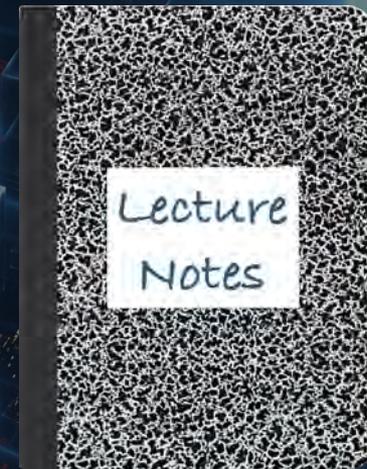


CS 417 – DISTRIBUTED SYSTEMS

**Week 7:** Decentralized Lookup  
Part 3: Domain Name System (DNS)



Paul Krzyzanowski

© 2026 Paul Krzyzanowski. No part of this content may be reproduced or reposted in whole or in part in any manner without the permission of the copyright owner.

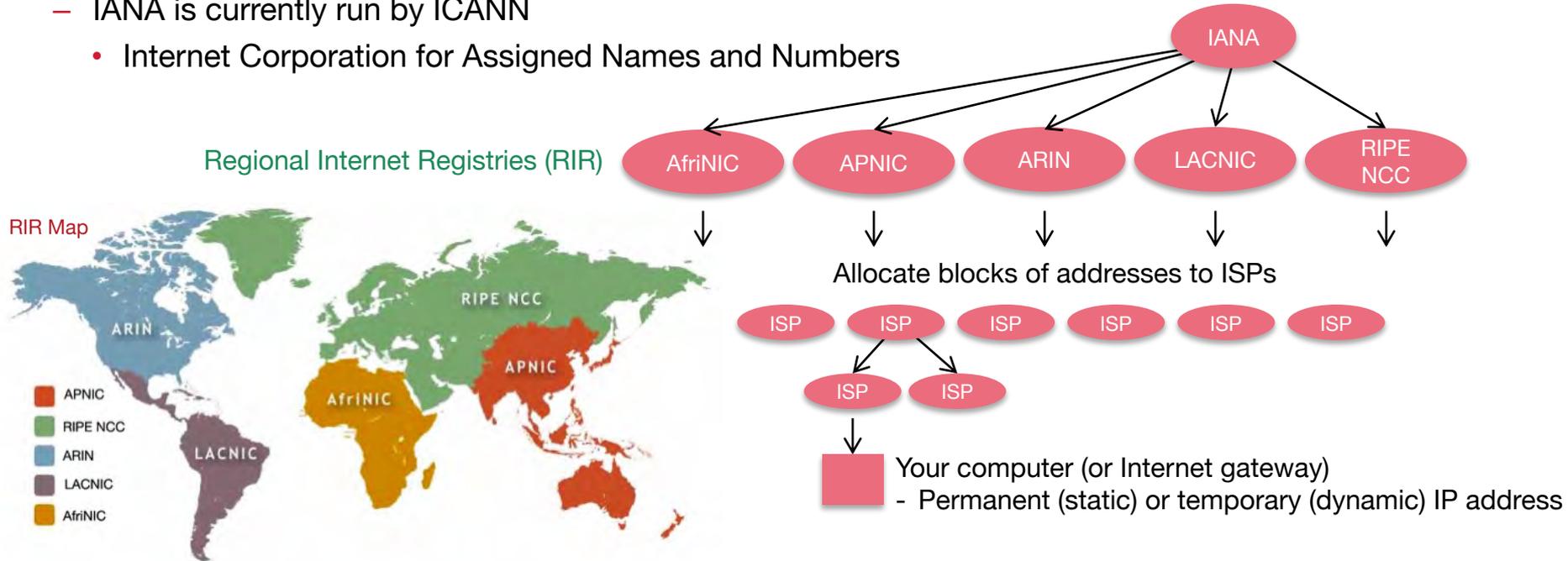
# How are IP addresses assigned?

IP addresses are distributed hierarchically

Internet Assigned Numbers Authority (IANA) at the top

– IANA is currently run by ICANN

- Internet Corporation for Assigned Names and Numbers



# How are machine names assigned?

## Early ARPANET

- Globally unique names per machine (e.g., UCBVAX)
- Kept track at the Network Information Center (NIC) at the Stanford Research Institute (SRI)

## *That doesn't scale!*

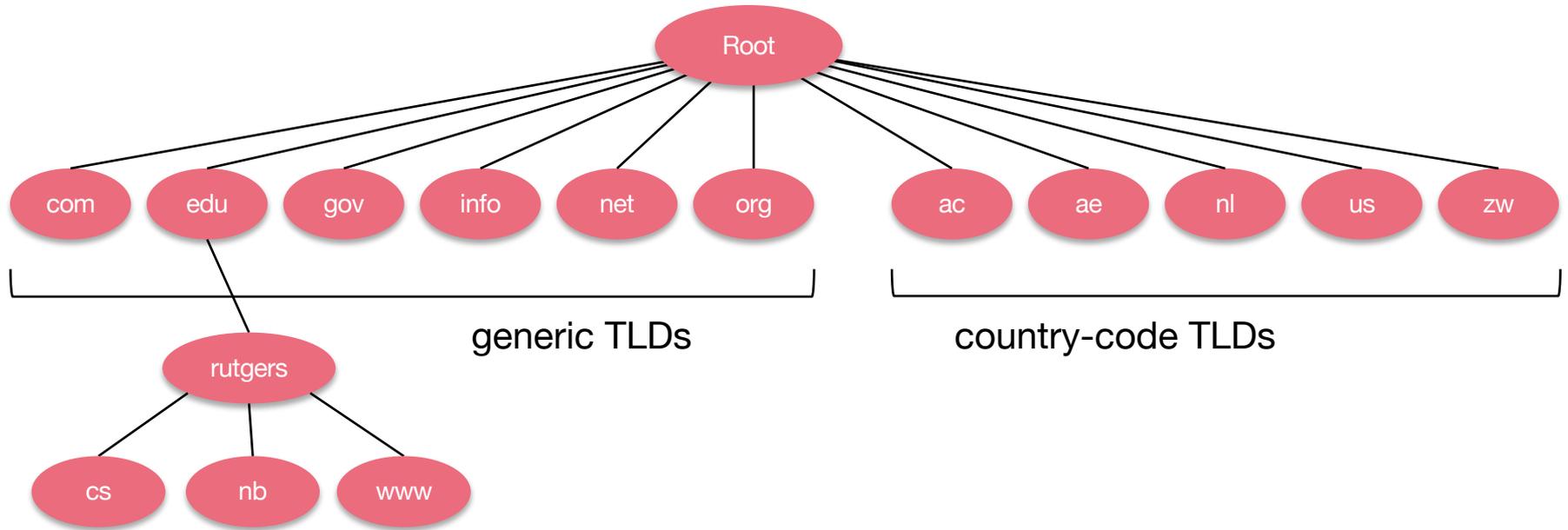
- A **domain hierarchy** was created in 1984 (RFC 920)
  - Domains are administrative entities: divide name management
  - Tree-structured global name space
  - Textual representation of domain names

`www.cs.rutgers.edu`



Jon Postel manually maintained, approved and tracked computer names & addresses on the Internet

# Domain Name Hierarchy



# Top Level Domains (TLDs)

## ccTLD

**Country-code** domains  
ISO 3166 codes  
e.g., .us, .de, .ca, .es

## IDN ccTLD

**Internationalized**  
country-code domains  
e.g., .السعودية, .中國, .рф

## gTLD

**Generic** top-level domains  
e.g., .biz, .com, .edu,  
.gov, .info, .net, .org,  
.audio, .catering, .网络

There are currently 1,436 top-level domains (as of March 9, 2026)

Each top-level domain has an administrator assigned to it

Assignment is delegated to various organizations by the Internet Assigned Numbers Authority (IANA)

IANA keeps track of the **root servers**

See <http://www.iana.org/domains/root/db> for the latest count

# Shared registration

- **Domain name registry:** *this is the database*
  - Keeps track of all domain names registered under a top-level domain
- **Domain name registry operator:** *this is the company that runs the DB*
  - NIC = **Network Information Center** – the organization that keeps track of the registration of domain names under a top-level domain
    - Keeps the database of domain names
    - See <https://www.icann.org/resources/pages/listing-2012-02-25-en>
- **Domain name registrar:** *this is the company you use to register*
  - Company that lets you register a domain name
  - Registrars update the registry database at the NIC
  - See <https://www.iana.org/assignments/registrar-ids/registrar-ids.xhtml>

\* I don't know why the latest registry list in 2023 is from 2012

# Shared registration

- Multiple domain **registrars** provide domain **registration services**
  - 3,020 registrars as of March 2026, including 1201 unique DropCatch.com registrars
- The registrar you choose becomes the **designated registrar** for your domain
  - Maximum period of registration for a domain name = 10 years
- The **registry operator** keeps the **central registry database** for the top-level domain
- Only the designated registrar can change information about domain names
  - A domain name owner may invoke a domain transfer process

## Example

- *Namecheap* is the designated registrar for `poopybrain.com`
- *VeriSign, Inc.* is the registry operator for the `.com` gTLD

See <https://www.icann.org/registrar-reports/accredited-list.html> for the latest list of registrars

# The problem

Every device connected to the internet has a unique Internet Protocol (IP) address

How do you **resolve** user-friendly machine names to IP addresses?

`www.cs.rutgers.edu` → `128.6.4.24`

# Original solution

In the early days (up through the 1980s)

- Search `/etc/hosts` file for machine name (see RFC 606)
- File periodically downloaded from Network Information Center (NIC) at the Stanford Research Institute (SRI)
- This was not sustainable with millions of hosts on the Internet
  - A lot of data
  - A lot of churn in the data: new hosts added, deleted, addresses changed
  - Maintenance
  - Traffic volume

*The solution doesn't scale!*

# DNS: Domain Name System

Distributed database: a hierarchy of **name servers**

**DNS** is an application-layer protocol

- Name-address resolution is handled at the edge
- The network core is unaware of host names ... and does not care
- There is no special relationship between names and addresses

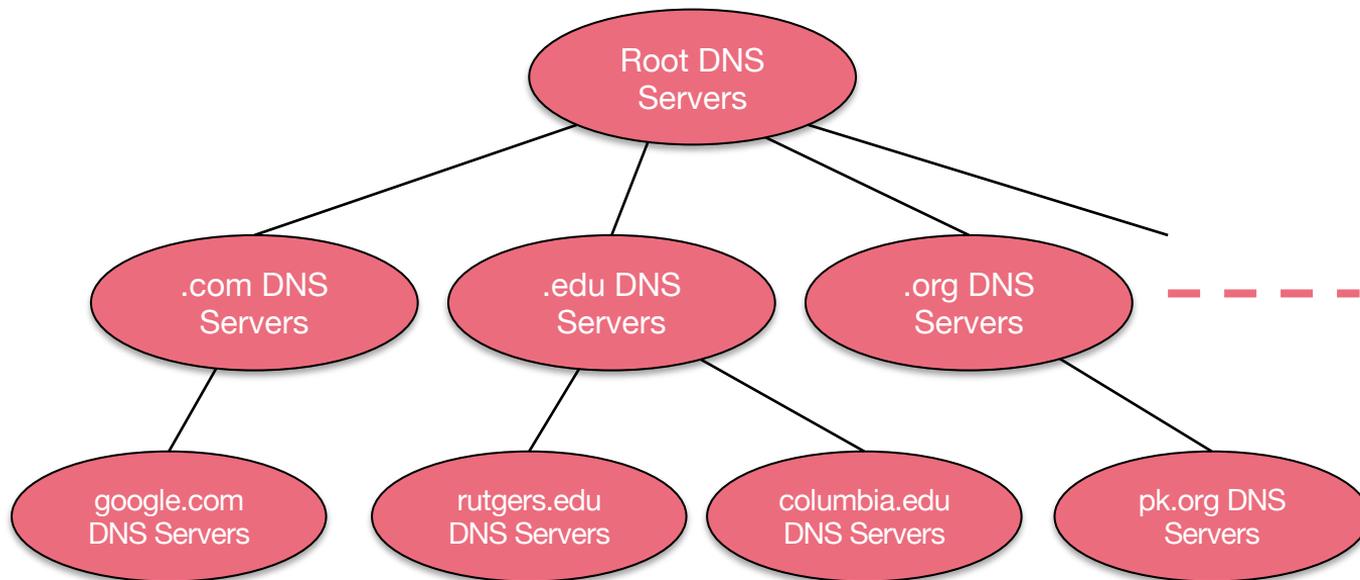
Example: `cs.poopybrain.com` can resolve to `cs.rutgers.edu`

`cs.poopybrain.com` → `cs.rutgers.edu`

# DNS servers provide...

- Name to IP address translation
- Aliasing of names (called **canonical** names)
- Identification of name servers
- Names of mail servers
- Load distribution:
  - Multiple name servers may handle a query for a domain
  - Caching – store past look-ups
  - Ability to provide a set of IP addresses for a name

# DNS is a distributed, hierarchical database



A collection of DNS servers

# Authoritative DNS server

- An **authoritative name server** is responsible for answering queries about its zone
  - Provides *real* answers vs. *cached* answers
  - Configured by the administrator
- **Zone** = group of machines under a node in the tree  
E.g., rutgers.edu

# A DNS server returns answers to queries

## Key data that a DNS server maintains (partial list)

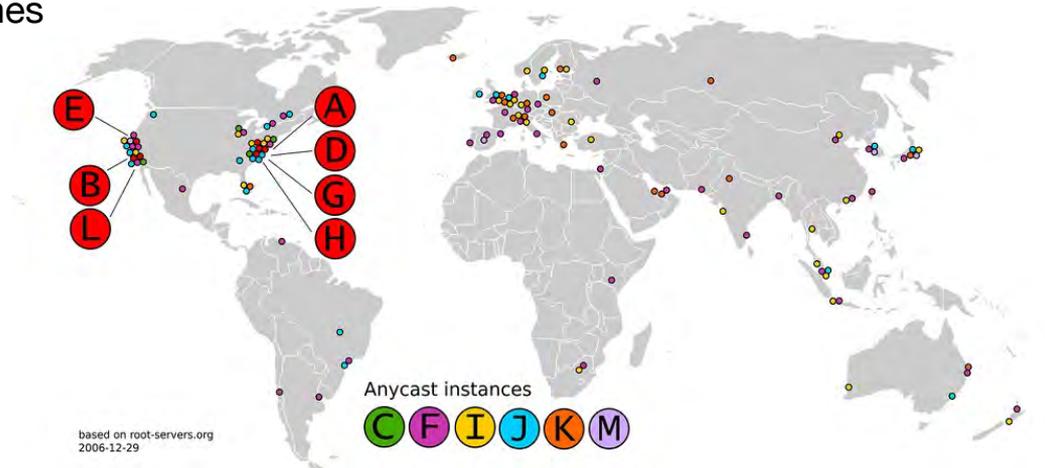
Information	Abbreviation	Description
Host	A	Host address (name to address) Includes name, IP address, time-to-live (TTL)
Canonical name	CNAME	Name for an alias
Mail exchanger	MX	Host that handles email for the domain
Name server	NS	Identifies the name server for the zone: tell other servers that yours is the authority for info within the domain
Start of Zone Authority	SOA	Specifies authoritative server for the zone. Identifies the zone, time-to-live, and primary name server for the zone

# Finding your way

- How do you find the DNS Server for rutgers.edu?
  - That's what the **domain registry** keeps track of
  - When you register a domain,
    - You supply the addresses of at least two DNS servers that can answer queries for your zone
    - You give this to the **domain registrar**, who updates the database at the **domain registry**
- So how do you find the right DNS server?
  - Start at the root

# Root name servers

- The **root name server** answers can return a list of authoritative name servers for top-level domains
- 13 root name servers
  - A.ROOT-SERVERS.NET, B.ROOT-SERVERS.NET, ...
  - Each has redundancy (via *anycast* routing or load balancing)
    - Each server is really a set of machines



Download the latest list at <http://www.internic.net/domain/named.root>

# DNS Queries

- **Iterative** (non-recursive) name resolution: *may not give you a final answer*
  - DNS server will return a definitive answer or a **referral** to another DNS server
    - *referral* = reference to a DNS server for a lower level of the queried namespace
  - Advantage: stateless
- **Recursive** DNS name resolution: *responsible for finding the final answer*
  - Name server will take on the responsibility of fully resolving the name – it processes referrals
    - May query multiple other DNS servers on your behalf
      1. Send query to a **root** name server (get a referral)
      2. Send query to an **edu** name server (get a referral)
      3. Send query to a **rutgers** name server (get an answer)
  - Disadvantage: name server has more work; has to keep track of state
  - Advantages: Caching opportunities, less work for the client!

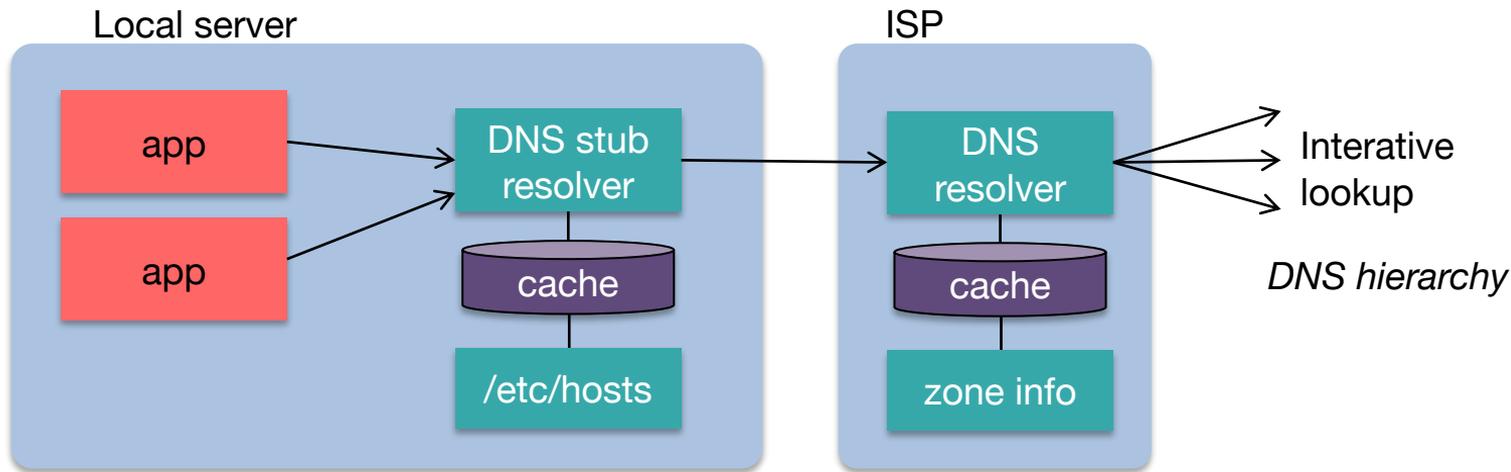
*Most top-level DNS servers only support iterative queries*

# DNS Resolvers: local name server

**DNS Resolver** = client side of DNS

- Not really a part of the DNS hierarchy
- Acts as an intermediary between programs that need to resolve names and the name servers
- A resolver is responsible for performing the full resolution of the query
- Where are the resolvers?
  - Each local system has one: that's what applications contact
    - Local cache; may be a process or a library
    - On Linux & Windows, these are limited DNS servers (called **stub resolvers**)
      - Usually not capable of handling referrals and expect to talk with a name server that can handle recursion (full resolution)
  - ISPs (and organizations) run them on behalf of their customers
    - Including a bunch of free ones (OpenDNS, Google Public DNS)
- Resolvers cache past lookups – they are not responsible for zones

# DNS Resolvers in action



## Local stub resolver:

- check local cache
- check local hosts file
- send request to external resolver

E.g., on Linux: resolver is configured via the `/etc/resolv.conf` file

## External resolver

- DNS server that accepts recursion
- Running at ISP, Cloudflare, Google Public DNS, OpenDNS, etc.

# Sample Query

- Rutgers registered `rutgers.edu` with the `.edu` domain
  - `educause.net` is the domain registry for the `.edu` gTLD
  - Registration includes defining the name servers for `.rutgers.edu`
    - `ns124.a2.incapsecuredns.net: 192.230.123.124`
    - `ns8.a1.incapsecuredns.net: 192.230.122.8`
    - `ns87.a0.incapsecuredns.net: 192.230.121.87`
- EDUCAUSE registered its name servers with root name servers
  - `ns1.twtelecom.net`
  - `ns1.educause.edu`
  - `ns1.twtelecom.net`

} Name servers for `.edu`
- We know how to get to root name servers
  - Download <http://www.internic.net/domain/named.root>

# Sample Query

Submit query to a local *DNS resolver*:

1. *query(cs.rutgers.edu)* → any root name server  
send query to f.root-servers.net: 192.5.5.241
2. Receive *referral* to a list of DNS servers for *edu*  
a.edu-servers.net: 192.5.6.30 ... d.edu-servers.net: 192.31.80.30 ...
3. *query(cs.rutgers.edu)* → edu name server  
send query to d.edu-servers.net: 192.31.80.30
4. Receive *referral to rutgers.edu* name servers:
  - dns2.rutgers.edu. 192.230.121.86
  - ns1.rutgers.edu. 192.230.122.7
  - ru-ufl.rutgers.edu. 192.230.123.123
  - ns6.dnsmadeeasy.com. 208.80.124.13
5. *query(cs.rutgers.edu)* → rutgers name server  
send query to 208.80.124.13
6. The rutgers name server returns  
A 128.6.48.178 *address record*  
MX 10 cs-rutgers-edu.mail.protection.outlook.com. *mx (mail exchange): domain name for email*

# Caching

- Starting every query at the root would place a huge load on root name servers
- A name server can **cache** results of previous queries
  - Save query results for a *time-to-live* amount of time
  - The time-to-live value is specified in the domain name record by an authoritative name server

The End