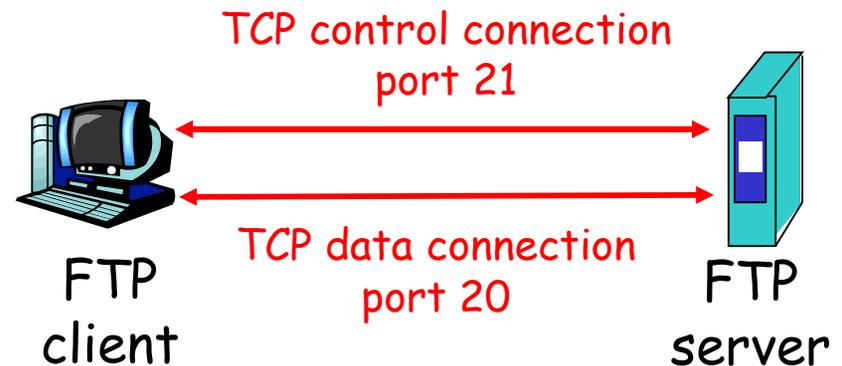


FTP, SMTP and DNS

FTP: separate control, data connections

- ❑ FTP client contacts FTP server at port 21, specifying TCP as transport protocol
- ❑ Client obtains authorization over control connection
- ❑ Client browses remote directory by sending commands over control connection.
- ❑ When server receives a command for a file transfer, the server opens a TCP data connection to client
- ❑ After transferring one file, server closes connection.



- ❑ Server opens a second TCP data connection to transfer another file.
- ❑ Control connection: "out of band"
- ❑ FTP server maintains "state": current directory, earlier authentication

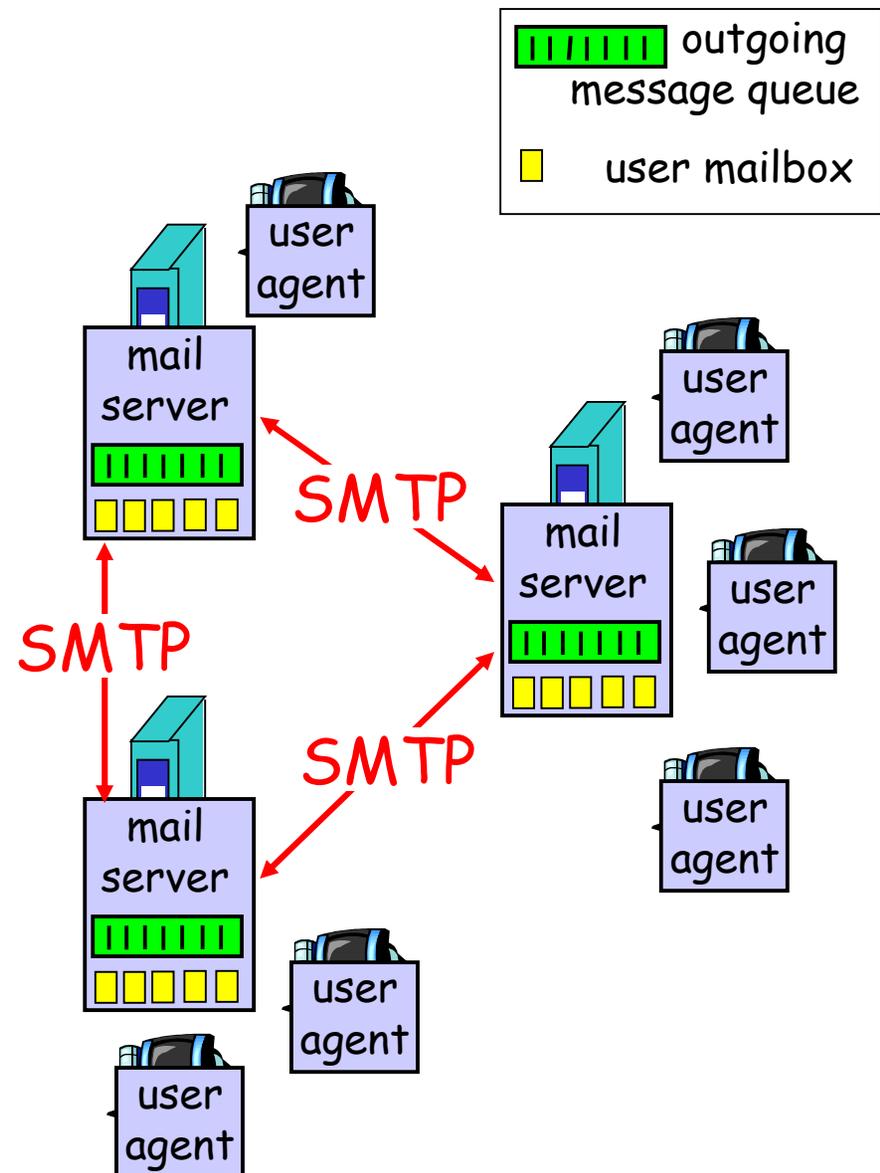
Electronic Mail

Three major components:

- ❑ user agents
- ❑ mail servers
- ❑ simple mail transfer protocol: SMTP

User Agent

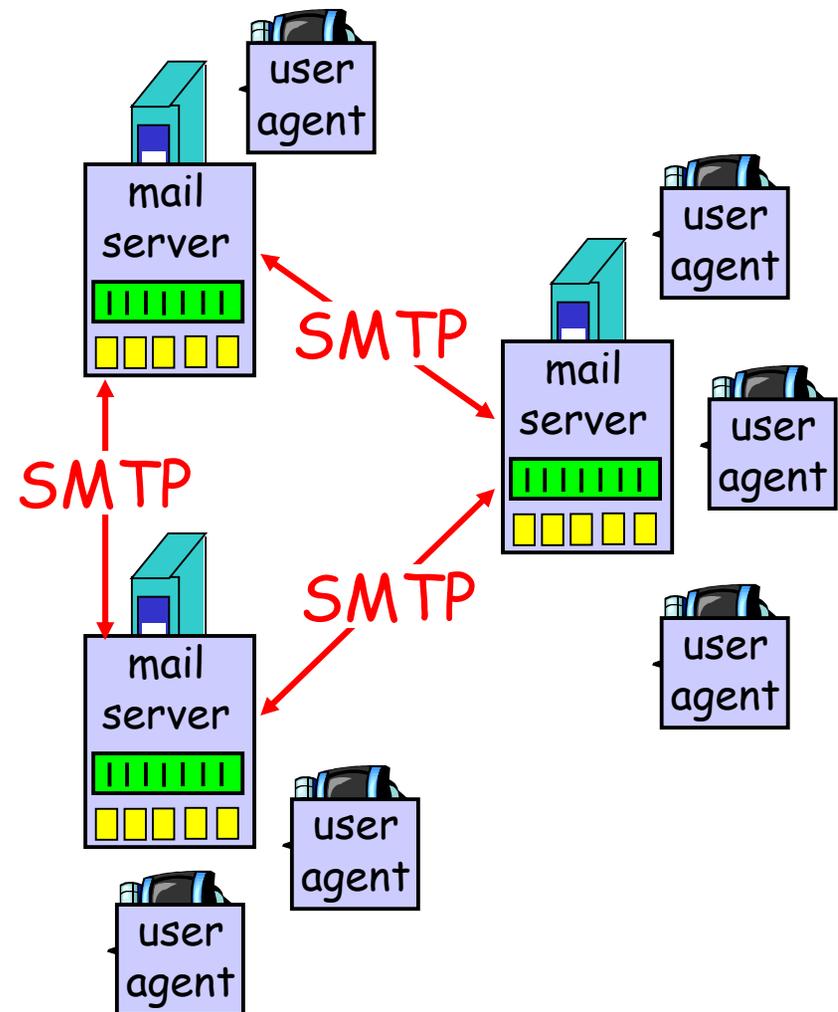
- ❑ a.k.a. "mail reader"
- ❑ composing, editing, reading mail messages
- ❑ e.g., Eudora, Outlook, elm, Netscape Messenger
- ❑ outgoing, incoming messages stored on server



Electronic Mail: mail servers

Mail Servers

- **mailbox** contains incoming messages for user
- **message queue** of outgoing (to be sent) mail messages
- **SMTP protocol** between mail servers to send email messages
 - client: sending mail server
 - "server": receiving mail server

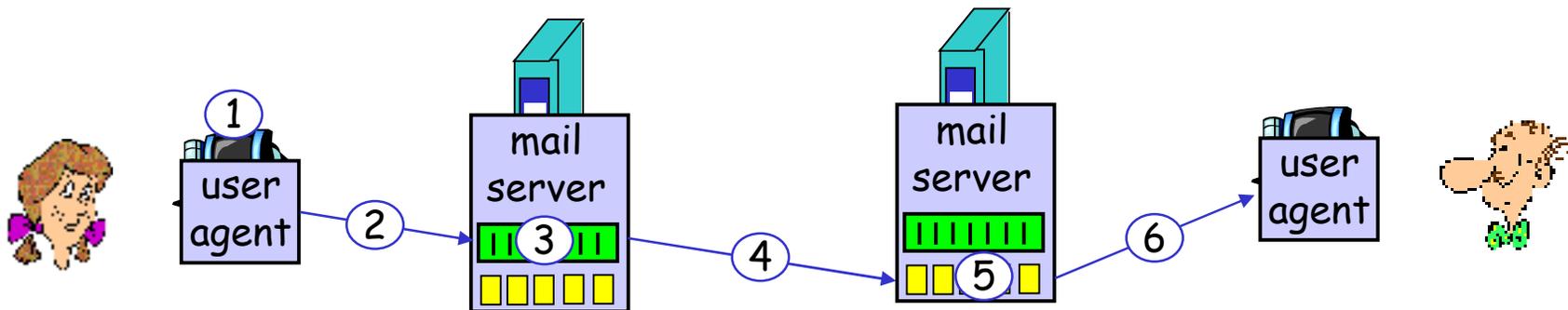


Electronic Mail: SMTP [RFC 2821]

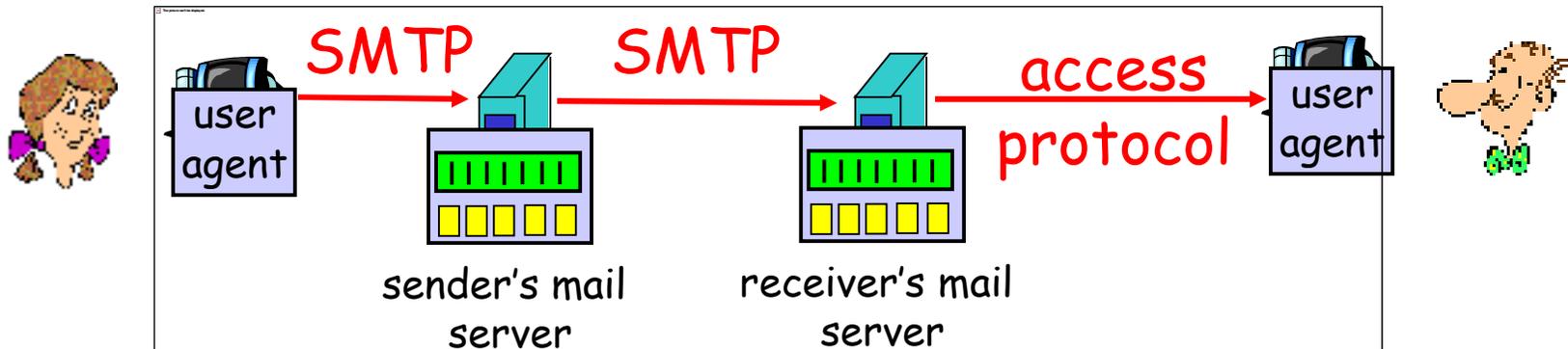
- ❑ uses TCP to reliably transfer email message from client to server, port 25
- ❑ direct transfer: sending server to receiving server
- ❑ three phases of transfer
 - handshaking (greeting)
 - transfer of messages
 - closure
- ❑ command/response interaction
 - **commands:** ASCII text
 - **response:** status code and phrase
- ❑ messages must be in 7-bit ASCII

Scenario: Alice sends message to Bob

- 1) Alice uses UA to compose message and "to" `bob@someschool.edu`
- 2) Alice's UA sends message to her mail server; message placed in message queue
- 3) Client side of SMTP opens TCP connection with Bob's mail server
- 4) SMTP client sends Alice's message over the TCP connection
- 5) Bob's mail server places the message in Bob's mailbox
- 6) Bob invokes his user agent to read message



Mail access protocols



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

DNS: Domain Name System

People: many identifiers:

- SSN, name, passport #

Internet hosts, routers:

- IP address (32 bit) - used for addressing datagrams
- "name", e.g.,
ww.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*
host, routers, name servers to communicate to *resolve* names (address/name translation)
 - note: core Internet function, implemented as application-layer protocol
 - complexity at network's "edge"

DNS and Applications

- Which applications use DNS?

- HTTP
 - Browser extracts hostname
 - Sends hostname to DNS
 - DNS does lookup and returns IP address
 - Browser sends HTTP GET to IP address

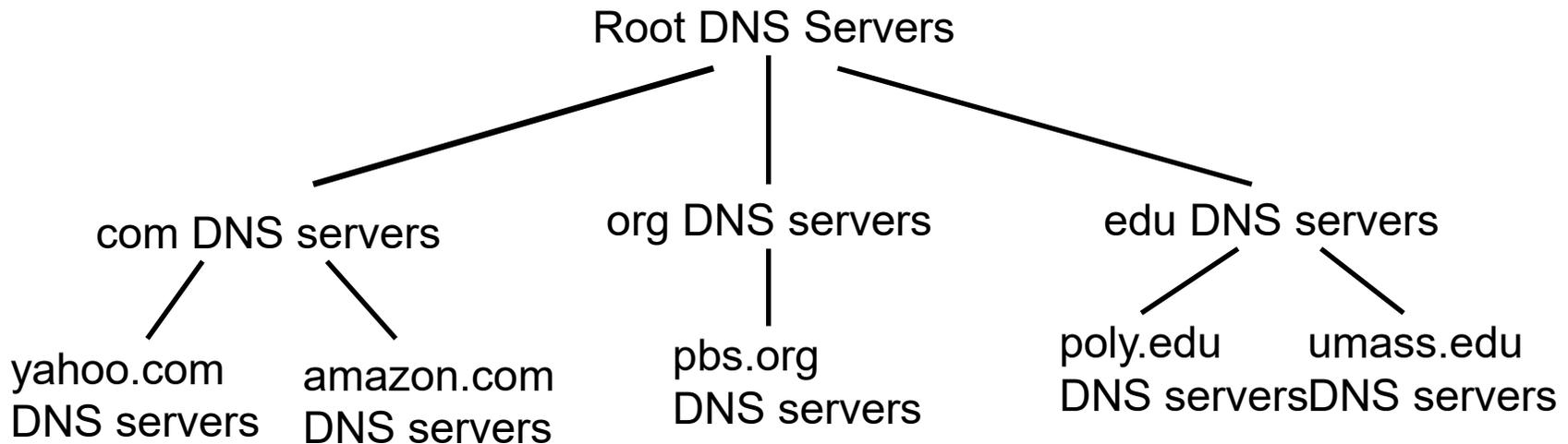
DNS

Why not centralize DNS?

- ❑ single point of failure
- ❑ traffic volume
- ❑ distant centralized database
- ❑ maintenance

doesn't scale!

Distributed, Hierarchical Database

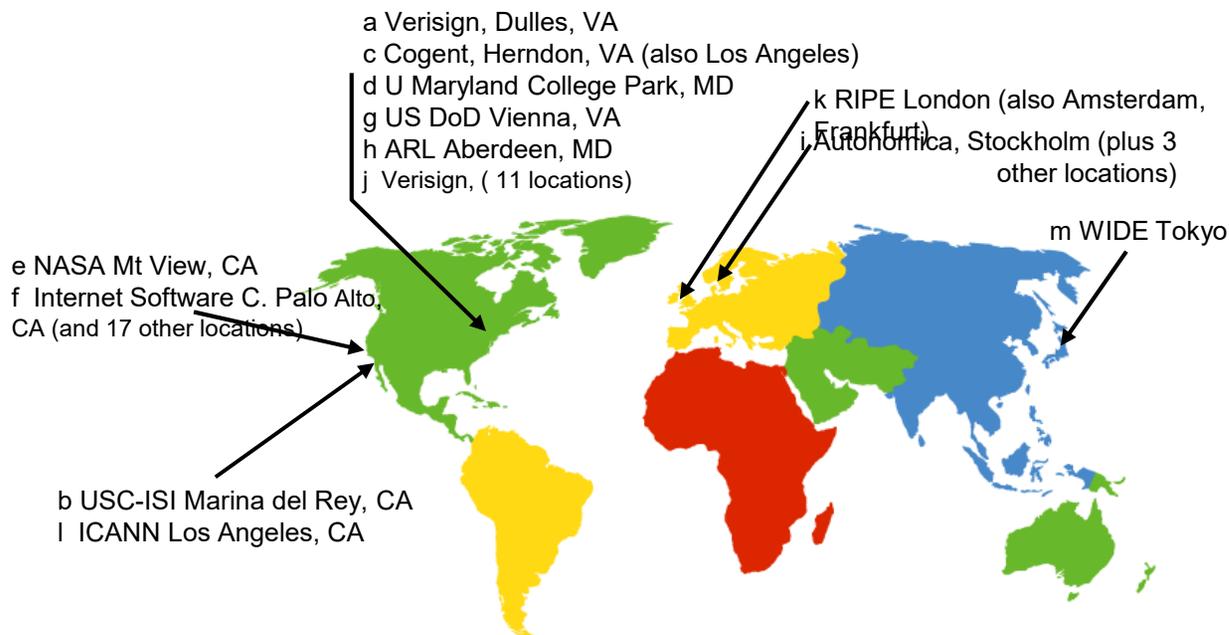


Client wants IP for www.amazon.com; 1st approx:

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

DNS: Root name servers

- ❑ contacted by local name server that can not resolve name
- ❑ root name server:
 - contacts authoritative name server if name mapping not known
 - gets mapping
 - returns mapping to local name server



13 root name servers worldwide

TLD and Authoritative Servers

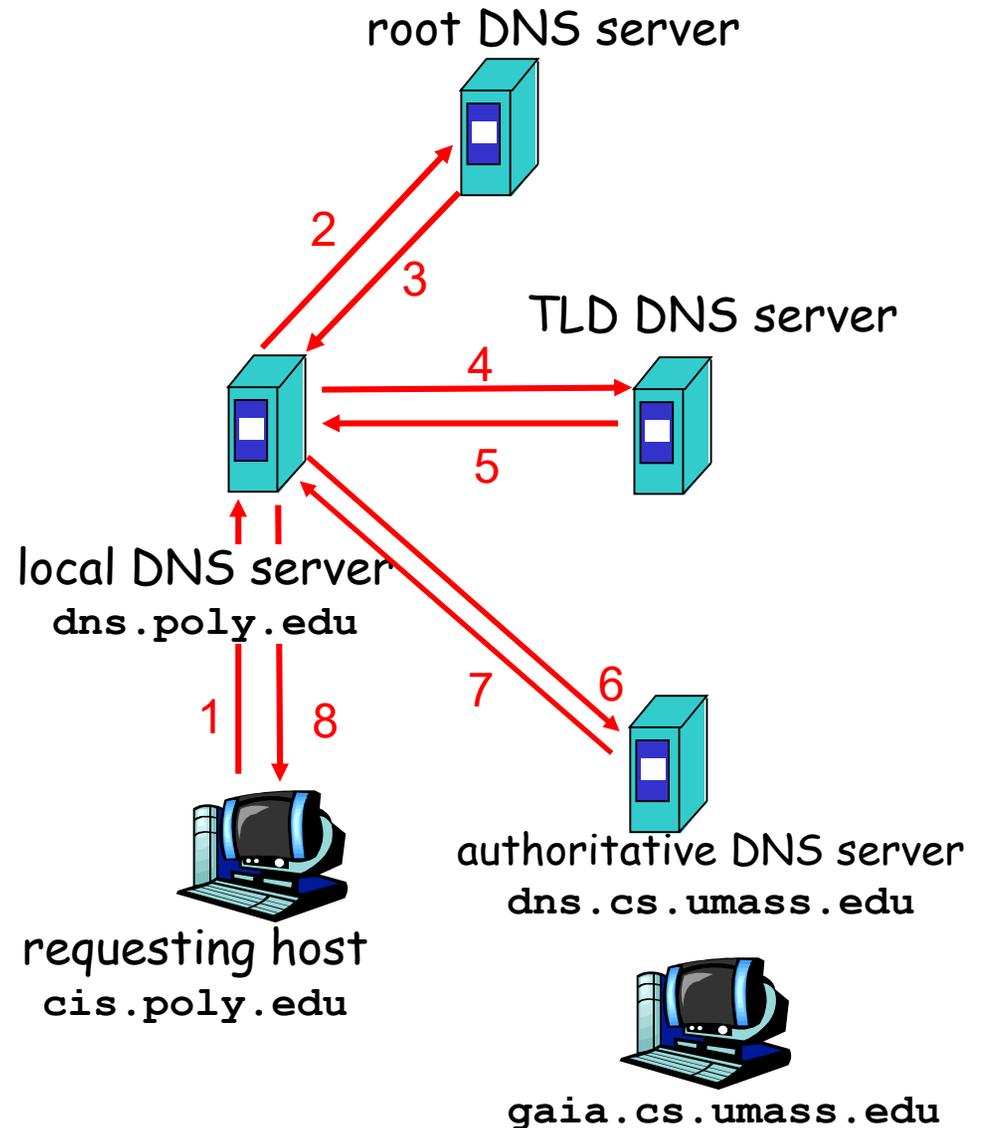
- ❑ **Top-level domain (TLD) servers:** responsible for com, org, net, edu, etc, and all top-level country domains uk, fr, ca, jp.
 - Network solutions maintains servers for com TLD
 - Educause for edu TLD
- ❑ **Authoritative DNS servers:** organization's DNS servers, providing authoritative hostname 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 hierarchy
- ❑ Each ISP (residential ISP, company, university) has one.
 - Also called "default name server"
- ❑ When a host makes a DNS query, query is sent to its local DNS server
 - Acts as a proxy, forwards query into hierarchy.

Example

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



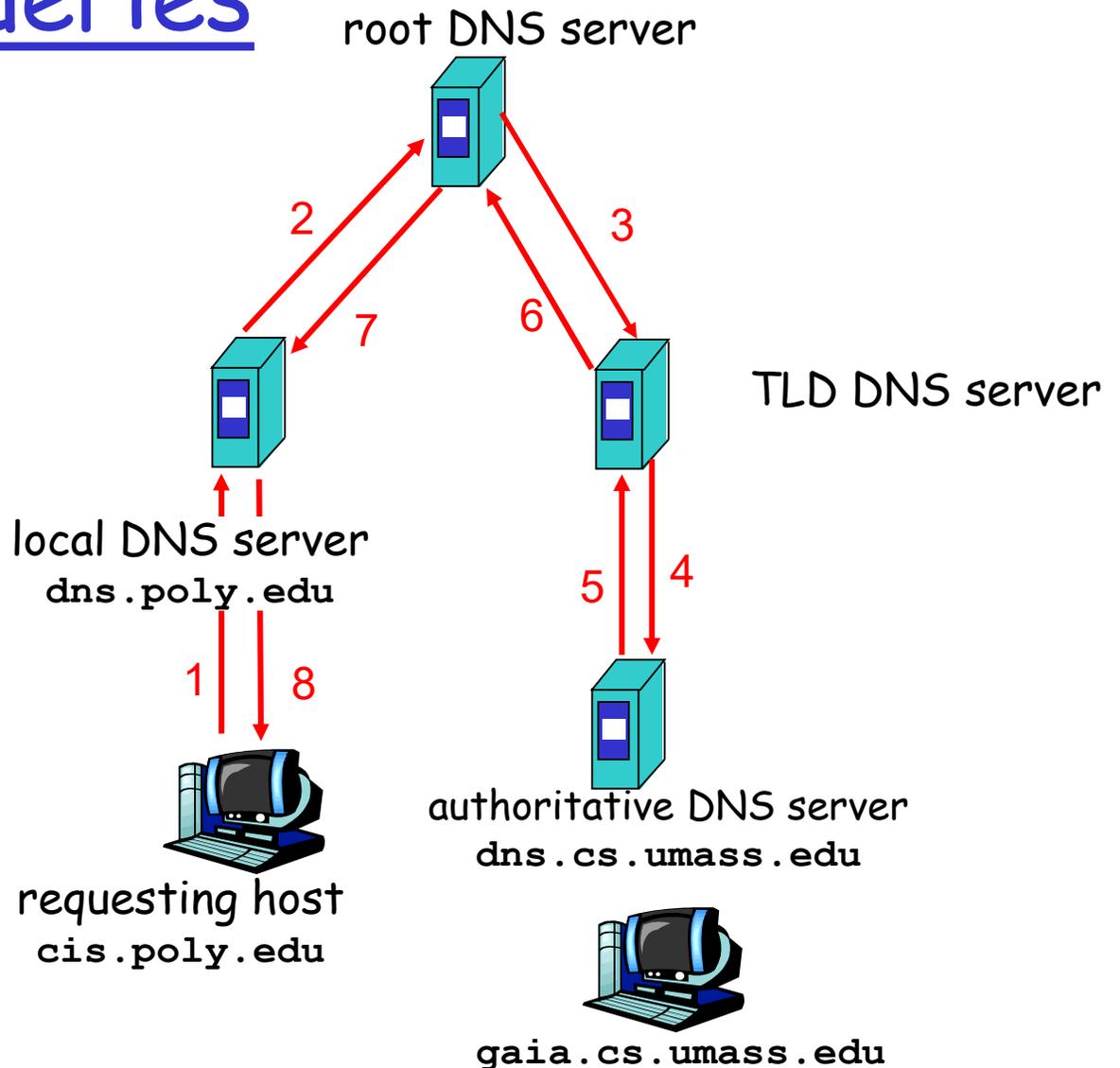
Recursive queries

recursive query:

- ❑ puts burden of name resolution on contacted name server
- ❑ heavy load?

iterated query:

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



DNS: caching and updating records

- ❑ once (any) name server learns mapping, it *caches* mapping
 - cache entries timeout (disappear) after some time
 - TLD servers typically cached in local name servers
 - Thus root name servers not often visited
- ❑ update/notify mechanisms under design by IETF
 - RFC 2136
 - <http://www.ietf.org/html.charters/dnsind-charter.html>

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 IP address 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