

CS 43: Computer Networks

Naming and DNS

September 19, 2019



Reading Quiz

Last class

- Distributed systems architectures
 - Client-Server
 - Peer-to-Peer
- Challenges in designing distributed systems
 - Partial failures

Today

- Identifiers and addressing
- Domain Name System
 - History
 - Query sequences
 - Record types
 - Load balancing

DNS: domain name system

People: many identifiers:

- SSN, name, passport #

Internet hosts, routers:

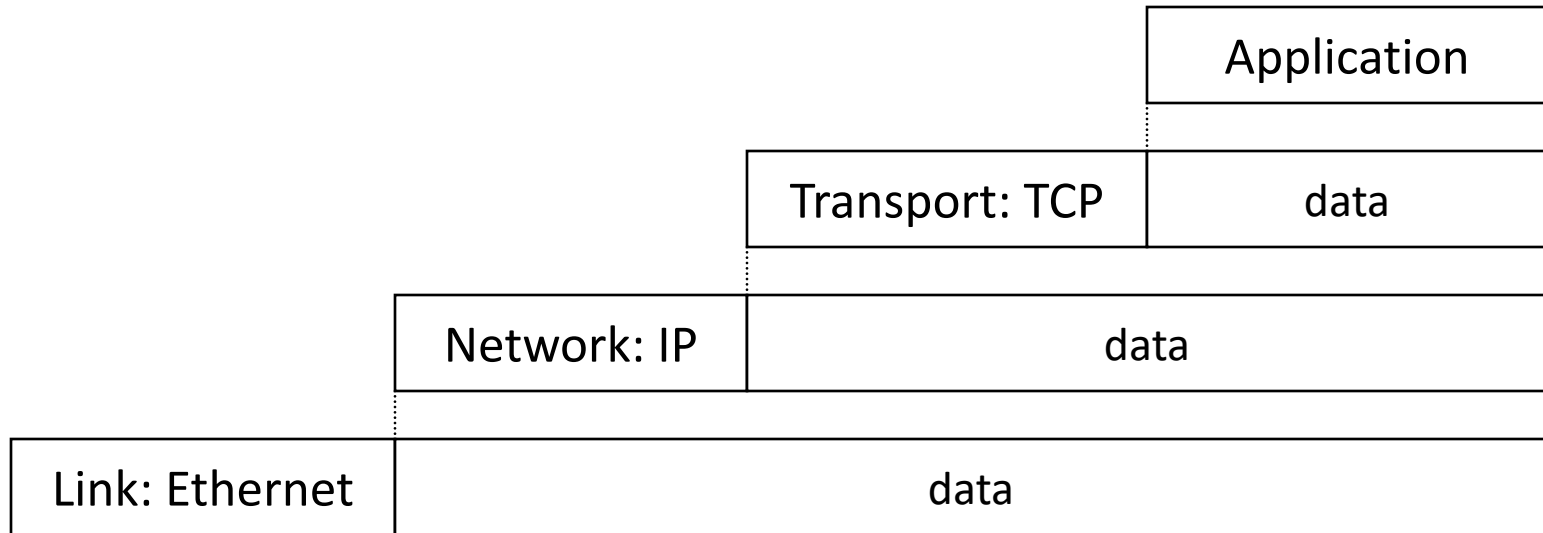
- IP address (32 bit) - used for addressing datagrams
- “name”, e.g., www.google.com - used by humans

How do we map between IP address and name, and vice versa ?

DNS: domain name system

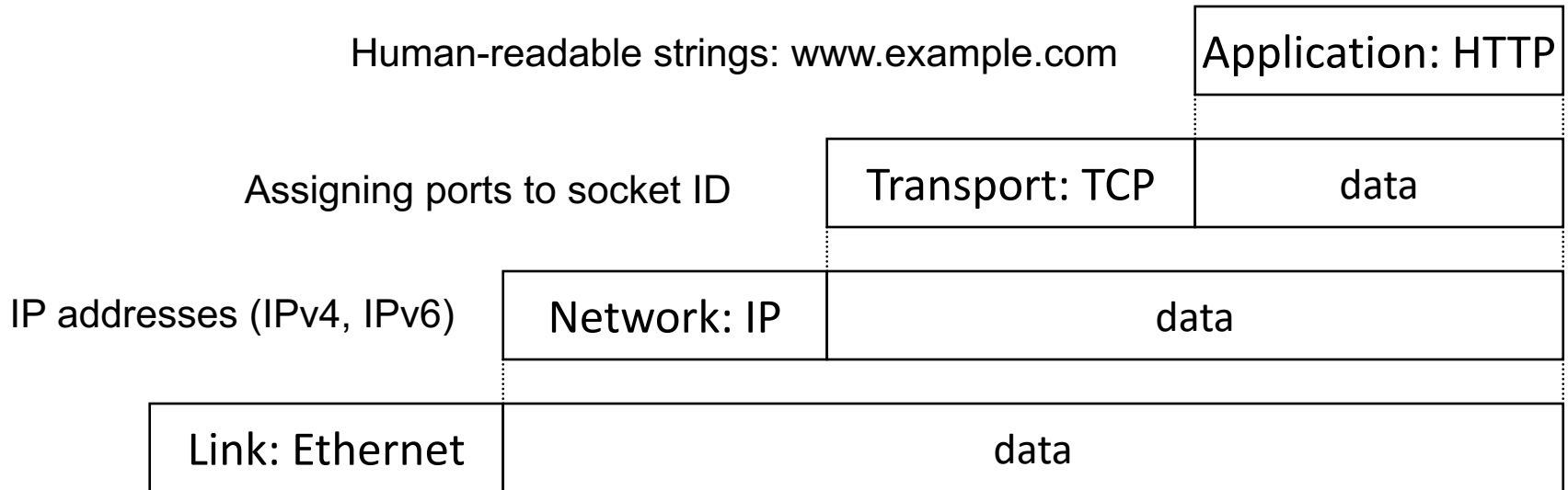
- **distributed database** implemented in hierarchy of many **name servers**.
- **application-layer protocol**: hosts, name servers communicate to **resolve** names → addresses
 - core Internet function, implemented as application-layer protocol
 - complexity at network's "edge"

Recall: Addressing and Encapsulation



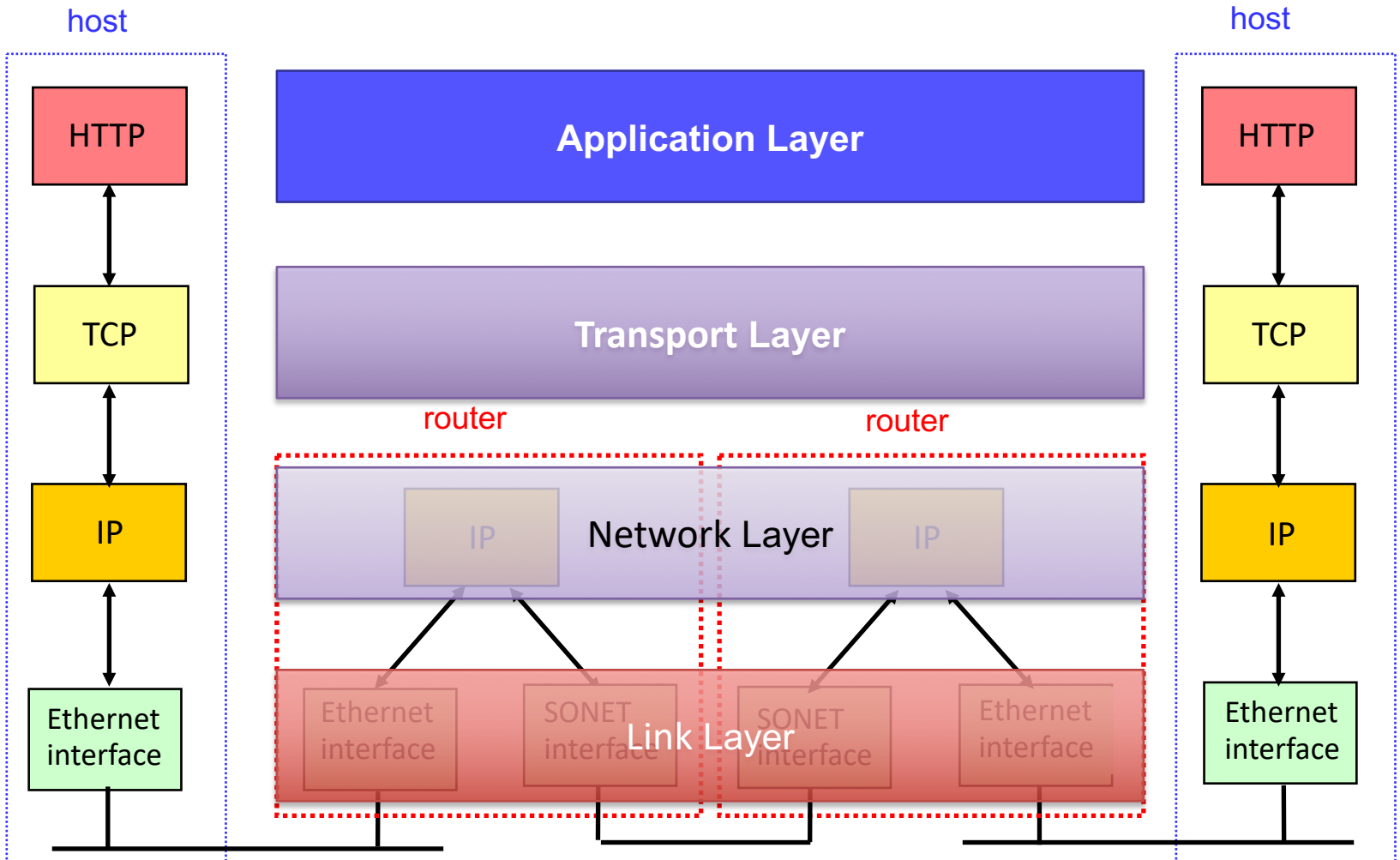
- Higher layer within lower layer
- Each layer has different concerns, provides abstract services to those above

Recall: Addressing and Encapsulation

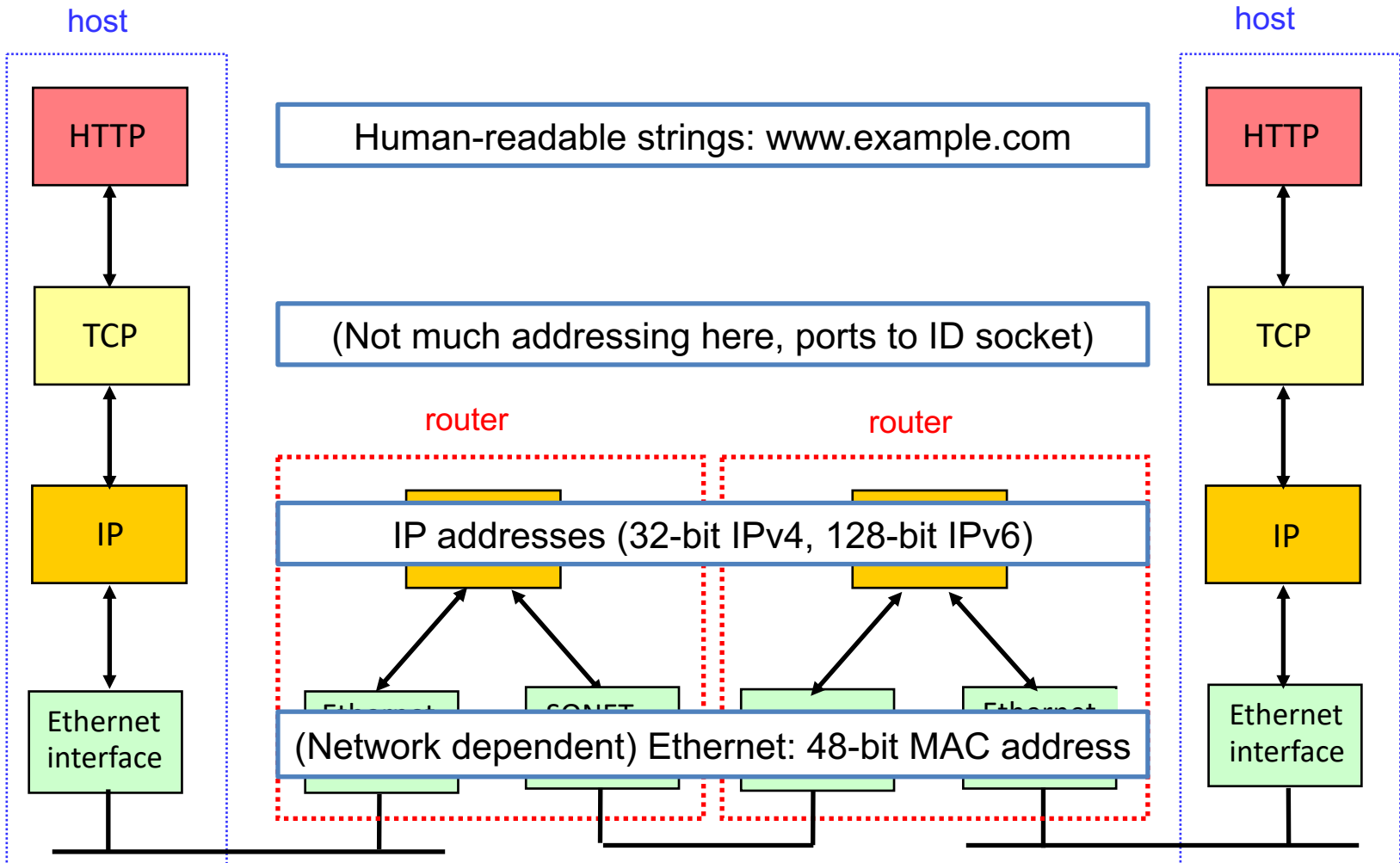


(Network dependent) Ethernet:
48-bit MAC address

TCP/IP Protocol Stack



Recall: TCP/IP Protocol Stack



Why do we need to map names to IP addresses? Why not route on names at the network layer?

- A. Domain names are hierarchical, so we can route on domain names too.
- B. Domain names are variable length, vs IP are fixed length, some changes will be required to switch.
- C. With domain names we wouldn't know where to route to geographically.
- D. Some other reason.

What's in a name?

- **Host name** (e.g., www.swarthmore.edu)
 - Used by humans to specify host of interest
 - Unique, selected by host administrator
 - Hierarchical, **variable-length string** of alphanumeric characters

What's in a name?

- **IP address** (e.g., 130.58.68.164)
 - Used by routers to forward packets
 - Unique, **topologically meaningful** locator
 - Hierarchical namespace of **32 bits**

What's in a name?

- **MAC address** (e.g., D8:D3:85:94:5F:1E)
 - Used by network adaptors to identify frames
 - Unique, **hard-coded identifier** burned into network adaptor
 - Flat name space (of 48 bits in Ethernet)

Identifiers

- Host name: **web.cs.swarthmore.edu**
 - **Domain**: registrar for each top-level domain (e.g., .edu)
 - **Host name**: local administrator assigns to each host
- IP addresses: **130.58.68.164**
 - **Prefixes**: ICANN, regional Internet registries, and ISPs
 - **Hosts**: static configuration, or dynamic using DHCP
- MAC addresses: **D8:D3:85:94:5F:1E**
 - **OIDs**: assigned to vendors by the IEEE
 - **Adapters**: assigned by the vendor from its block

Identifiers

- Host name: **web.cs.swarthmore.edu** (today)
 - **Domain**: registrar for each top-level domain (e.g., .edu)
 - **Host name**: local administrator assigns to each host
- IP addresses: **130.58.68.164** (a few weeks)
 - **Prefixes**: ICANN, regional Internet registries, and ISPs
 - **Hosts**: static configuration, or dynamic using DHCP
- MAC addresses: **D8:D3:85:94:5F:1E** (after fall break)
 - **OIDs**: assigned to vendors by the IEEE
 - **Adapters**: assigned by the vendor from its block

Mapping Between Identifiers

- Domain Name System (**DNS**)
 - Given a host name, provide the IP address
 - Given an IP address, provide the host name
- Address Resolution Protocol (**ARP**)
 - Given an IP address, provide the MAC address
 - To enable communication within the Local Area Network
- Dynamic Host Configuration Protocol (**DHCP**)
 - Automates host boot-up process
 - Given a MAC address, assign a unique IP address
 - ... and tell host other stuff about the Local Area Network

What's the biggest challenge for DNS?

What's the biggest challenge for DNS?

- A. It's old.
- B. The fact that the Internet is global.
- C. The fact that DNS is now critical infrastructure.
- D. The sheer number of name lookups happening at any given time.
- E. How and when the name -> IP address mapping should change.

In the old days...

- Pre-1982, everyone downloads a “hosts.txt” file from SRI
- Pre-1998, Jon Postel, researcher at USC, runs the **Internet Assigned Numbers Authority (IANA)**
 - RFCs 882 & 883 in 1983
 - RFCs 1034 & 1035 in 1987



Emailed 8/12 root DNS servers, asked change to his authority. They did.

<http://www.wired.com/wiredenterprise/2012/10/joe-postel/>

Since 1998...

- Control of Internet Assigned Numbers Authority (IANA) transferred to **Internet Corporation for Assigned Names and Numbers (ICANN)**
 - ICANN is a private non-profit (formerly) blessed by US DOC
 - Global advisory committee for dealing with international issues
 - 2000's: Many efforts for UN control, US resisted
 - 2016: ICANN no longer partnered with DOC

Who should control DNS?

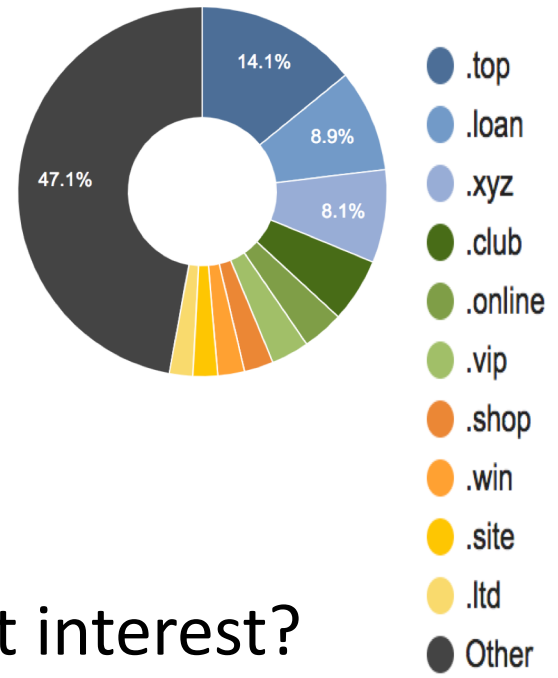
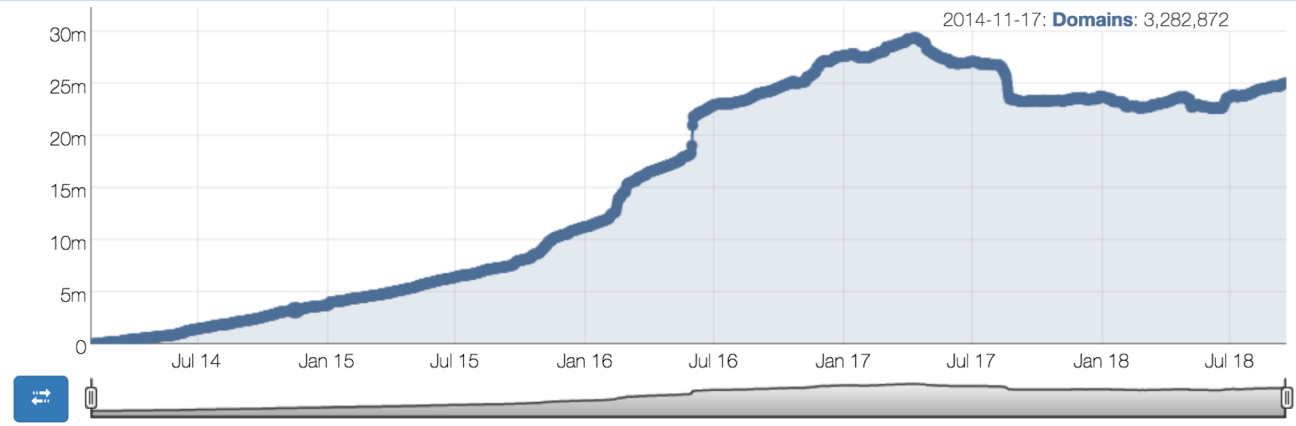
- A. US government
- B. UN / International government
- C. Private corporation
- D. Someone else

Recent Controversy

Advertisement

new gTLD Domains

new gTLD distribution



- Is ICANN working in the world's best interest?
- New "top level domains" added, for auction

Contributing to change?

🏠 > Internet standards >

RFCs

Memos in the RFC document series contain technical and organizational notes about the Internet.

Division of Computer and Network Systems (CNS)

CNS invents new computing and networking technologies, while ensuring their security and privacy, and finds new ways to make use of current technologies.



<https://www.whitehouse.gov/ostp/> <https://www.nsf.gov/cise/cns/about.jsp>

<https://www.icann.org/public-comments#open-public>

<https://www.ietf.org/standards/rfc/>

DNS Services

- DNS is an **application-layer protocol**. E2E design!
- It provides:
 - **Hostname to IP address translation**
 - Host aliasing (canonical and alias names)
 - Mail server aliasing
 - Load distribution (one name may resolve to multiple IP addresses)
 - Lots of other stuff that you might use a directory service to find. (Wikipedia: List of DNS record types)

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

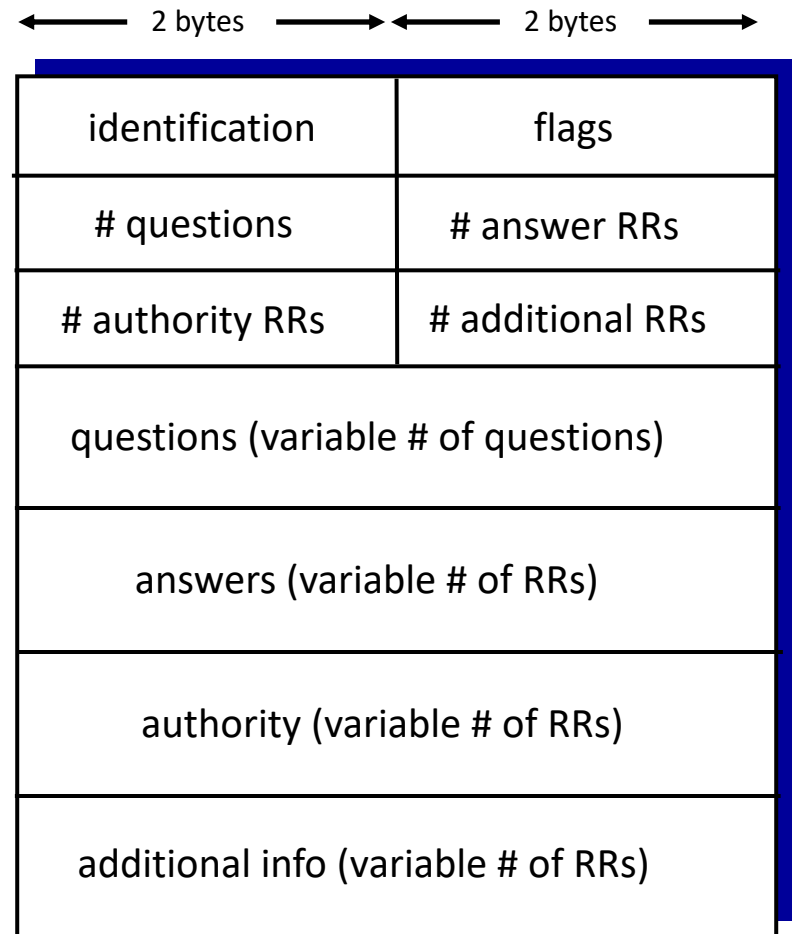
- **name** is alias name for some “canonical” (the real) name
- **www.ibm.com** is really servereast.backup2.ibm.com
- **value** is canonical name

type=MX

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

DNS protocol, messages

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



DNS protocol, messages

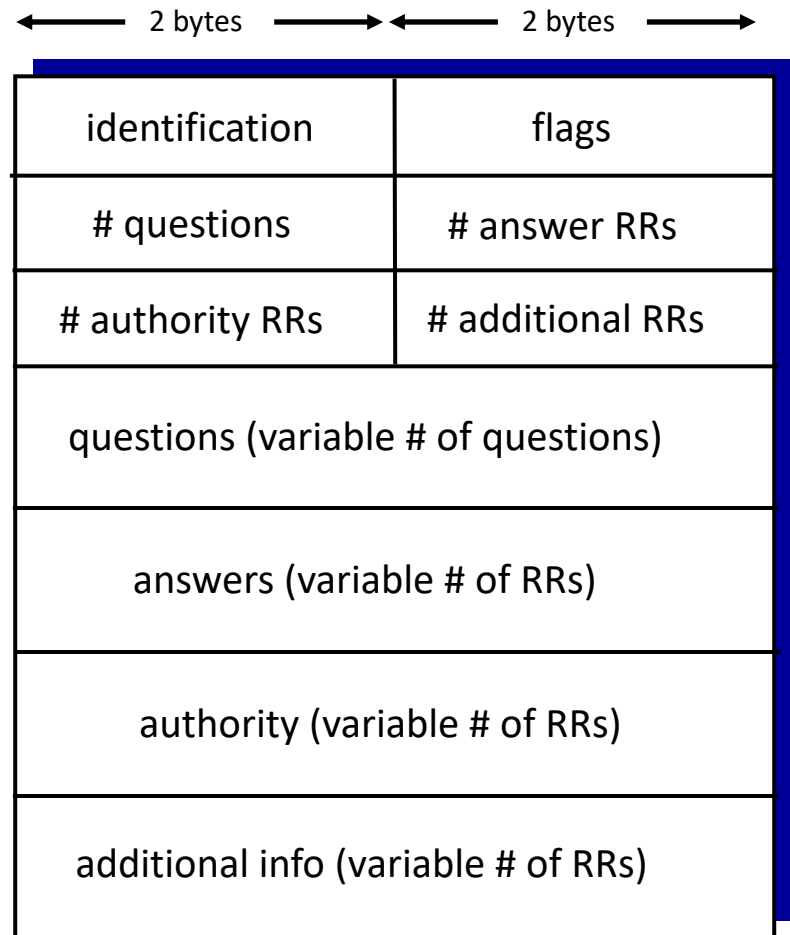
- **query** and **reply** messages, both with same **message format**

Message header

- **identification**: 16 bit # for query, reply to query uses same #
- **flags**:
 - query or reply
 - recursion desired
 - recursion available
 - reply is authoritative

Sent via UDP!

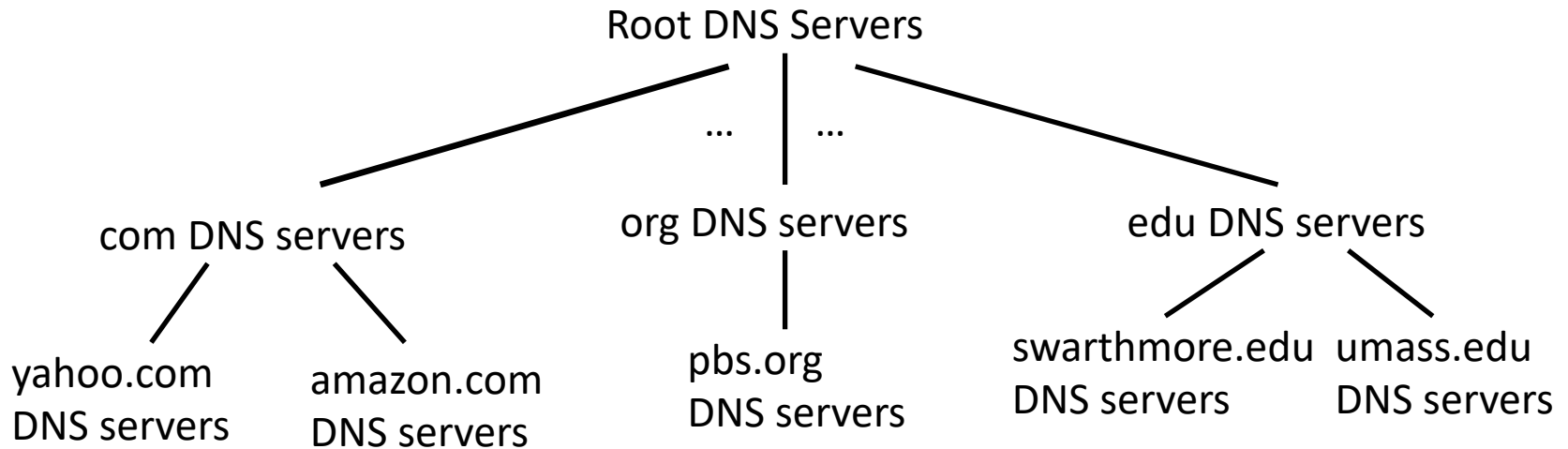
- No connection established
- Not reliable



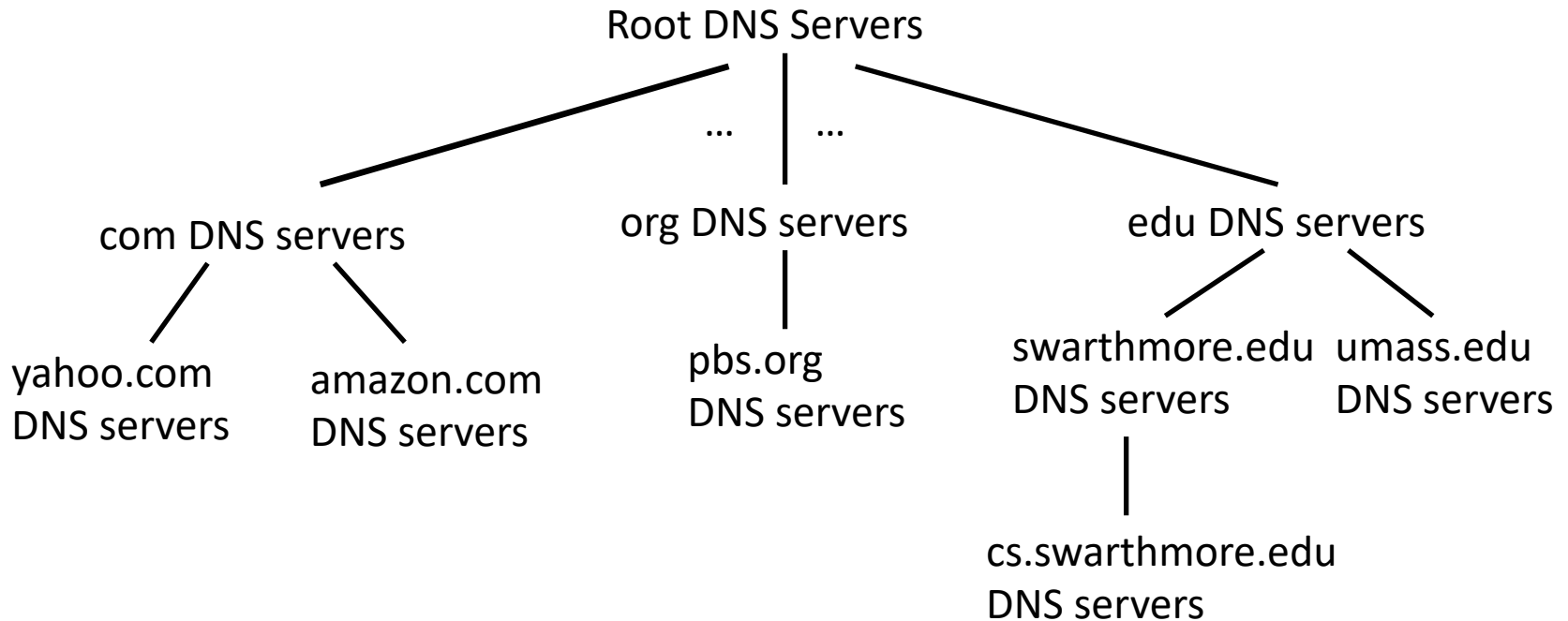
Domain Name System (DNS)

- Distributed administrative control
 - Hierarchical name space divided into zones
 - Distributed over a collection of DNS servers
- Hierarchy of DNS servers
 - Root servers
 - Top-level domain (TLD) servers
 - Authoritative DNS servers
- Performing the translations
 - Local DNS servers
 - Resolver software

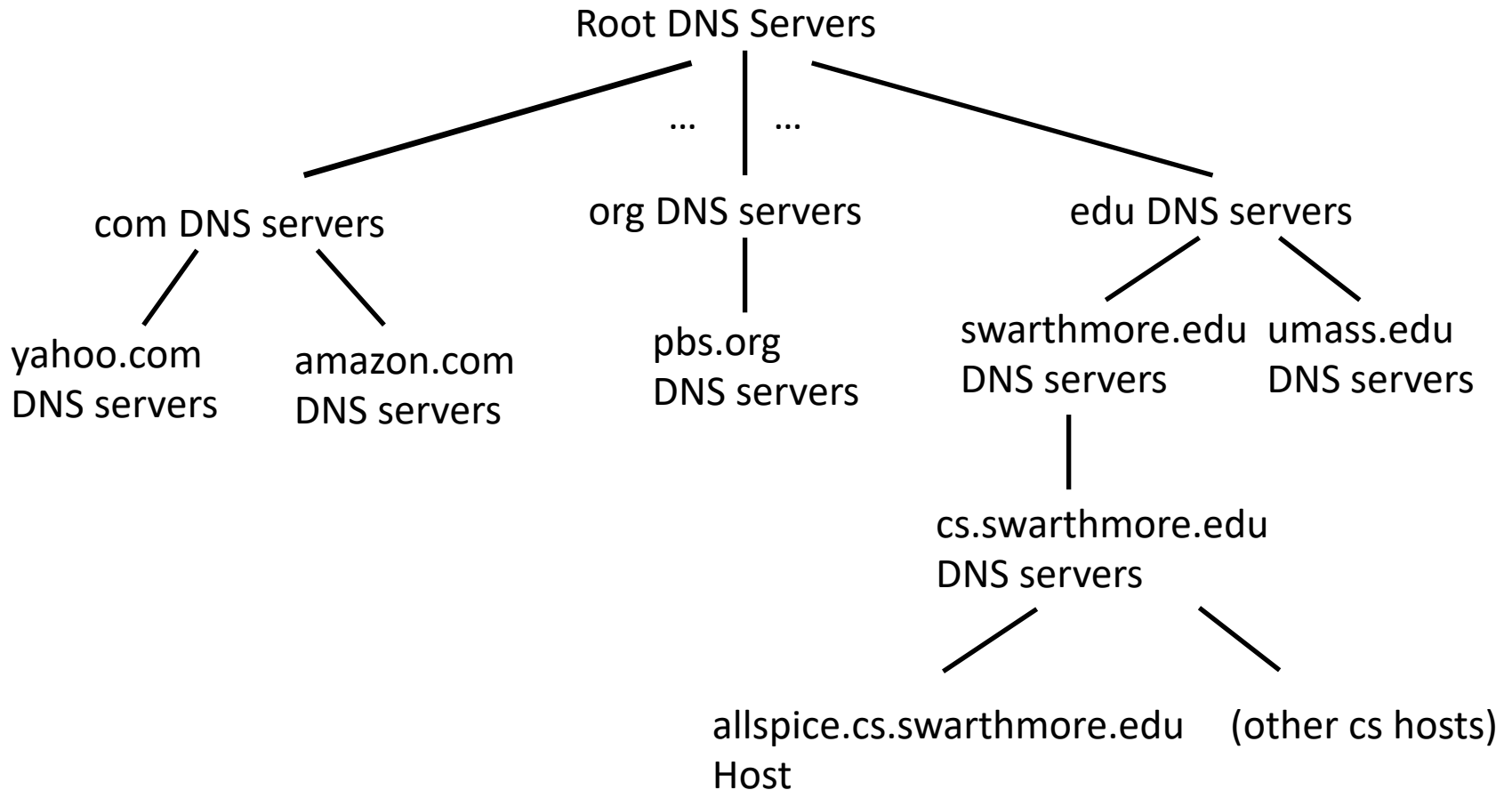
DNS: a distributed, hierarchical database



DNS: a distributed, hierarchical database



DNS: a distributed, hierarchical database



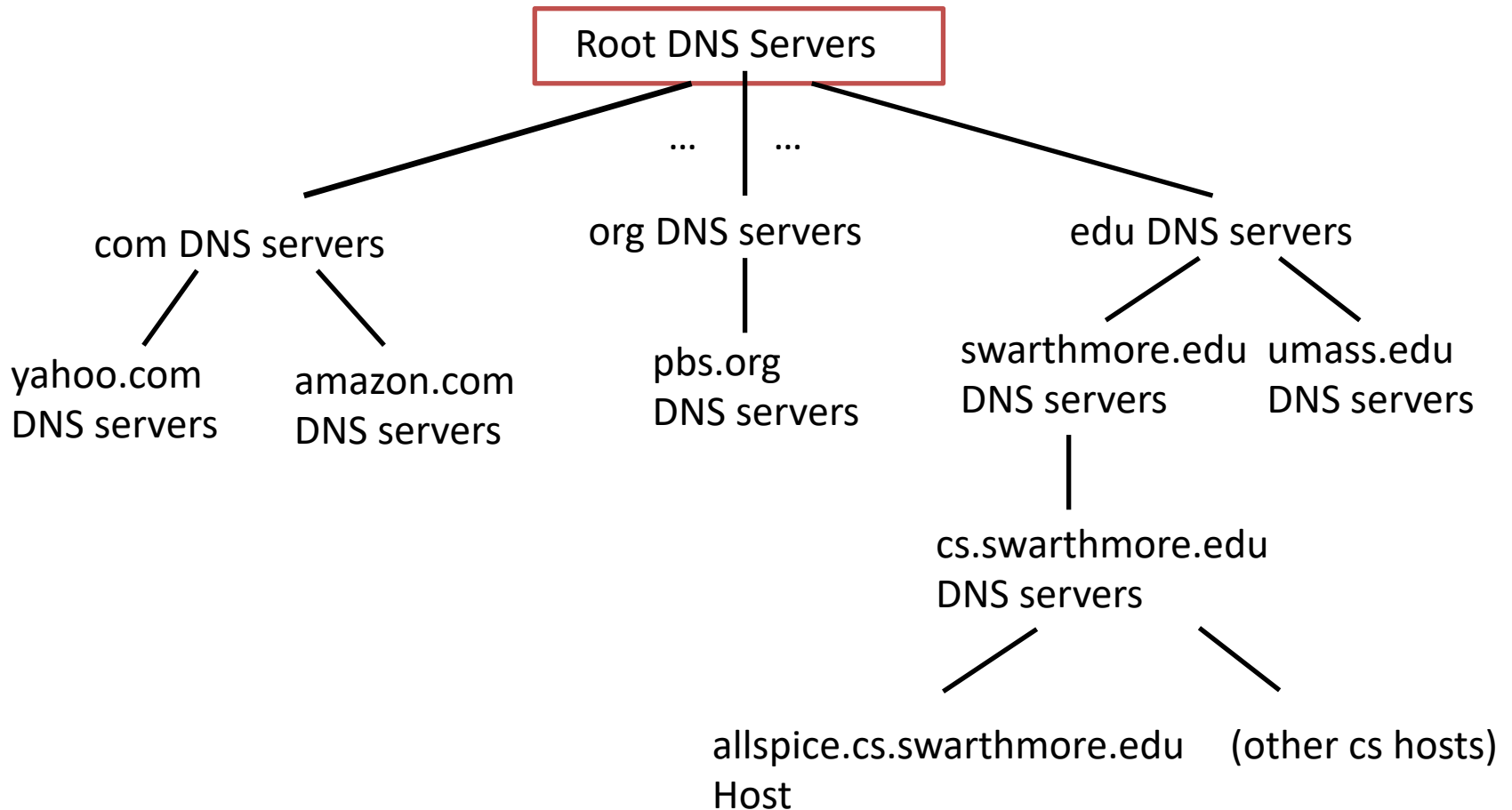
- allspice.cs.swarthmore.edu.

Nameless root,
Usually implied.

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

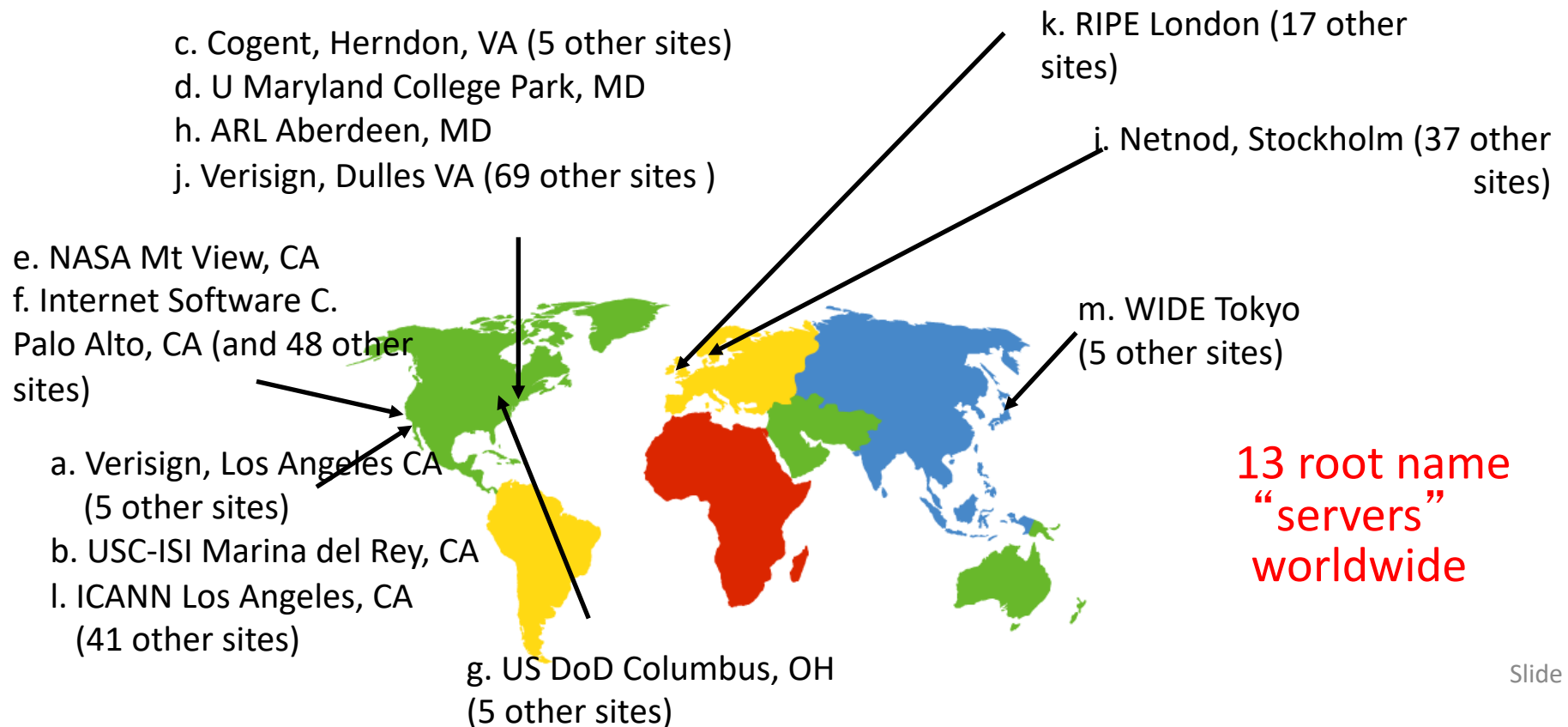
- A. It divides up responsibility among parties.
- B. It improves performance of the system.
- C. It reduces the size of the state that a server needs to store.
- D. Some other reason.

DNS: a distributed, hierarchical database



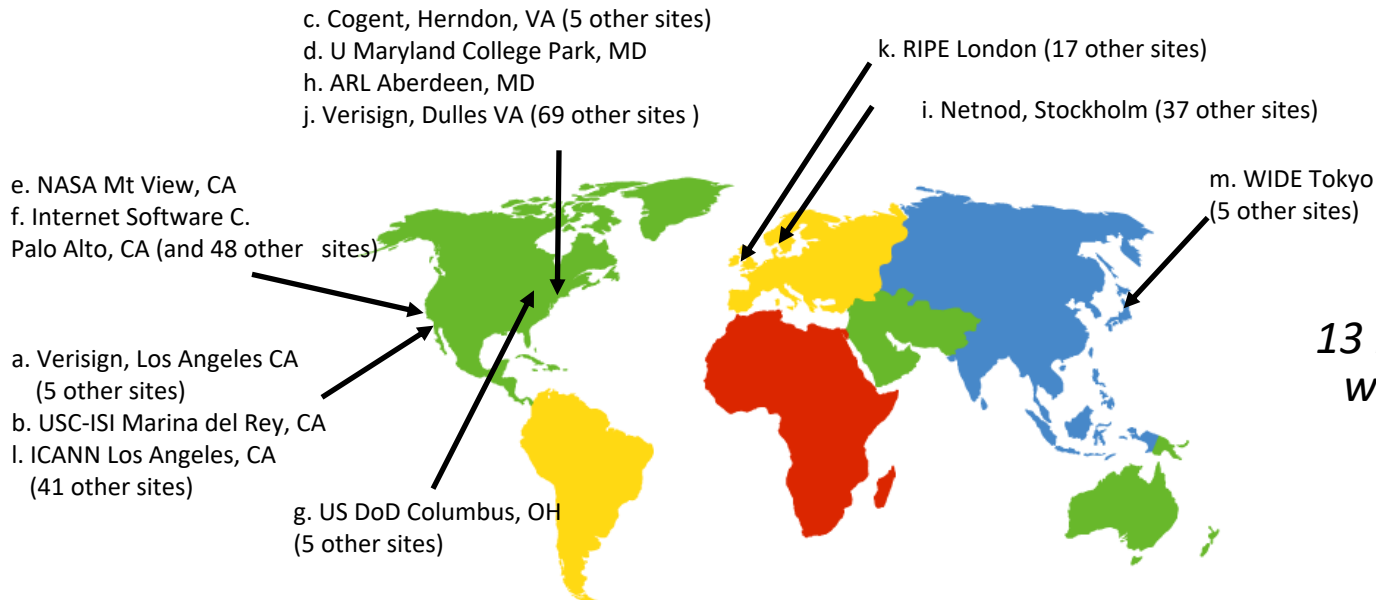
DNS: Root Name Servers

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



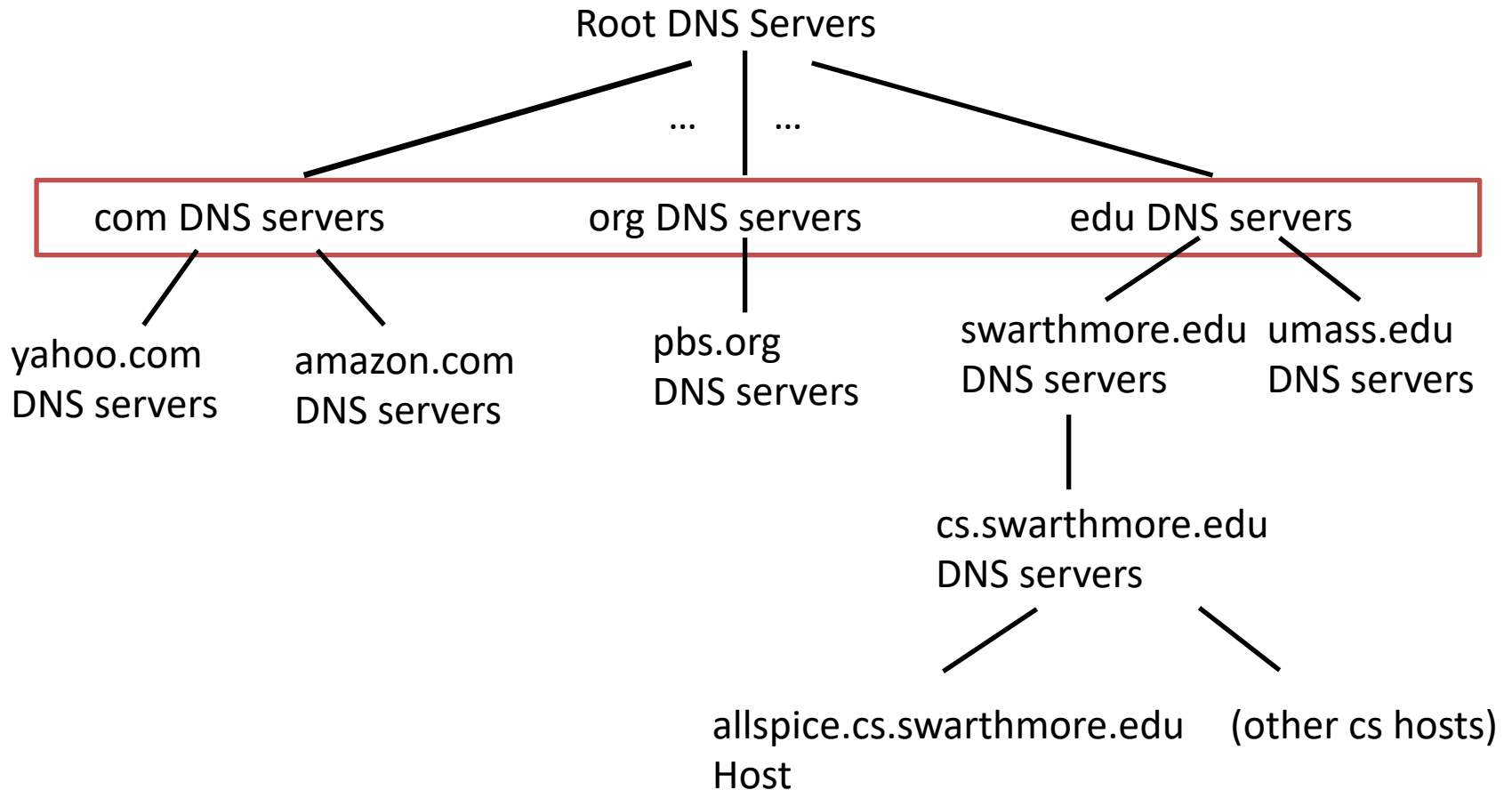
DNS: Root Name Servers

- Root name server:
 - Knows how to find top-level domains (.com, .edu, .gov, etc.)
 - How often does the location of a TLD change?
 - approx. 400 total root servers
 - Significant amount of traffic is not legitimate



*13 root name “servers”
worldwide*

DNS: a distributed, hierarchical database

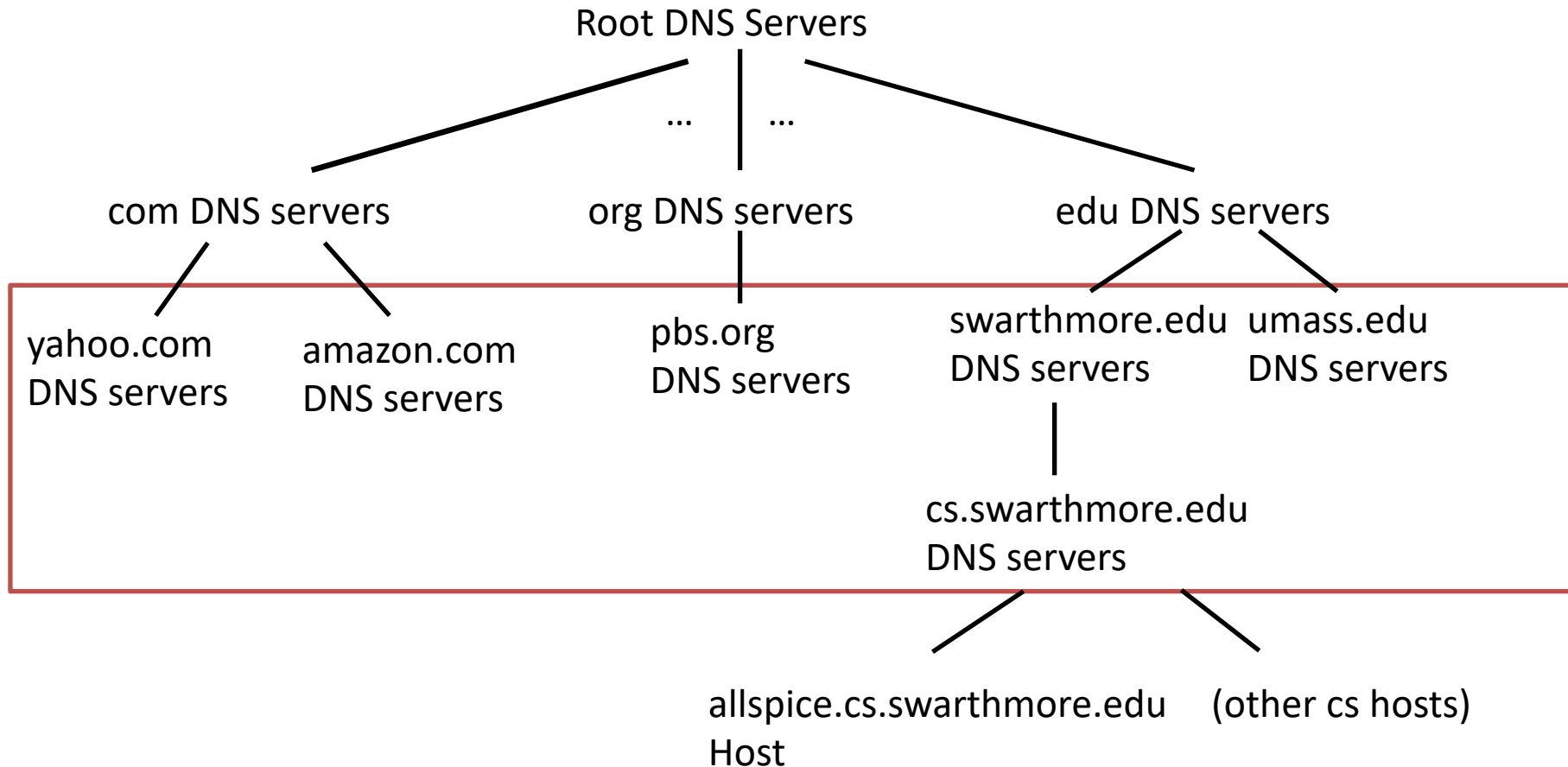


Top Level Domains

Top-level domain (TLD) servers:

- Responsible for com, org, net, edu, gov, aero, jobs, museums, and all top-level country domains, e.g.: uk, fr, de, ca, jp, etc.
- Verisign maintains servers for .com and .net TLD
- Educause for .edu TLD (Verisign actually runs backend)
- Others managed by corresponding entity (e.g., local governments or companies)

DNS: a distributed, hierarchical database



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

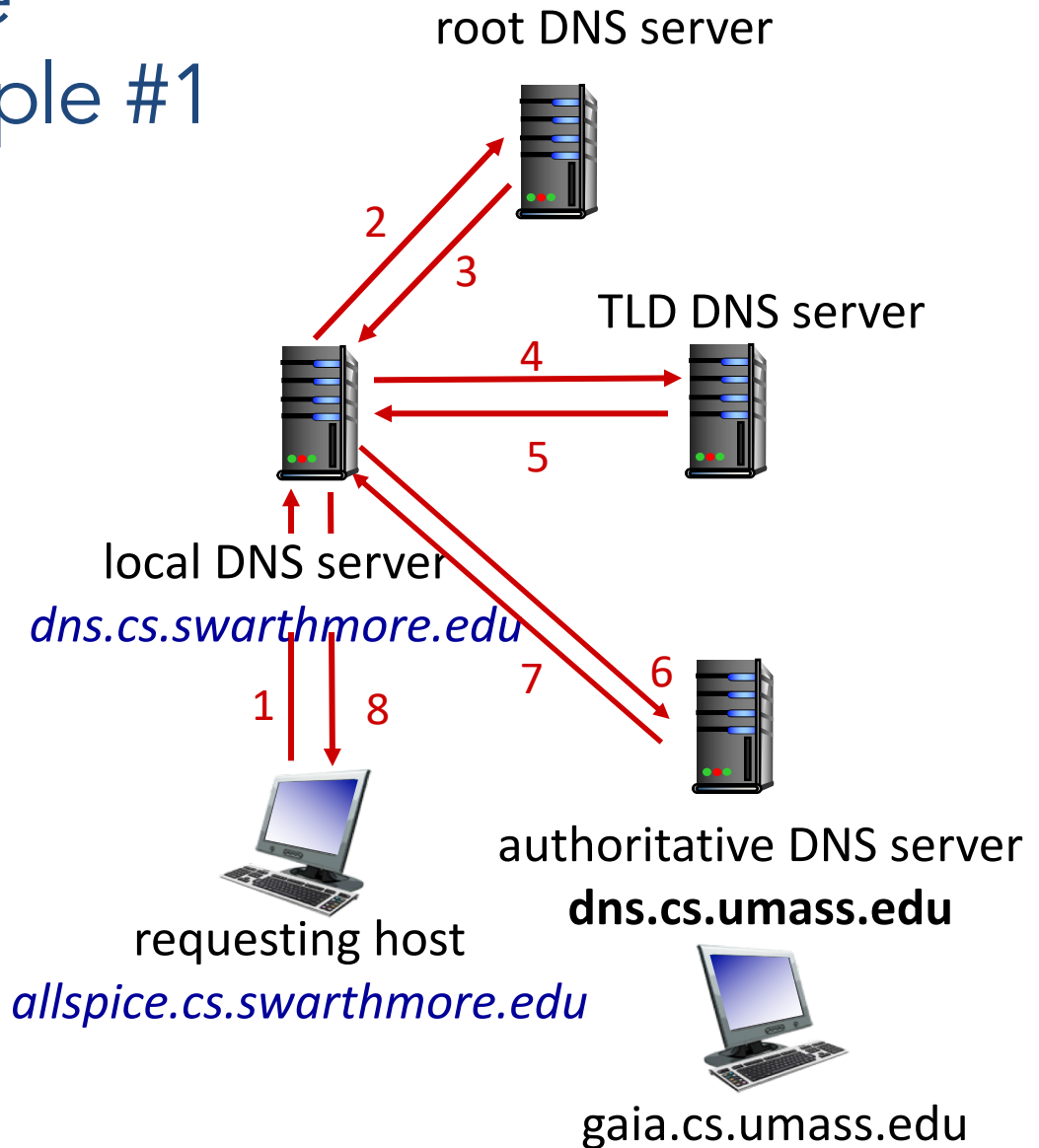
- Each ISP (residential ISP, company, university) has (at least) one
 - also called “default name server”
- When host makes DNS query, query is sent to its local DNS server
 - has local cache of recent name-to-address translation pairs (but may be out of date!)
 - acts as proxy, forwards query into hierarchy

DNS name resolution example #1

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

iterative query:

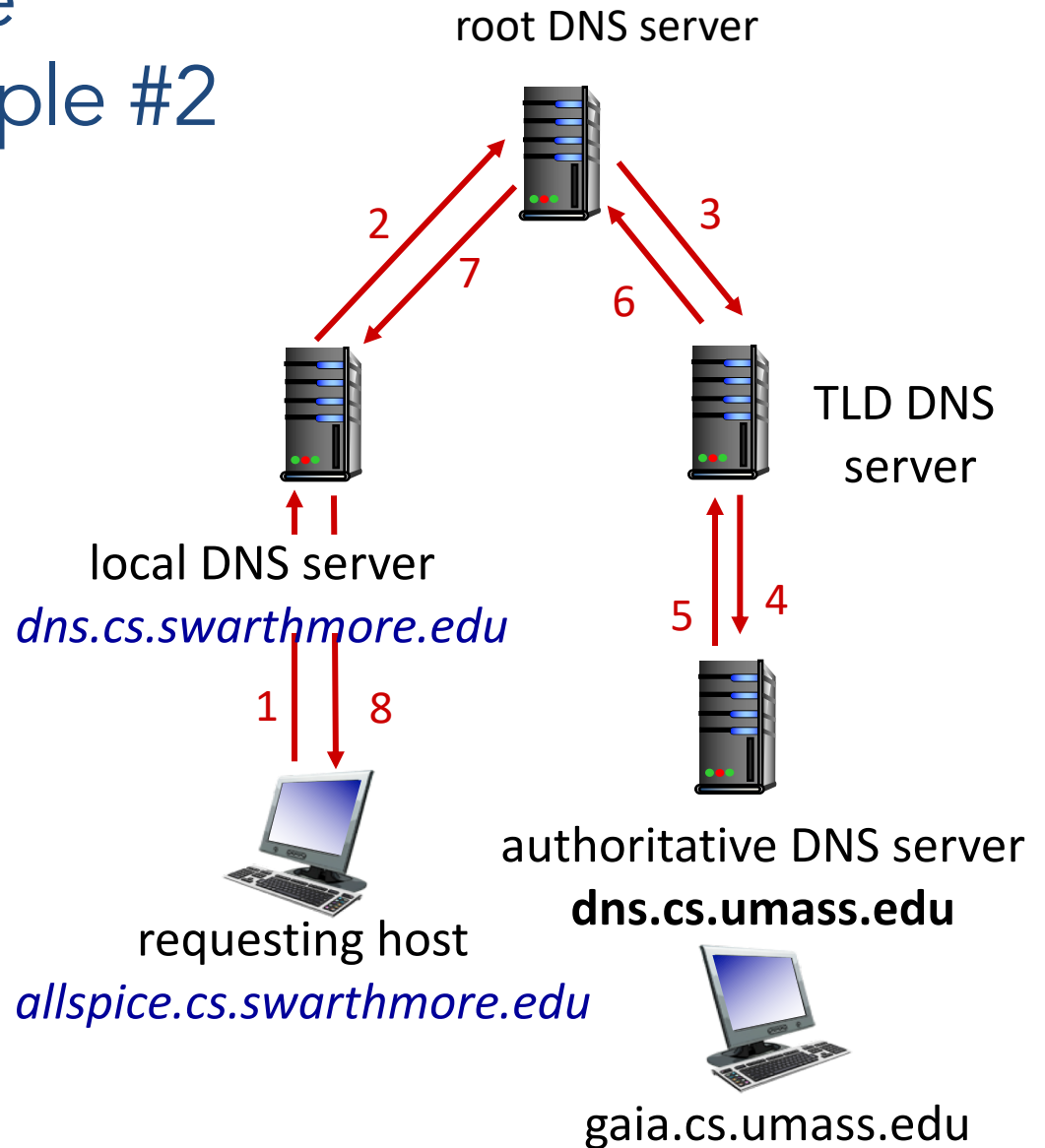
- contacted server replies with name of server to contact
- “I don’t know this name, but ask this server”



DNS name resolution example #2

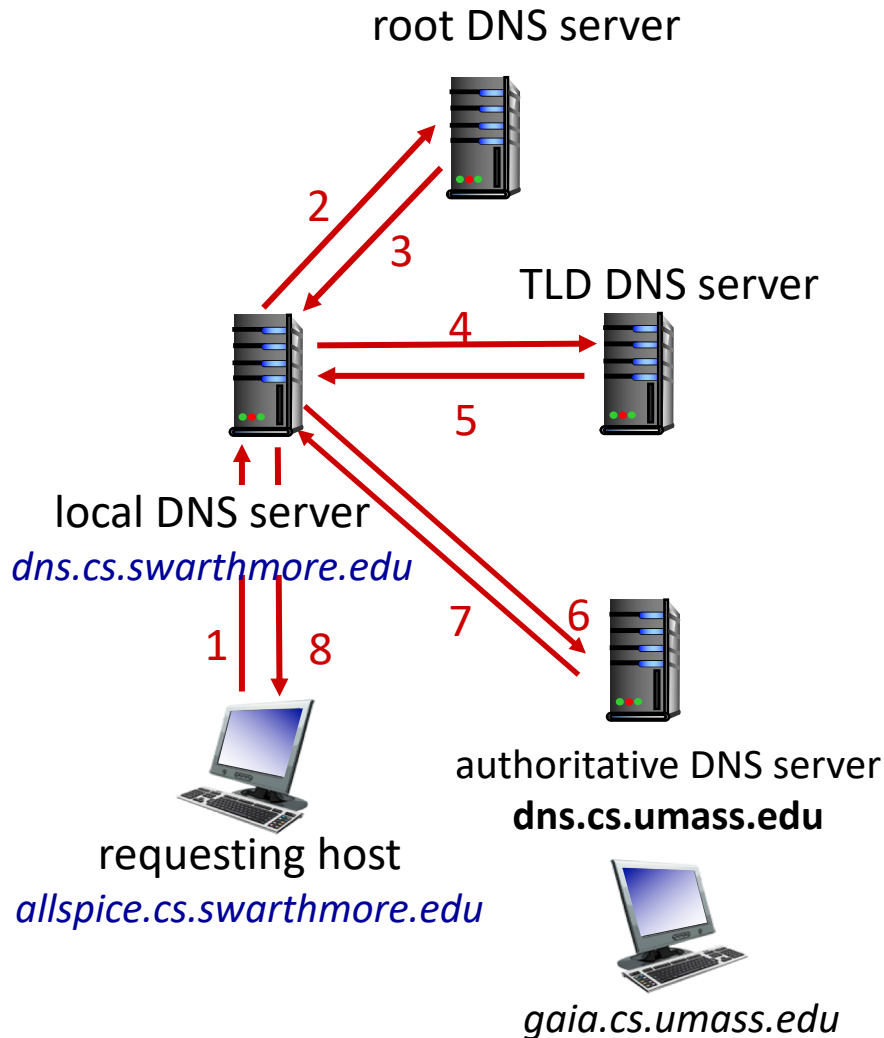
recursive query:

- each server asks the next one, in a chain

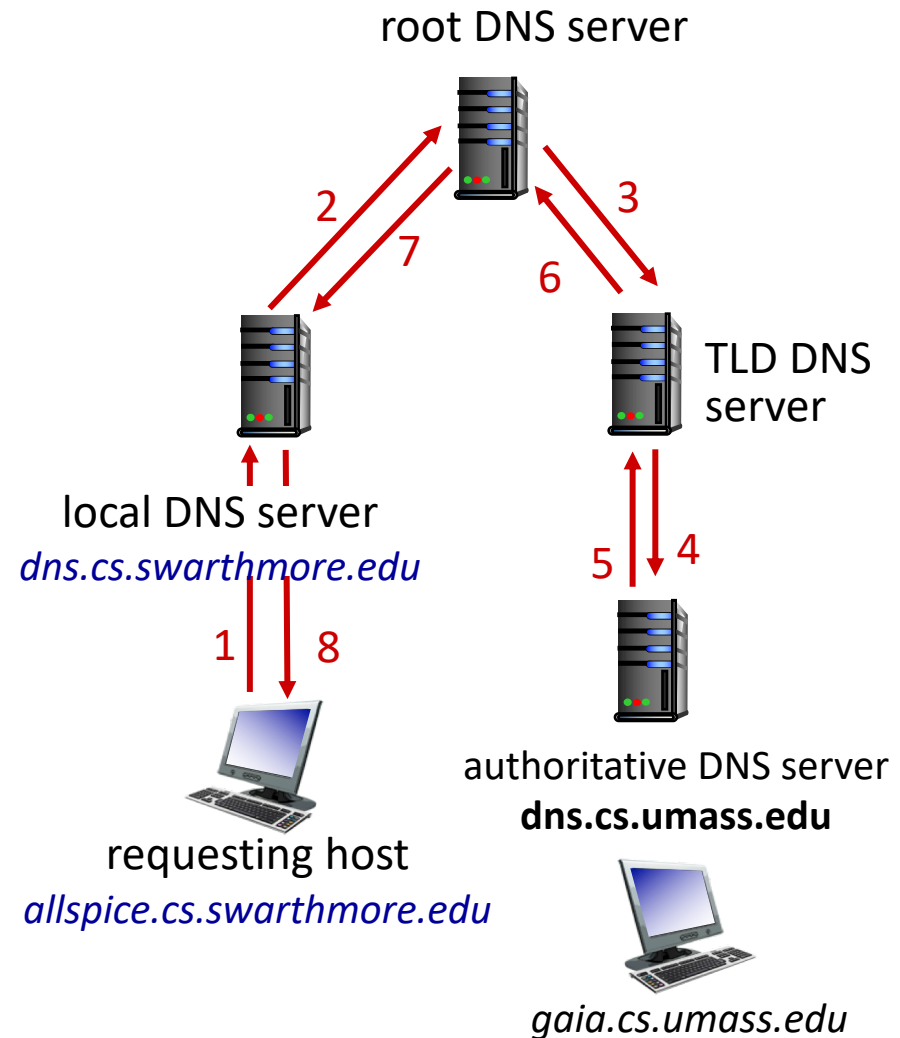


Which would you use? Why?

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
```

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
 - Thus root name servers not often (legitimately) visited

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

Inserting (or changing) records

- Example: new startup “Network Utopia”
- Register networkutopia.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.networkutopia.com
 - **type MX record** for @networkutopia.com email

Attacking DNS

DDoS attacks

- Bombard root servers with traffic
 - Not successful to date
 - Traffic Filtering
 - Local DNS servers cache IPs of TLD servers, bypassing root
- Bombard TLD servers
 - Potentially more dangerous

Redirect attacks

- Man-in-middle
 - Intercept queries
- DNS poisoning
 - Send bogus replies to DNS server that caches

Exploit DNS for DDoS

- Send queries with spoofed source address: target IP
- Requires amplification

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