CSCI 491-01
Topics: Internet Programming
Fall 2008

Application Layer

Derek Leonard
Hendrix College

September 17, 2008

Chapter 2: Roadmap

2.1 Principles of network applications

2.2 Web and HTTP

2.3 FTP

2.4 Electronic Mail

- SMTP, POP3, IMAP

2.5 DNS

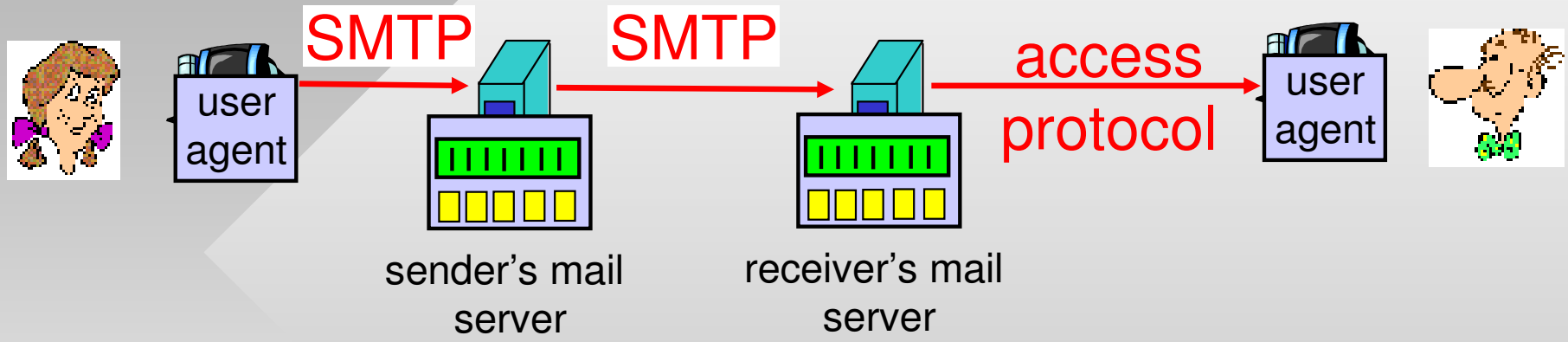
2.6 P2P file sharing

2.7 Socket programming with TCP

2.8 Socket programming with UDP

2.9 Building a Web server

Mail Access Protocols



- SMTP: delivery/storage to receiver's server
- Mail access protocol: retrieval from server
  - POP: Post Office Protocol [RFC 1939] – port 110
    - Authorization (agent <-->server) and download
  - IMAP: Internet Mail Access Protocol [RFC 1730] – port 143
    - More features (more complex)
    - Manipulation of stored messages on server
  - HTTP: Hotmail, Yahoo! Mail, etc.

POP3 Protocol

Authorization phase

- Client commands:
  - `user`: declare username
  - `pass`: password
- Server responses
  - `+OK`
  - `-ERR`

Transaction phase, client:

- `list`: list message #s
- `retr`: retrieve message by number
- `dele`: delete
- `quit`

```
telnet mail.hendrix.edu 110
+OK POP3 server ready
user bob
+OK
pass hungry
+OK user successfully logged on
```

```
list
1 498
2 912
.
retr 1
<message 1 contents>
.
dele 1
retr 2
<message 2 contents>
.
dele 2
quit
+OK POP3 server signing off
```

POP3 (More) and IMAP

More about POP3

- Example used “download and delete” mode
- “Download-and-keep”
  - Multiple copies of message on different clients
- POP3 is stateless across sessions
  - Server assigns unique IDs to each message
  - Command UIDL lists IDs
  - UA determines new messages by remembering IDs of downloaded email

IMAP

- Keep all messages in one place: the server
- Allows user to organize messages in folders
- IMAP keeps user state across sessions:
  - Names of folders and mappings between message IDs and folder names

Chapter 2: Roadmap

2.1 Principles of network applications

2.2 Web and HTTP

2.3 FTP

2.4 Electronic Mail

- SMTP, POP3, IMAP

2.5 DNS

2.6 P2P file sharing

2.7 Socket programming with TCP

2.8 Socket programming with UDP

2.9 Building a Web server

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.yahoo.com – used by humans

Q: map between IP addresses and name?

Domain Name System:

- Distributed database
  - Implemented in hierarchy of many *name servers*
- Application-layer protocol
  - Hosts/routers/name servers communicate to *resolve* names (address/name translation)
  - Core Internet function, implemented as an application-layer protocol

DNS

DNS services

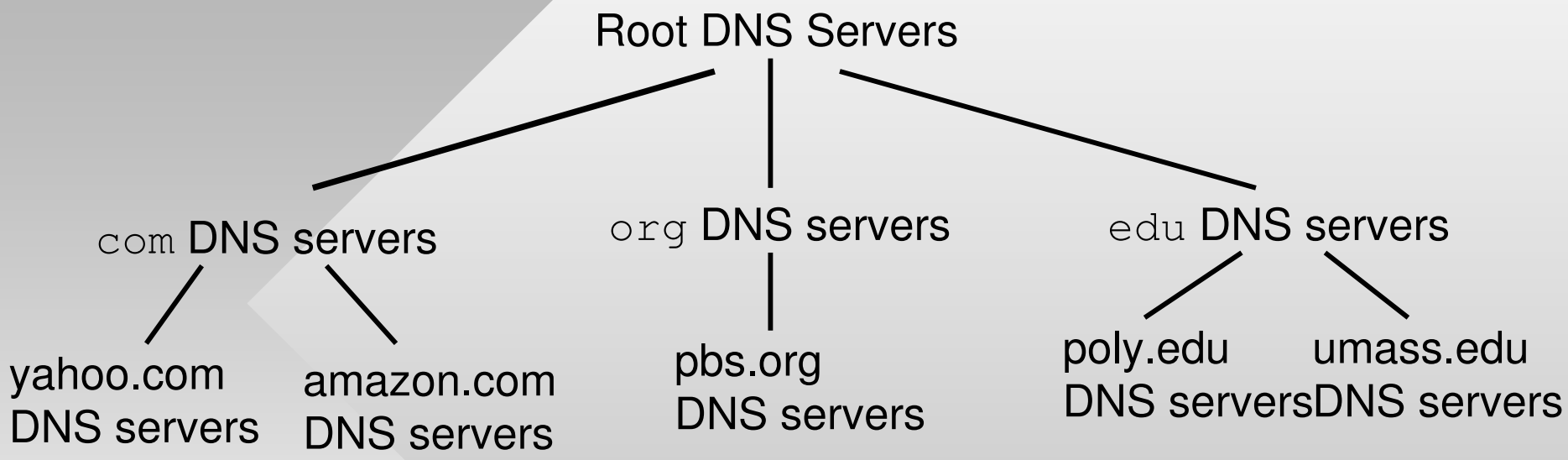
- Forward: hostname to IP address translation; reverse: IP to hostname
- Host aliasing
- Mail server lookup
- Load distribution
  - Replicated Web servers: set of IP addresses for one DNS name

Why not centralize DNS?

- Single point of failure
- Traffic volume
- Maintenance is a nightmare

Doesn't *scale!*

Distributed, Hierarchical Database

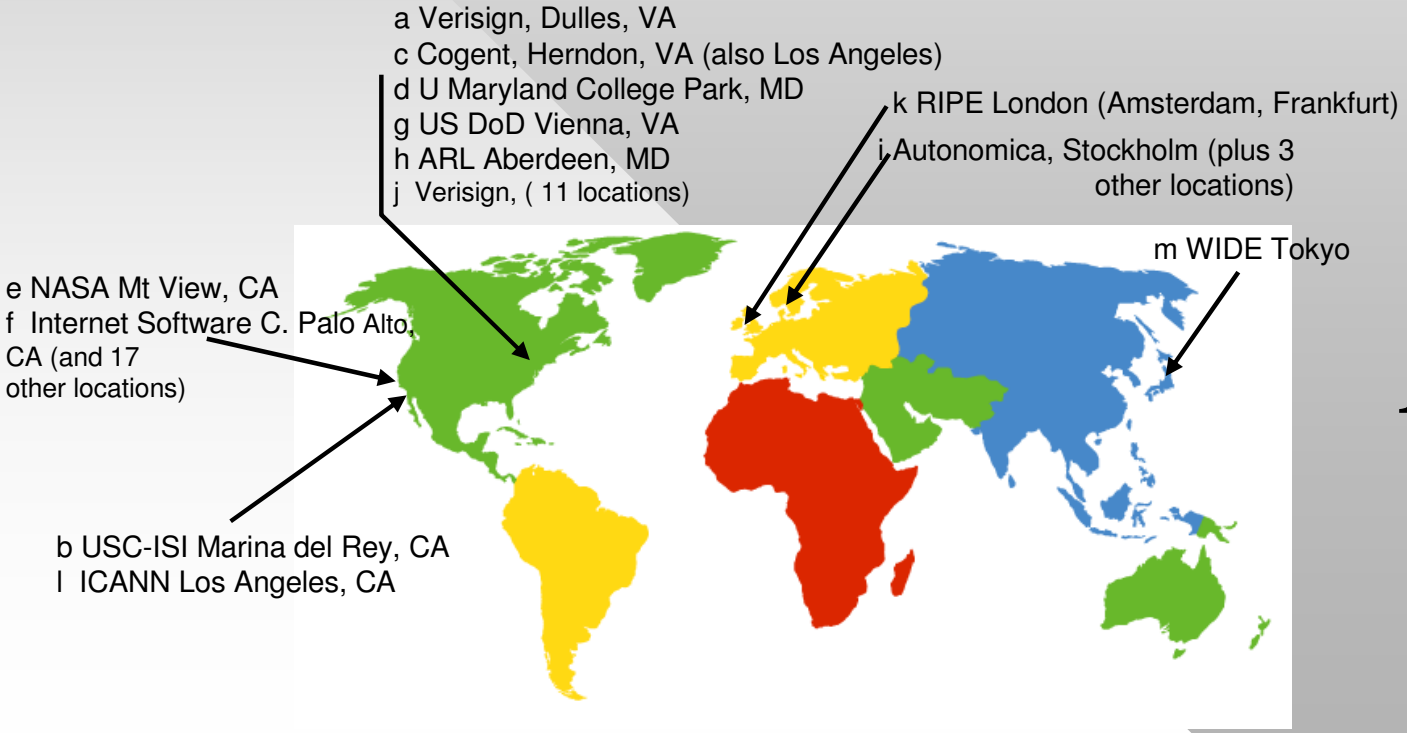


Client wants IP for www.amazon.com:

- Client queries a root server to find the `com` DNS server
- Client queries the `com` DNS server to get the `amazon.com` DNS server
- Client queries the `amazon.com` DNS server to get IP address for `www.amazon.com`

DNS: Root Name Servers

- Contacted by local name server that cannot resolve name
- Root name servers
  - Replicated server farm
  - Set of hardwired IP addresses
  - More info: http://root-servers.org/



13 root name servers worldwide

TLD and Authoritative Servers

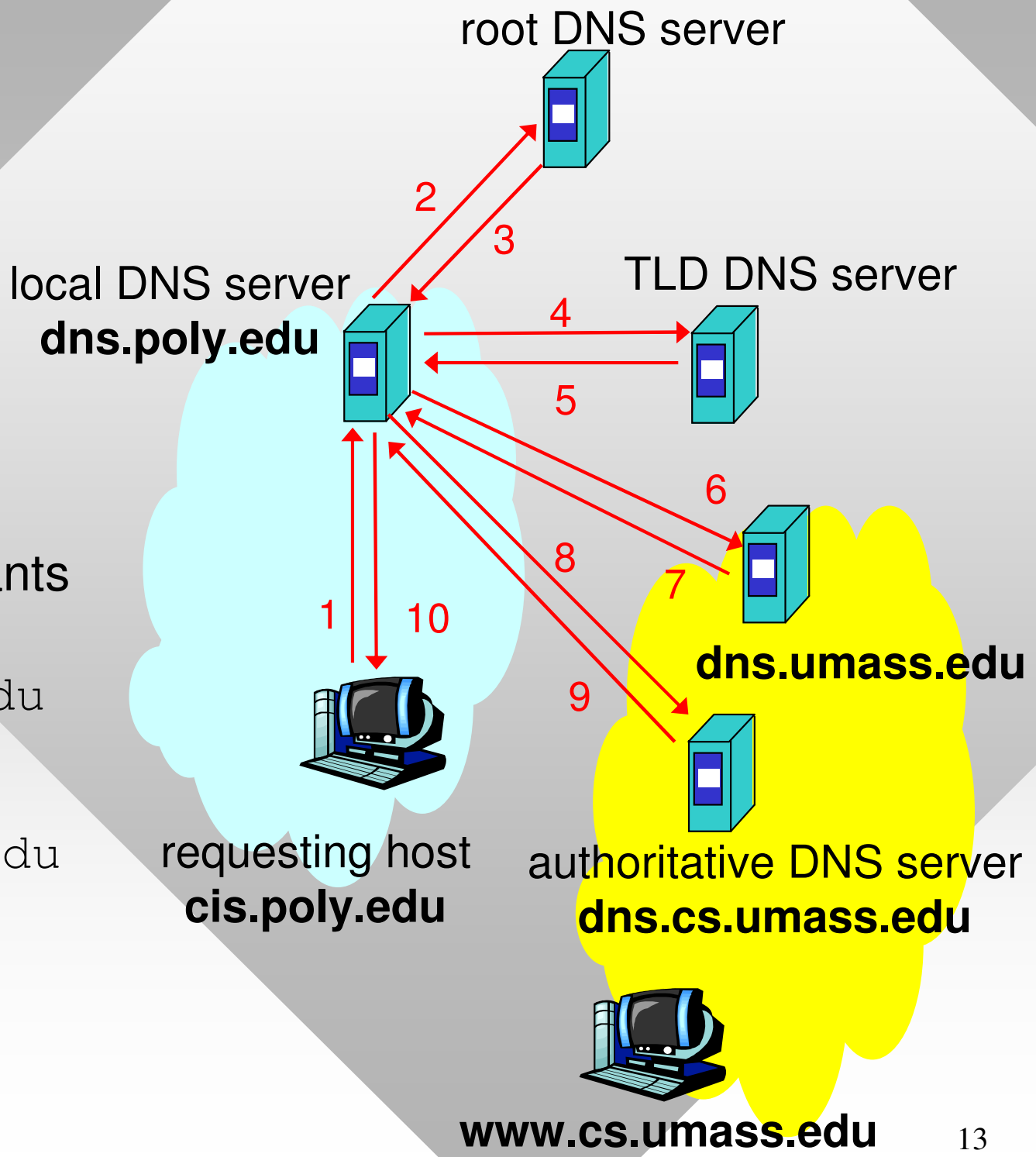
- Top-level domain (TLD) servers: responsible for com, org, net, edu, etc, and all top-level country domains uk, fr, ca, jp.
  - Network Solutions maintains servers for the `com` TLD
  - Educause maintains the `edu` TLD
- *Authoritative* DNS servers: organization's DNS servers, providing authoritative hostname to IP mappings for organization's servers (e.g., Web and mail)
  - Can be maintained by organization or service provider

Local Name Server

- Does not strictly belong to the hierarchy
  - Any computer that accepts requests and then finds out the answer by traversing the DNS tree
- Each network (ISP, company, university) has a few
  - Also called “primary DNS server” in network options (“secondary” is used for backup in case primary fails)
  - Your own computer may be one
- When a host makes a DNS query (gethostbyname), query is sent to its local DNS server
  - Local server acts as a proxy (cache) and forwards query into hierarchy if it cannot answer it from cache

Example

- `cis.poly.edu` wants the IP address for `www.cs.umass.edu`
- Next request for `joe.cs.umass.edu` – what happens?



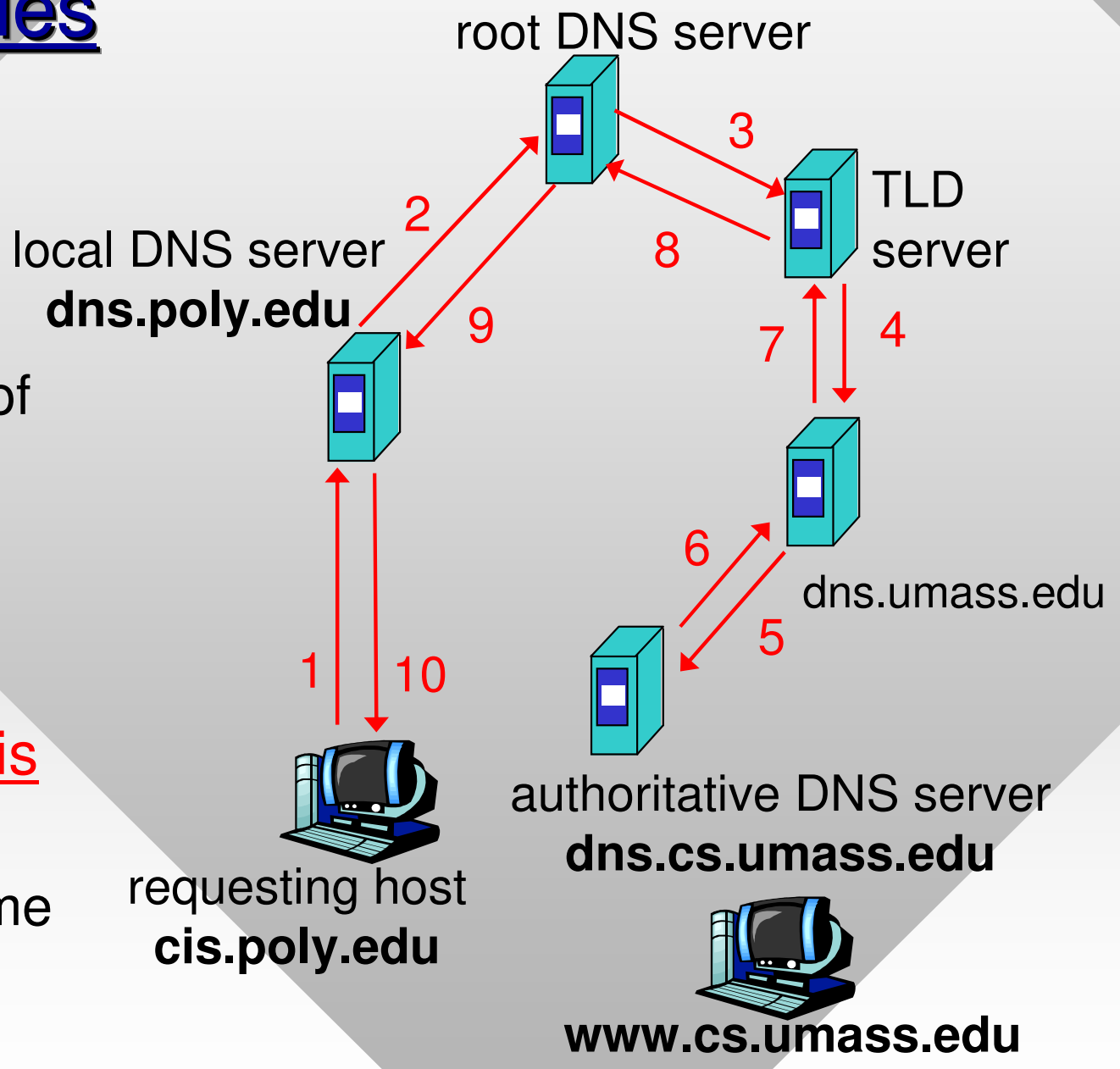
Recursive queries

Iterated query (previous page):

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

Recursive query (this page):

- Puts burden of name resolution on contacted name server



DNS: Caching and Updating Records

- Once (any) name server learns a mapping, it *caches* the mapping
  - Cache entries timeout (disappear) after some time (TTL)
  - Unexpired entries are served directly from cache, in which case they are called non-authoritative
  - If the original DNS server is contacted, the response is authoritative
  - TLD servers are typically cached in local name servers
  - Thus root name servers not visited often

DNS Records

DNS: distributed database of 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 "cannonical" (real) name `www.cs.tamu.edu` is really `web1.cs.tamu.edu`
  - **`value`** is cannonical name
- Type = MX
  - **`value`** is name of mailserver associated with domain **`name`**

DNS Records (Unix)

primary DNS server

root@mydomain.com

```
; zone file for mydomain.com
@       IN SOA          ns1.mydomain.com. root.mydomain.com. (
        2003080800      ; se = serial number
        3h              ; ref = refresh to secondary DNS
        15m             ; ret = retry zone transfer
        3w              ; ex = zone expire on secondary
        3h              ; default TTL )
        IN NS           ns1.mydomain.com.
        IN MX 10        mail.anotherdomain.com.
joe     IN A            192.168.254.3
www     IN CNAME        joe
```

SOA = start of authority; IN = Internet protocol; mail exchanger has priority 10; @ = shortcut for domain name

DNS Records

- Reverse DNS lookups are performed using a special construction of a “fake” DNS name
- The IP address is reversed and is followed by “in-addr.arpa”
  - Example: 128.194.135.65 is requested as 65.135.194.128.in-addr.arpa
  - The query type must be set to PTR
- Use nslookup to query DNS servers
- RFC 1035 describes the basic format of the header