Computer Networks and Applications

COMP 3331/COMP 9331

Week 3

Application Layer II, DNS)

Reading Guide: Chapter 2, Sections 2.3, 2.4

Application Layer: outline

2.1 principles of network applications
- app architectures
- app require

2.2 Web and H

2.3 electronic mail
- SMTP, POP3, IMAP

2.4 DNS

2.5 P2P applications

2.6 video streaming and content distribution orks (CDNs)

et programming

DP and TCP

Self study

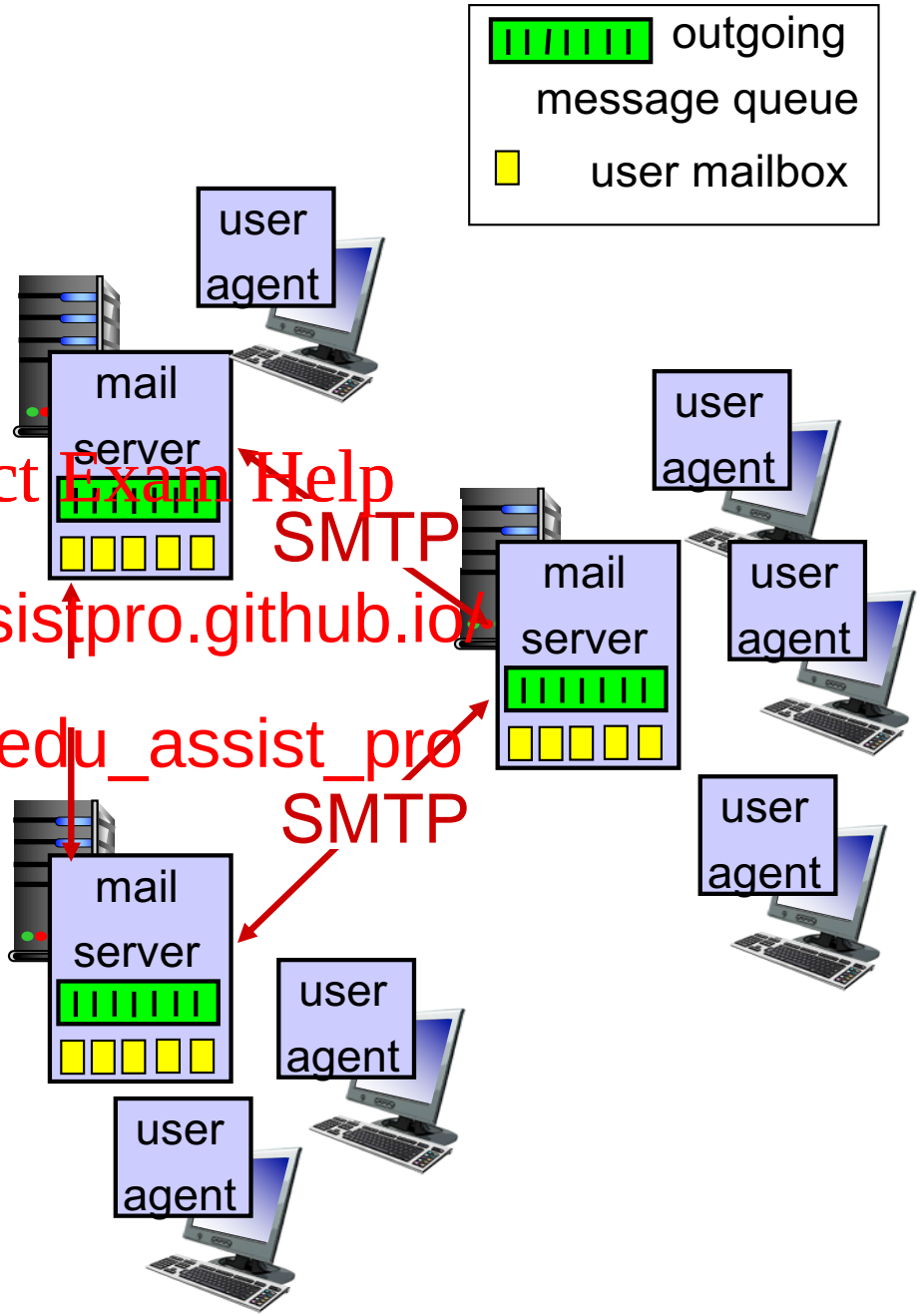
Electronic mail

Three major components:

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

User Agent

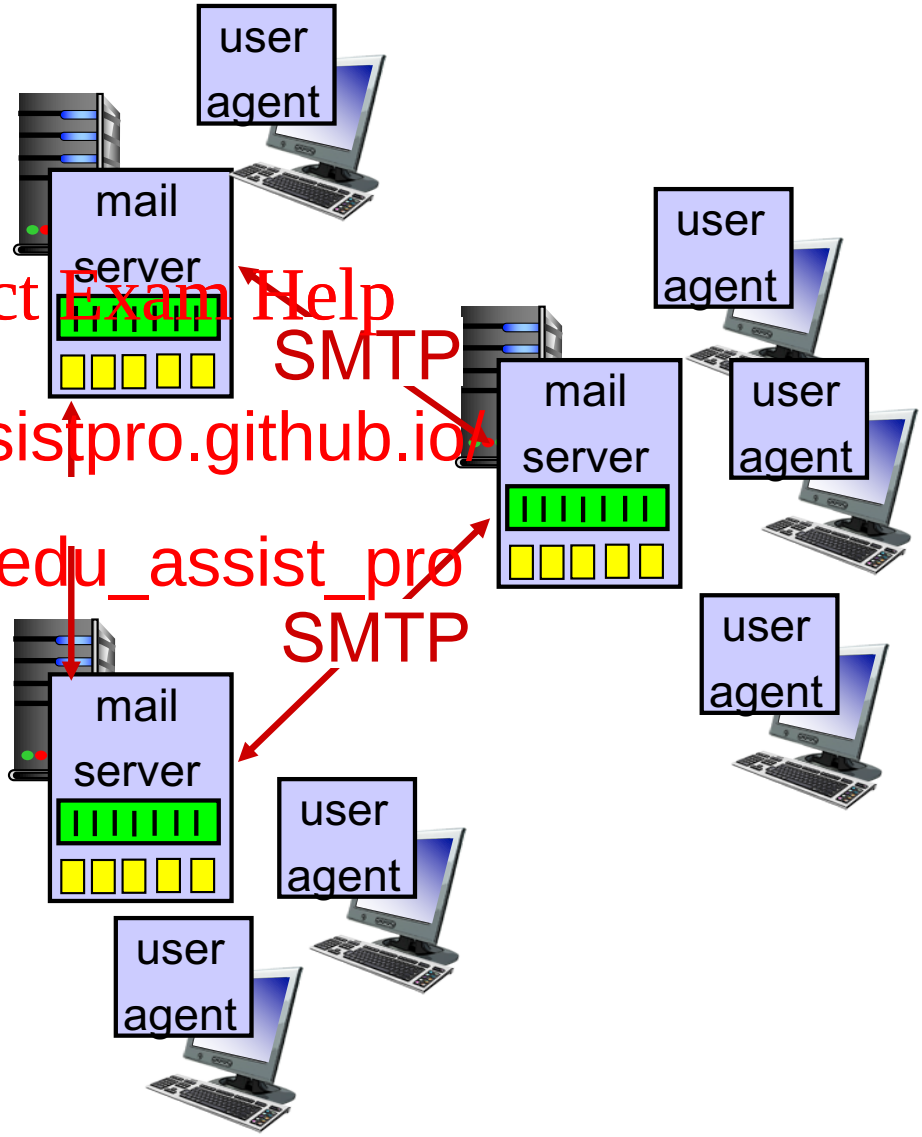
- a.k.a. "mail reader"
- composing, editing, reading mail messages
- e.g., Outlook, Thunderbird, iPhone mail client
- outgoing, incoming messages stored on server



Electronic mail: mail servers

mail servers:

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

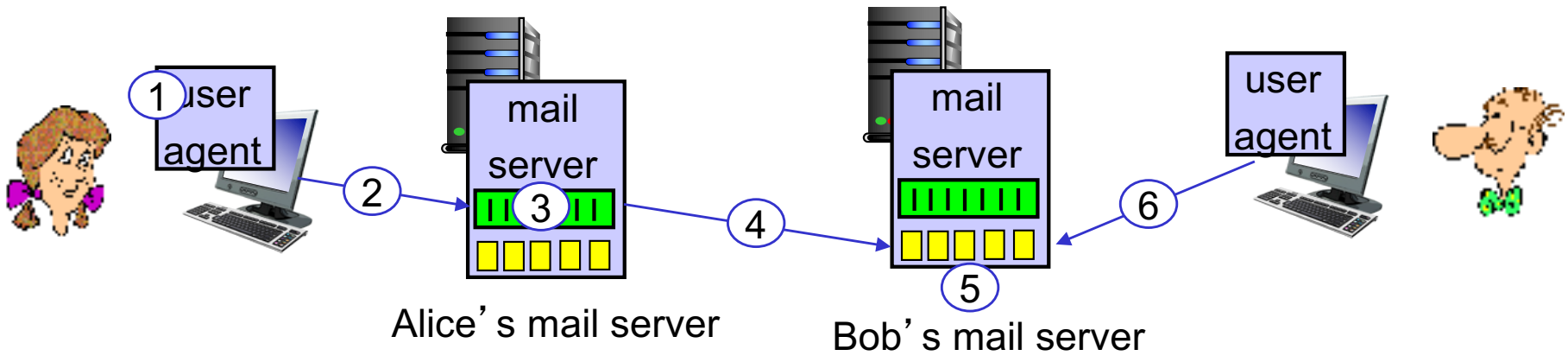


Electronic Mail: SMTP [RFC 2821]

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

Scenario: Alice sends message to Bob

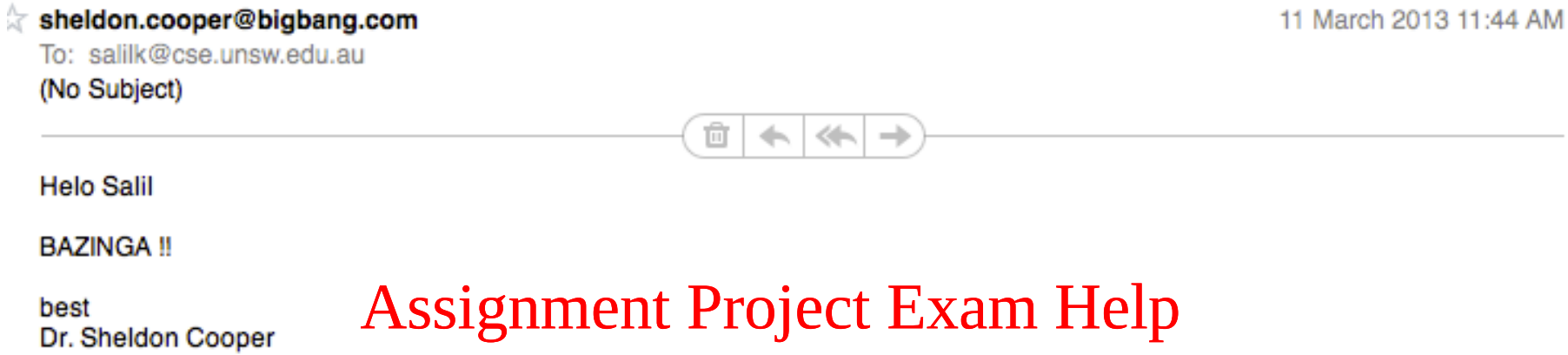
1) Alice uses UA to compose message "to" `bob@someschool.edu`
2) Alice's UA sends message to her mail server; message placed in message
3) client side of SMT TCP connection with Bob's mail server
4) SMTP client sends Alice's message over the TCP connection
5) Bob's mail server places the message in Bob's mailbox
kes his user agent
essage



Sample SMTP interaction

```
S: 220 hamburger.edu
C: HELO crepes.fr
S: 250  Hello crepes.fr, pleased to meet you
C: MAIL FROM: <alice@crepes.fr>
S: 250 alice@crepes.fr... Sender ok
C: RCPT TO: <b
S: 250 bob@hamburger.edu .          ent ok
C: DATA
S: 354 Enter mail, end with "." on a line by itself
C: Do you like ketchup?
C: How about pickles?
C: .
S: 250 Message accepted for delivery
C: QUIT
S: 221 hamburger.edu closing connection
```

How to tell a fake email?



sheldon.cooper@bigbang.com 11 March 2013 11:44 AM
To: salilk@cse.unsw.edu.au
(No Subject)

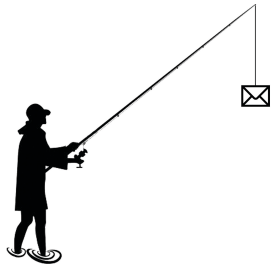
Helo Salil

BAZINGA !!

best
Dr. Sheldon Cooper

Further reading: http://www.millersmiles.co.uk/identitytheft/spoofemail-060603.htm

Phishing



- Spear phishing
  - Phishing attempts directed at specific individuals or companies
  - Attackers may gather personal information (social engineering) about their targets to increase their probability of success
  - Most popular a attacks
- Clone phishing
  - A type of phishing attack whereby a legitimate, and previously delivered email containing an attachment or link has had its content and recipient address(es) taken and used to create an almost identical or cloned email.
  - The attachment or link within the email is replaced with a malicious version and then sent from an email address spoofed to appear to come from the original sender.

SMTP: final words

- SMTP uses persistent connections
- SMTP requires message (header & body) to be in 7-bit ASCII
- SMTP server use `CRLF.CRLF` to determine end of message

comparison with HTTP:

- HTTP: pull
- SMTP: push
- e ASCII
- d/response
- status codes
- HTTP: each object encapsulated in its own response msg
- SMTP: multiple objects sent in multipart msg

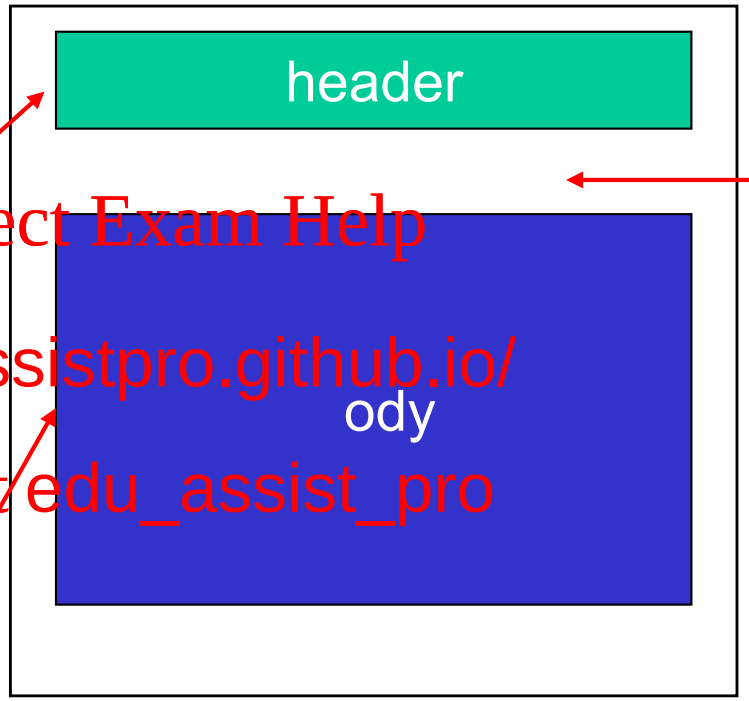
Mail message format

SMTP: protocol for exchanging email msgs

RFC 5322 (822,2822): standard for text message format (Internet Format, IMF):

- header lines, e.g.,
  - To:
  - From:
  - Subject:

  different from SMTP MAIL FROM, RCPT TO: commands!
- Body: the "message"
  - ASCII characters only



Quiz: SMTP

Why do we have Sender's mail server?

- User agent can directly connect with recipient mail server without the need of sender's mail server? What's the catch?

Why do we have ’s mail server?

- Can't the recipie n own e

Quiz: E-mail attachments?

- IF SMTP only allows 7-bit ASCII, how do we send pictures/videos/files via email?

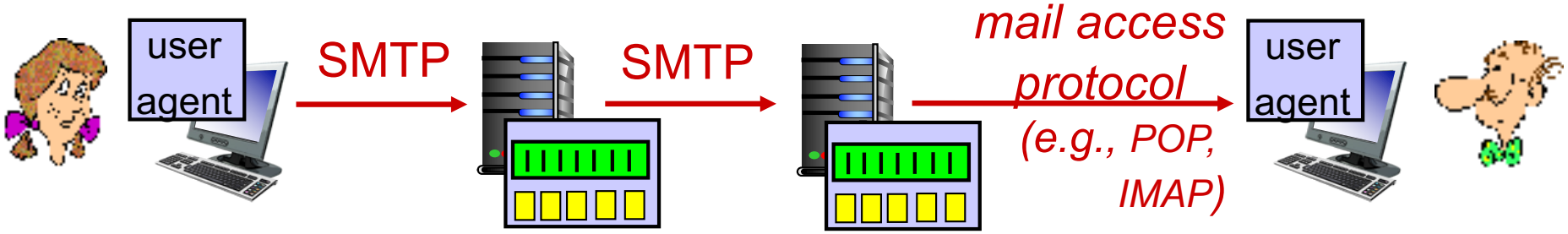
A: We use a dif of SMTP

B: We encode these objects a ll

C: We're really sending links to the objects, rather than the objects themselves

D: Like HTTP, we can send these in binary

Mail access protocols



- SMTP: delivery/ r
- mail access protocol: retrieval r
  - POP: Post Office Protocol [R thorization, download
  - IMAP: Internet Mail Access Protocol [RFC 1730]: more features, including manipulation of stored msgs on server
  - HTTP(S): Gmail, Yahoo! Mail, etc.

Quiz: HTTP vs SMTP

- Which of the following is not true?

A. HTTP is pull-based, SMTP is push-based

B. HTTP uses object, SMTP uses a multipart message f

C. SMTP uses persistent connections

D. HTTP uses client-server communication but SMTP does not

2. Application Layer: outline

2.1 principles of network applications
- app architectures
- app require

2.2 Web and H

2.3 electronic mail
- SMTP, POP3, IMAP

2.4 DNS

2.5 P2P applications

2.6 video streaming and content distribution orks (CDNs)

et programming

DP and TCP

A nice overview: https://webhostinggeeks.com/guides/dns/

DNS: domain name system

people: many identifiers:

- TFN, name, passport #

Internet hosts, routers:

- IP address (32 bit) - used for addr 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*
- *n-layer protocol:* hosts, rvers communicate to es (address/name)

- note: core Internet function, implemented as application-layer protocol
- complexity at network's "edge"

DNS: History

- Initially all host-address mappings were in a hosts.txt file (in /etc/hosts):
  - Maintained by the Stanford Research Institute (SRI)
  - Changes were submitted to SRI by email
  - New versions of om SRI
  - An administrator cretion

Jon Postel

- As the Internet grew this system n:
  - SRI couldn't handle the load; names were not unique; hosts had inaccurate copies of hosts.txt

- The Domain Name System (DNS) was invented to fix this

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

DNS: services, structure

DNS services

- hostname to IP address translation
- host aliasing
  - canonical, alias names
- mail server ali
- load distribution
  - replicated Web servers: many IP addresses correspond to one name
  - Content Distribution Networks: use IP address of requesting host to find best suitable server
    - Example: closest, least-loaded, etc

why not centralize DNS?

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

doesn't scale!

Goals

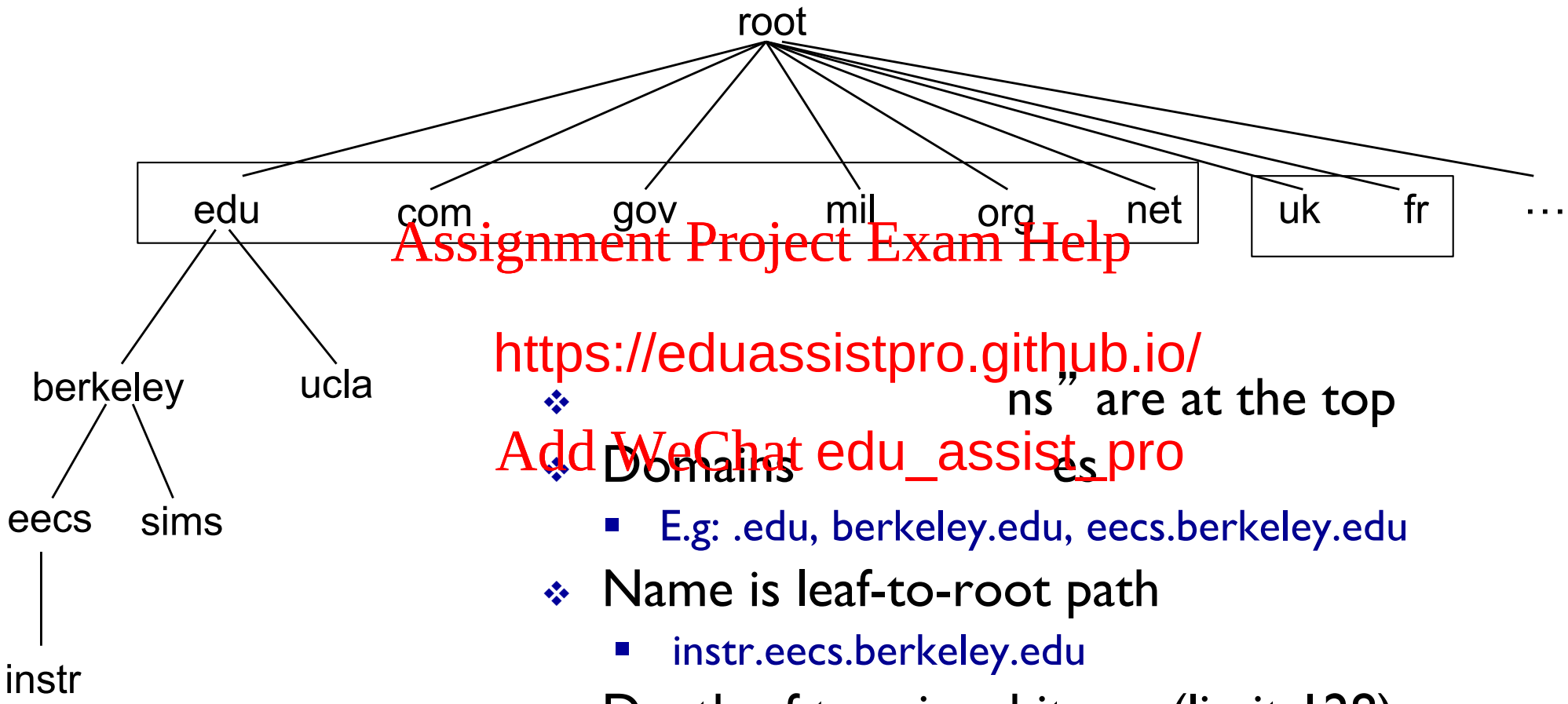
- No naming conflicts (uniqueness)
- Scalable
  - many names
  - (secondary) fr
- Distributed, a ation
  - Ability to update my own (n ames
  - Don't have to track everybody's updates
- Highly available
- Lookups should be fast

Key idea: Hierarchy

Three intertwined hierarchies

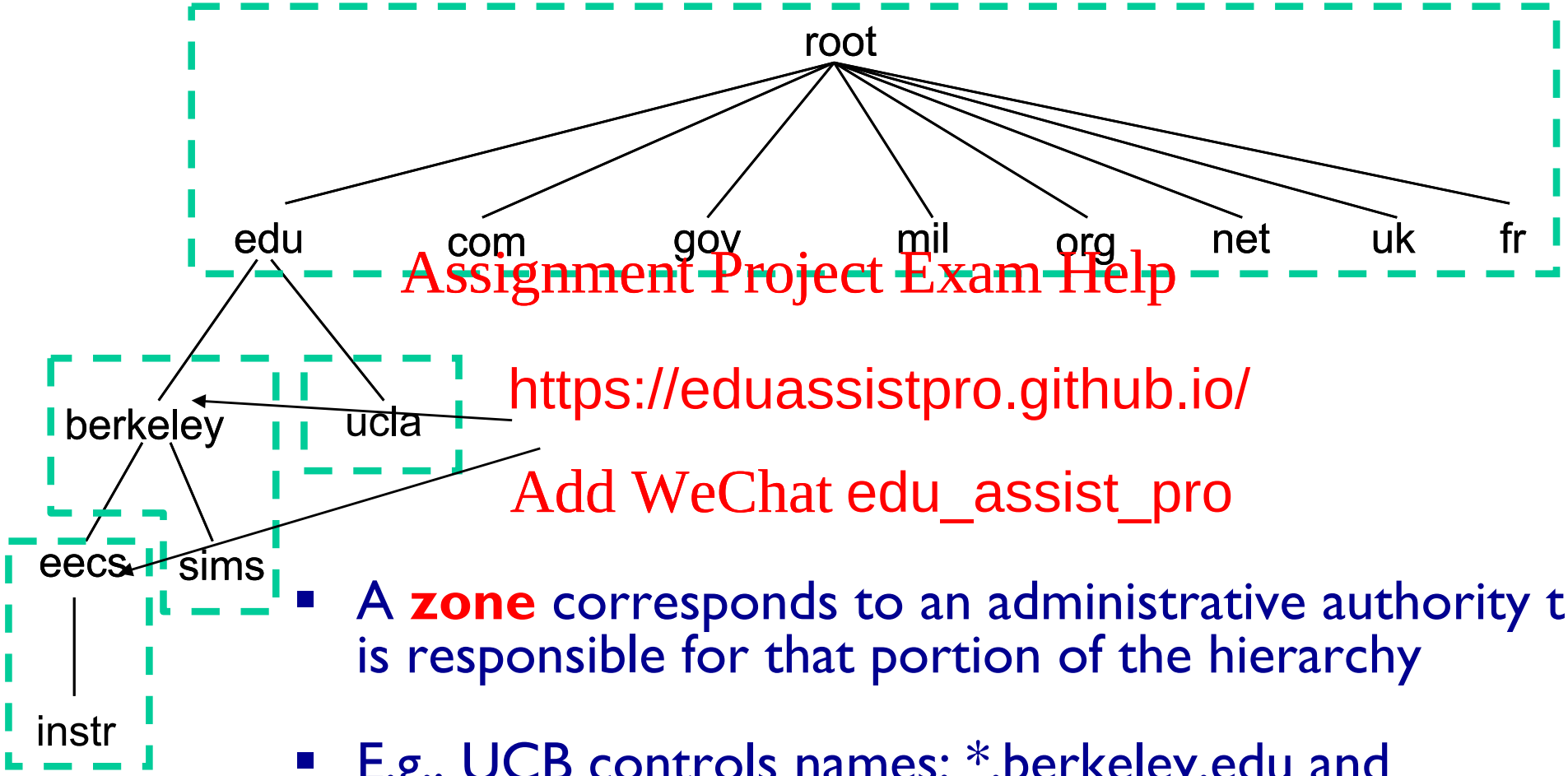
- Hierarchical namespace
  - As opposed to original flat namespace
- Hierarchicall
  - As opposed to centralised
- (Distributed) hierarchy of servers
  - As opposed to centralised storage

Hierarchical Namespace



- ns" are at the top
- Domains es
  - E.g: .edu, berkeley.edu, eecs.berkeley.edu
- Name is leaf-to-root path
  - instr.eecs.berkeley.edu
- Depth of tree is arbitrary (limit 128)
- Name collisions trivially avoided
  - each domain is responsible

Hierarchical Administration



- A **zone** corresponds to an administrative authority that is responsible for that portion of the hierarchy
- E.g., UCB controls names: *.berkeley.edu and *.sims.berkeley.edu
- E.g., EECS controls names: *.eecs.berkeley.edu

Server Hierarchy

- Top of hierarchy: Root servers
  - Location hardwired into other servers

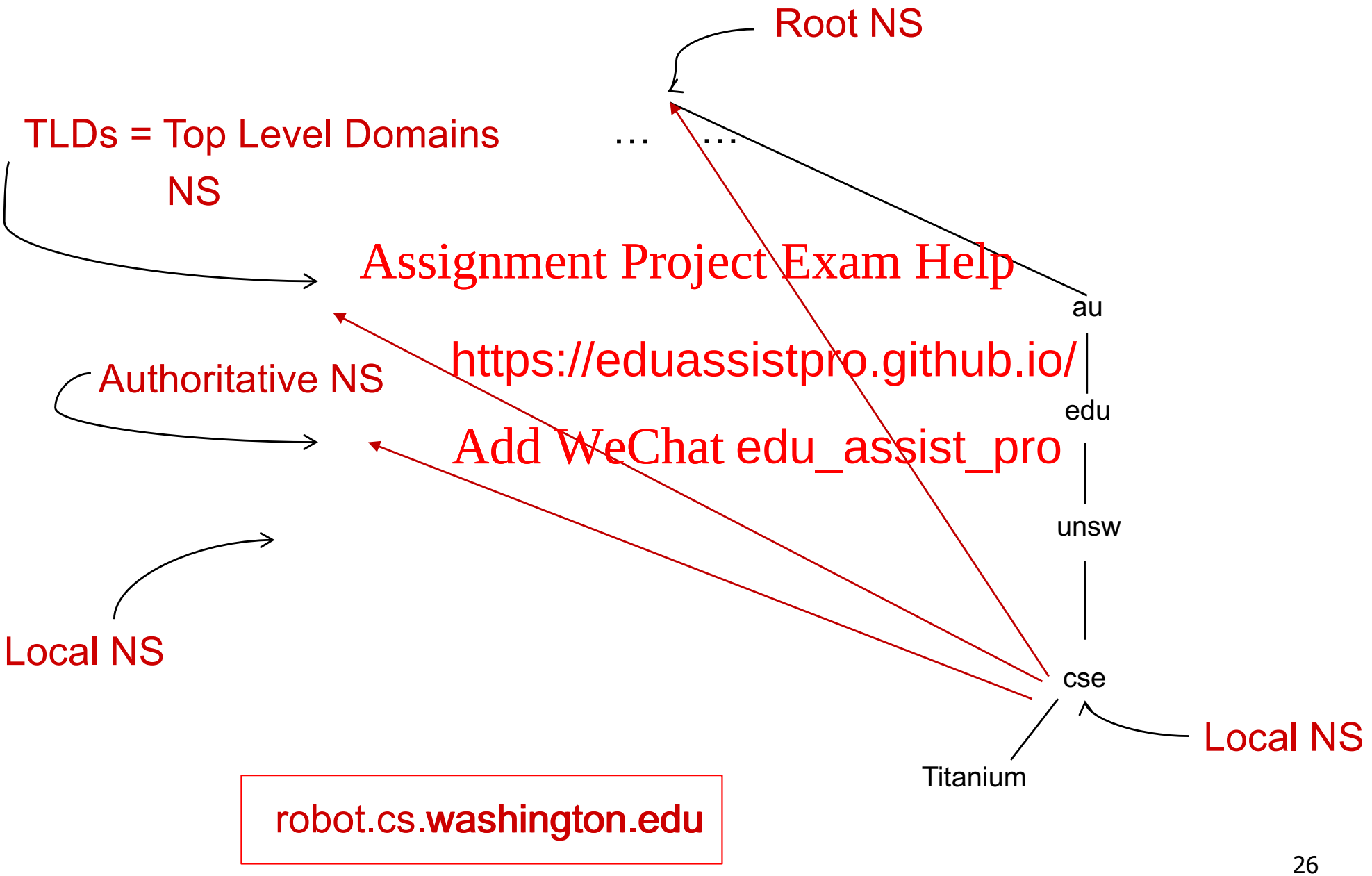
- Next Level: T) servers
  - .com, .edu, et
  - Managed professionally

- Bottom Level: Authoritative DNS servers
  - Actually store the name-to-address mapping
  - Maintained by the corresponding administrative authority

Server Hierarchy

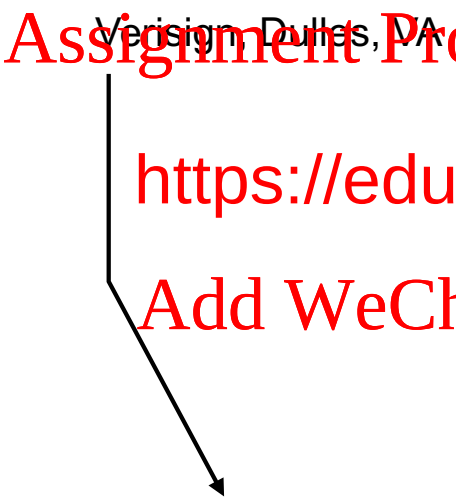
- Each server stores a (small!) subset of the total DNS database
- An authoritative DNS server stores "resource records" for all DNS names in the domain that it has authority for
- Each server needs to know oth hat are responsible for the other portions of the hi
  - Every server knows the root
  - Root server knows about all top-level domains

DNS: a distributed, hierarchical database



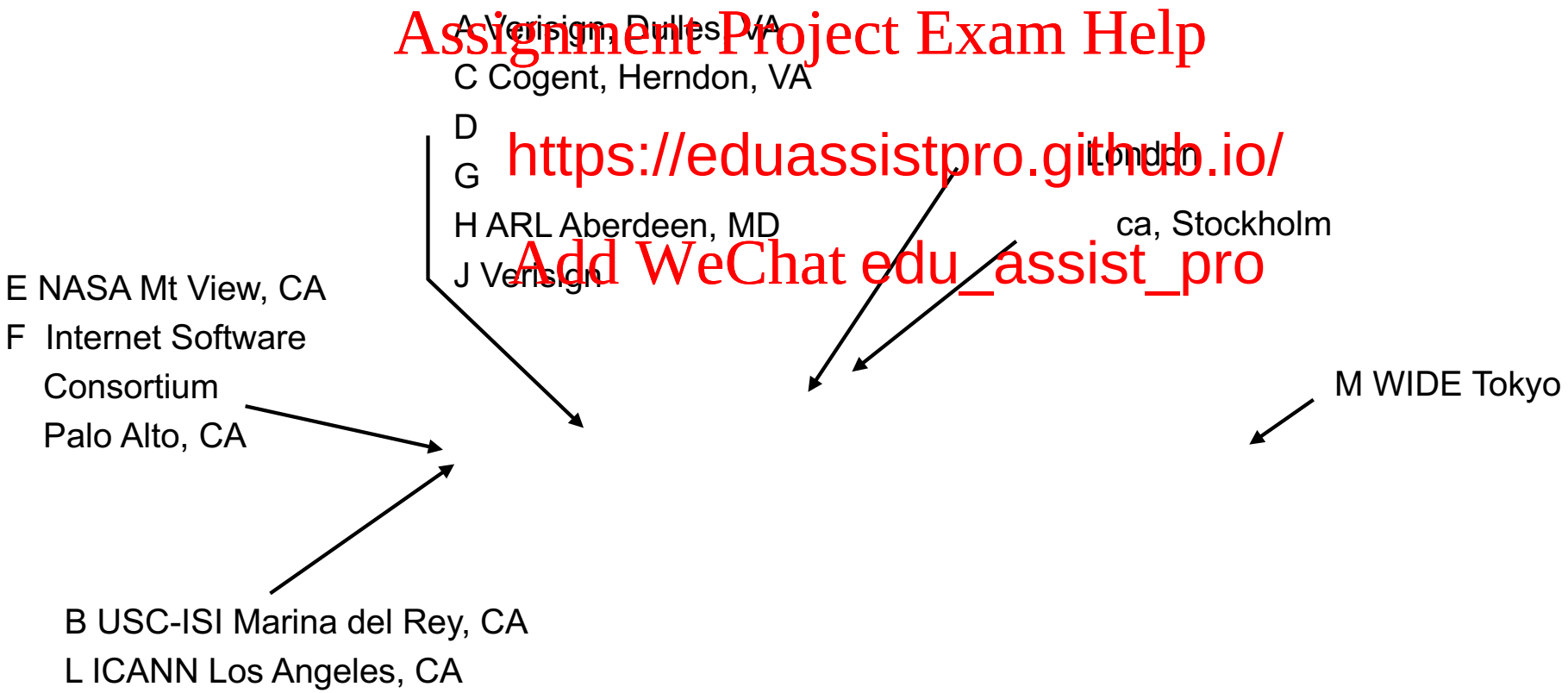
DNS Root

- Located in Virginia, USA
- How do we make the root scale?



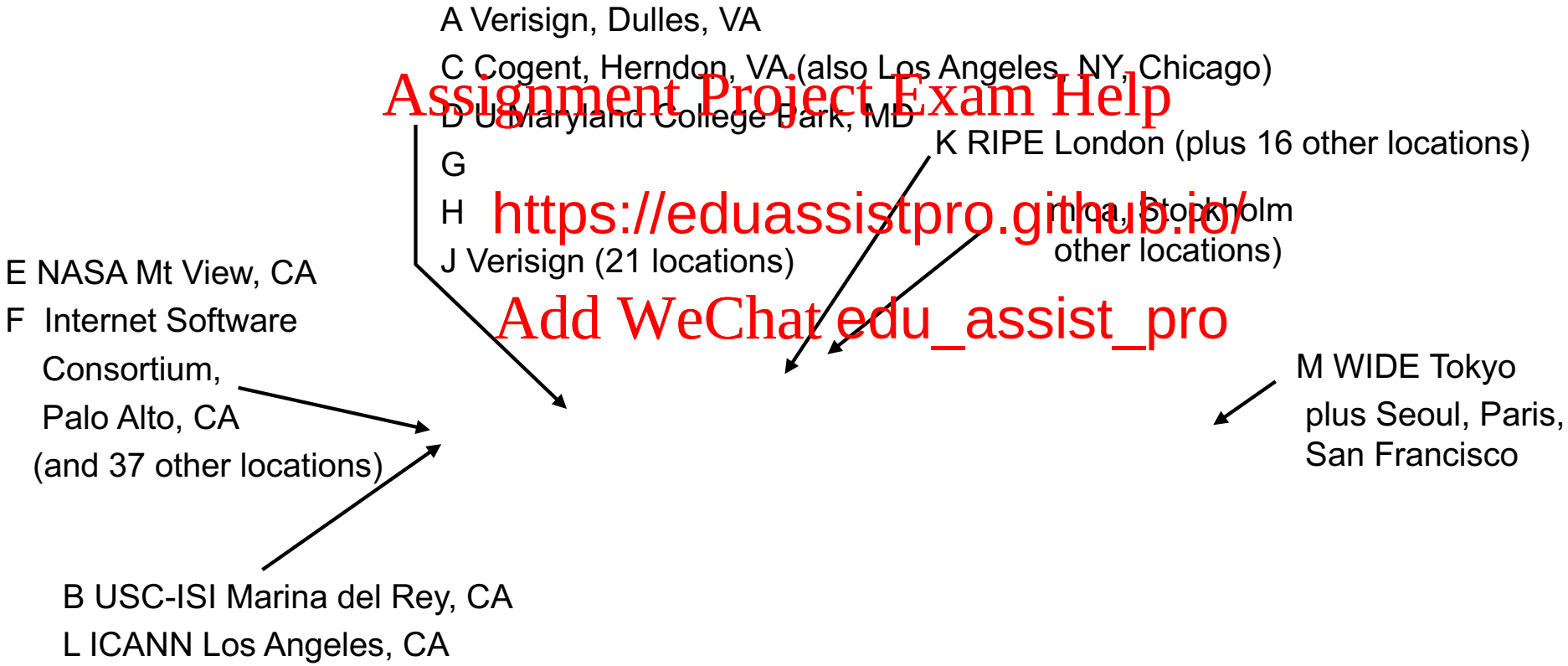
DNS Root Servers

- 13 root servers (labeled A-M; see http://www.root-servers.org/)



DNS Root Servers

- 13 root servers (labeled A-M; see http://www.root-servers.org/)
- Replicated via any-casting



Root Server health: https://www.ultratools.com/tools/dnsRootServerSpeed

DNS: root name servers

www.root-servers.org

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

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

Local DNS name server

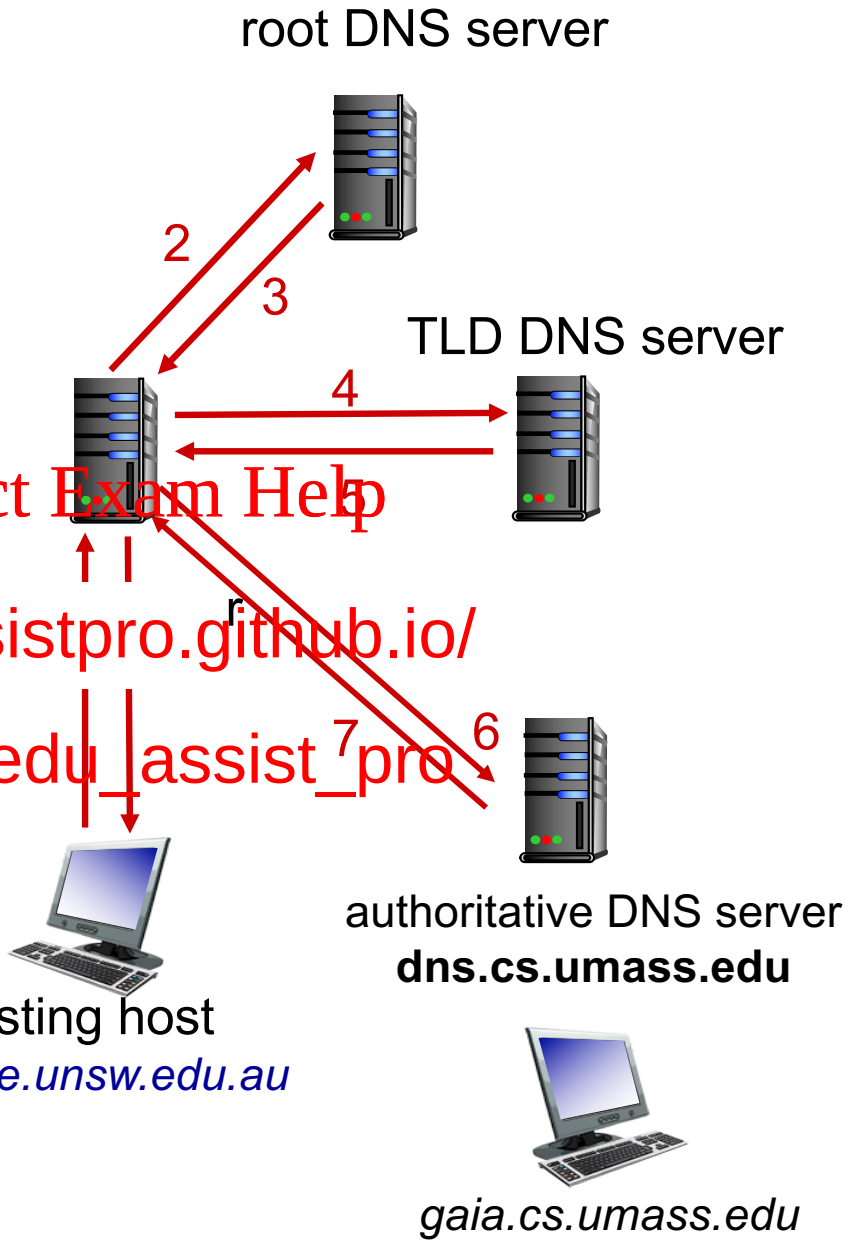
- does not strictly belong to hierarchy
- each ISP (residential ISP, company, university) has one
  - also called "default name server"
- Hosts configured with local DNS server address (e.g., /etc/resolv.conf) t configuration protocol (e.g., D
- Client application
  - Obtain DNS name (e.g., from URL
  - Do **gethostbyname()** to trigger DNS request to its local DNS 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

- host at wagner.cse.unsw.edu.au wants IP address for gaia.cs.umass.edu

iterated query:

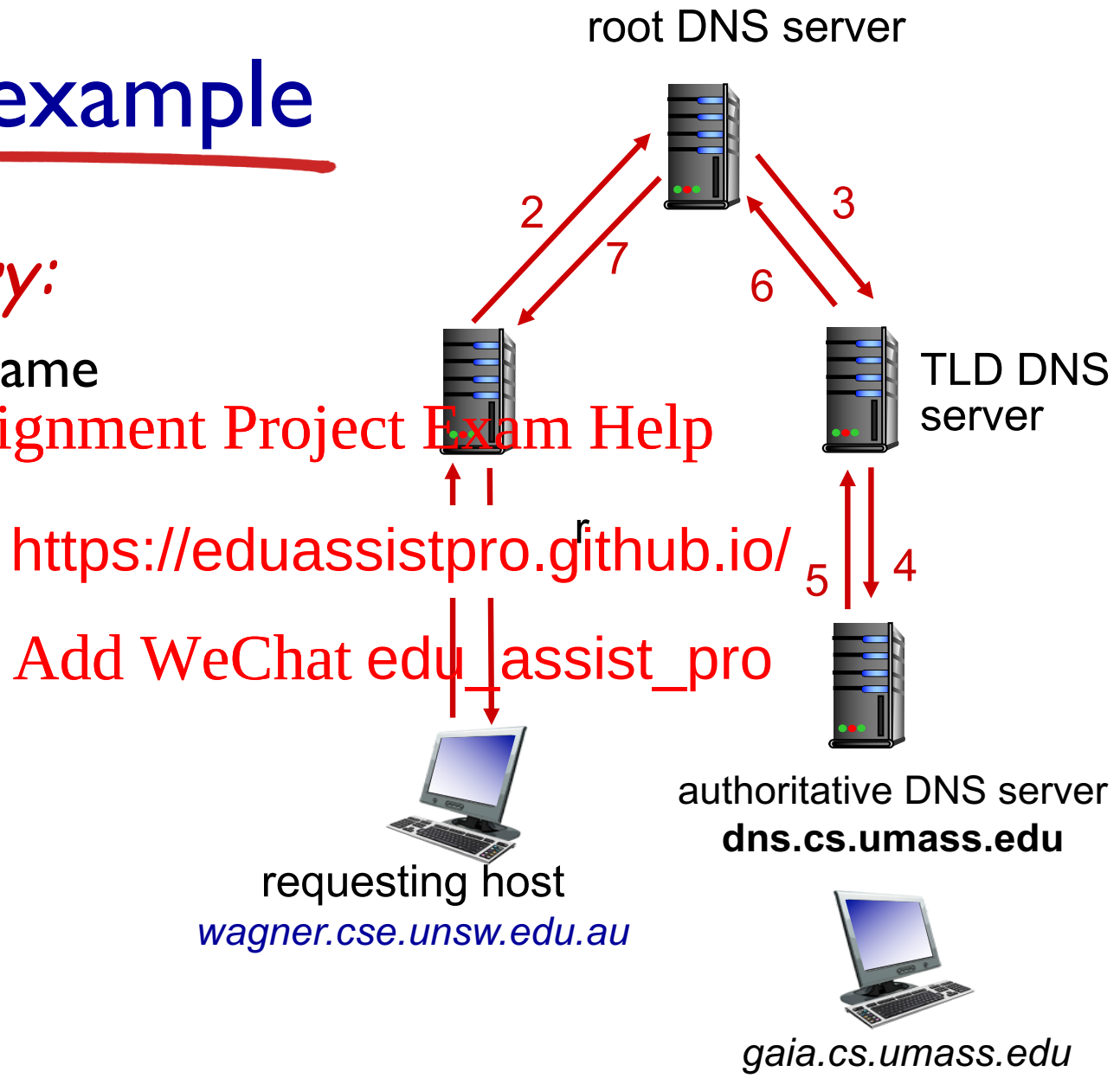
- contacted server replies with name of server to contact
- "I don't know this name, but ask this server"



DNS name resolution example

recursive query:

- puts burden of name resolution on contacted name server



DNS: caching, updating records

- once (any) name server learns mapping, it *caches* mapping
  - cache entries timeout (disappear) after some time (TTL)
  - TLD servers typically cached in local name servers
    - thus root name servers not often visited
- Subsequent r rden DNS
- cached entries may be *out* est effort name-to-address translati
  - if name host changes IP address, may not be known Internet-wide until all TTLs expire

DNS records

DNS: distributed db storing resource records (RR)

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

type=A

- **`name`** is hostna
- **`value`** is IP address

type=NS

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

ME

- alias name for some (the real) name

`com` is really **`servereast.backup2.ibm.com`**

- **`value`** is canonical name

type=MX

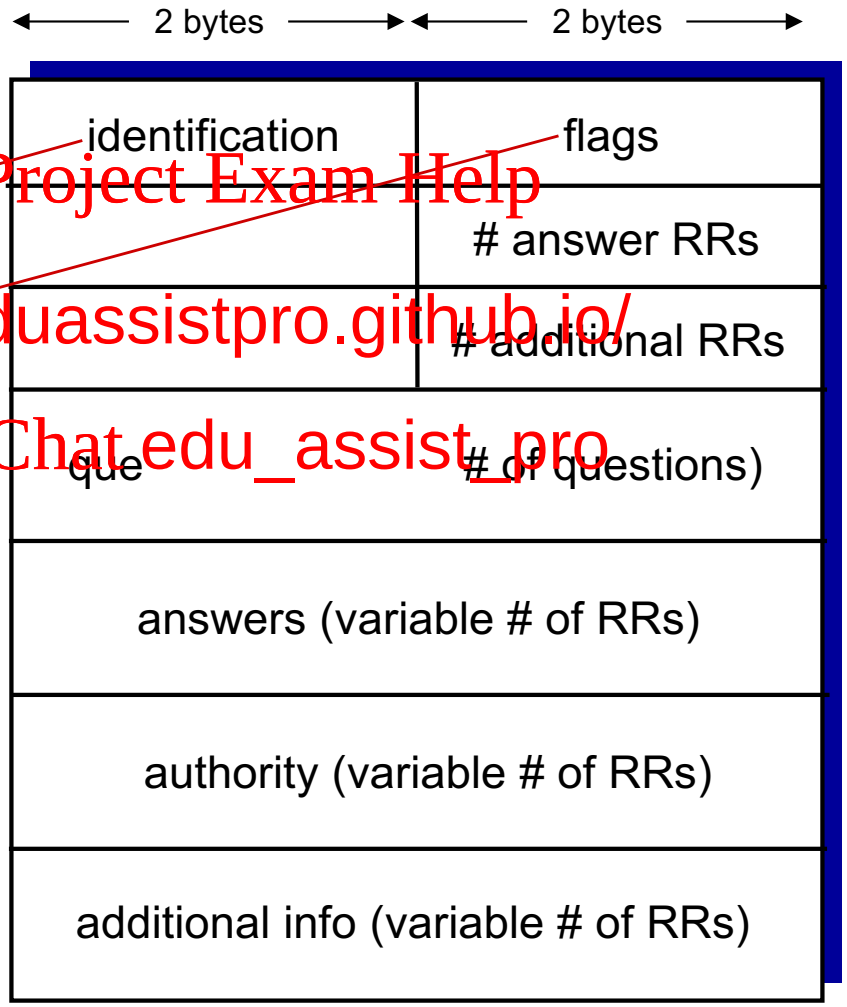
- **`value`** is name of mailserver associated with **`name`**

DNS protocol, messages

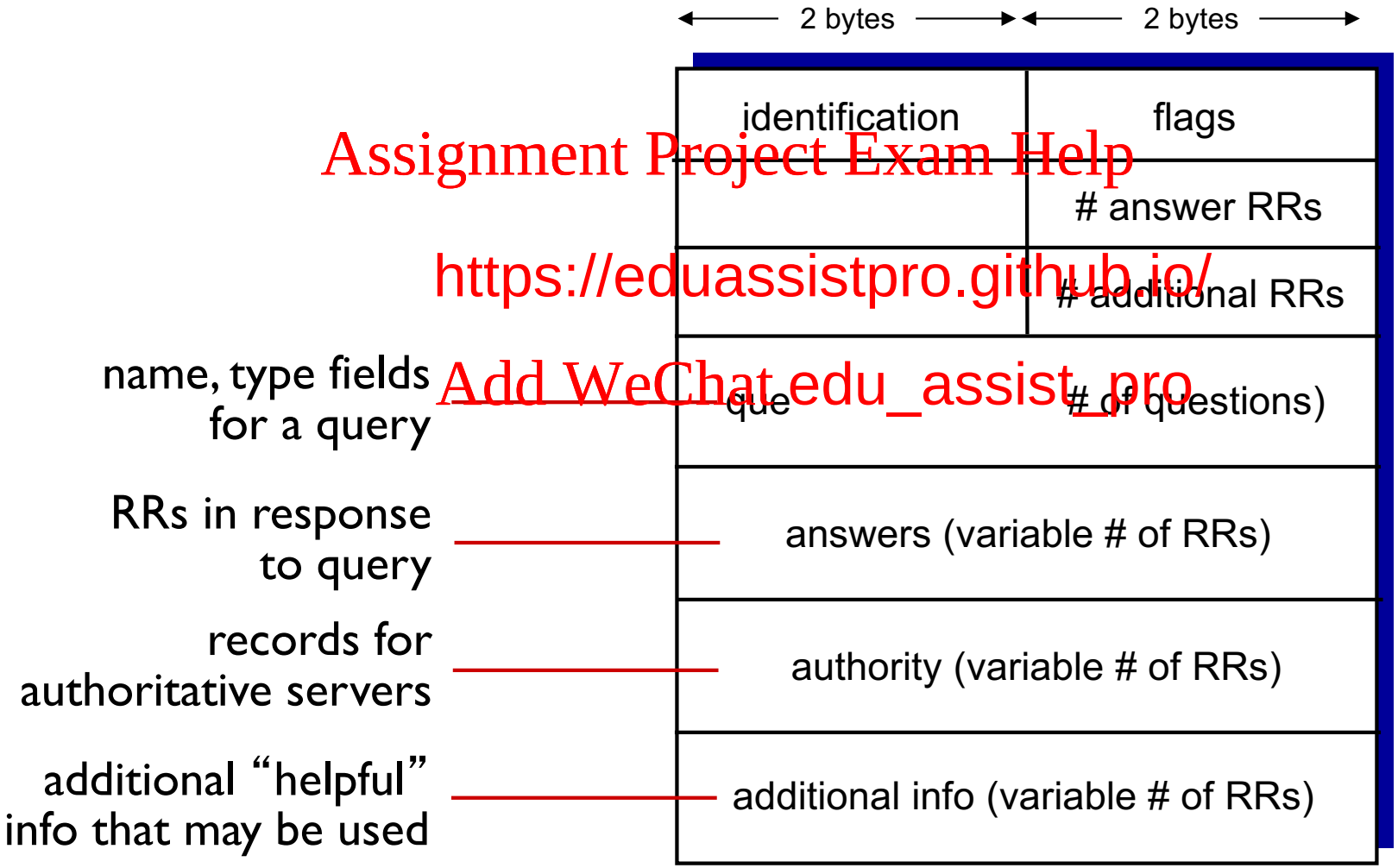
- *query* and *reply* messages, both with same *message format*

msg header

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



DNS protocol, messages



An Example

Try this out yourself. Part of one of the lab

Inserting records into DNS

- example: new startup "Network Utopia"
- register name networkutopia.com at *DNS registrar* (e.g., Network Solutions)
  - provide names, IP addresses of authoritative name server (primary and
  - registrar inse LD server:

```
(networkutopia.com, dns                utopia.com, NS)
(dns1.networkutopia.com                .212.1, A)
```


- create authoritative server type A record for www.networkuptopia.com; type MX record for networkutopia.com
- Q: Where do you insert these type A and type MX records?

  A: ??

Reliability

- DNS servers are replicated (primary/secondary)
  - Name service available if at least one replica is up
  - Queries can be load-balanced between replicas
- Usually, UDP
  - Need reliabili on top of UDP
  - Spec supports TCP too, but implemented
- Try alternate servers on ti
  - Exponential backoff when retrying same server
- Same identifier for all queries
  - Don't care which server responds

DNS provides Indirection

- Addresses can change underneath
  - Move www.cnn.com to 4.125.91.21
  - Humans/Apps should be unaffected
- Name could ma s
  - Enables
    - Load-balancing
    - Reducing latency by picking near
- Multiple names for the same address
  - E.g., many services (mail, www, ftp) on same machine
  - E.g., aliases like www.cnn.com and cnn.com
- But, this flexibility applies only within domain!

Reverse DNS

- IP address -> domain name
- Special PTR record type to store reverse DNS entries
- Where is rever
  - Troubleshooti te and ping
  - "Received" trace header field mail
  - SMTP servers for validating IP of originating servers
  - Internet forums tracking users
  - System logging or monitoring tools
  - Used in load balancing servers/content distribution to determine location of requester

Do you trust your DNS server?

- Censorship

- Logging
  - IP address, websites visited, geolocation data and more
  - E.g., Google DNS:

    https://developers.google.com/speed/public-dns/privacy

Attacking DNS

DDoS attacks

- Bombard root servers with traffic
  - Not successful to date
  - Traffic Filtering
  - Local DNS serv IPs of TLD serv root server to be bypassed
- Bombard TLD servers
  - Potentially more dangerous

Redirect attacks

- Man-in-middle
  - Intercept queries
- DNS poisoning
  - ogus replies to DNS , which caches
- E for DDoS
  - ies with spoofed source address: target IP
- Requires amplification

Want to dig deeper?

http://www.networkworld.com/article/2886283/security0/top-10-dns-attacks-likely-to-infiltrate-your-network.html

Detailed Report at - http://www.verizonenterprise.com/resources/reports/rp_data-breach-digest-2017-sneak-peek_xg_en.pdf

DNS Cache Poisoning

- Suppose you are a bad guy  and you control the name server for drevil.com. Your name server receives a request to resolve www.drevil.com. and you respond as follows:

;; QUESTION SECTION:
;www.drevil.com. IN A

;; ANSWER SECTION:
www.drevil.com 300 IN A 129.45.212.42

;; AUTHORITY SECTION:
drevil.com 86400 IN NS dns1.drevil.com.
drevil.com 86400 IN NS google.com

A drevil.com machine, **not** google.com

;; ADDITIONAL SECTION:
google.com 600 IN A 129.45.212.222

- Solution: Do not allow DNS servers to cache IP address mappings unless they are from authoritative name servers

Dig deeper?

DNS Cache Poisoning Test

https://www.grc.com/dns/dns.htm

DNSSEC: DNS

http://www.dnss

Quiz: DNS

- If a name server has no clue about where to find the address for a hostname then

A. Server asks server

B. Server asks its root name

C. Request is not processed

D. Server asks another name server in its domain

Quiz: DNS

- Which of the following is an example of a Top Level Domain?

A. yoda.jedi.st

B. jedi.starwars.com

C. starwars.com

D. .com

Quiz: DNS

- A web browser needs to contact www.cse.unsw.edu.au. The minimum number of DNS requests sent is:

A. 0

B. 1

C. 2

D. 3