

# NAMING AND THE DOMAIN NAME SYSTEM (DNS)

George Porter  
Module 1  
Fall 2020



# ATTRIBUTION

- These slides are released under an Attribution-NonCommercial-ShareAlike 3.0 Unported (CC BY-NC-SA 3.0) Creative Commons license
- These slides incorporate material from:
  - Michael Freedman and Kyle Jamieson, Princeton University (also under a CC BY-NC-SA 3.0 Creative Commons license)
  - Andrew Moore, Univ. of Cambridge
  - Computer Networks, 6<sup>th</sup> ed

# DNS HOSTNAME VERSUS IP ADDRESS

- **DNS host name** (e.g. www.cs.ucsd.edu)
  - **Mnemonic** name appreciated by humans
  - **Variable length**, full alphabet of characters
  - Provides **little** (if any) information about **location**
- **IP address** (e.g. 128.112.136.35)
  - Numerical address appreciated by **routers**
  - **Fixed length**, decimal number
  - **Hierarchical** address space, related to host **location**

# MANY USES OF DNS

- Hostname to IP address translation
  - IP address to hostname translation (reverse lookup)
- Host name aliasing: other DNS names for a host
  - Alias host names point to canonical hostname
- **Email:** Lookup domain's mail server by domain name

# ORIGINAL DESIGN OF DNS

- Per-host file named /etc/hosts (1982)
  - Flat namespace: each line = IP address & DNS name
  - SRI (Menlo Park, California) kept the master copy
  - Everyone else downloads regularly
- *But, a single server doesn't scale*
  - Traffic implosion (lookups and updates)
  - Single point of failure
- Need a distributed, hierarchical **collection** of servers

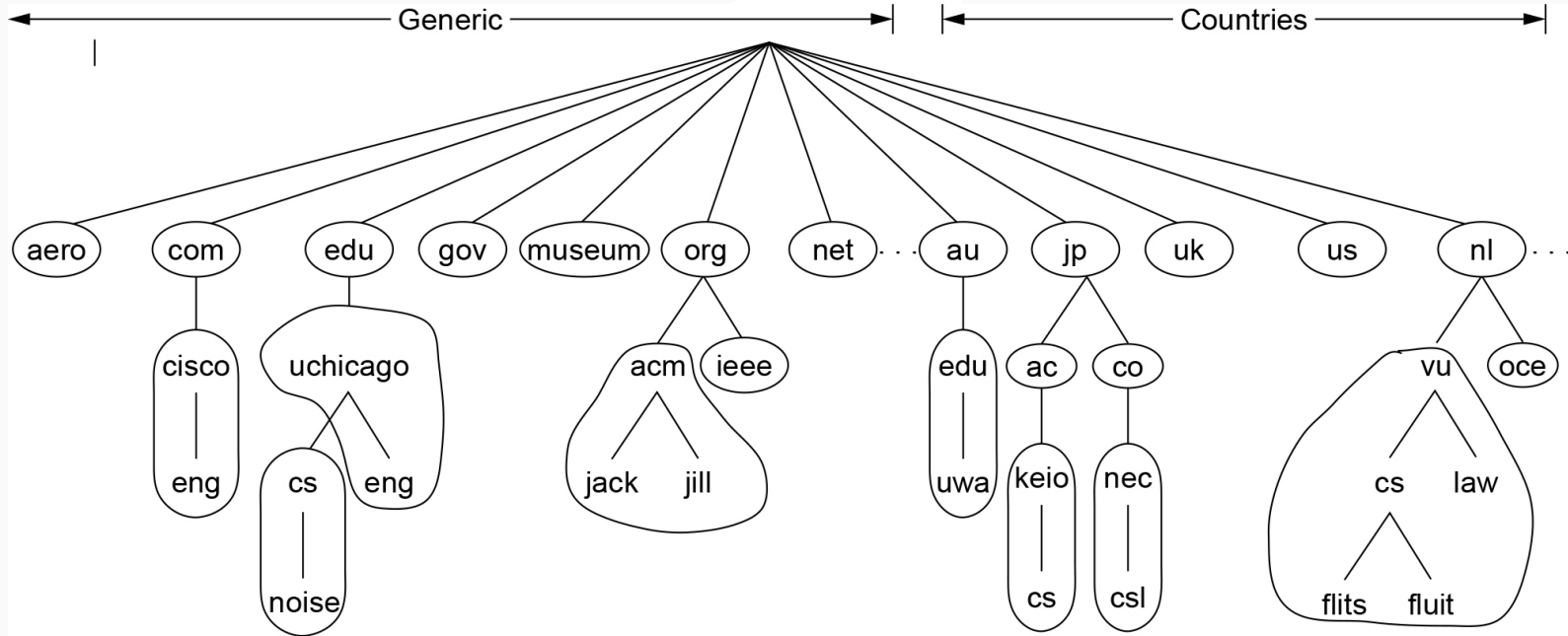
# DNS: GOALS AND NON-GOALS

- A wide-area **distributed database**
- Goals:
  - **Scalability**; decentralized maintenance
  - **Robustness**
  - Global scope
    - Names mean the same thing everywhere
  - Distributed updates/queries
  - Good **performance**

# DOMAIN NAME SYSTEM (DNS)

- Hierarchical name space divided into contiguous sections called **zones**
  - Zones are distributed over a collection of DNS servers
- Hierarchy of DNS servers:
  - **Root** servers (identity hardwired into other servers)
  - **Top-level domain (TLD)** servers
  - **Authoritative** DNS servers
- Performing the translations:
  - **Local DNS servers** located near clients
  - **Resolver** software running on clients

# DNS IS HIERARCHICAL

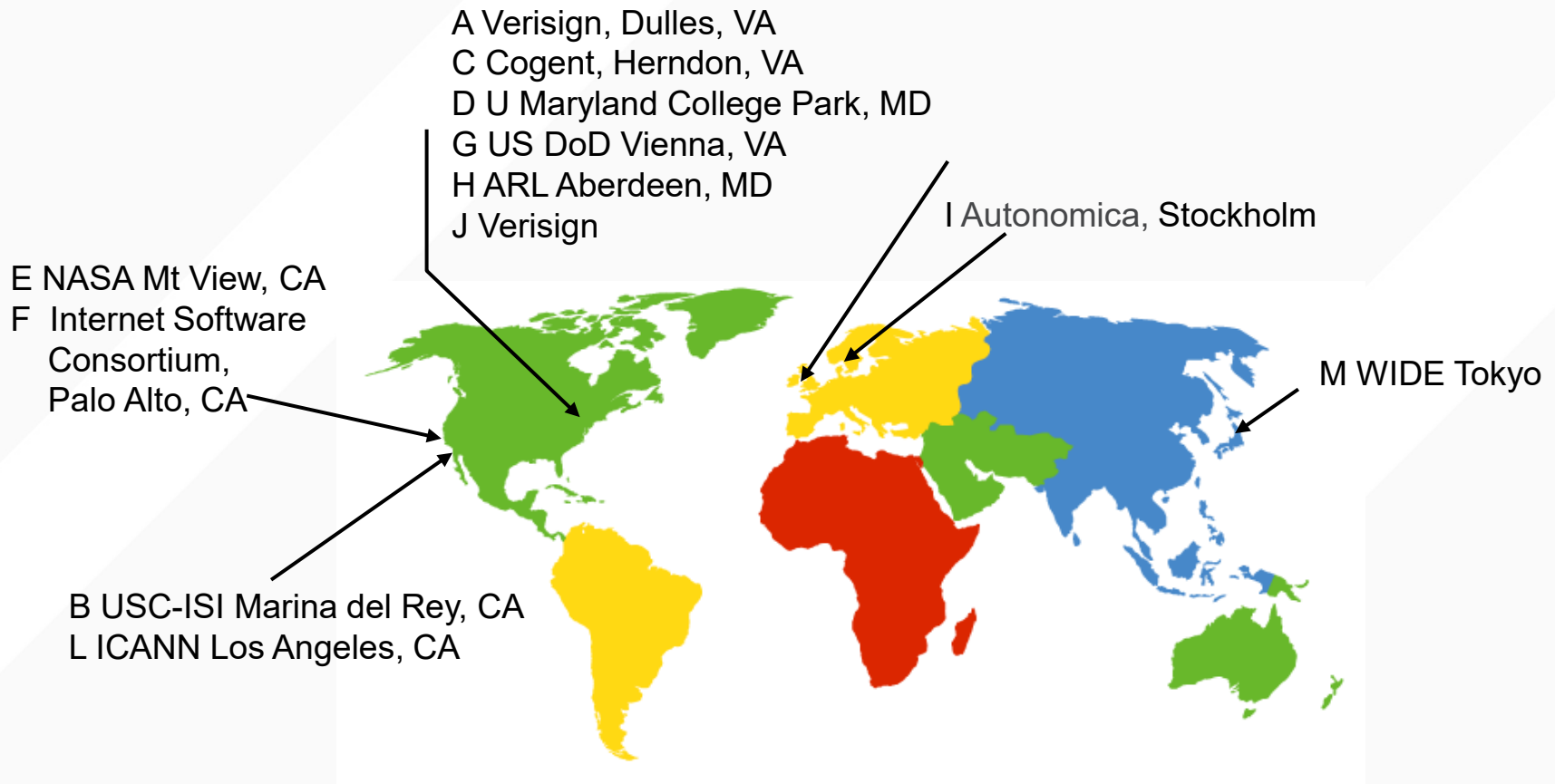


- Hierarchy of namespace matches hierarchy of servers
- Set of nameservers answers queries for names within zone
- Nameservers store names and links to other servers in tree



# DNS ROOT NAMESERVERS

- 13 root servers



# TLD AND AUTHORITATIVE SERVERS

- <ftp://ftp.internic.net/domain/named.root>
- Top-level domain (TLD) servers
  - Responsible for com, org, net, edu, etc, and all top-level country domains: uk, fr, ca, jp
  - Network Solutions maintains servers for com TLD
  - Educause non-profit for edu TLD
- Authoritative DNS servers
  - An organization's DNS servers, providing authoritative information for that organization
  - May be maintained by organization itself, or ISP

# COMMON TLDS

Domain	Intended use	Start date	Restricted?
com	Commercial	1985	No
edu	Educational institutions	1985	Yes
gov	Government	1985	Yes
int	International organizations	1988	Yes
mil	Military	1985	Yes
net	Network providers	1985	No
org	Non-profit organizations	1985	No
aero	Air transport	2001	Yes
biz	Businesses	2001	No
coop	Cooperatives	2001	Yes
info	Informational	2002	No
museum	Museums	2002	Yes
name	People	2002	No
pro	Professionals	2002	Yes
cat	Catalan	2005	Yes
jobs	Employment	2005	Yes
mobi	Mobile devices	2005	Yes
tel	Contact details	2005	Yes
travel	Travel industry	2005	Yes
xxx	Sex industry	2010	No

# LOCAL NAME SERVERS

- Do not strictly belong to hierarchy
- Each ISP (or company, or university) has one
  - Also called **default** or **caching** name server
- When host makes DNS query, query is sent to its local DNS server
  - Acts as proxy, forwards query into hierarchy
  - Does work for the client

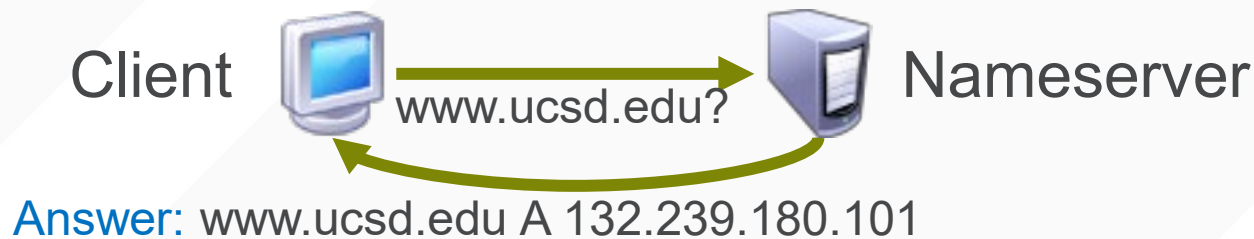
# DNS RESOURCE RECORDS

- DNS is a distributed database storing **resource records**
- Resource record includes: (**name**, type, **value**, time-to-live)

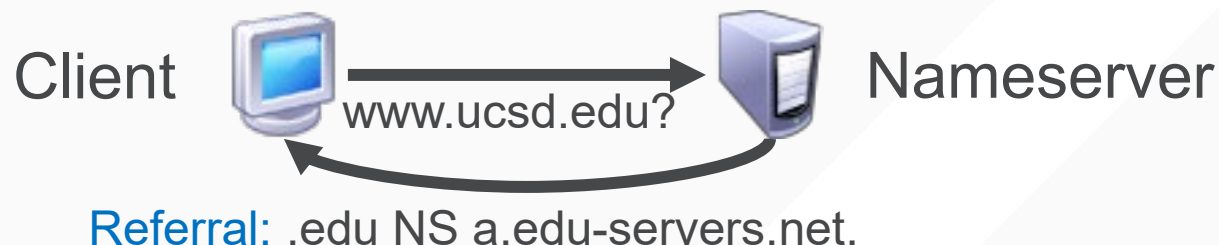
Type	Meaning	Value
SOA	Start of authority	Parameters for this zone
A	IPv4 address of a host	32-Bit integer
AAAA	IPv6 address of a host	128-Bit integer
MX	Mail exchange	Priority, domain willing to accept email
NS	Name server	Name of a server for this domain
CNAME	Canonical name	Domain name
PTR	Pointer	Alias for an IP address
SPF	Sender policy framework	Text encoding of mail sending policy
SRV	Service	Host that provides it
TXT	Text	Descriptive ASCII text

# DNS IN OPERATION

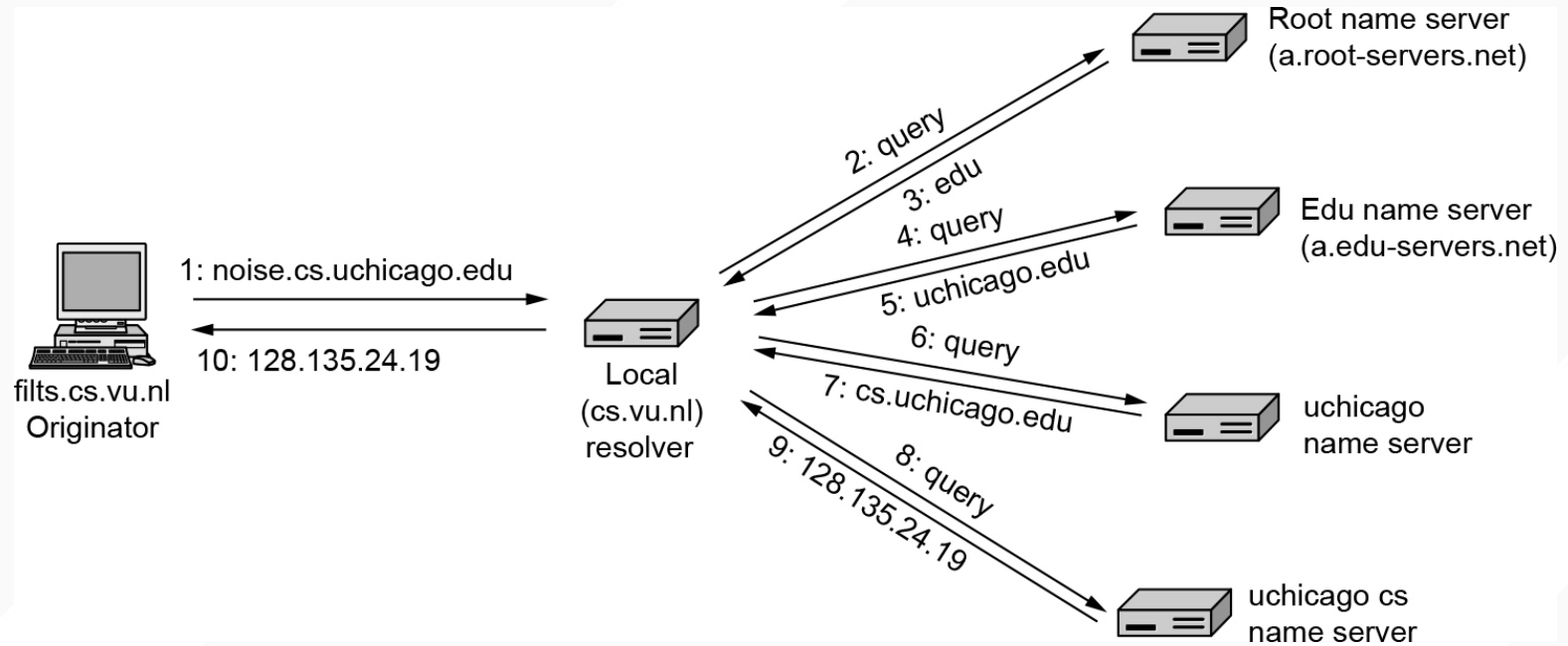
- Most queries and responses are UDP datagrams
- Two types of queries:
- **Recursive:** Nameserver responds with answer or error



- **Iterative:** Nameserver may respond with a referral



# ITERATIVE LOOKUP



# DNS CACHING

- Performing all these queries takes time
  - And all this **before actual communication** takes place
- Caching can **greatly reduce overhead**
  - The top-level servers very rarely change
    - Popular sites visited often
  - Local DNS server often has the information cached
- How DNS caching works
  - All DNS servers **cache responses to queries**
  - Responses include a time-to-live (TTL) field
    - Server deletes cached entry after TTL expires



# JULIA EVAN'S GUIDE TO *DIG*



## dig

JULIA EVANS  
@b0rk

dig makes  
DNS queries!

```
$ dig google.com
```

```
google.com 208 IN A  
           ↑ TTL  
ip address! → 172.217.13.110
```

dig *TYPE* domain.com

this lets you choose which  
DNS record to query for!

types to try: *SRV* default  
*MX* *TXT* *AAAA* *A*

dig @8.8.8.8 domain  
 ↑ Google DNS server  
dig @server lets you  
pick which DNS server  
to query! Useful to  
check if your system  
DNS is misbehaving ☺

dig *+trace* domain

traces how your domain  
gets resolved, starting  
at the root nameservers

dig *-x* 172.217.13.174

makes a reverse  
DNS query - find  
which domain resolves  
to an IP!

dig *+short* domain

Usually dig prints lots of  
output! With +short it  
just prints the IP address/  
value of the DNS record

UC San Diego