eecs489 COMPUTER NETWORKS

Lecture 5: Domain Name System

Names vs. Addresses

Names are easier for human to remember
- www.umich.edu vs. 141.213.4.4

Addresses can be changed without changing names
- move www.umich.edu to 128.212.5.5
- useful for renumbering when changing providers

Name could map to multiple addresses
- google.com maps to multiple replicas of the Web site
- and to different "nearby" addresses in different geographies
  - to reduce latency or to provide localized content

Multiple names could map to the same address
- aliases such as graphics.eecs.umich.edu and www.eecs.umich.edu

Flat vs. Hierarchical Space

Example of flat name space:

Examples of hierarchical name space:

Examples of hierarchical address space:

Why form hierarchy?

Advantage of hierarchical space:

Domain Name System (DNS)

DNS consists of:

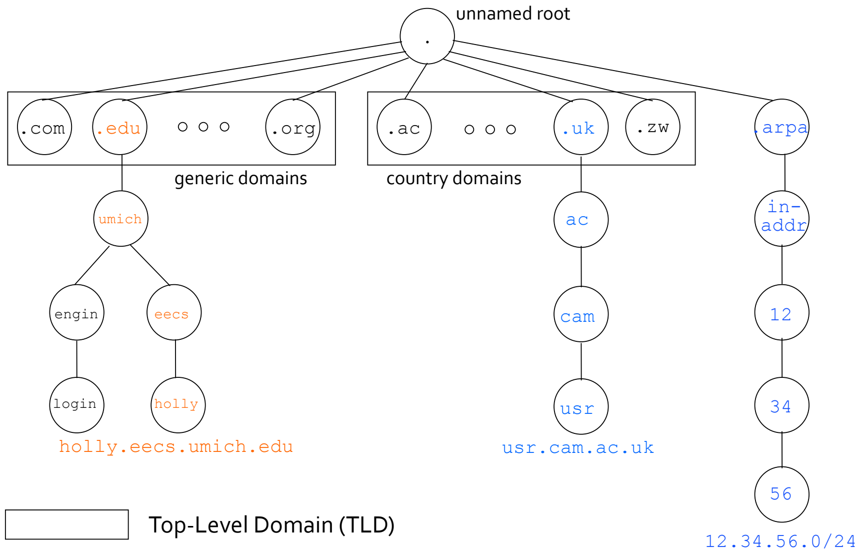
1. an hierarchical name space:
   name allocation decentralized to domains

   host.sub-subdomain.. . ..subdomain.domain[.ROOT]

   host: machine name, can be an alias
   sub-subdomain: department (engin, eecs, physics, math)
   subdomain: institution, company, geography, service provider (umich, mi, comcast)
   domain: most significant segment (edu, com, org, net, gov, us, it)

   Examples of Fully Qualified Domain Names (FQDNs):
   www.eecs.umich.edu, maps.google.com

DNS Hierarchical Name Space



Domain Name System (DNS)

DNS consists of:

2. an hierarchical name resolution infrastructure:
   - a distributed database storing resource records (RRs)
   - client-server, query-reply protocol

Berkeley Internet Name Domain (BIND): the most common implementation of the DNS name resolution architecture

DNS Resource Record

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

`type=A`
- `name` is hostname
- `value` is IP address

`type=NS`
- `name` is domain (e.g., `umich.edu`)
- `value` is IP address of authoritative name server for this domain

`type=CNAME`
- `name` is alias name for some "canonical" (real) name
  for example: `graphics.eecs.umich.edu` is really `www.eecs.umich.edu`
- `value` is canonical name

`type=MX`
- `value` is name of mail exchange server associated with `name`

DNS Resource Record

DNS lookup returns only entries matching type: Hence when web browser couldn't find an Address entry, mail may still find a Mail eXchange entry

Try:

```
% dig smtp.eecs.umich.edu A
% dig smtp.eecs.umich.edu MX
```

DNS Name Servers

DNS database is partitioned into zones

A zone holds one or more domains, analogy:

DNS	File System
domains	folders
zones	volumes

Name server: a process managing a zone

Authoritative or primary name server:
the "owner" of a zone

- providing authoritative mappings for organization's server names (e.g., web and mail)
  - can be maintained by an organization or its service provider

DNS Name Servers

Zones may be replicated (for what purpose?)

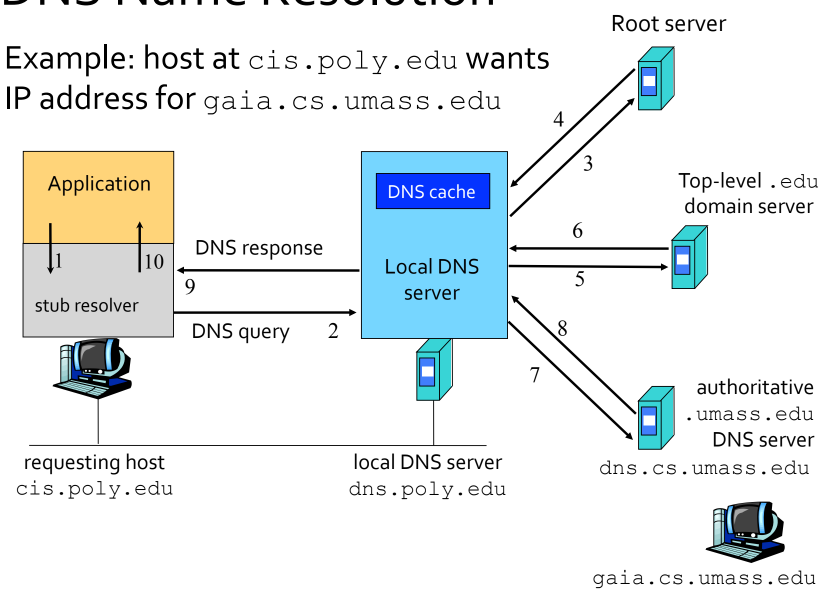
- secondary servers: replicas

Zone transfer: downloading a zone from the primary server to the replicas

A name server can be the primary server for one or more zones, and the secondary server for one or more zones

DNS Name Resolution

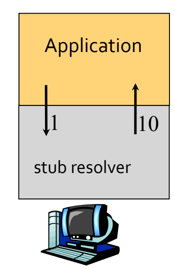
Example: host at `cis.poly.edu` wants IP address for `gaia.cs.umass.edu`



DNS Name Resolution: Client Side

Client:

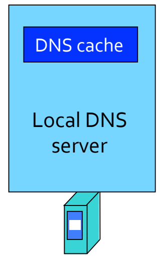
- has stub resolver linked in
- consults `/etc/resolv.conf` to find local name server
- forms FQDN
- queries up to 3 local name servers in turn
- if no response, double timeout and retry for 4 rounds



DNS Name Resolution: Client Side

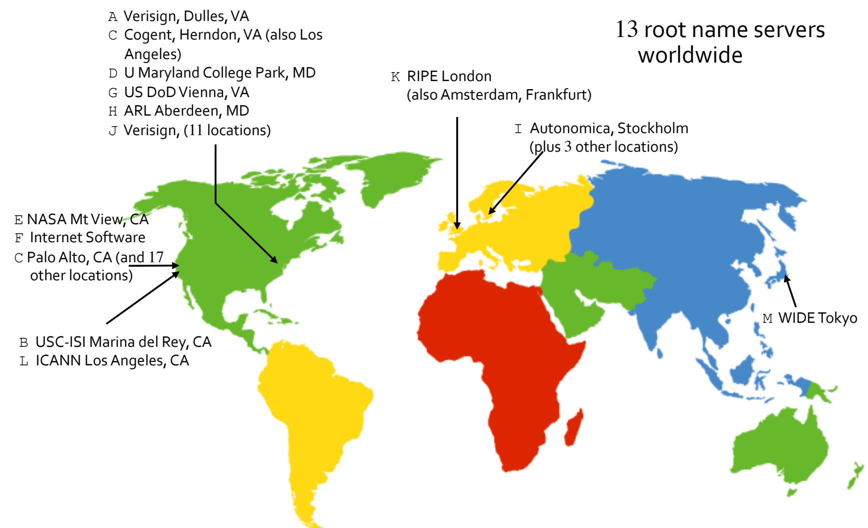
Local name server:

- when a host makes a DNS query, the query is sent to its local name server
- each ISP (residential ISP, company, university) has one
  - also called "default name server"
- acts as a proxy, forwards query into the DNS hierarchy
- parses FQDN from right to left
  - always goes to ROOT first
- consults `/etc/named.conf`, `named.root`, and `zonefile` to find name servers
- caches resolved name



DNS Root Name Servers

13 root name servers worldwide



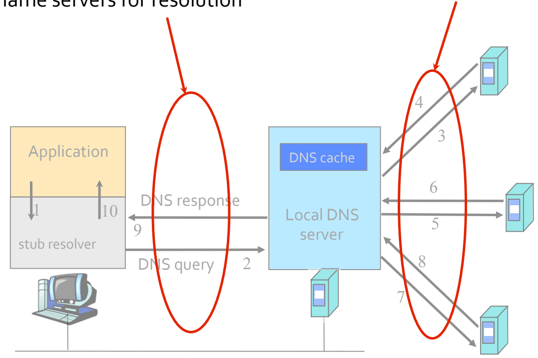
Recursive vs. Iterative Query

Recursive query:

- local name server must resolve the name (or return "not found"); if necessary, by asking other name servers for resolution

Iterative query:

- contacted server replies with the name of server address of sub-domain
  - "I don't know this name, but ask this other name server"
- requesting name server visits each name server referred to



Why not always do recursive resolution?

DNS Caching

Once a (any) name server learns of a mapping, it caches the mapping

- to reduce latency in DNS translation

Cache entries timeout (disappear) after some time-to-live (TTL)

- TTL is assigned by the authoritative server (owner of the host name)

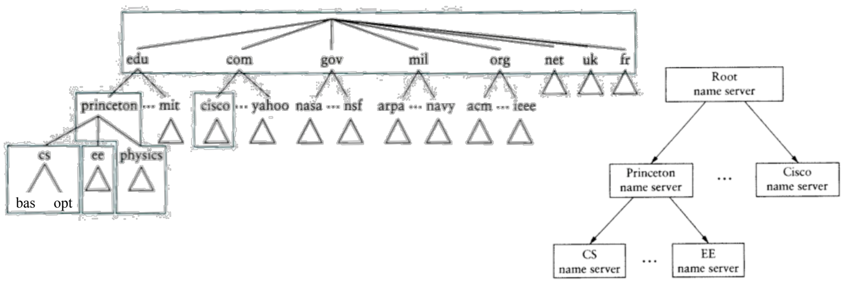
Local name servers typically also cache

- TLD name servers cache to reduce visits to root name servers
- all other name servers cache referrals
- cache both positive and negative results

DNS Name Resolution Exercises

Show the DNS resolution paths, assuming the DNS hierarchy shown and assuming caching, starting with empty caches:

- `thumper.cisco.com` looks up `bas.cs.princeton.edu`
- `thumper.cisco.com` looks up `opt.cs.princeton.edu`
- `thumper.cisco.com` looks up `cat.ee.princeton.edu`
- `thumper.cisco.com` looks up `ket.physics.princeton.edu`
- `bas.cs.princeton.edu` looks up `dog.ee.princeton.edu`
- `opt.cs.princeton.edu` looks up `cat.ee.princeton.edu`



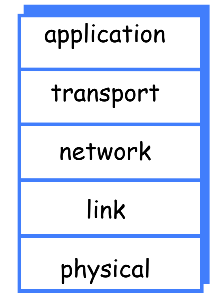
Peterson & Davie 2nd. ed., pp. 627, 628

DNS Design Points

DNS serves a core Internet function

At which protocol layer does the DNS operate?

- host, routers, and name servers communicate to resolve names (name to address translation)
- complexity at network's "edge"



Why not centralize DNS?

DNS is "exploited" for server load balancing, how?

DNS Protocol, Message Format

Same message format for both query and reply messages

flags:
- query or reply
- recursion desired
- recursion available
- reply is authoritative