Chapter 2: outline

2.1 principles of network applications

2.2 Web and HTTP

2.3 electronic mail

- SMTP, POP3, IMAP

2.4 DNS

2.5 P2P applications

2.6 video streaming and content distribution networks

2.7 socket programming with UDP and TCP

DNS: domain name system

people: many identifiers:

- SSN, name, passport #

Internet hosts, routers:

- IP address - used for addressing datagrams
  - 4 bytes or 32 bits
  - e.g., 129.23.4.51
  - used for routing
- "name",
  - e.g., www.uci.edu
  - used by humans
  - variable length

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)
  - runs on top of UDP, port 53
  - note: core Internet function, implemented as application-layer protocol
  - complexity at network's "edge"

DNS services

- hostname to IP address translation

```
nslookup (or host, dig, whois) athina.calit2.uci.edu

Name:       athina.calit2.uci.edu
Address:    128.195.177.83
```

- host aliasing: canonical vs. alias names

```
nslookup  (or dig) www.cnn.com
www.cnn.com   canonical name = www.cnn.com.vgtf.net
www.cnn.com.vgtf.net canonical name = cnn-56m.gslb.vgtf.net.
 Name:      cnn-56m.gslb.vgtf.net
Address:    157.166.249.11
Name:       cnn-56m.gslb.vgtf.net
Address:    157.166.248.10
```

- mail server aliasing

```
Nslookup –type=mx stanford.edu
Stanford.edu   mail exchanger = 40 mx1.stanford.edu.
stanford.edu   mail exchanger = 20 mx2.stanford.edu.
stanford.edu   mail exchanger = 20 mx3.stanford.edu.
```

- load distribution
  - replicated Web servers: set of IP addresses for one canonical name
  - rotating

```
nslookup (or dig) google.com

Name:       google.com
Address:    74.125.227.167
Name:       google.com
Address:    74.125.227.168
Name:       google.com
Address:    74.125.227.169
.....
```

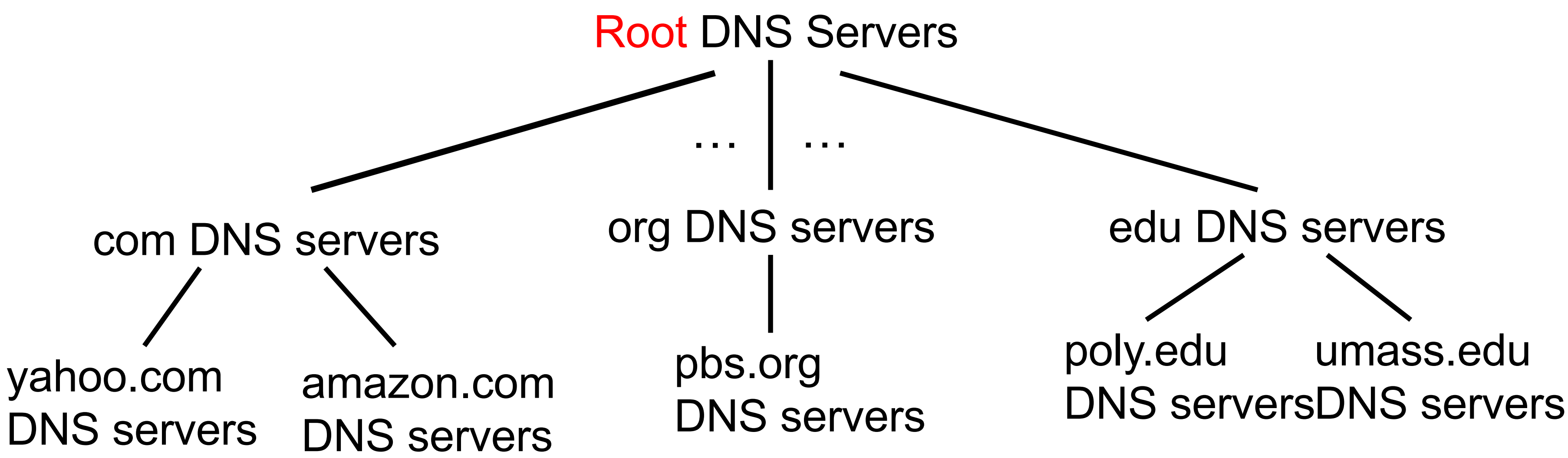
DNS structure

why not centralize DNS?

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

A: doesn't scale!

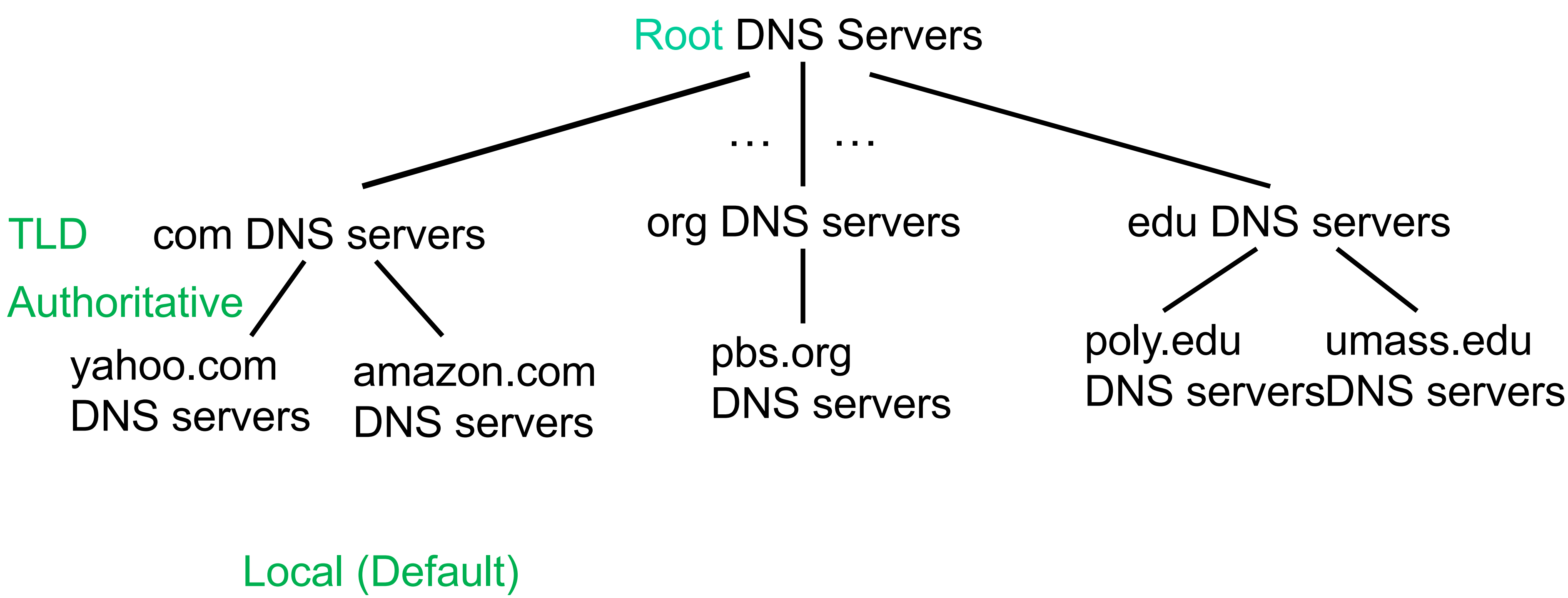
DNS: a distributed, hierarchical database



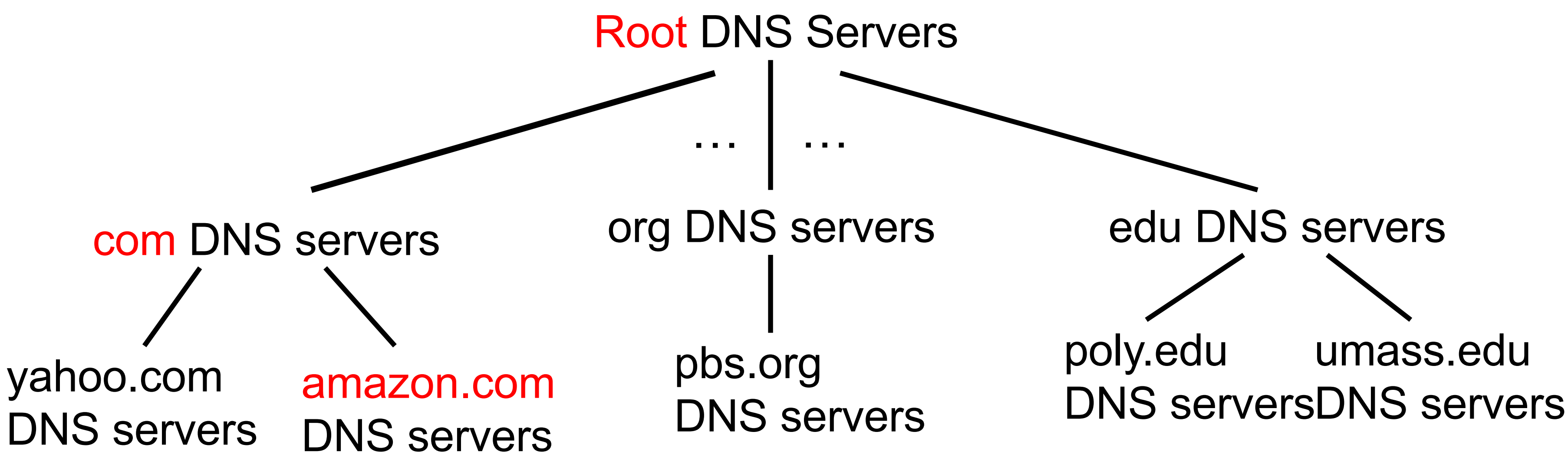
client wants IP for www.amazon.com; 1st approximation:

- 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

DNS: a distributed, hierarchical database



DNS: a distributed, hierarchical database

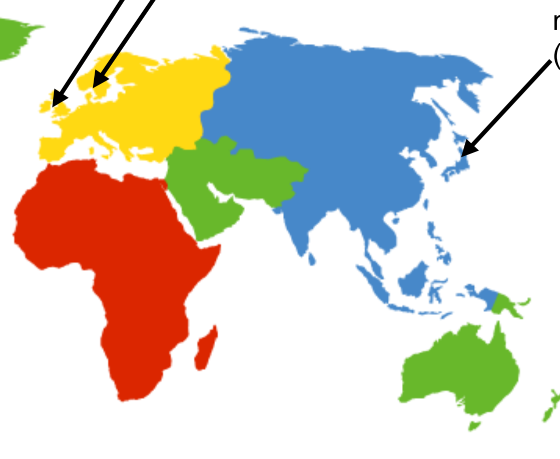


client wants IP for www.amazon.com; 1st approximation:

- 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

DNS: root name servers

- 13 root name servers worldwide: a, b…m
  - in fact replicated: 247 root servers as of 2011
  - https://www.iana.org/domains/root/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



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
  - http://www.iana.org/domains/root/db
- Network Solutions maintains servers for .com TLD
- Educause for .edu TLD: http://whois.educause.edu

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"

- 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
    - caches records

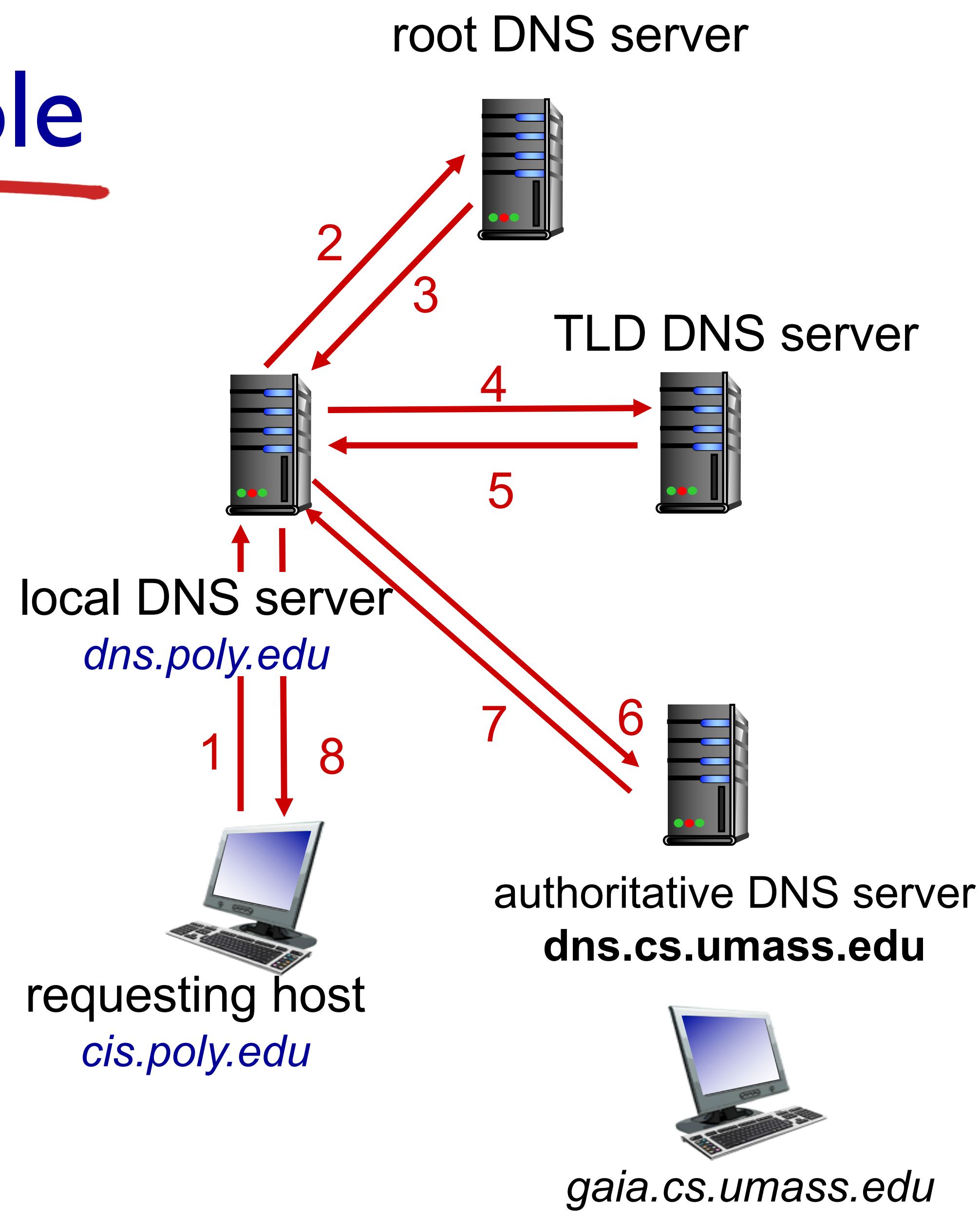
- Ex: more /etc/resolv.conf

DNS name resolution example

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

iterated query:

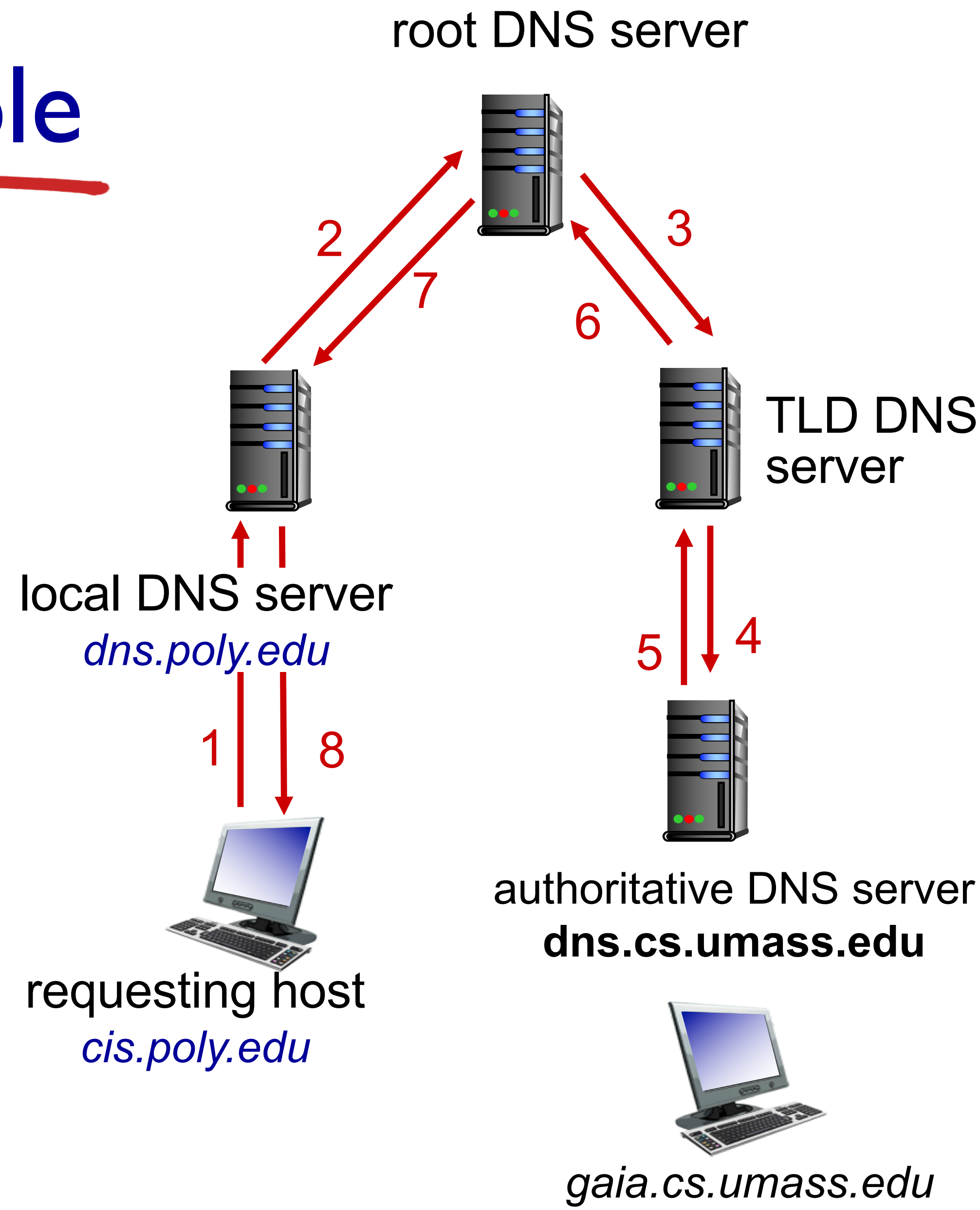
- contacted server replies with name of server to contact
- “I don’t know this name, but ask this server”



DNS name resolution example

recursive query:

- puts burden of name resolution on contacted name server
- heavy load at upper levels of hierarchy?



DNS: caching, updating records

- once (any) name server learns a mapping, it *caches it*
  - cache entries timeout (disappear) after some time (TTL)
    - Time-to-live (TTL) by default is 2 days
    - Needed because records change often
  - TLD servers typically cached in local name servers
    - thus root name servers are not visited often

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

- How to configure the records in the database
  - statically
  - update/notify mechanisms RFC 2136

DNS records -Summary

DNS: distributed database 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 hostname 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`**

DNS records

DNS: distributed db storing resource records (RR)

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

Type=A

- **`name`** is hostname
- **`value`** is IP address
- Stored at authoritative server of that domain

Example

- (**`odysseas.calit2.uci.edu, 128.195.185.112,`** A)
  - You can lookup this info (both directions)
    - by command line, e.g.: nslookup or dig or host
    - or on the web, e.g.
      - http://www.kloth.net/services/nslookup.php
      - http://www.iana.org/domains/root/db
      - http://whois.educause.edu
    - or in the old days: gethostbyname(), gethostbyaddr()

DNS records

DNS: distributed db storing resource records (RR)

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

Type=NS
- **`name`** is domain (e.g., foo.com)
- **`value`** is hostname of authoritative name server for this domain
- this record is used to route a request further

Example
- **`(uci.edu, ns1.service.uci.edu, NS)`**
  - type "`nslookup -ty=ns uci.edu`"
  - Or simply "`nslookup uci.edu`"

DNS records

DNS: distributed db storing resource records (RR)

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

Type=MX

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

Example

- **`(uci.edu, mta.service.uci.edu, MX)`**
  - type "`nslookup -ty=mx uci.edu`"
  - type "`nslookup -ty=mx stanford.edu`"
  - Can have multiple `NS` and `MS` records
  - several `MX` records, allow for load balancing

DNS records

DNS: distributed db storing resource records (RR)

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

Type=CNAME

- **`name`** is alias name for some "canonical" (the real) name
- **`value`** is canonical name

Example

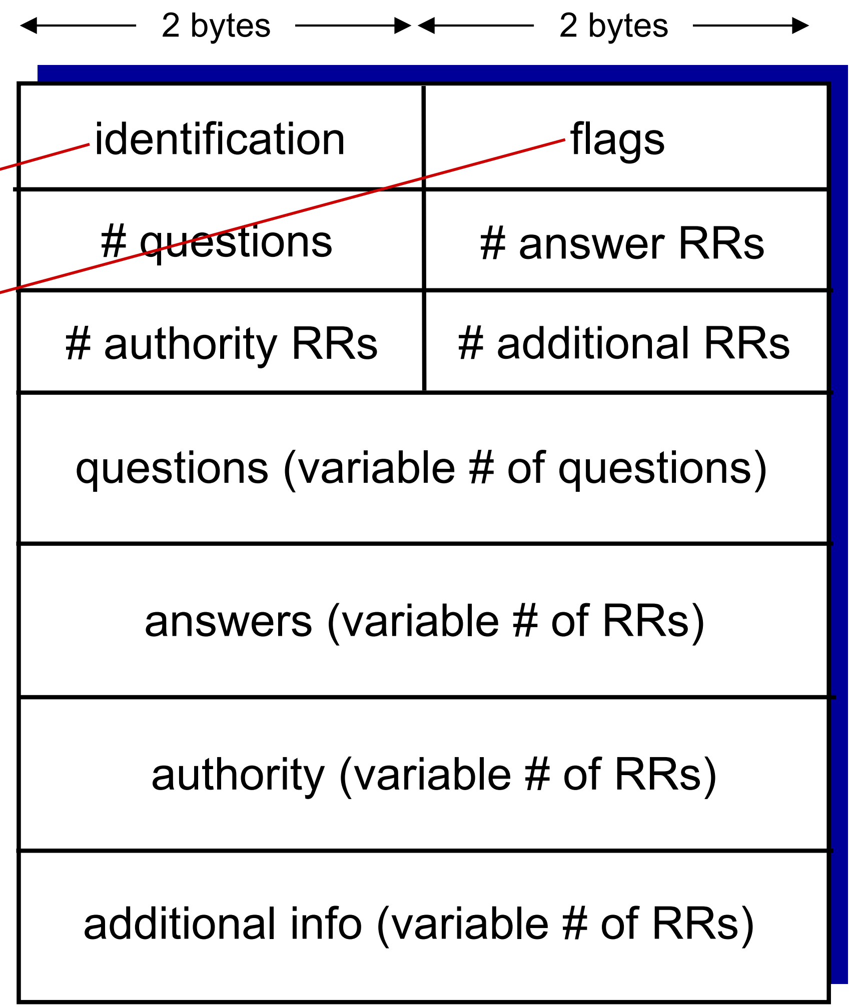
- `(www.networkedsystems.uci.edu, odysseas.calit2.uci.edu, CNAME)`
  - type "`nslookup -type=cname www.networkedsystems.uci.edu`"
  - Or simply "`nslookup www.networkedsystems.uci.edu`"
  - "`nslookup -type=cname www.ibm.com`"
  - alias, and potential for load balancing
  - a company can have the same alias for several servers…

DNS protocol, messages

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

message header

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



"dig" command returns those fields explicitly

DNS protocol, messages

2 bytes	2 bytes
identification	flags
# questions	# answer RRs
# authority RRs	# additional RRs
questions (variable # of questions)	
answers (variable # of RRs)	
authority (variable # of RRs)	
additional info (variable # of RRs)	

Name queried, query type — questions (variable # of questions)

(>=1) RRs in response to query — answers (variable # of RRs)

(>=1) records pointing to other authoritative servers — authority (variable # of RRs)

additional "helpful" info that may be used — additional info (variable # of RRs)

http://www.ietf.org/rfc/rfc1035.txt

Inserting records into DNS

- example: new startup "Network Utopia"
- register name networkuptopia.com at *DNS registrar(*)*
  - provide names, IP addresses of authoritative name servers (primary and secondary)
  - registrar inserts two RRs into .com TLD server:
    (networkutopia.com, dns1.networkutopia.com, NS)
    (dns1.networkutopia.com, 212.212.212.1, A)
- create authoritative server type A record for www.networkuptopia.com; type MX record for networkutopia.com

- (*) E.g. Network Solutions for .com, see www.internic.net for registrars accredited by ICANN)

Inserting records into DNS

- Example: new startup "Network Utopia"
- Register name networkuptopia.com at *DNS registrar* (e.g., Network Solutions, see www.internic.net for approved registrars by ICANN)
  - provide names, IP addresses of authoritative name server (primary and secondary), verifies uniqueness, puts into database for a small fee, acredited by ICANN
  - registrar inserts two RRs into .com TLD server:

```
(networkutopia.com, dns1.networkutopia.com, NS)
(dns1.networkutopia.com, 212.212.212.1, A)
(networkutopia.com, dns2.networkutopia.com, NS)
(dns2.networkutopia.com, 212.212.212.2, A)
```

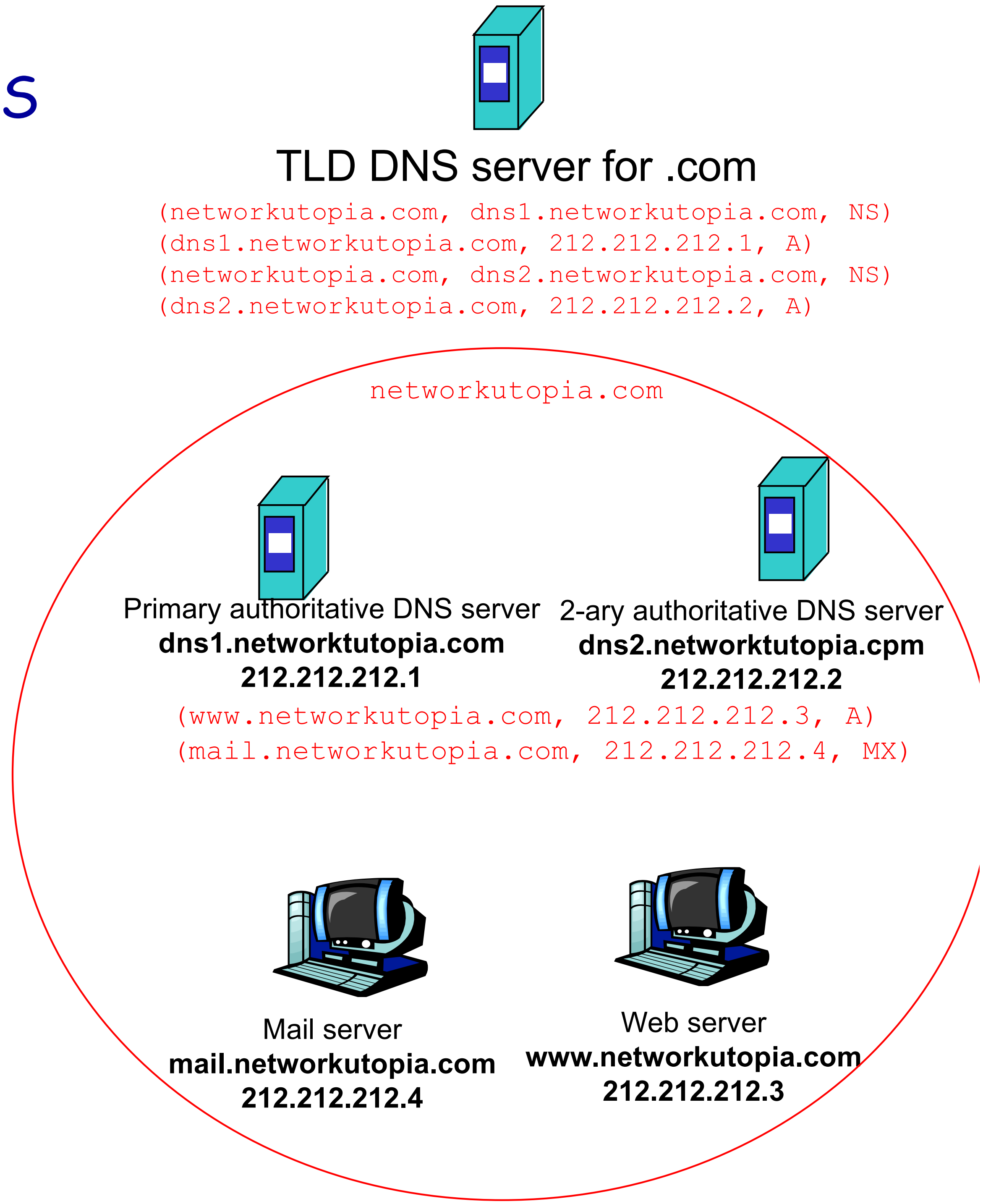
  - Create Type A record in your own authoritative server

```
(www.networkutopia.com, 212.212.212.3, A)
```

  - Create Type MX record in your own authoritative server

```
(mail.networkutopia.com, 212.212.212.4, MX)
```

Inserting records into DNS



Example cont'd: quering DNS records

- Q: How do people visit the website www.networkutopia.com?
- A:
  - Host: sends query to local DNS server
  - Local DNS server: asks TLD server [or root , if TLD not in cache]
  - TLD server: provides A and NS records for dns1.networkutopia.com

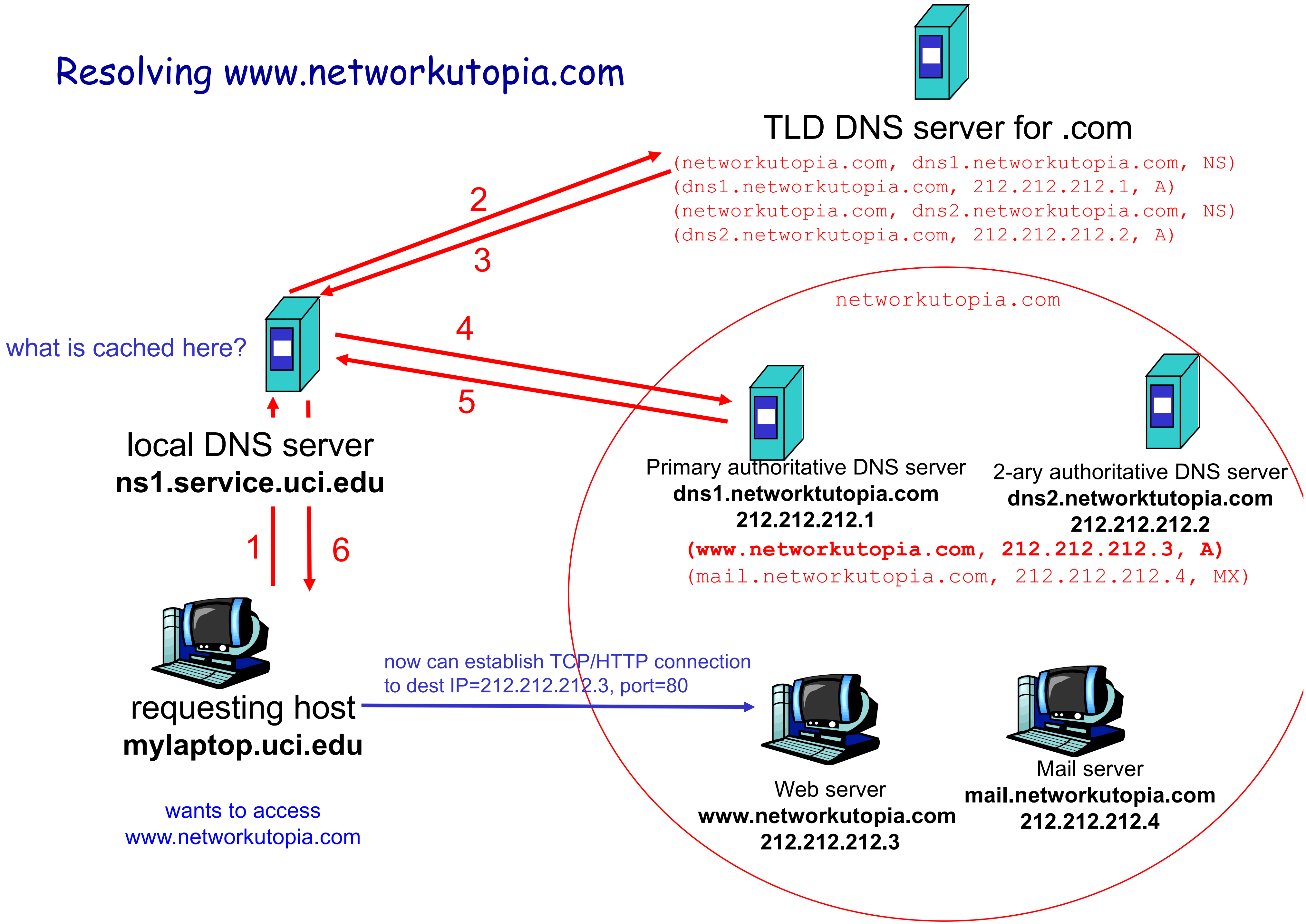
    ```
    (networkutopia.com, dns1.networkutopia.com, NS)
    (dns1.networkutopia.com, 212.212.212.1, A)
    ```

  - Local DNS server: sends query to authoritative server (212.212.212.1)
  - Authoritative name server: provides type A record

    ```
    (www.networkutopia.com, 212.212.212.3, A)
    ```

  - Local DNS server: returns this info to host (and caches RR for future use)
  - Host: establishes TCP/HTTP connection to (IP: 212.212.212.3, port 80)

Resolving www.networkutopia.com



DNS Load balancing

- DNS may return many RRs in same response
  - E.g. try "dig www.amazon.com" multiple times
- Clients:
  - by default, they pick the first one
  - could also choose not to – this part is not standard
- Order of multiple records: Unspecified
  - most often: Round Robin
  - or static or preference to numerically "closer"networks
  - or taking into account load or RTT, or other metric computed by the client or by other (non DNS) servers
- Some references
  - http://en.wikipedia.org/wiki/Domain_Name_System
  - http://en.wikipedia.org/wiki/Round-robin_DNS
  - RFC1794: http://tools.ietf.org/html/rfc1794
  - http://technet.microsoft.com/en-us/library/cc787484(v=ws.10).aspx

Attacking DNS

- ICANN: http://www.icann.org/

- Attacks against root servers (2002, 2007):
  - DNS root servers proved robust (to pings or queries).Traffic filtering, caching, anycast load balancing: http://www.icann.org/en/announcements/factsheet-dns-attack-08mar07_v1.1.pdf
  - DDoS attacks to TLD more dangerous
  - 98% of TLD DNS queries are invalid: http://www.caida.org/publications/papers/2008/root_internet/root_internet.pdf
- Redirect/Man in the middle
  - Cache poisoning: send bogus replies to DNS servers that cache
    - Using DNS to redirect traffic
      - 2008: Kaminsky vulnerability: http://www.unixwiz.net/techtips/iguide-kaminsky-dns-vuln.html
      - 2009: Twitter Blackout: http://www.theregister.co.uk/2009/12/18/dns_twitter_hijack/
      - 2015: Tesla's DNS hacked (CPS threat): http://www.tripwire.com/state-of-security/security-data-protection/teslas-dns-hacked-leads-website-and-twitter-hijack/
  - Intercept queries: E.g. to block access to Facebook in China

- Using DNS to launch DDoS attacks
  - Send requests with spoofed source address (target) – responses flood target
  - Requires amplification

DNS Summary

- Core Internet functionality

- Implemented as a network application
  - On top of UDP (or TCP) port 53
  - Defined in RFCs: 1034, 1035 (1987)
    - http://www.ietf.org/rfc/rfc1035.txt
  - Proposed by Mockapertis (UCI PhD 1982)
    - http://en.wikipedia.org/wiki/Paul_Mockapetris
  - Many extensions, e.g. DNSSEC

Practice

- Interactive Exercise:
  - Delay when browsing: DNS+TCP+HTTP:
    - https://gaia.cs.umass.edu/kurose_ross/interactive/DNS_HTTP_delay.php
- Interactive Animation
  - Recursive vs iterative DNS Queries
    - https://media.pearsoncmg.com/aw/ecs_kurose_compnetwork_7/cw/content/interactiveanimations/recursive-iterative-queries-in-dns/index.html
- Problems from the book