Application Layer

CS5700 Fall 2019

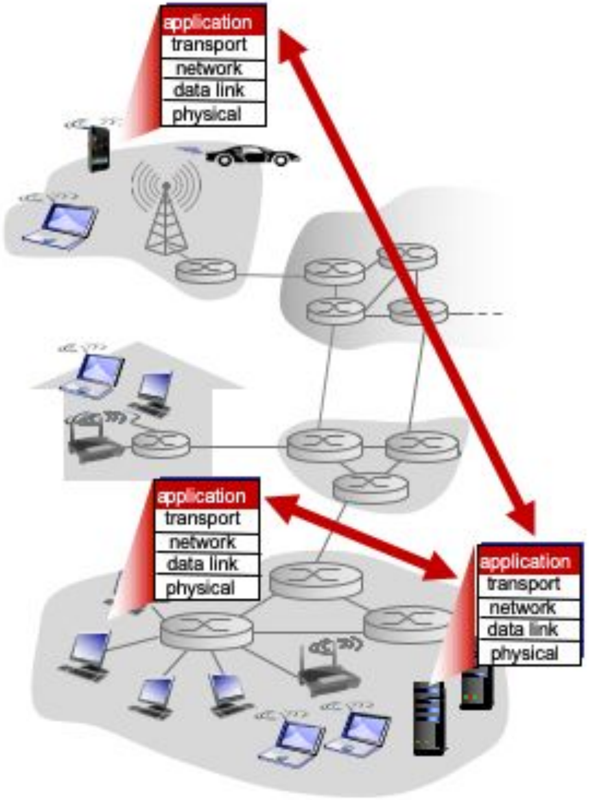
Agenda

- Principles of network applications
- DNS
- Web and HTTP
- SMTP
- CDN

Principles of network applications

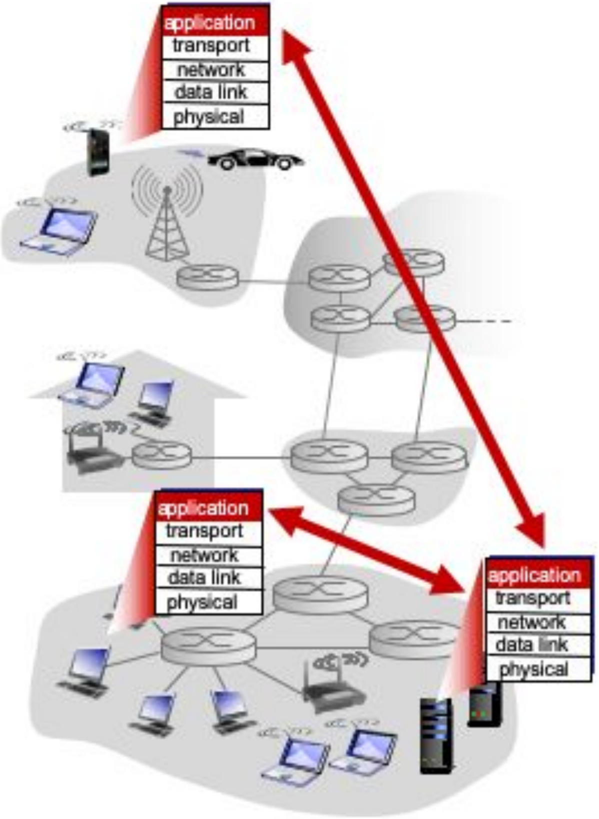
Principle - intelligence at the edge

- Internet does not provide services. It only provides communication.
- Application programs provide all services.



Principle - intelligence at the edge

- Write application programs that
  - Run on hosts
  - Communicate over network
- No need to change network core
  - Network core devices do not run user applications



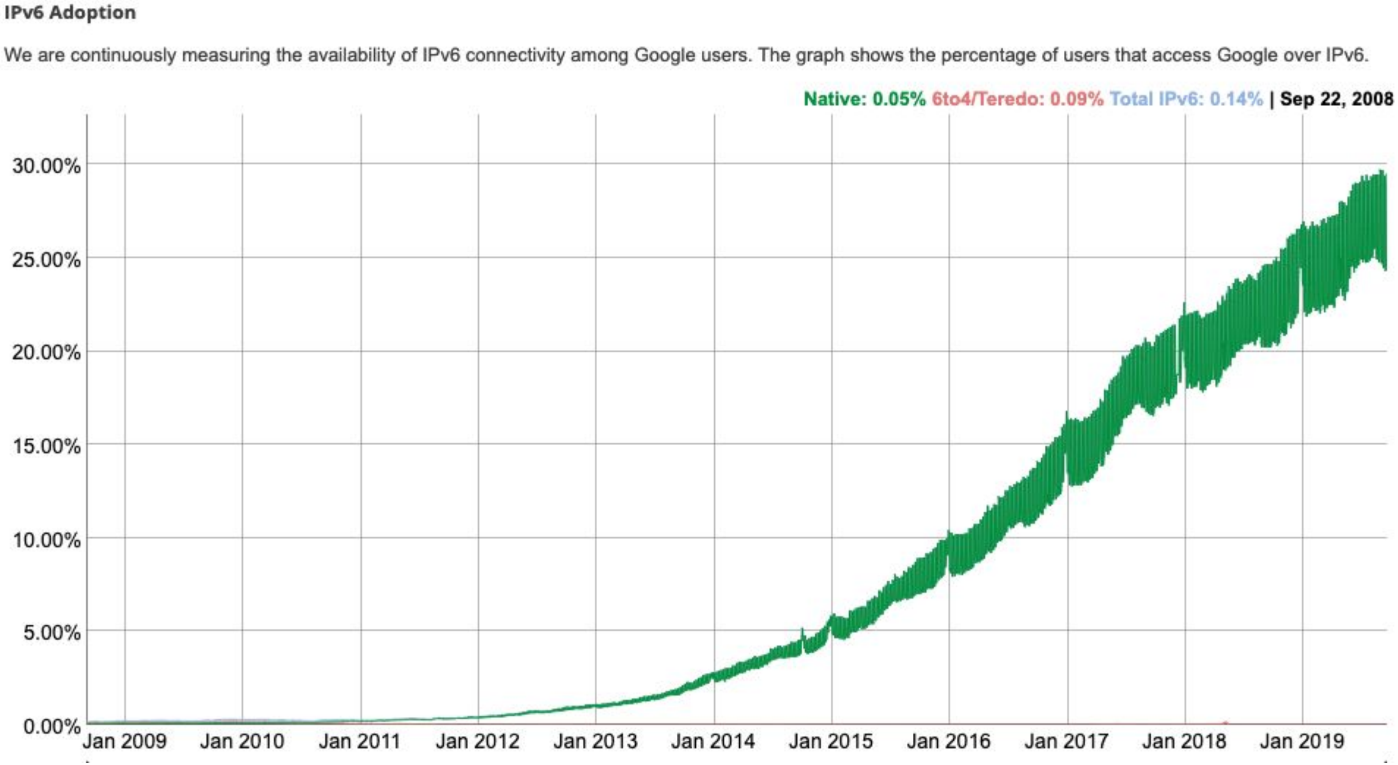
Principle - intelligence at the edge

- Web
- Email
- Network games
- Streaming videos (Youtube, Netflix, Hulu, etc.)
- Realtime video conferencing
- Social networking
- …
- All require no change at the network core

Why is it a good principle?

How long does it take us to adopt IPv6?

How long does it take us to adopt IPv6?

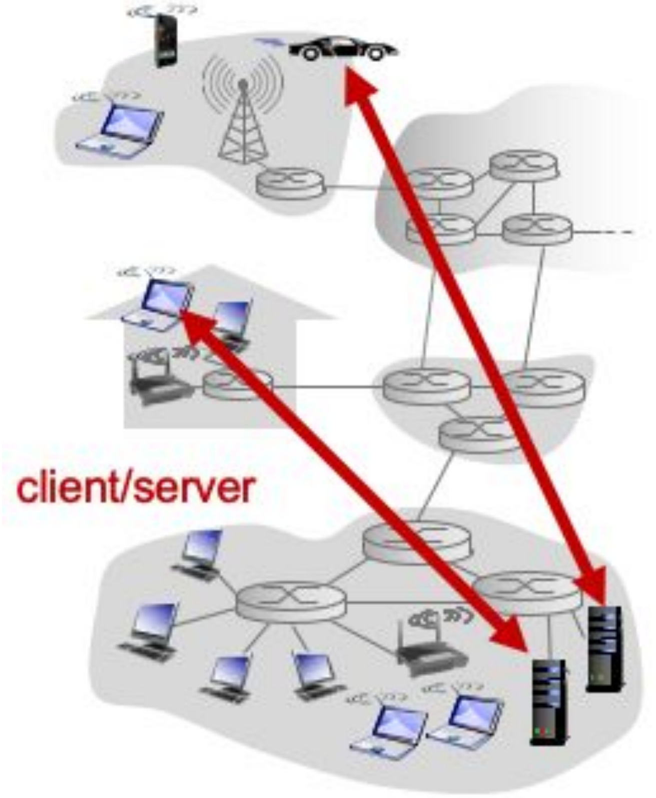


IPv6 Adoption

We are continuously measuring the availability of IPv6 connectivity among Google users. The graph shows the percentage of users that access Google over IPv6.

Client-server architecture

- Server
  - Always-on host
  - Permanent IP address
  - Data center for scaling
- Client
  - Communicate with server
  - May have dynamic IP address
  - Do not communicate directly with each other



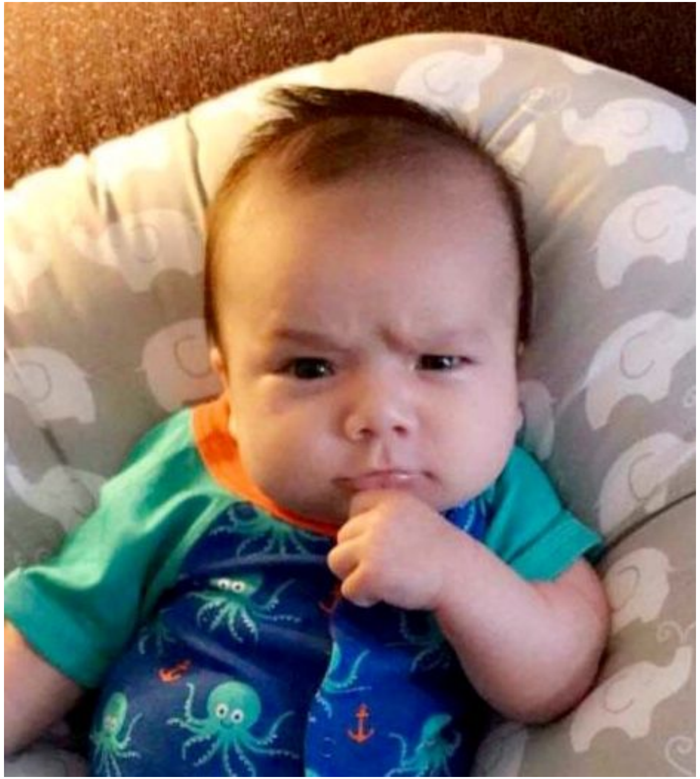
Transport layer service model - TCP

- Reliable data transfer
  - No loss, in-order
- Flow control: sender won’t overwhelm receiver
- Congestion control: throttle sender when network is overloaded
- Connection oriented: setup required between client and server

Transport layer service model - UDP

- Unreliable data transfer
  - Loss, out-of-order, duplicate
- That’s it!

Any service you'd like transport layer to have?



Other important services

- Timing (aka bounded latency)
  - E.g. Internet telephony, interactive games
- Throughput
  - E.g. multimedia
- Security
  - Encryption, data integrity, etc.
- …
- None of the above is provided in transport layer! :(

DNS

DNS - domain name system

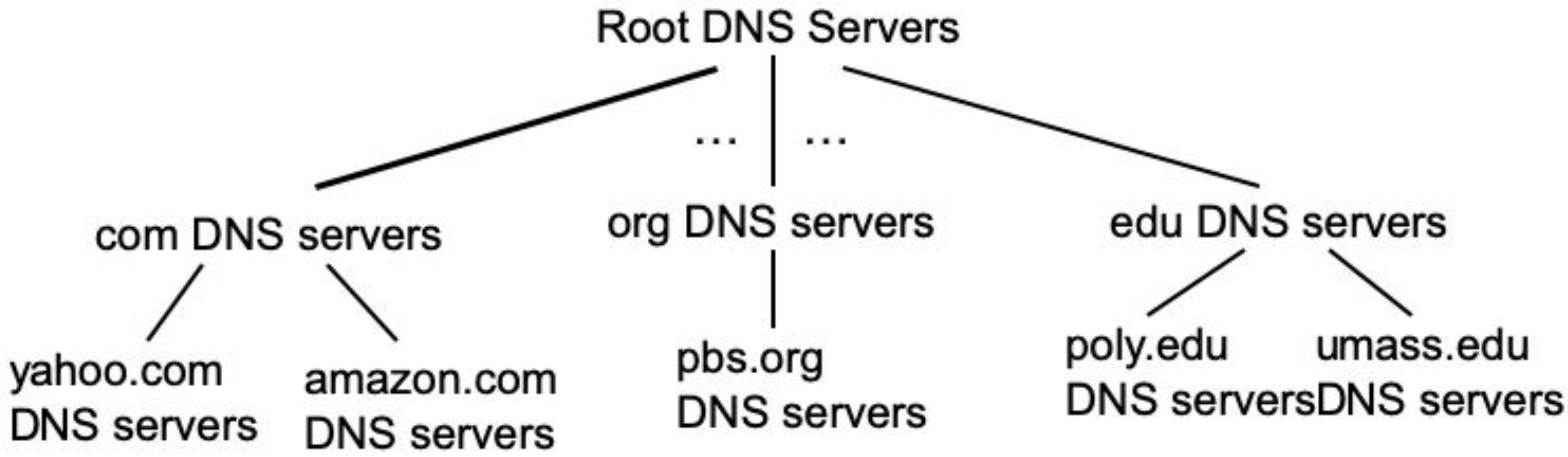
- Important piece of Internet infrastructure
- Runs at the application layer
- Translate human-readable names into IP addresses
- Distributed database
    - Centralized DNS doesn’t scale!

DNS

- Names are hierarchical
- Each name divided into segments by period char
  - Read as “dot”
- Most significant segment is on the right
- Rightmost segment known as a top-level domain (TLD)
- E.g. neu.edu

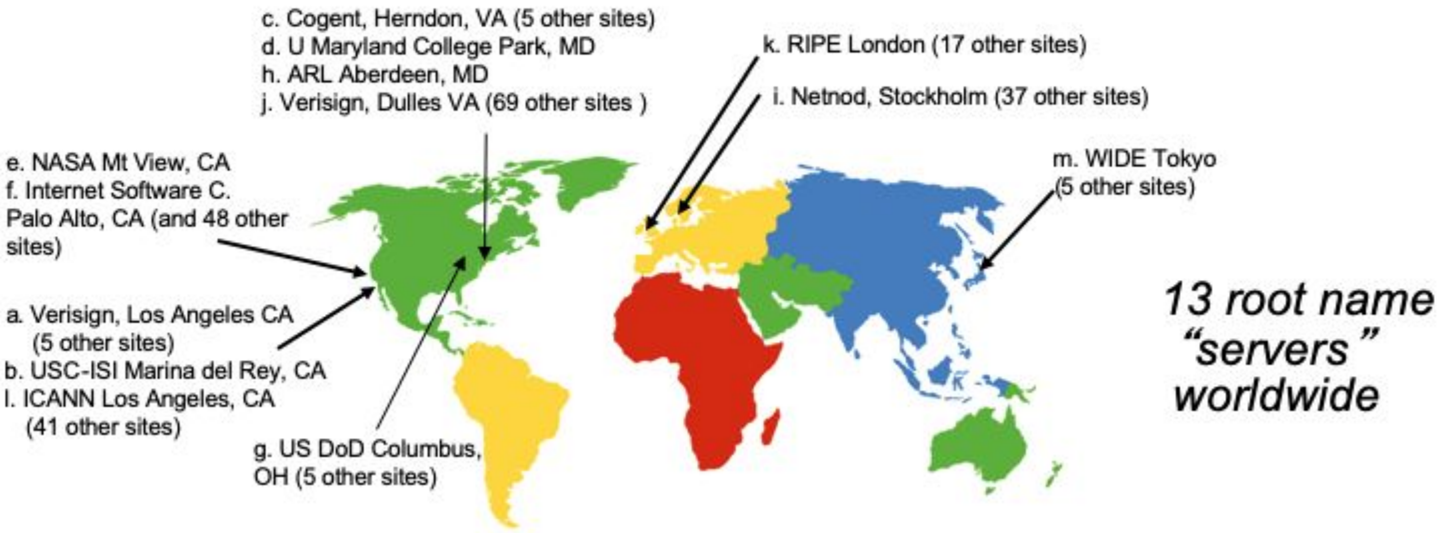
DNS - hierarchical database

- How do you get IP address for www.neu.edu?



DNS - root name servers

- 13 logical root name servers. ([a-m].root-servers.net)
- Provide which TLD name server to ask next


c. Cogent, Herndon, VA (5 other sites)
d. U Maryland College Park, MD
h. ARL Aberdeen, MD
j. Verisign, Dulles VA (69 other sites)
k. RIPE London (17 other sites)
i. Netnod, Stockholm (37 other sites)
e. NASA Mt View, CA
f. Internet Software C.
Palo Alto, CA (and 48 other
sites)
m. WIDE Tokyo
(5 other sites)
a. Verisign, Los Angeles CA
(5 other sites)
b. USC-ISI Marina del Rey, CA
l. ICANN Los Angeles, CA
(41 other sites)
g. US DoD Columbus,
OH (5 other sites)
13 root name
"servers"
worldwide

DNS - TLD name servers

- Responsible for com, org, net, edu, …, and all top-level country domains
- Provide which authoritative name server to ask next

DNS - authoritative name servers

- Organization’s own name servers
- Provide authoritative hostname to IP mappings for organization’s named hosts

Summary so far...

- How many DNS queries you need?
  - 1 for root name server
  - 1 for TLD name server
  - 1 for authoritative name server
- Is there any issue?

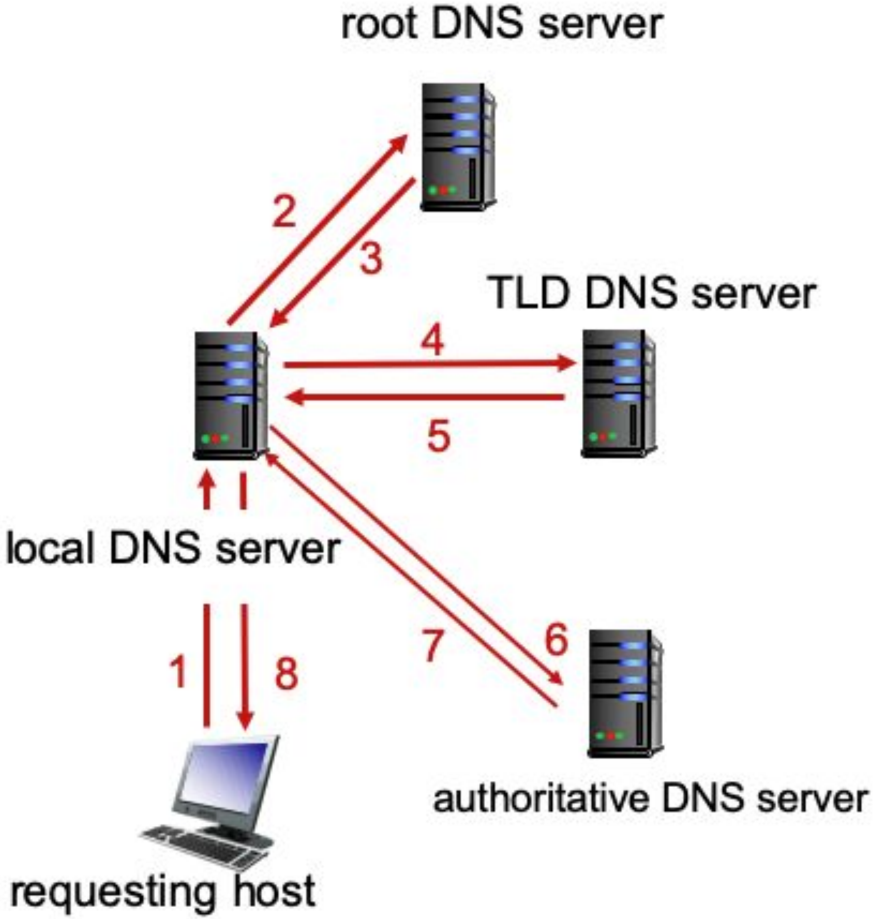
Too slow!!

DNS - local name server

- Does not belong to hierarchy
- Each ISP (residential ISP, company, university) has one
- When host makes DNS query, query is sent to its local name server
    - Acts as proxy, forwards query into hierarchy
    - Has local cache of recent name-to-address map

Put all together

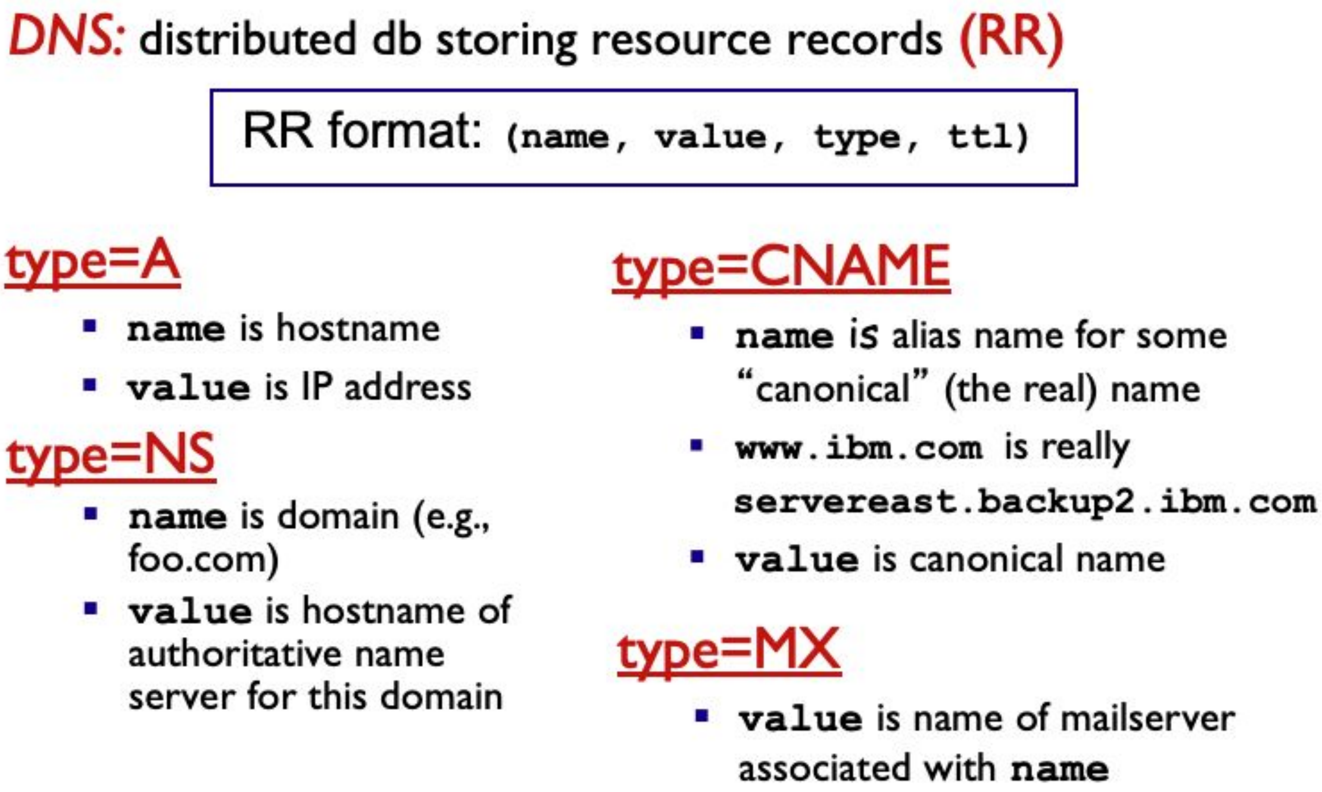
- Can you see this is more efficient?



DNS - caching

- Cache entries timeout after TTL
    - What is reasonable TTL? Who decide?
- TLD name servers typically cached in local name servers
    - Thus root name servers not often visited
- Cached entries may be out-of-date

DNS records



DNS: distributed db 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

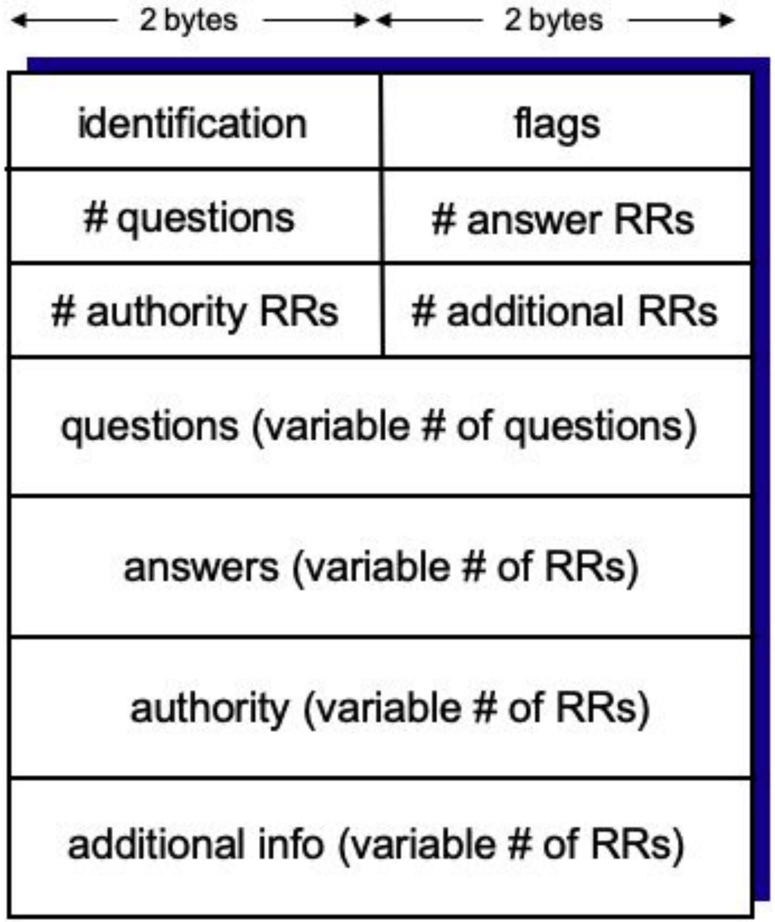
- **`name`** is alias name for some “canonical” (the real) name
- **`www.ibm.com`** is really **`servereast.backup2.ibm.com`**
- **`value`** is canonical name

type=MX

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

DNS - message format

- Both query and reply messages have the same format
- Flags:
  - Query or reply
  - Recursion desired
  - Recursion available
  - Reply is authoritative



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)	

DNS

- Is DNS using TCP or UDP as transport layer protocol?

Demo wireshark

Inserting records into DNS

- New startup “Network Utopia”
- Register name networkutopia.com at DNS registrar
  - Provide names, IP addresses of authoritative name server (primary and secondary)
  - Registrar inserts two RRs into .com TLD name server (networkutopia.com, dns1.networkutopia.com, NS) (dns1.networkutopia.com, 10.1.1.1, A)
- Create type A record for www.networkutopia.com in authoritative name server.