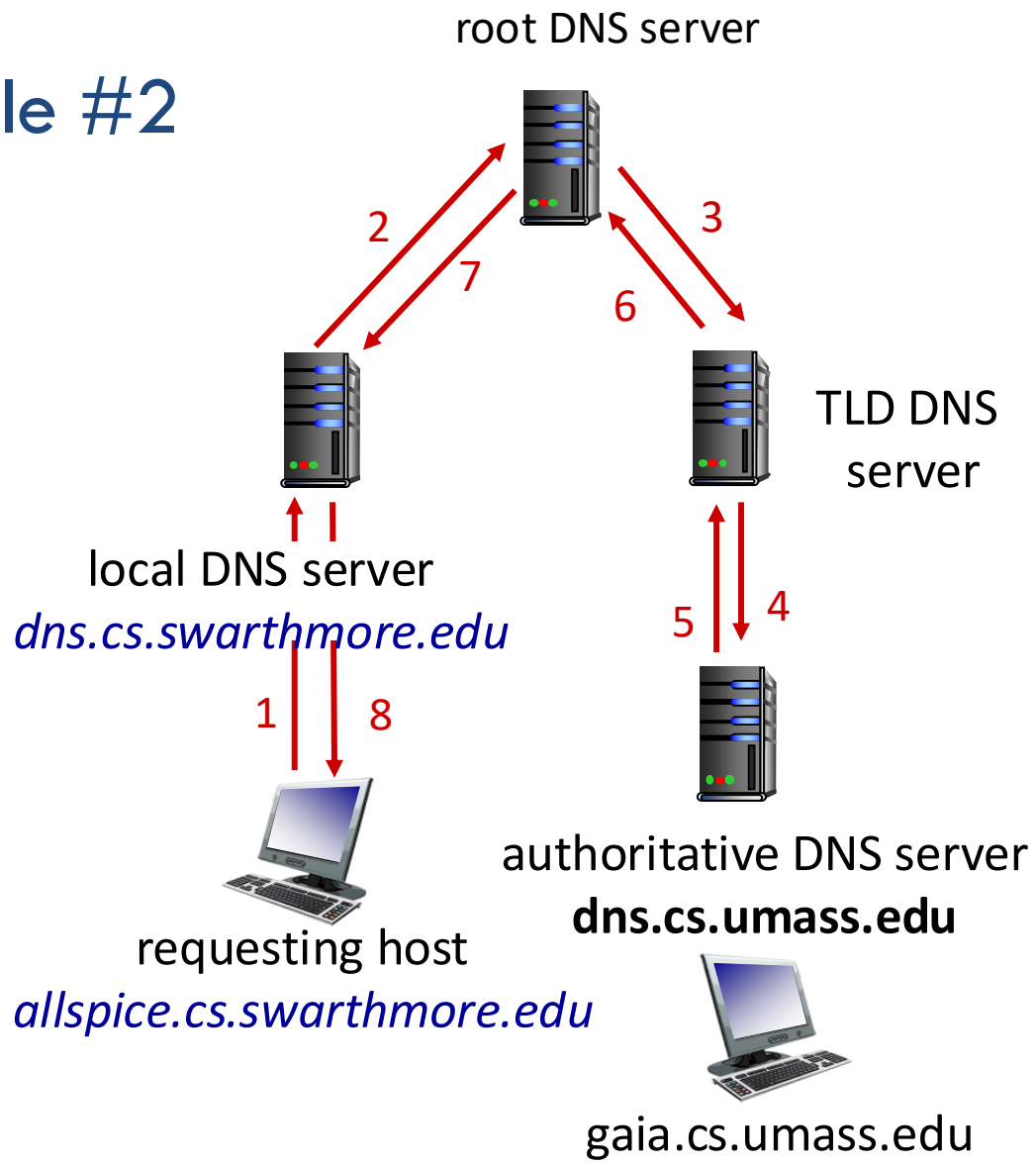
- contacted server replies with name of server to contact
- "I don't know this name, but ask this server"



DNS name resolution example #2

recursive query:

- each server asks the next one, in a chain

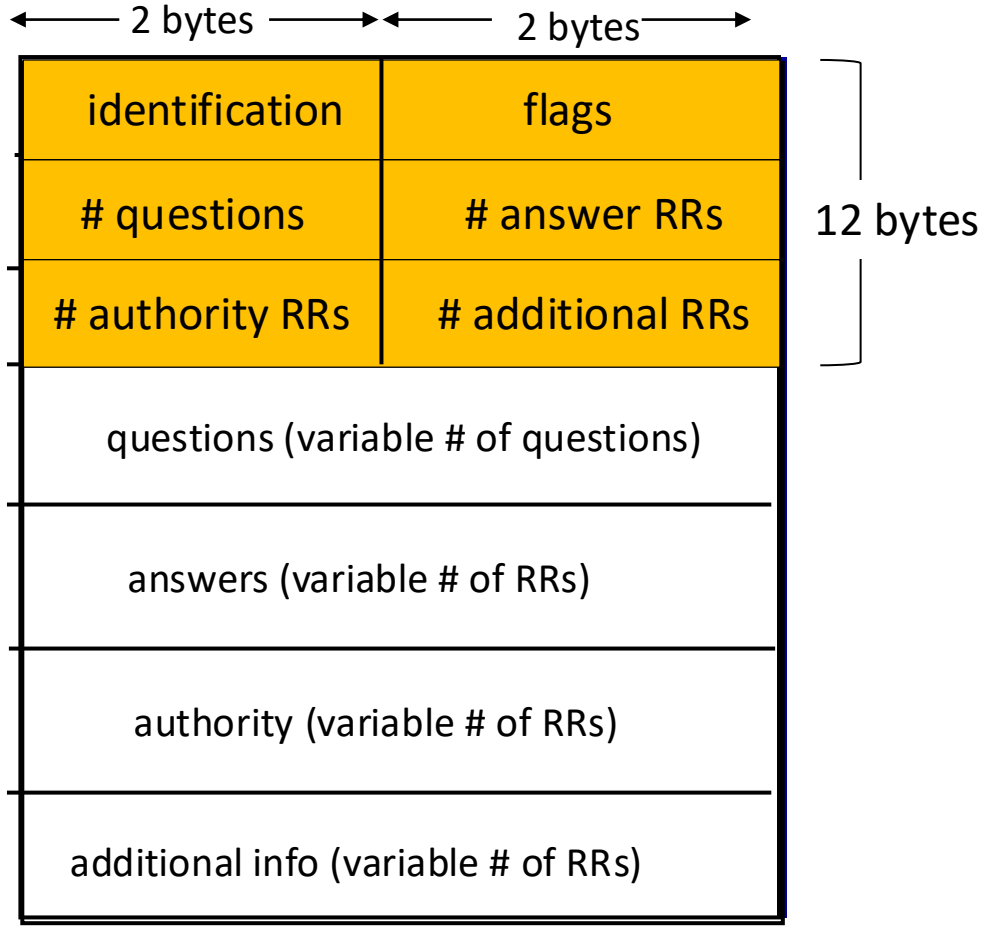


DNS protocol, messages

- query and reply messages, both with same message format

Message header

- identification: 16 bit id for query, reply to query uses same id.
- flags: recursion, query/reply
- # Resource Records to follow



DNS protocol, messages

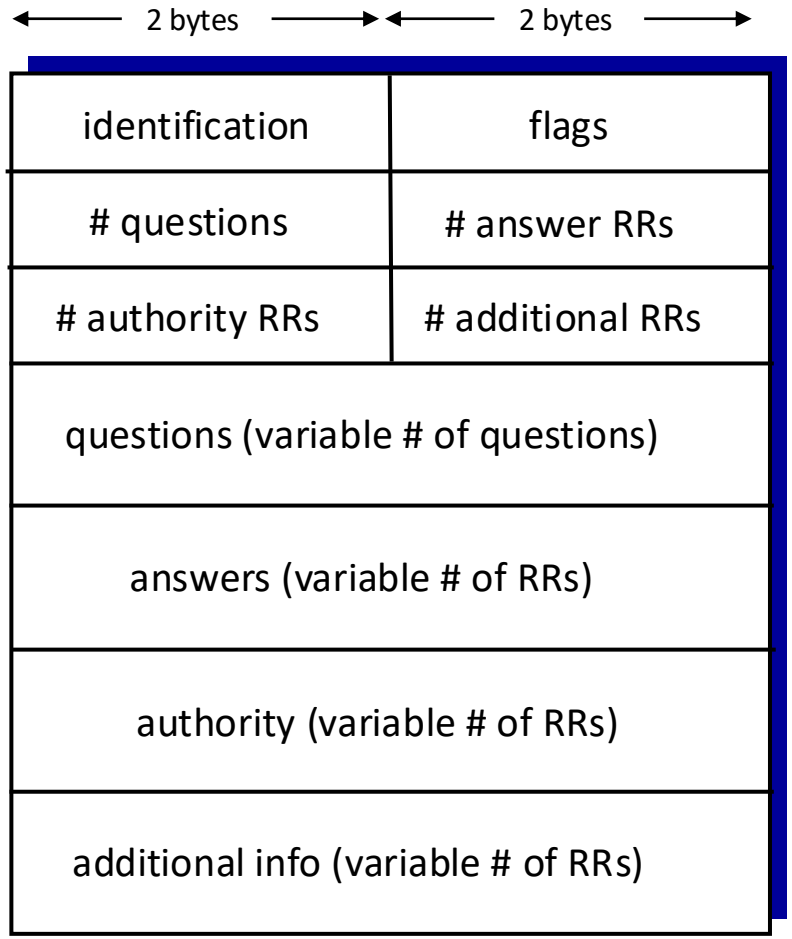
- query and reply messages, both with same message format!

Binary Protocol!

- Delimiters: pre-defined lengths/field
- Names: <len><name>

Sent via UDP (User Datagram Protocol)

- No connection established
- Not reliable!



DNS Types

RR format: `(name, value, type, ttl)`

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

Query
Name: cs.swarthmore.edu
Type: A

Resp.
Name: cs.swarthmore.edu
Value: 130.58.68.9

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

Query
Name: cs.swarthmore.edu
Type: NS

Resp.
Name: cs.swarthmore.edu
Value: dns.cs.swarthmore.edu

DNS Types, Continued

RR format: `(name, value, type, ttl)`

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

Query
Name: foo.mysite.com
Type: CNAME

Resp.
Name: foo.mysite.com
Value: bar.mysite.com

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

Query
Name: cs.umass.edu
Type: MX

Resp.
Name: cs.umass.edu
Value: barramail.cs.umass.edu.

DNS Records

DNS: distributed DB storing resource records (RR)

RR format: `(name, value, type, ttl)`

type=A

- **name** is hostname
- **value** is IP address

type=NS

- **name** is domain (e.g., foo.com)
- **value** is hostname of authoritative name server for this domain

type=CNAME

- **name** is alias name for some "canonical" (the real) name
- **www.ibm.com** is really
- servereast.backup2.ibm.com
- **value** is canonical name

type=MX

- **value** is name of mailserver associated with name

Tools

- dig
  - $ dig cs.swarthmore.edu
  - $ dig cs.swarthmore.edu ns
  - $ dig @dns.cs.swarthmore.edu cs.swarthmore.edu mx
  - $ man dig
- host
  - $ host cs.swarthmore.edu
  - $ host -t ns cs.swarthmore.edu
  - $ host -t mx cs.swarthmore.edu dns.cs.swarthmore.edu
  - $ man host

Tools (cont)

- nslookup
  - $ nslookup cs.swarthmore.edu
  - $ nslookup cs.swarthmore.edu dns.cs.swarthmore.edu

- whois
  - $ whois google.com
  - $ whois swarthmore.edu

DNS Query Process and Cache