CSE 486/586 Distributed Systems

The Domain Name System

Slides by Steve Ko
Computer Sciences and Engineering
University at Buffalo

Domain Name System (DNS)

Proposed in 1983 by Paul Mockapetris

Separating Names and IP Addresses

- Names are easier (for humans!) to remember
  - www.cnn.com *vs.* 151.101.125.67
- IP addresses can change underneath
  - Move www.cnn.com to 54.84.95.176
  - *E.g.*, renumbering when changing providers
- Name could map to multiple IP addresses
  - www.cnn.com to multiple replicas of the Web site
- Map to different addresses in different places
  - Address of a nearby copy of the Web site
  - *E.g.*, to reduce latency, or return different content
- Multiple names for the same address
  - *E.g.*, aliases like www.cse.buffalo.edu and alfred.cse.buffalo.edu

Two Kinds of Identifiers

- Host name (e.g., www.cnn.com)
  - Mnemonic name appreciated by humans
  - Provides little (if any) information about location
  - Hierarchical, variable # of alpha-numeric characters
- IP address (*e.g.*, 151.101.21.67)
  - Numerical address appreciated by routers
  - Related to host's current location in the topology
  - IPv4: Hierarchical name space of 32 bits
  - IPv6: Hierarchical name space of 128 bits
    - *E.g.*, 2a04:4e42:5:0:0:0:0:323

Hierarchical Assignment Processes

- Host name: www.cse.buffalo.edu
  - Domain: registrar for each top-level domain (e.g., .edu)
  - Host name: local administrator assigns to each host
- IP addresses: 128.205.32.58
  - Prefixes: ICANN, regional Internet registries, and ISPs
  - Hosts: static configuration; or dynamic using DHCP, IPv6 autoconfig, *etc.*

Overview: Domain Name System

- A client-server architecture [2]
  - The server-side is distributed for scalability.
  - But the servers are still a hierarchy of clients and servers
- Computer science concepts underlying DNS:
  - Indirection: names in place of addresses
  - Hierarchy: in names, addresses, and servers
  - Caching: of mappings from names to/from addresses
- DNS software components
  - DNS resolvers
  - DNS servers
- DNS queries
  - Iterative queries
  - Recursive queries
- DNS caching based on time-to-live (TTL)

Strawman Solution #1: Local File

- Actually used as the original name to address mapping
  - Flat namespace
  - /etc/hosts
  - SRI kept the main copy
  - Remote sites downloaded it regularly
- The count of hosts began increasing rapidly
  - One machine per domain → one machine per user
  - Many more downloads
  - Many more updates
  - The hosts file just didn't scale!

Strawman Solution #2: Central Server

- Central server
  - One place where all mappings are stored
  - All queries go to the central server
- Many practical problems
  - Single point of failure
  - High traffic volume
  - Distant, centralized database
  - Single point of update
  - Does not scale

Need a distributed, hierarchical collection of servers

Domain Name System (DNS)

- Properties of DNS
  - Hierarchical name space divided into zones
  - Distributed over a collection of DNS servers
- DNS servers form a hierarchy
  - Root servers
  - Top-level domain (TLD) servers
  - Authoritative DNS servers
- Name-to-address translations performed near the user
  - Local DNS servers
  - Resolver software

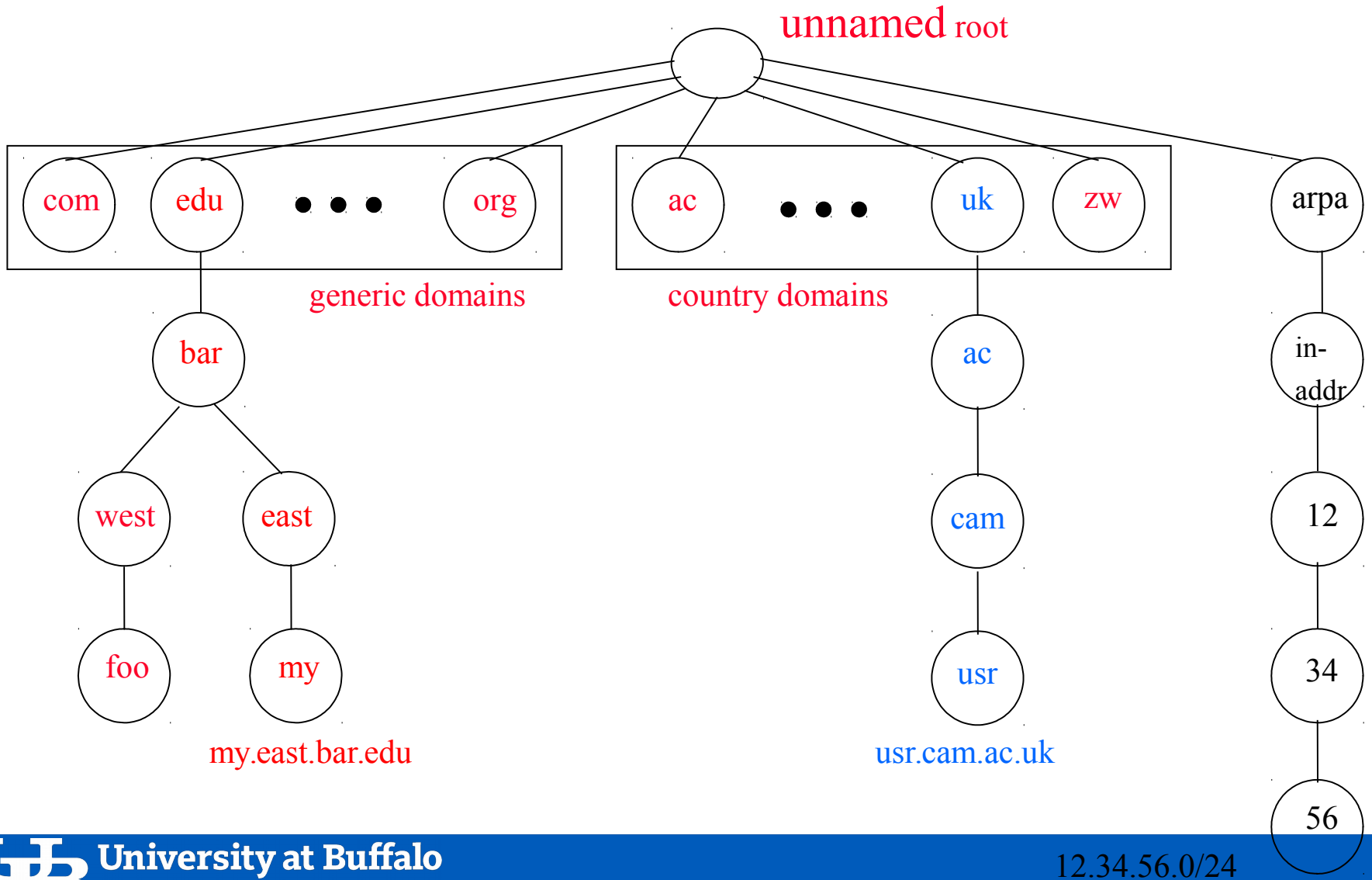
DNS Root Servers

- 13 root servers (see http://www.root-servers.org/)
- Labeled A through M

TLD and Authoritative DNS Servers

- Top-level domain (TLD) servers
  - Generic domains (*e.g.*, com, org, edu)
  - Country domains (*e.g.*, uk, fr, ca, jp)
  - Now custom TLDs (*e.g.*, google, संगठन, 移动)
  - Typically managed professionally
    - » Network Solutions maintains servers for “com”
    - » Educause maintains servers for “edu”
- Authoritative DNS servers
  - Provide public records for hosts at an organization
  - For the organization’s servers (*e.g.*, Web and mail)
  - Can be maintained locally or by a service provider

Distributed, Hierarchical Database

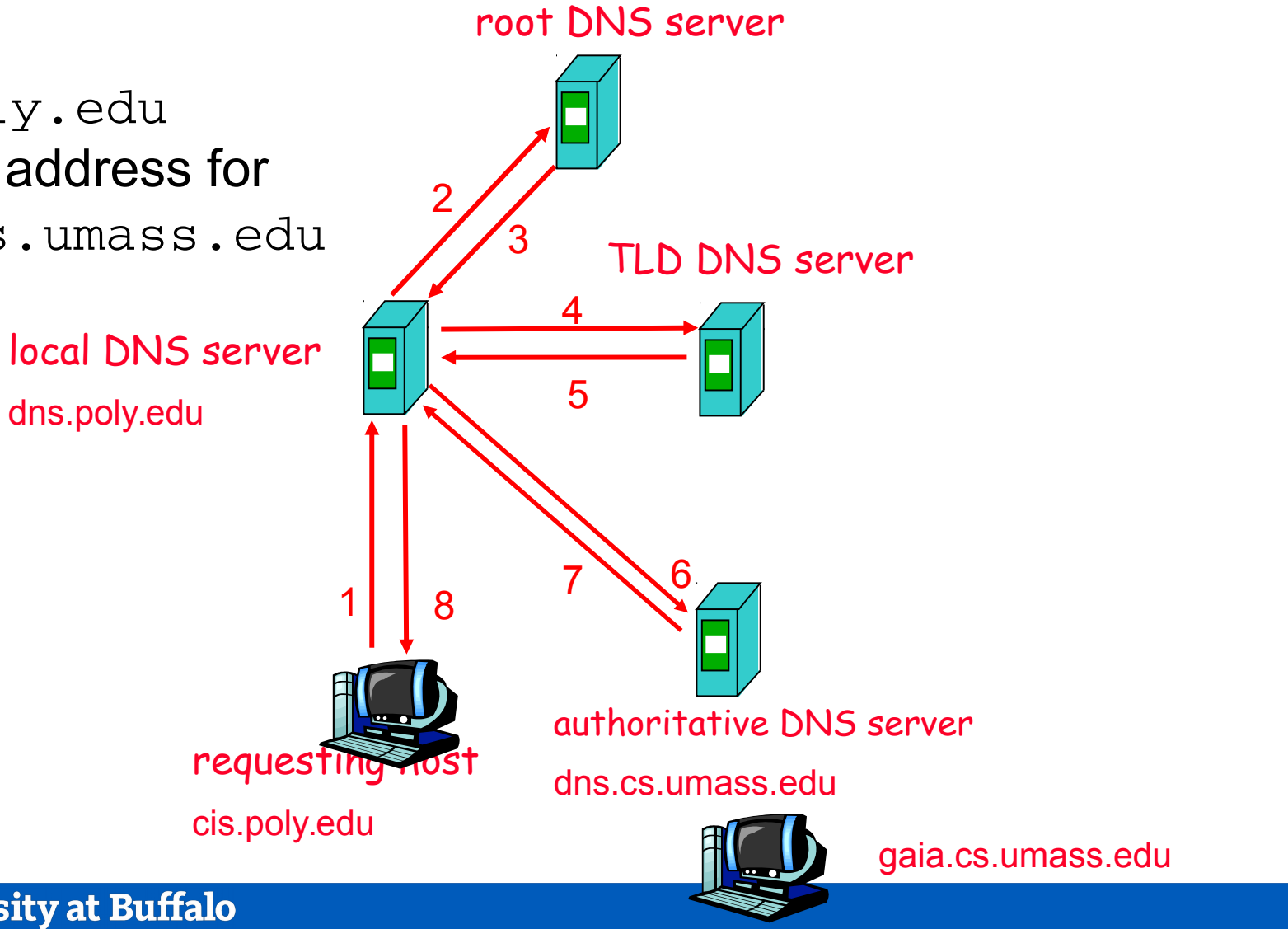


Using DNS

- Local DNS server ("default name server")
  - Usually near the end hosts who use it
  - Local hosts configured with local server
  - *e.g.*, /etc/resolv.conf or learn the server via DHCP
- Client application
  - Extract server name (*e.g.*, from the URL)
  - Do *gethostbyname()* to trigger resolver code
- Server application
  - Extract client IP address from socket
  - Optional *gethostbyaddr()* to translate into name

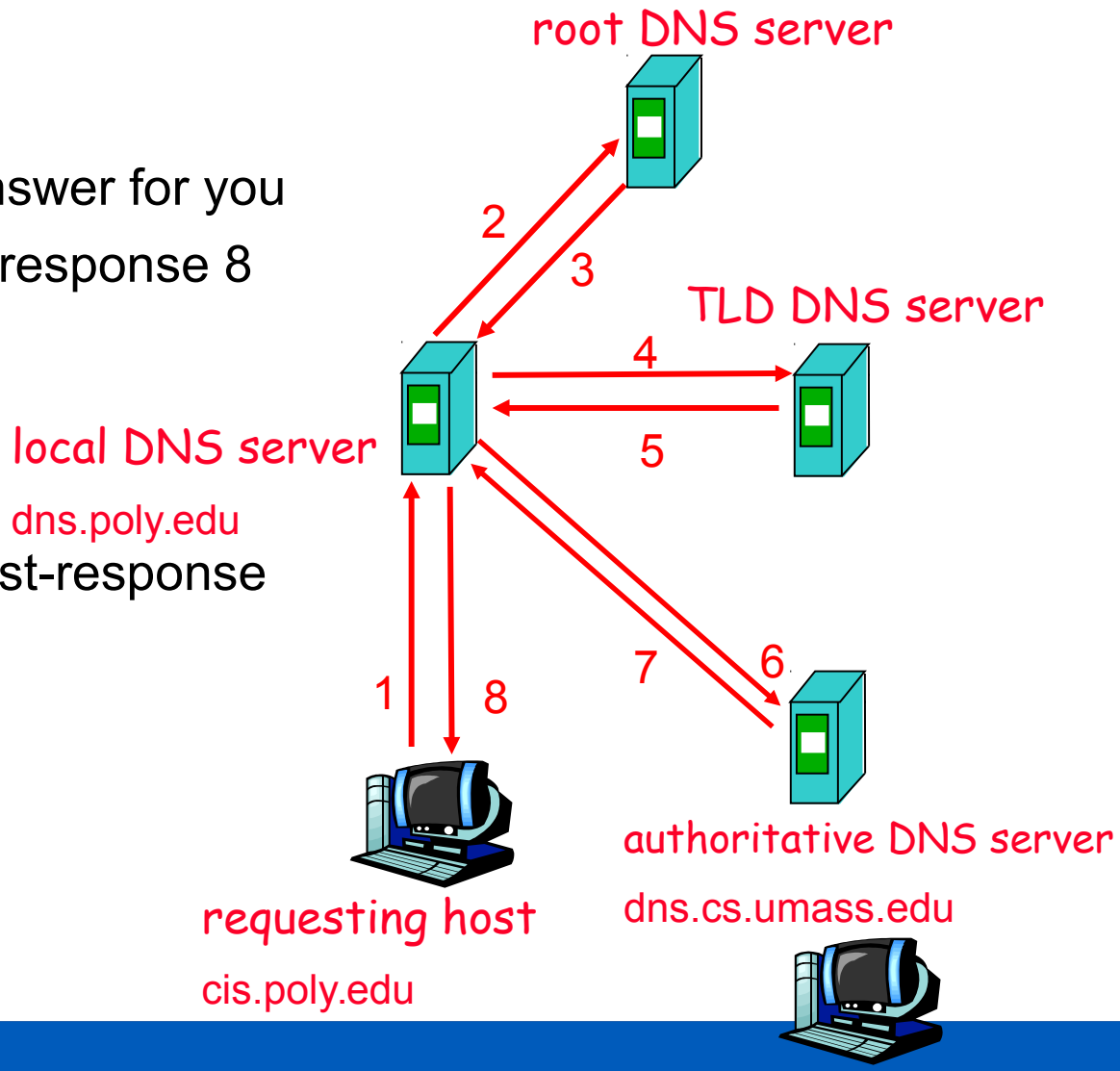
Example

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



Recursive vs. Iterative Queries

- Recursive query
  - Ask server to get answer for you
  - *E.g.*, request 1 and response 8
- Iterative query
  - Ask server who to ask next
  - *E.g.*, all other request-response pairs



DNS Caching

- Performing all these queries take time
  - And all this before the actual communication takes place
  - *E.g.*, 1-second latency before starting Web download
- Caching can substantially reduce overhead
  - The top-level servers very rarely change
  - Popular sites (*e.g.*, www.cnn.com) visited often
  - Local DNS server often has the information cached
- How DNS caching works
  - DNS servers cache responses to queries
  - Responses include a “time to live” (TTL) field
  - Server deletes the cached entry after TTL expires

Negative Caching

- Remember things that don't work
  - Misspellings like www.cnn.comm and www.cnnn.com
  - These can take a long time to fail the first time
  - Good to remember that they don't work
  - … so the failure takes less time the next time around

DNS Resource Records

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
  - Value is hostname of authoritative nameserver for the domain
- Type CNAME
  - Name is alias for some canonical name; *e.g.*, www.cse.buffalo.edu is a CNAME pointer for alfred.cse.buffalo.edu
- Type MX
  - Value is a name of a mailserver associated with the name

Reliability

- DNS servers are replicated
  - Name service available if at least one replica is up
  - Queries can be load balanced between replicas
- UDP is often used for queries
  - Reliability must be implemented on top of UDP
- Try alternate servers on timeout
  - Exponential backoff when retrying same server
- The same identifier is used for all queries
  - The client doesn't care which server responds

Inserting Resource Records into DNS

- Example: just created startup “FooBar”
- Register foobar.com at a DNS registrar for com
  - Provide the registrar with the names and IP addresses of your authoritative name servers (primary and secondary)
  - Registrar inserts two RRs into the com TLD server:
    - (foobar.com, dns1.foobar.com, NS)
    - (dns1.foobar.com, 212.212.212.1, A)
- Put records in the authoritative server dns1.foobar.com
  - Type A record for www.foobar.com
  - Type MX record for foobar.com
- Play with “dig” on UNIX

```
$ dig nytimes.com ANY

;; ANSWER SECTION:
nytimes.com.        300 IN  SOA ns1.p24.dynect.net. hostmaster.nytimes.com.
   2017091238 300 150 1209600 300
nytimes.com.        300 IN  NS  ns4.p24.dynect.net.
nytimes.com.        300 IN  NS  ns3.p24.dynect.net.
nytimes.com.        300 IN  NS  ns2.p24.dynect.net.
nytimes.com.        300 IN  NS  ns1.p24.dynect.net.
nytimes.com.        500 IN  A   151.101.1.164
nytimes.com.        500 IN  A   151.101.193.164
nytimes.com.        500 IN  A   151.101.129.164
nytimes.com.        500 IN  A   151.101.65.164
nytimes.com.        300 IN  MX  5 ALT1.ASPMX.L.GOOGLE.com.
nytimes.com.        300 IN  MX  10 ASPMX2.GOOGLEMAIL.com.
nytimes.com.        300 IN  MX  5 ALT2.ASPMX.L.GOOGLE.com.
nytimes.com.        300 IN  MX  10 ASPMX3.GOOGLEMAIL.com.
nytimes.com.        300 IN  MX  1 ASPMX.L.GOOGLE.com.
nytimes.com.        500 IN  TXT "253961548-4297453"
nytimes.com.        500 IN  TXT "v=spf1 mx ptr ip4:170.149.160.0/19
   ip4:209.11.220.51/32 include:alerts.wallst.com include:authsmtp.com
   include:sendgrid.net include:_spf.google.com include:inyt.com
   include:_spf.e.sparkpost.com ~all"
```

```
$ dig nytimes.com +norec @a.root-servers.net

;; QUESTION SECTION:
;nytimes.com.                   IN   A

;; AUTHORITY SECTION:
com.            172800  IN  NS  e.gtld-servers.net.
com.            172800  IN  NS  b.gtld-servers.net.
com.            172800  IN  NS  j.gtld-servers.net.
com.            172800  IN  NS  m.gtld-servers.net.
com.            172800  IN  NS  i.gtld-servers.net.
com.            172800  IN  NS  f.gtld-servers.net.
com.            172800  IN  NS  a.gtld-servers.net.
com.            172800  IN  NS  g.gtld-servers.net.
com.            172800  IN  NS  h.gtld-servers.net.
com.            172800  IN  NS  l.gtld-servers.net.
com.            172800  IN  NS  k.gtld-servers.net.
com.            172800  IN  NS  c.gtld-servers.net.
com.            172800  IN  NS  d.gtld-servers.net.
```

```
$ dig nytimes.com +norec @k.gtld-servers.net

;; QUESTION SECTION:
;nytimes.com.                   IN   A

;; AUTHORITY SECTION:
nytimes.com.            172800  IN   NS   ns3.p24.dynect.net.
nytimes.com.            172800  IN   NS   ns1.p24.dynect.net.
nytimes.com.            172800  IN   NS   ns2.p24.dynect.net.
nytimes.com.            172800  IN   NS   ns4.p24.dynect.net.

;; ADDITIONAL SECTION:
ns3.p24.dynect.net. 172800  IN   A    208.78.71.24
ns1.p24.dynect.net. 172800  IN   A    208.78.70.24
ns2.p24.dynect.net. 172800  IN   A    204.13.250.24
ns4.p24.dynect.net. 172800  IN   A    204.13.251.24

;; Query time: 71 msec
;; SERVER: 2001:503:d2d::30#53(2001:503:d2d::30)
;; WHEN: Thu Apr 19 15:50:21 EDT 2018
;; MSG SIZE  rcvd: 190
```

```
$ dig nytimes.com ANY +norec @ns1t.nytimes.com

;; ANSWER SECTION:
nytimes.com.            300 IN  SOA ns1.p24.dynect.net. hostmaster.nytimes.com.
2017091238 300 150 1209600 300
nytimes.com.            300 IN  NS  ns1.p24.dynect.net.
nytimes.com.            300 IN  NS  ns2.p24.dynect.net.
nytimes.com.            300 IN  NS  ns4.p24.dynect.net.
nytimes.com.            300 IN  NS  ns3.p24.dynect.net.
nytimes.com.            500 IN  A   151.101.1.164
nytimes.com.            500 IN  A   151.101.193.164
nytimes.com.            500 IN  A   151.101.129.164
nytimes.com.            500 IN  A   151.101.65.164
nytimes.com.            300 IN  MX  5 ALT2.ASPMX.L.GOOGLE.COM.
nytimes.com.            300 IN  MX  5 ALT1.ASPMX.L.GOOGLE.COM.
nytimes.com.            300 IN  MX  1 ASPMX.L.GOOGLE.COM.
nytimes.com.            300 IN  MX  10 ASPMX3.GOOGLEMAIL.COM.
nytimes.com.            300 IN  MX  10 ASPMX2.GOOGLEMAIL.COM.
nytimes.com.            500 IN  TXT "google-site-
verification=jZcmQFxPEP38yqYpmRvo0v_9hQFAdBZPUEBwTNUPUF8"
nytimes.com.            500 IN  TXT "253961548-4297453"
nytimes.com.            500 IN  TXT "adobe-idp-site-verification=5ce4d99c-af0a-4b7
9217-bd49d3336df0"
nytimes.com.            500 IN  CAA 0 issue "comodoca.com"
```

Summary

- DNS is a distributed client-server architecture
- Why have DNS?
  - Names are easier (for us!) to remember
  - IP addresses can change while names remain stable
  - Name could map to multiple IP addresses
  - Map to different addresses in different places
  - Multiple names for the same address
- Properties of DNS
  - Distributed over a collection of DNS servers
- Hierarchy of DNS servers
  - Root servers, top-level domain (TLD) servers, authoritative DNS servers

References

[1] Textbook sections 13.1-13.2. **Required Reading.**

[2] Paul Mockapetris. *Domain Names – Concepts and Facilities.* RFC 1034. November 1987. https://www.rfc-editor.org/rfc/rfc1034.txt

[3] Paul Mockapetris and Kevin Dunlap. *Development of the Domain Name System.* Proceedings of ACM SIGCOMM. August 1988. http://ccr.sigcomm.org/archive/1995/jan95/ccr-9501-mockapet.pdf

Acknowledgements

- These slides by Steve Ko, lightly modified and used with permission by Ethan Blanton.
- These slides contain material developed and copyrighted by Indranil Gupta (UIUC), Michael Freedman (Princeton), and Jennifer Rexford (Princeton).