This lecture will be recorded!!!

Welcome to

CS 3516:
Computer Networks

Prof. Yanhua Li

Time: 9:00am –9:50am M, T, R, and F

Zoom Lecture

Fall 2020 A-term

Updates

- Quiz 3
  - On Monday 9/14
  - 1 bonus question
  - Topics: HTTP basics, cookies, transmission time in persistent vs non-persistent.
- Project 1
  - Due Next Thursday 9/17
- Extra Zoom office hours (will be announced later today)
  - Prof Li and a TA

Chapter 2: outline

2.1 principles of network applications
- app architectures
- app requirements

2.2 Web and HTTP

2.5 DNS

Service Overview, Structure

Resolution process

Data Format

DNS: domain name system

people: many identifiers:

- SSN, name, passport #

Internet hosts, routers:

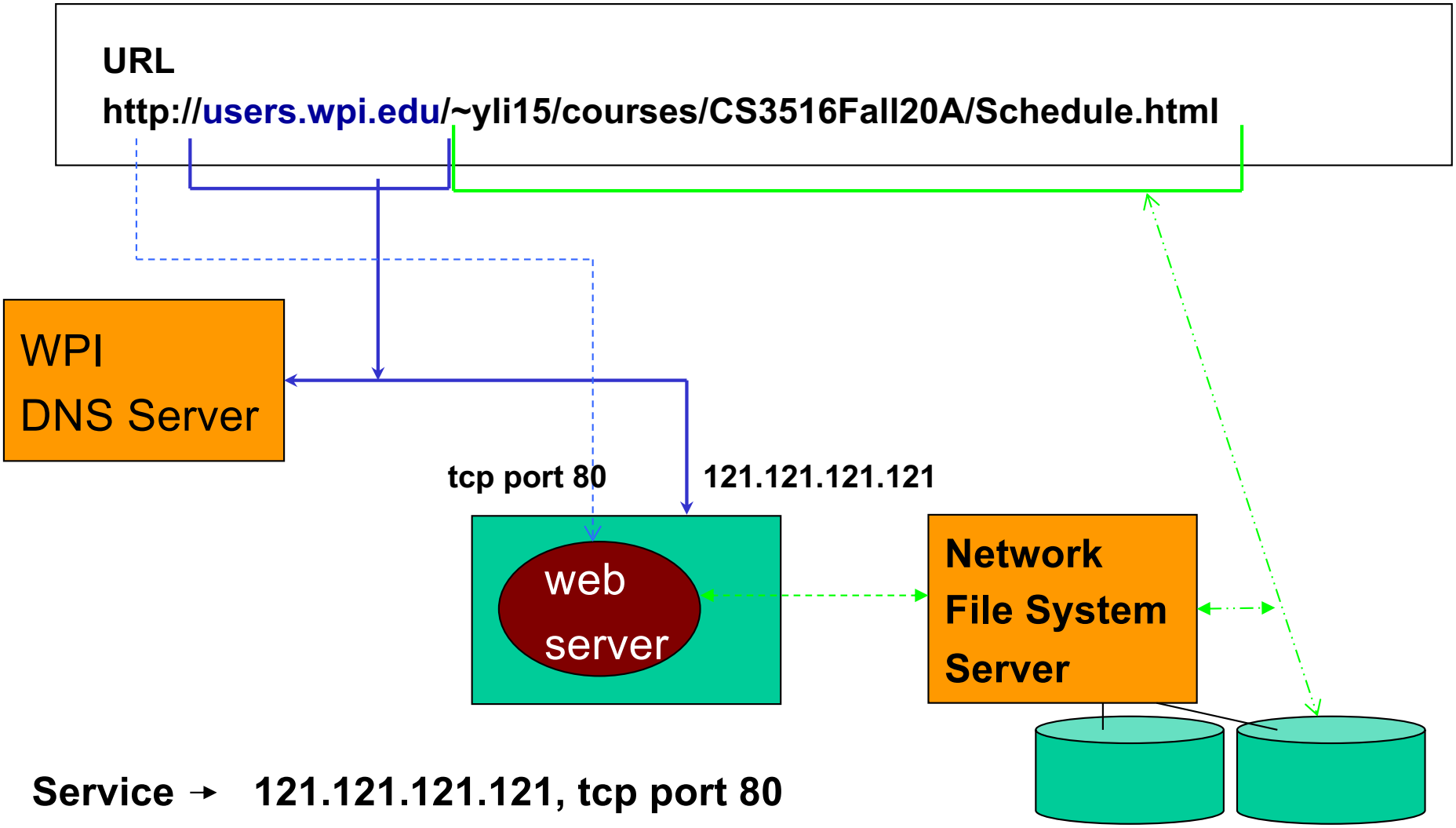
- IP address (32 bit) - used for addressing datagrams
- "name", e.g., www.yahoo.com - used by humans

Q: how to map between IP address and name, and vice versa ?

Domain Name System:

- *distributed database* implemented in hierarchy of many *name servers*
- *application-layer protocol:* hosts, name servers communicate to *resolve* names (address/name translation)
  - note: core Internet function, implemented as application-layer protocol
  - complexity at network's "edge"

Resolving Name, Locating Service/Object



Service → 121.121.121.121, tcp port 80

Object → ~yli15/courses/CS3516Fall20A/Schedule.html

DNS: services, structure

DNS services

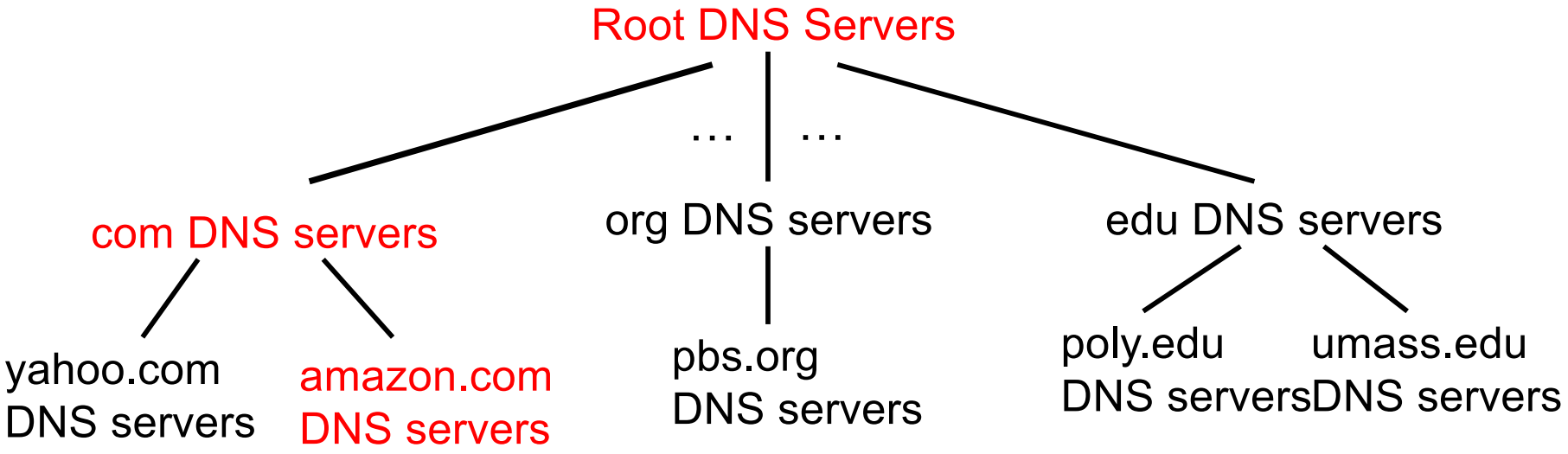
- hostname to IP address translation
- host aliasing
  - canonical, alias names
- mail server aliasing
- load distribution
  - replicated Web servers: many IP addresses correspond to one name

why not centralize DNS?

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

A: doesn't scale!

DNS: a distributed, hierarchical database



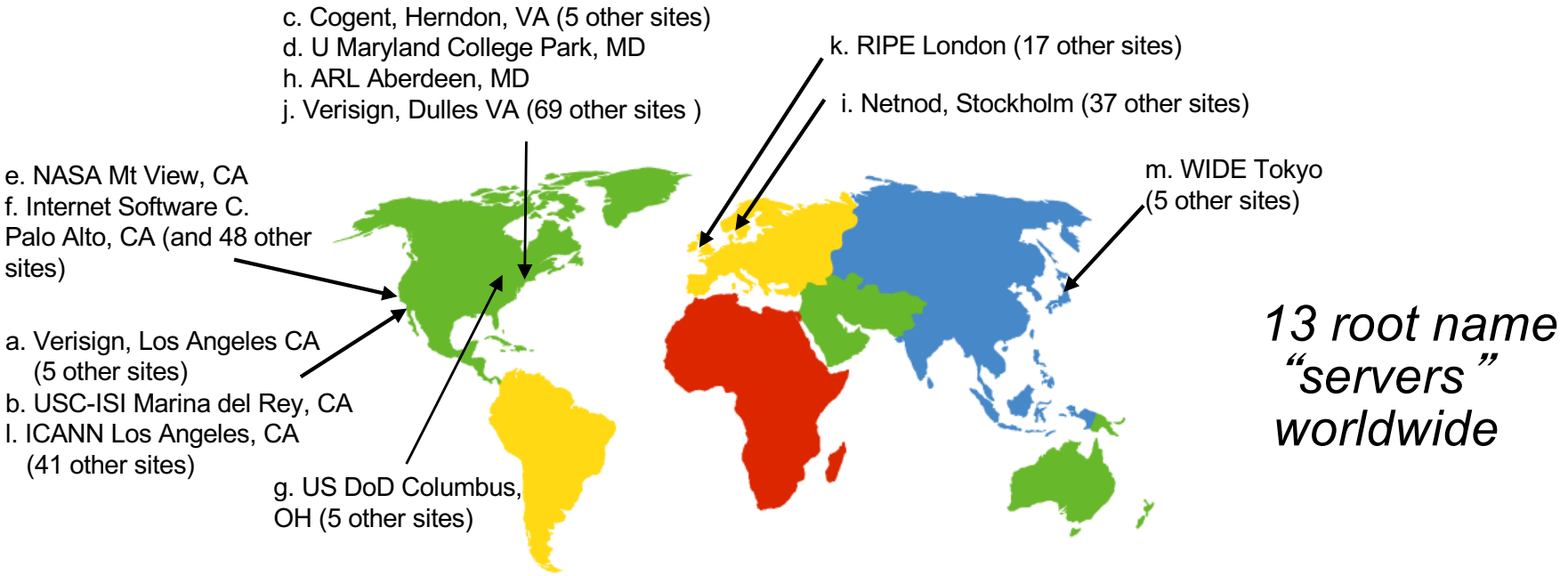
client wants IP for www.amazon.com; 1st approx:

- client queries 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

Analogy: Marshalls -> Physical Address

DNS: root name servers

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



13 root name "servers" worldwide

TLD, authoritative servers

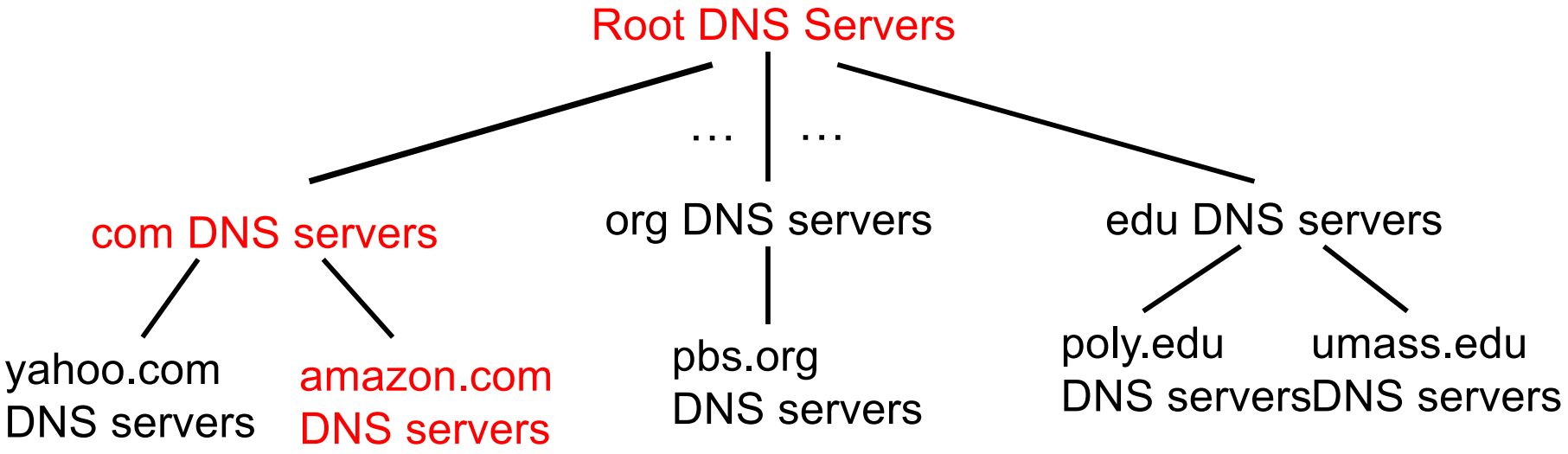
top-level domain (TLD) servers:

- responsible for com, org, net, edu, aero, jobs, museums, and all top-level country domains, e.g.: uk, fr, ca, jp
- Network Solutions maintains servers for .com TLD
- Educause for .edu TLD

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

DNS: a distributed, hierarchical database



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

- client queries 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

Analogy: Marshalls -> Physical Address

Local DNS name server

- does not strictly belong to hierarchy
- each ISP (residential ISP, company, university) has 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
- Difference btw Local DNS and Authoritative DNS server?
  - Given an organization, e.g., WPI, one for its internal users, one for external users

Chapter 2: outline

2.1 principles of network applications
- app architectures
- app requirements

2.2 Web and HTTP

2.5 DNS

Service Overview, Structure

Resolution process

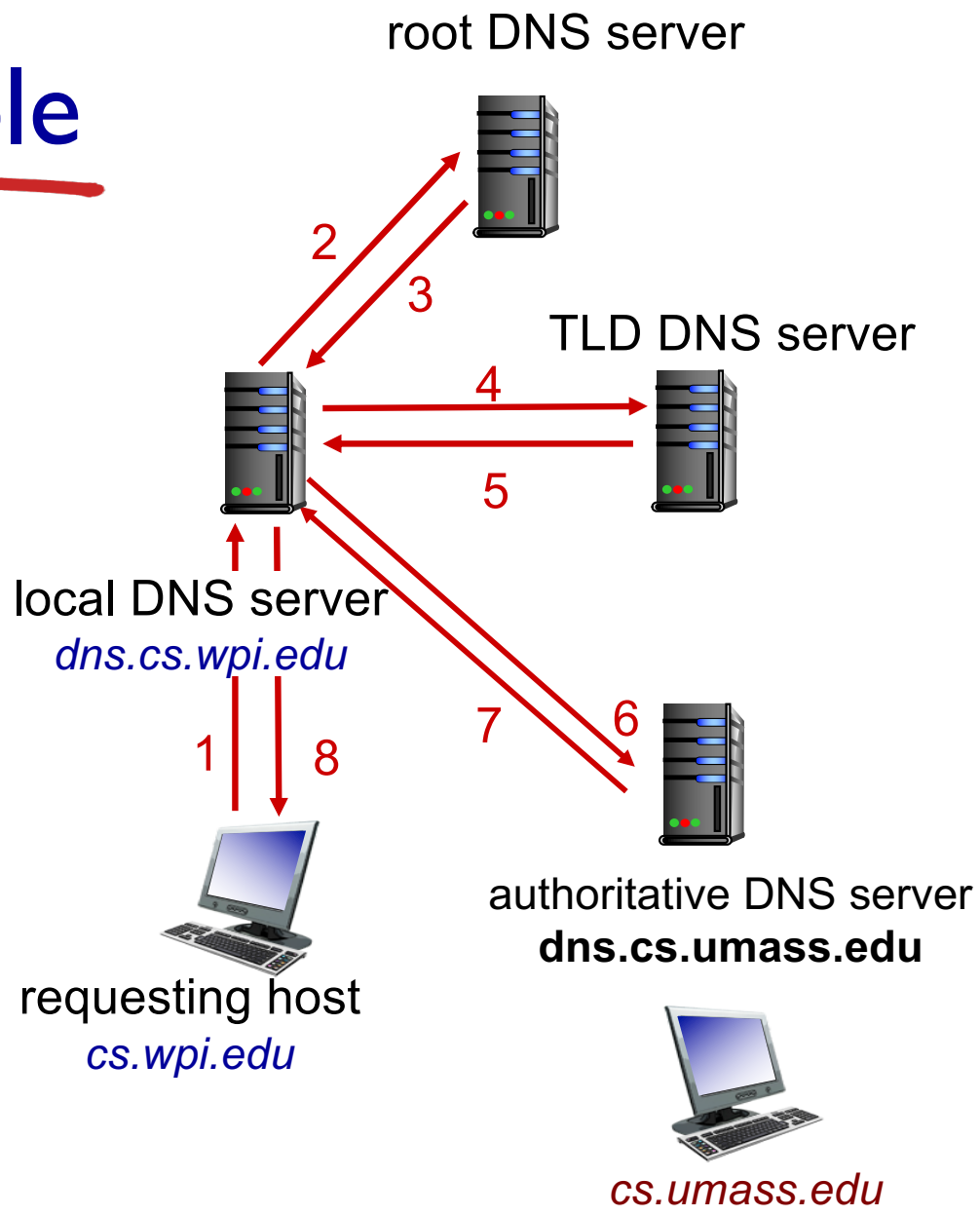
Data Format

DNS name resolution example

- host at *cs.wpi.edu* wants IP address for *cs.umass.edu*

iterated query:

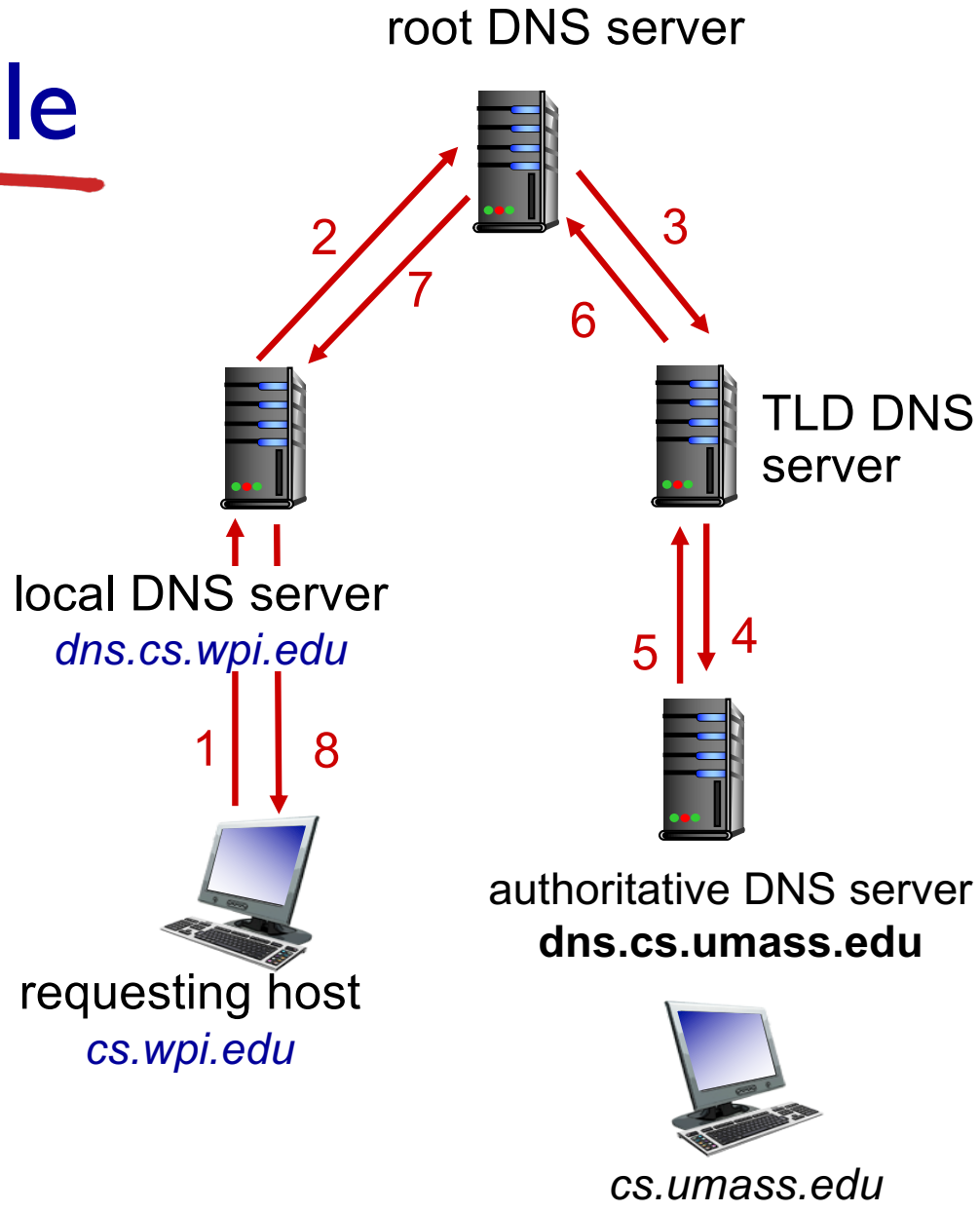
- contacted server replies with name of server to contact
- "I don't know this name, but ask this server"



DNS name resolution example

recursive query:

- puts burden of name resolution on contacted name server
- *Cons:* heavy load at upper levels of hierarchy



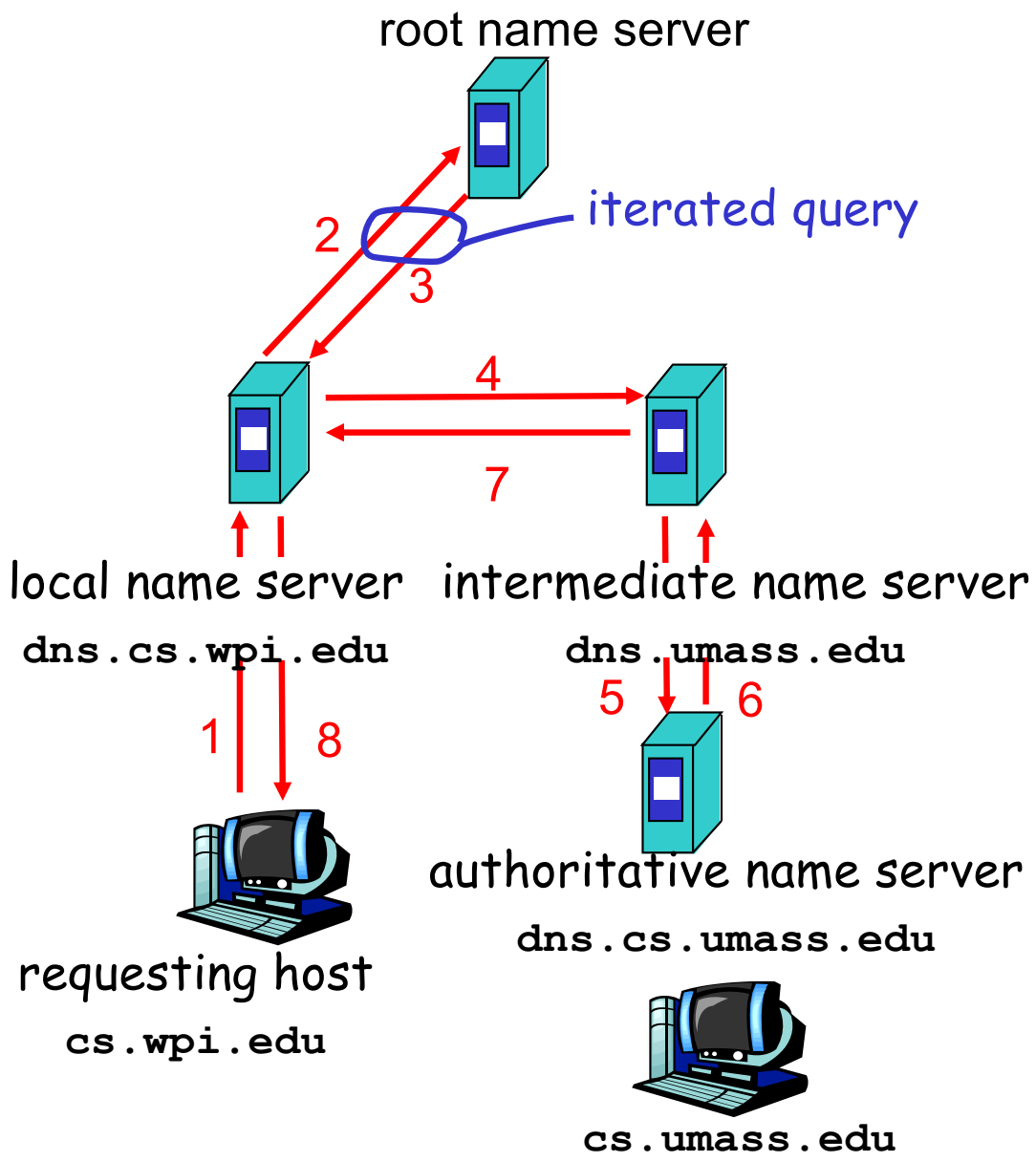
DNS queries

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, updating records

- once (any) name server learns 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 visited

- cached entries may be *out-of-date* (best effort name-to-address translation!)
  - if name host changes IP address, it may not be known Internet-wide until all TTLs expire

Chapter 2: outline

2.1 principles of network applications
- app architectures
- app requirements

2.2 Web and HTTP

2.5 DNS

Service Overview, Structure

Resolution process

Data Format

DNS records (of address mappings)

DNS: distributed db storing resource records (RR)

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

type=A

- **`name`** is hostname
- **`value`** is IP address
- **`(foo.com, 2.2.2.2, A, TTL)`**

type=NS

- **`name`** is domain (e.g., foo.com)
- **`value`** is hostname of authoritative name server for this domain
- **`(foo.com, dns1.foo.com, NS, TTL)`**

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
- **`(foo.com, rel1.foo.com, CNAME, TTL)`**

type=MX

- **`value`** is name of mailserver associated with **`name`**
- **`(foo.com, mail.foo.com, MX, TTL)`**

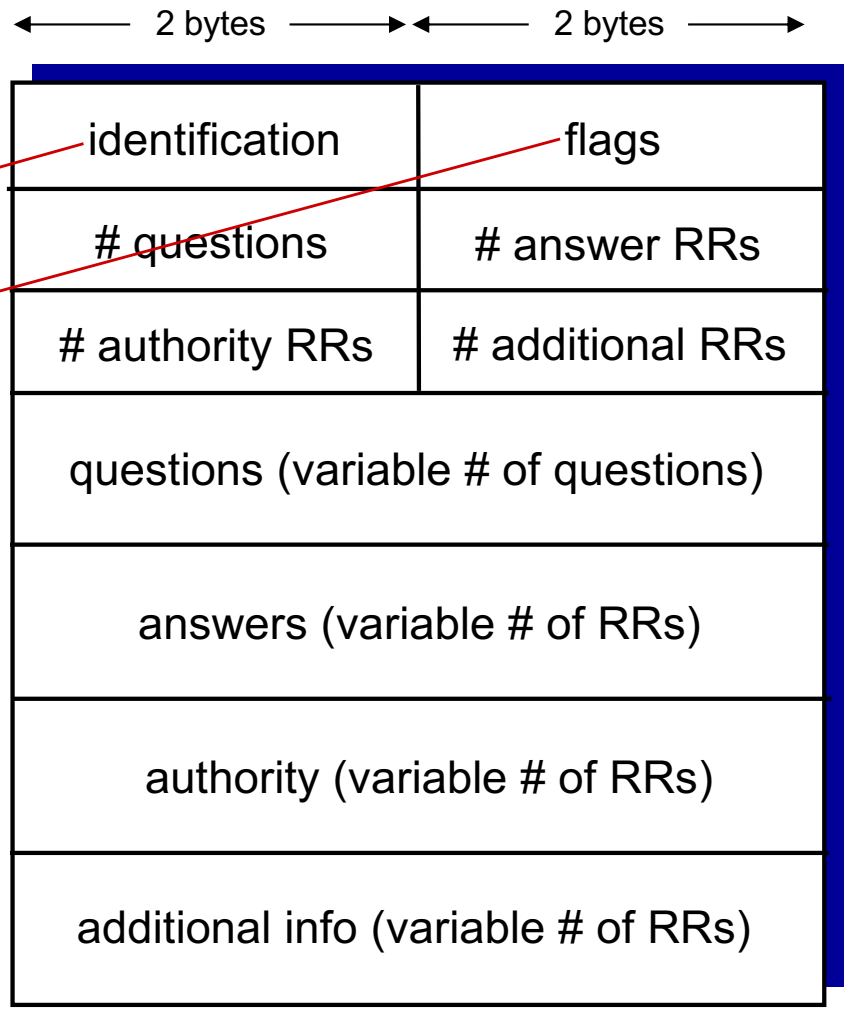
DNS protocol, messages

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

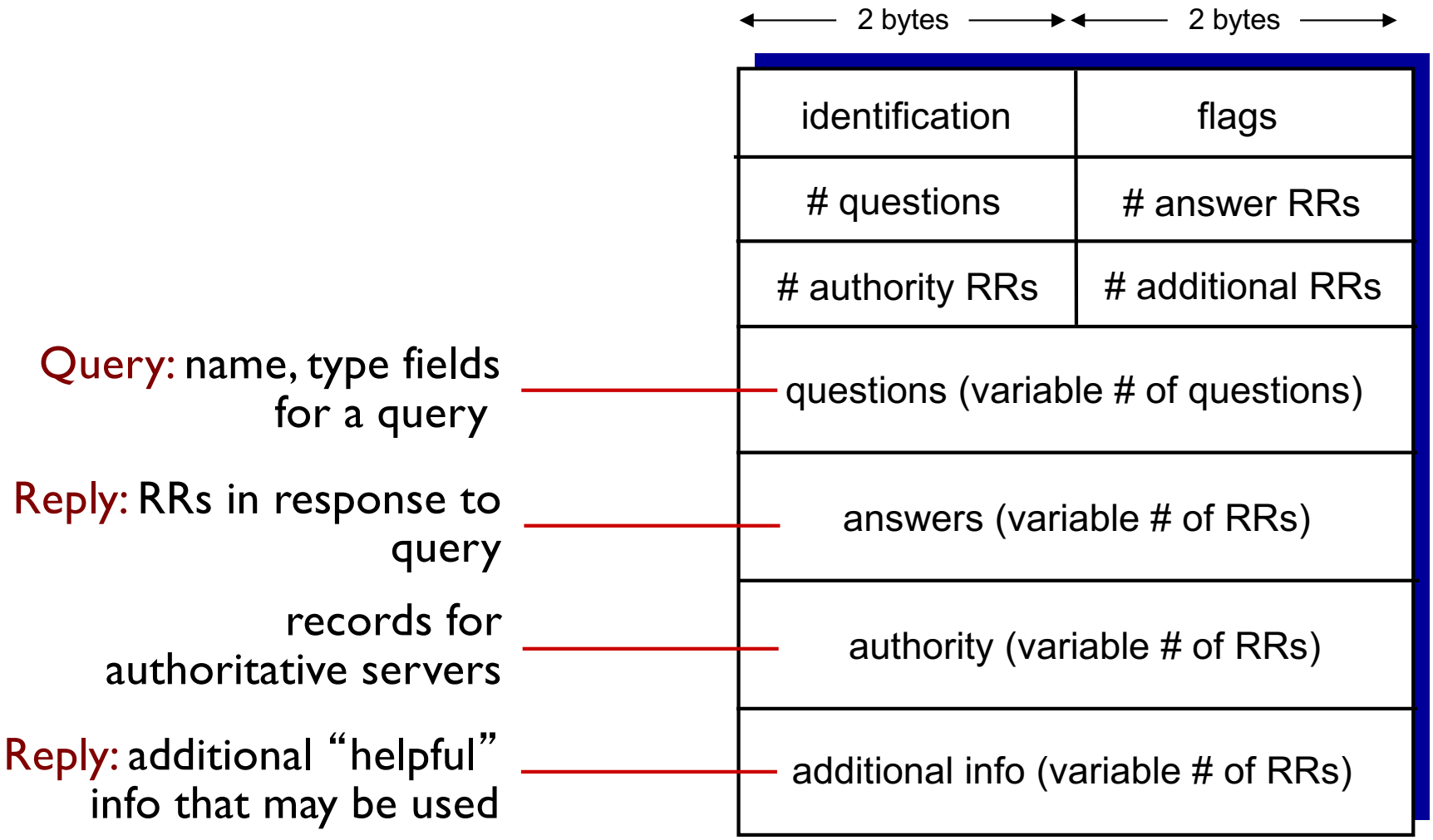
msg header

- identification: 16 bit # for query, reply to query uses same #
- flags:
  - query or reply (1 bit)
  - recursion desired (query) (1 bit)
  - recursion available (reply) (1 bit)
  - reply is authoritative (reply) (1 bit)

    (DNS is an authoritative DNS to a queried name)



DNS protocol, messages



Inserting records into DNS

- example: new startup "Networkabc"
- register name networkabc.com at *DNS registrar* (e.g., Network Solutions) (and pay a fee for it.)
  - provide names, IP addresses of authoritative name server (primary and secondary)
  - registrar inserts two RRs into .com TLD server:
    (networkabc.com, dns1.networkabc.com, NS)
    (dns1.networkabc.com, 212.212.212.1, A)
- Authoritative server
  - create type A record for www.networkabc.com;
    - **(networkabc.com, 212.212.221.5, A)**
  - create type MX record for networkabc.com
    - **(networkabc.com, mail.networkabc.com, MX)**

Attacking DNS

DDoS attacks

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

Questions?