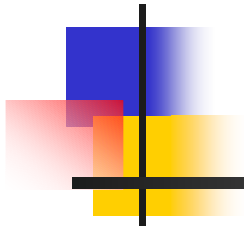
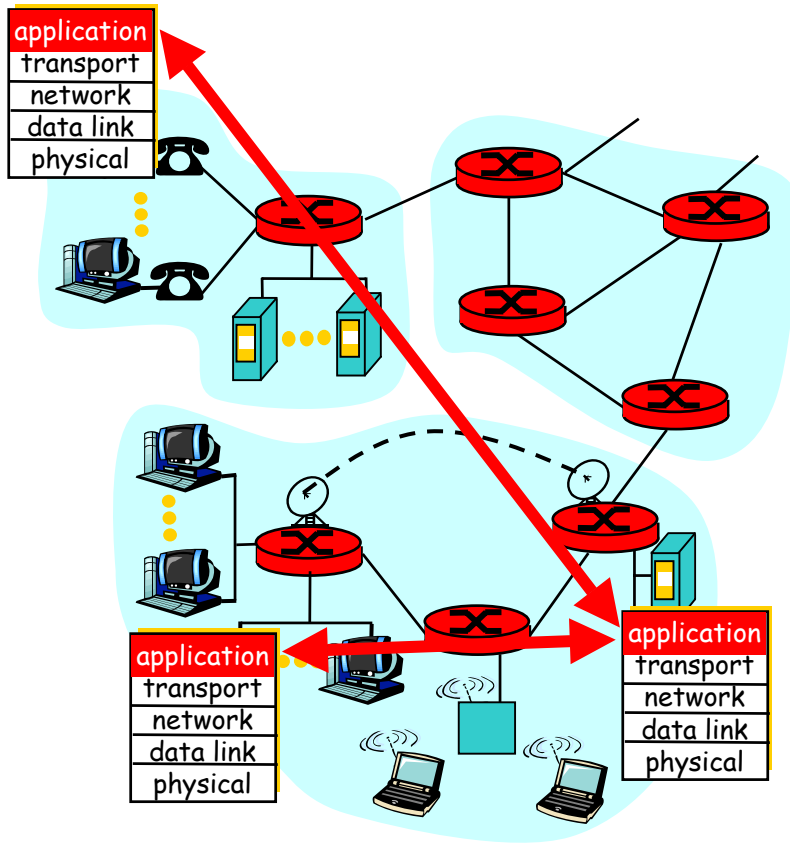


Application-layer Protocols



Dept. of Computer Science, University of Rochester

Network Applications and Application-layer Protocols



Network applications:

- running in end systems (hosts)
- distributed, communicating using network
- use communication services provided by lower layer protocols (TCP, UDP)
- e.g., e-mail, Web, FTP, instant messaging

Application-layer protocols

- define interface between application entities



Protocols vs. Implementations

- They specify:
 - Syntax and semantics of messages exchanged
 - Rules for when and how processes send & respond to messages
- They **don't** specify the implementation:
 - Programming, data structure, ...
- The goal:
 - Components implemented independently can inter-operate with each other as long as they follow the protocol specification

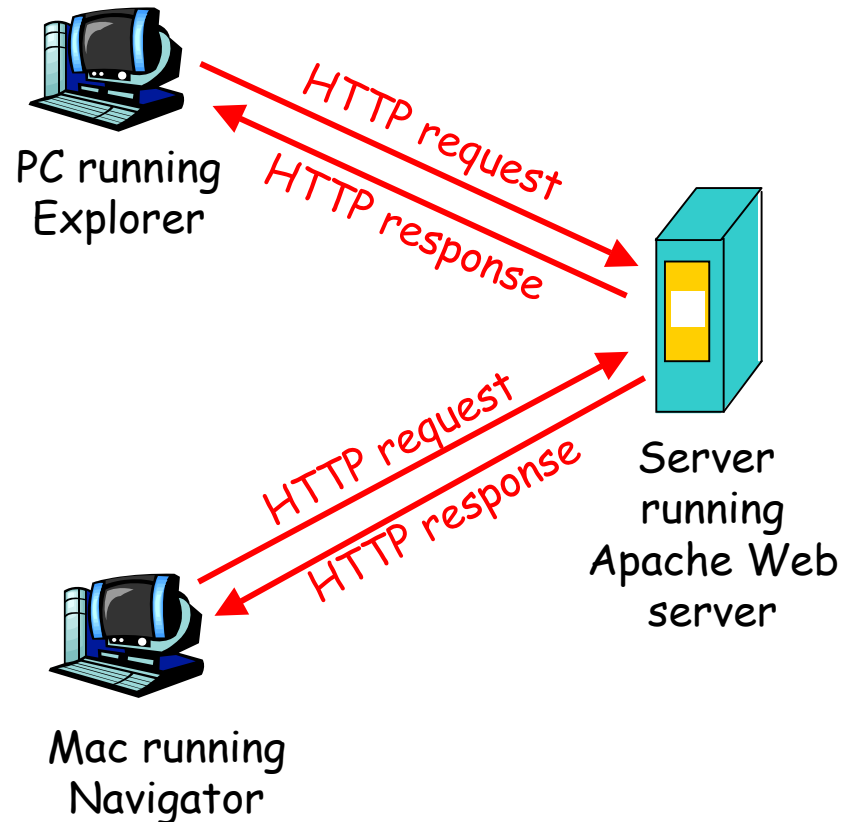


Network Applications

- Some commonplace applications
 - Web and HTTP
 - FTP, telnet, SSH
 - Electronic mail: SMTP, POP3, IMAP
 - DNS
 - Distributed file sharing
 - Multicast on overlay networks
 -

HTTP and Web

- **HTTP**: hypertext transfer protocol
- **Web**: the application using HTTP protocol
- Client/server model
 - client: browser that requests, receives, "displays" Web objects
 - server: Web server sends objects in response to requests





Performance of HTTP

Persistence

Non-persistent HTTP (1.0)

- At most one object is sent over a TCP connection.

Persistent HTTP (1.1)

- Multiple objects can be sent over single TCP connection between the browser and web server.

TCP performance overhead:

- connection establishment
- congestion control: slow start

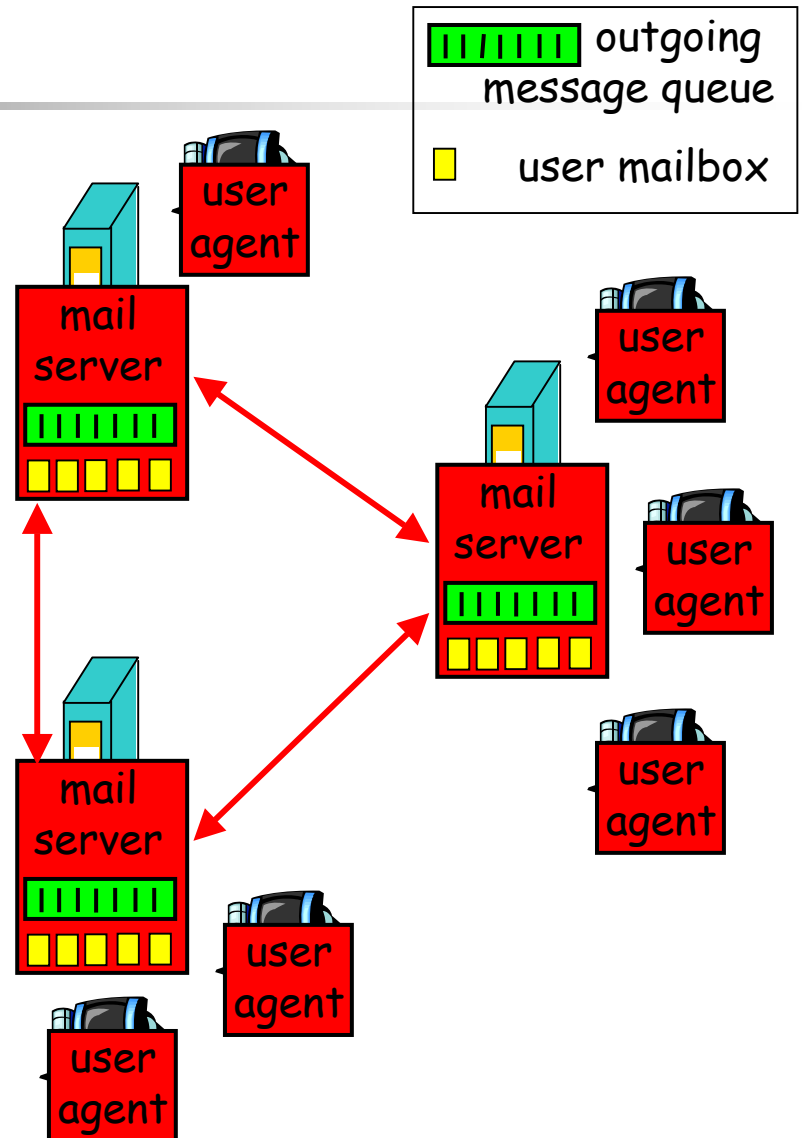
Non-persistent HTTP issues:

- pay these overhead for each object
- but browsers can open parallel TCP connections to fetch objects

Electronic Mail

Two types of entities:

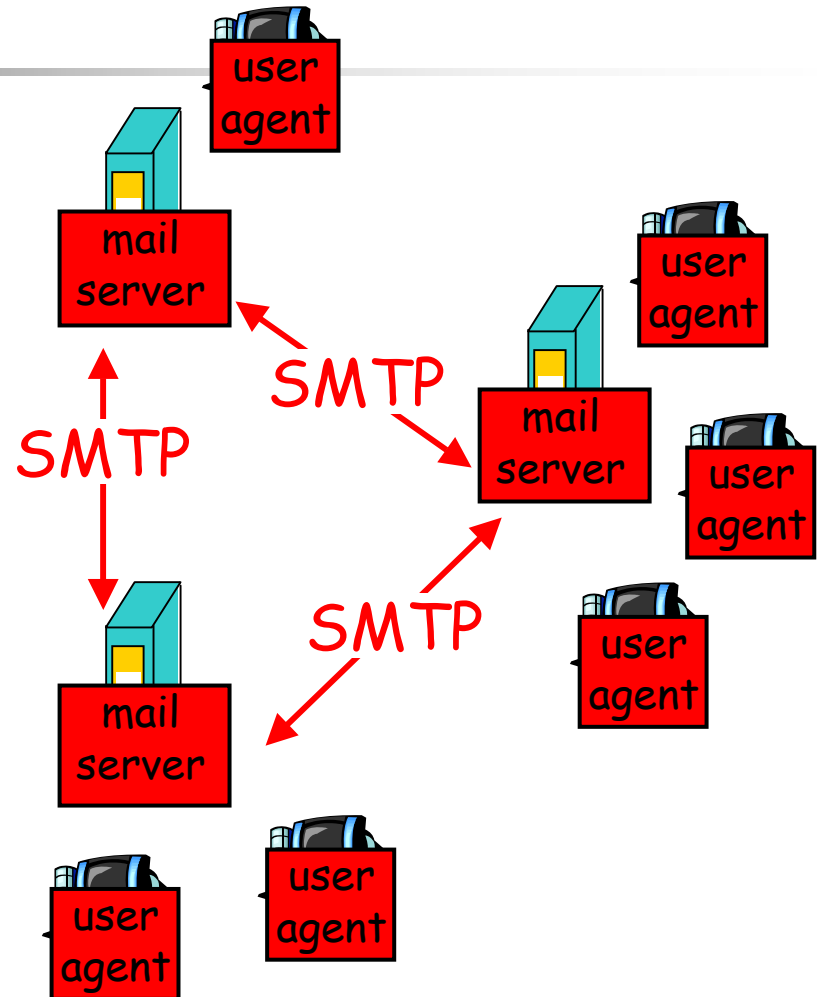
- **Mail servers**
 - **mailbox** contains incoming messages for users
 - **message queue** of outgoing (to be sent) mail messages
- **User agents**
 - a.k.a. "mail reader"
 - composing, editing, reading mail messages
 - e.g., pine, Outlook, Mozilla Thunderbird



Electronic Mail

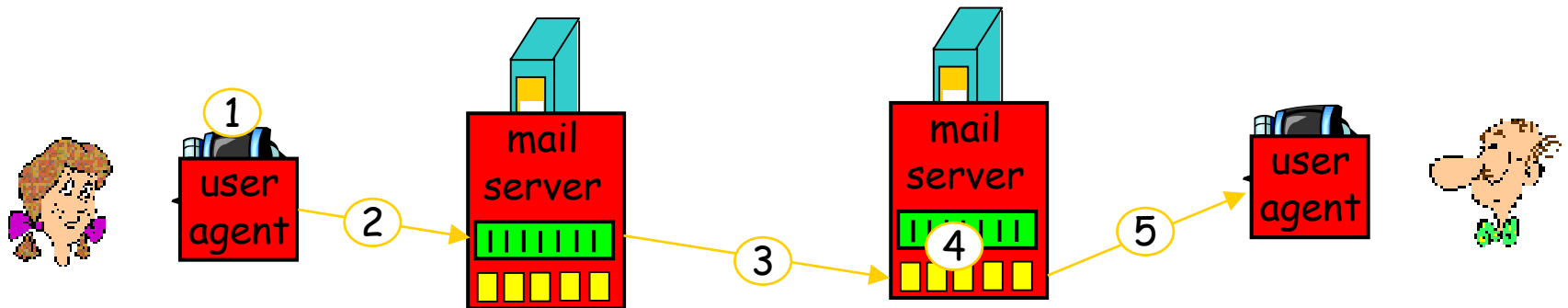
Two types of protocols:

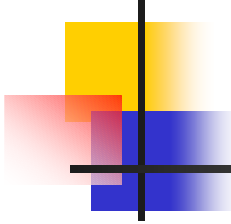
- **mail transfer protocol**
 - from sender agent to the receiver's mail server
 - SMTP (simple mail transfer protocol)
- **mail access protocol**
 - the receiver pulls mails from server to agent
 - POP3, IMAP, HTTP



Alice sends message to Bob

- 1) Alice uses UA to compose message with Bob's address
- 2) Alice's UA sends message to her mail server through SMTP; message placed in message queue
- 3) Alice's mail server sends message to Bob's mail server through SMTP
- 4) Bob's mail server places the message in Bob's mailbox
- 5) Bob invokes his user agent to read message





SMTP Interaction between mail servers at cs.rochester.edu (client) and hotmail.com (server)

```
S: 220 hotmail.com Don't send me any unsolicited bulk e-mail.
C: HELO cs.rochester.edu
S: 250 Hello cs.rochester.edu, pleased to meet you
C: MAIL FROM:<alice@cs.rochester.edu>
S: 250 alice@cs.rochester.edu... Sender ok
C: RCPT TO:<bob@hotmail.com>
S: 250 bob@hotmail.com ... Recipient ok
C: DATA
S: 354 Enter mail, end with "." on a line by itself
C: From: alice@cs.rochester.edu
C: Subject: ketchup
C:
C: Do you like ketchup?
C: How about pickles?
C: .
S: 250 Message accepted for delivery
C: QUIT
S: 221 hotmail.com closing connection
```

Try SMTP interaction for yourself

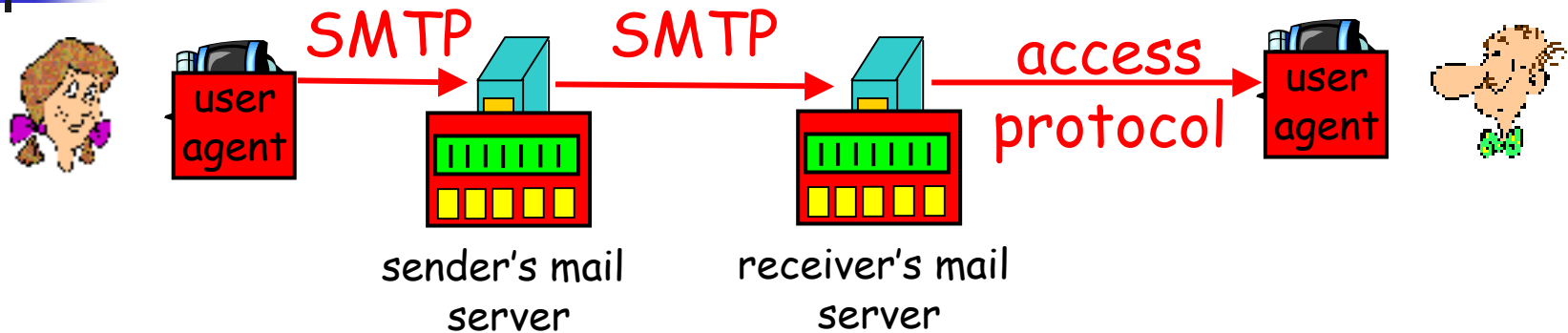


- `telnet servername 25`
- see 220 reply from server
- enter HELO, MAIL FROM, RCPT TO, DATA, QUIT commands

This lets you

- send email without using a normal email client
- ...

Mail Access Protocols



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



DNS: Domain Name System

People: multiple identifiers

- SSN - used for tax reporting, good for bookkeeping, indexing
- name - human friendly, easy to remember

Internet hosts:

- IP address - used for addressing, routing on the Internet
- "name", e.g., tallinn.cs.rochester.edu - human friendly

Q: map between IP addresses and name?

- **DNS query**: find the IP address for a given name
- not part of Internet as a network
- core function for Internet applications
 - "telnet tallinn.cs.rochester.edu" vs. "telnet 128.151.67.155"
 - <http://www.cnn.com> vs. <http://64.236.24.20>



Domain Name System: Overview

- distributed database
 - implemented with collaboration of many name servers distributed all over the network
- DNS queries
 - **name lookup**: find the IP address for a given name
 - others: find mail server for a domain, ...



DNS: Decentralized and Hierarchical

- Decentralized
 - no server has all name-to-IP address mappings
- Why not centralize DNS?
 - single point of failure, performance bottleneck
 - can only be close to few hosts \Rightarrow long latency to many others
- What if massively replicating it?
 - **no full replication**: too much data to fit into any single server
 - **no perfect consistency**: hard to make any changes on the data
- Hierarchical
 - there are different types of name servers

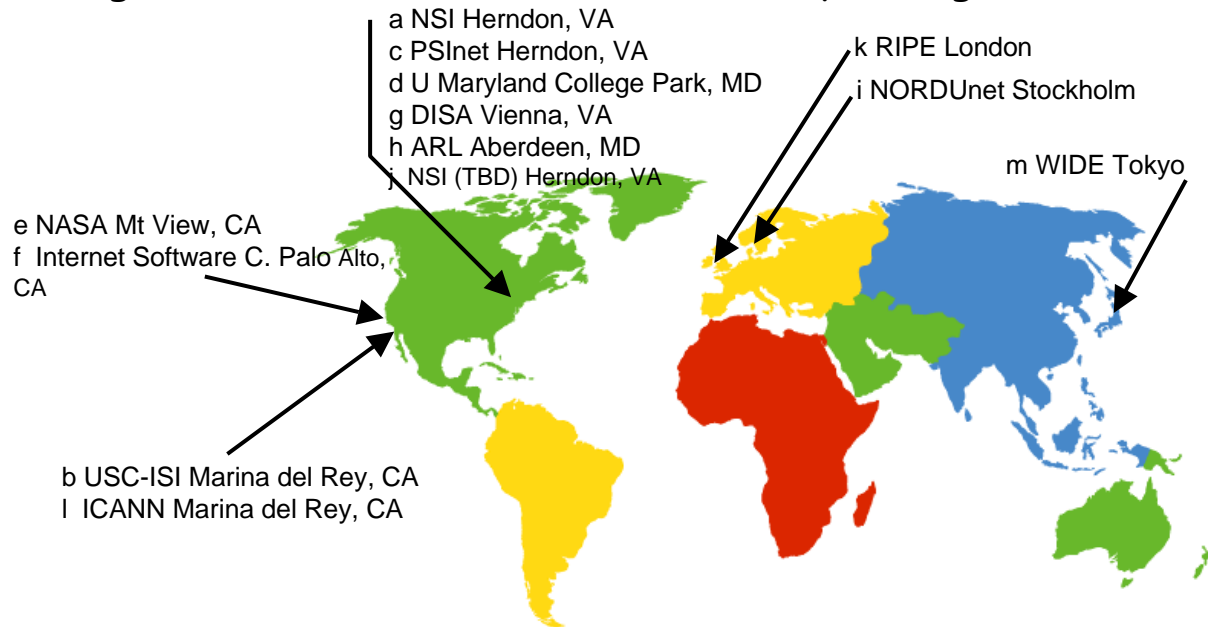


Types of Name Servers

- **local name server** (defined for a particular group of hosts):
 - each organization often has local (default) name server
 - clients often learn their default name server via DHCP
 - used when one of the hosts in its domain initiates a DNS query
 - first stop for a DNS query
- **authoritative name server** (defined for a particular group of hosts):
 - maintain "IP address - name" mapping for a group of hosts so it can answer DNS query for these hosts
 - used when one of the hosts in its domain is the target of a DNS query
 - last stop for a DNS query (if ever reached)
 - often the same as local name server for an organization
- each local NS may not know all authoritative NSs
 - something is missing ...

Root Name Servers

- **root name server** (defined globally)
 - bridging between local NSs and authoritative NSs
 - contacted by local name server that can not resolve name
- what it does:
 - contacts authoritative name server
 - gets result and returns it to the requesting local name server

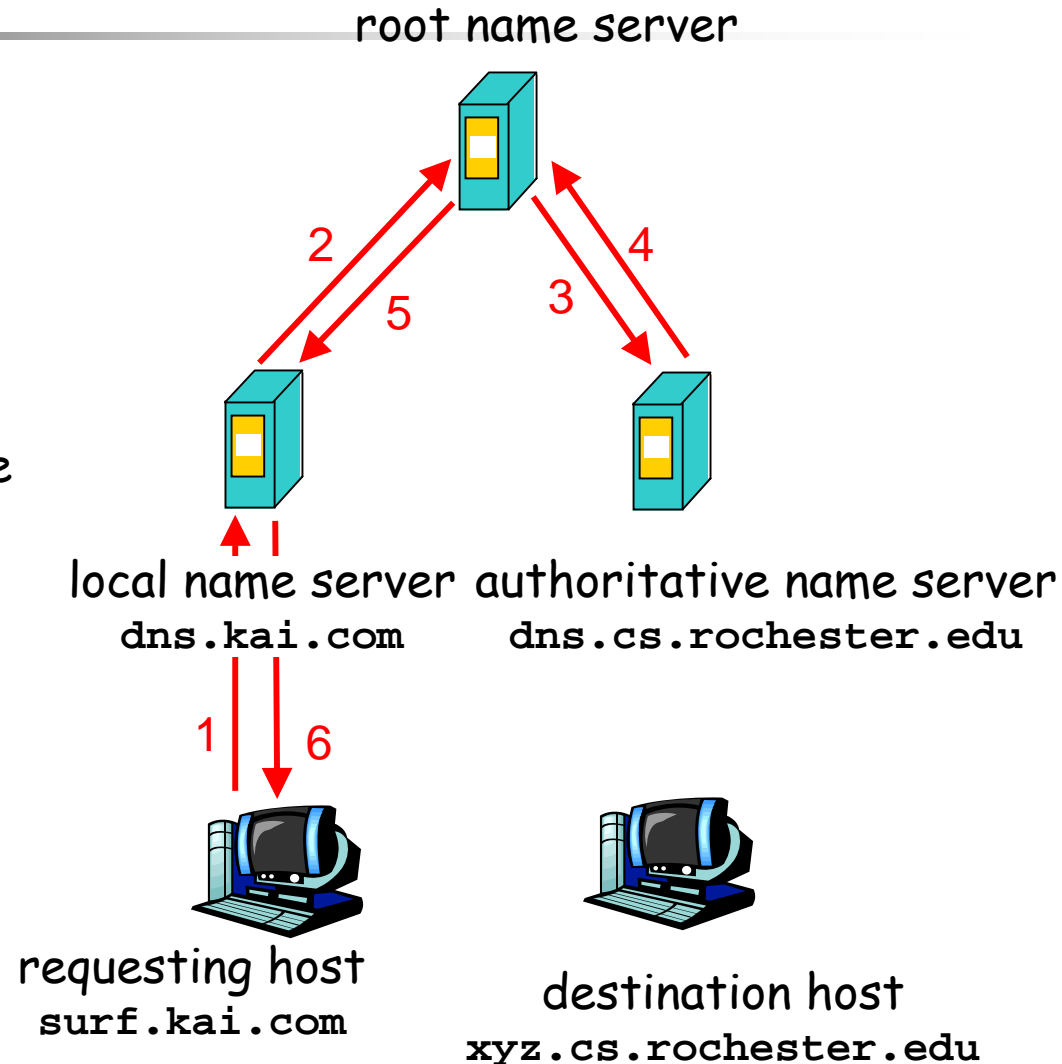


DNS Query in Action

host **surf.kai.com** wants IP address of **xyz.cs.rochester.edu**

1. contacts its local DNS server, **dns.kai.com**
2. **dns.kai.com** contacts root name server
3. root name server contacts authoritative name server, **dns.cs.rochester.edu**

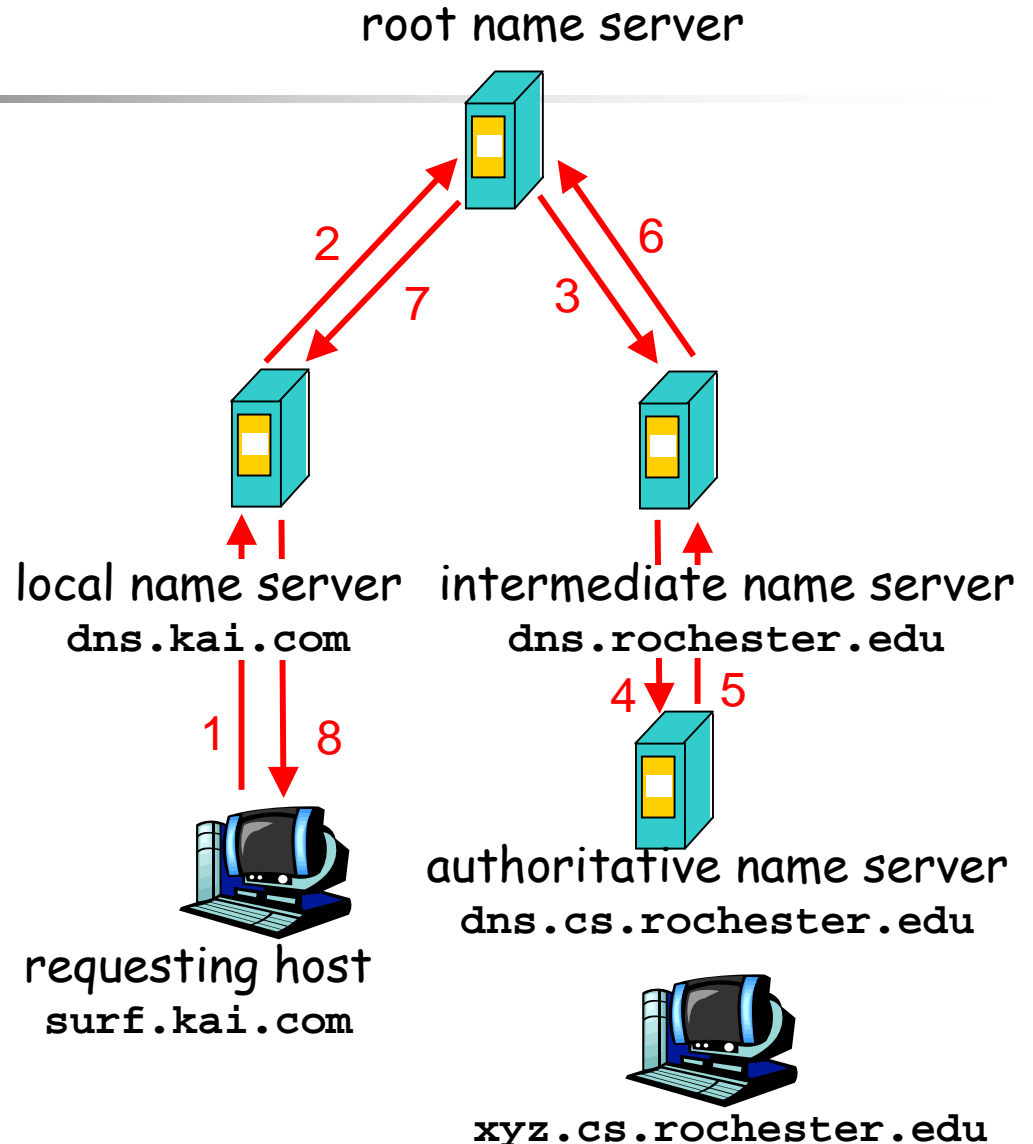
Note that the destination host is not involved in the lookup.



Intermediate Name Servers

Root name server:

- may not know authoritative name server
- may know intermediate name server: who to contact to find authoritative name server



Iterative Queries

root name server

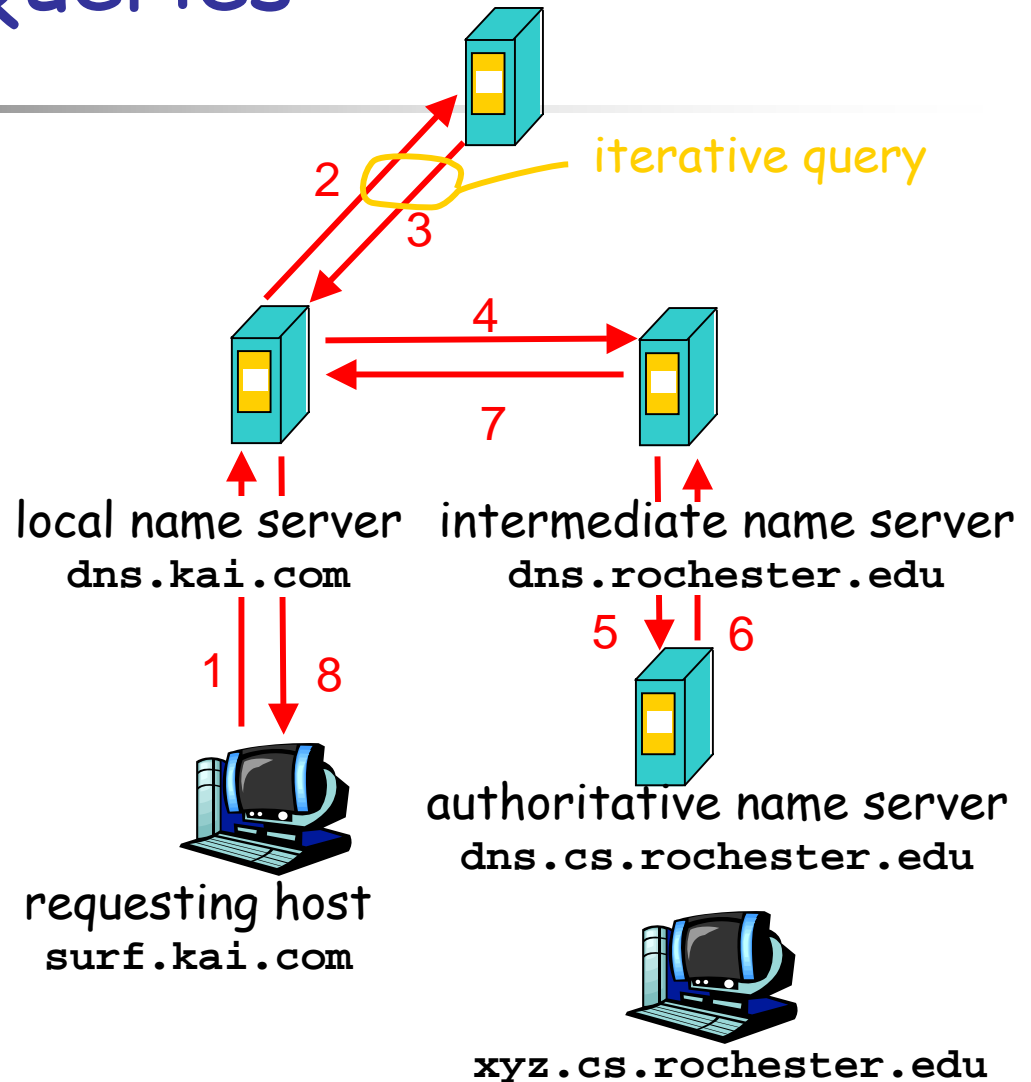
recursive query:

- puts burden of name resolution on contacted name server

iterative query:

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

What is the real difference here?





DNS Caching

- once (any) name server learns mapping, it cached mapping
 - next query for the same name can be answered directly
 - can be at any step of the name lookup (even an end host)
- cache entries timeout (and removed) after some time (e.g., a few minutes to a few days)
 - timeout is necessary because the mapping can change
 - switch Web server to a more powerful machine ⇒ change the mapping between www.cs.rochester.edu to IP address of the new machine
 - load balancing



DNS Scalability & Reliability

- Scalability
 - poor scalability at the few root name servers
 - made more scalable by caching
 - made more scalable by anycast
- Reliability
 - problems at (even a single) root name server(s) can cause big problem
 - again, caching helps improve reliability
- DNS caching is partial, not perfectly consistent replication
 - only feasible for stable name-IP address mapping, may not be suitable for more dynamic global mapping



TCP vs. UDP

- We saw applications:
 - Web/HTTP
 - Email/SMTP/POP/IMAP
 - DNS
- They build on some transport services.
 - TCP or UDP?



Disclaimer

- Parts of the lecture slides contain original work of James Kurose, Larry Peterson, and Keith Ross. The slides are intended for the sole purpose of instruction of computer networks at the University of Rochester. All copyrighted materials belong to their original owner(s).