DNS: Domain Name System

CS 360 Internet Programming

Daniel Zappala

Brigham Young University
Computer Science Department

Domain Name System

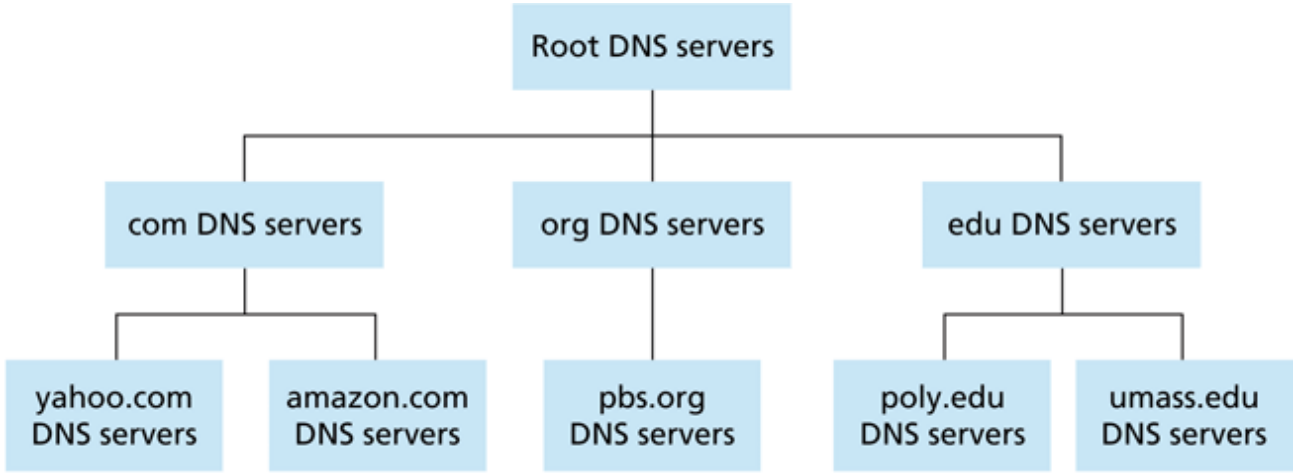
- people like to use names for computers (www.byu.edu), but computers need to use numbers (128.187.22.132)
- the Domain Name System (DNS) is a distributed database providing this service
  - a program send a query a local name server
  - the local name server contacts other servers as needed
- many DNS services
  - host name to IP address translation
  - host aliasing (canonical name versus alias names)
  - lookup mail server for a host
  - load distribution - can provide a set of IP addresses for one canonical name

Demonstration: DNS services

Names

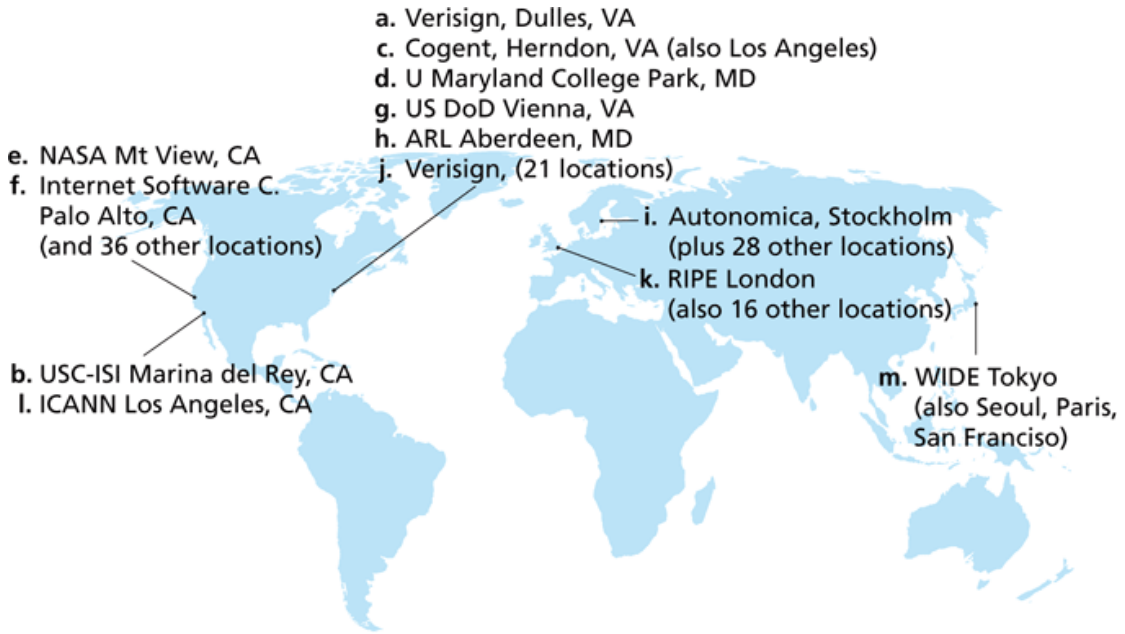
- domain name: top-level domain (TLD) + one or more subdomains
  - example: cs.byu.edu
- host name: a domain name with one or more IP addresses associated with it
- TLDs
  - ccTLD: country codes (.us, .uk, .tv)
  - gTLD: generic (.com, .edu, .org, .net, .gov, .mil) – see full list at Wikipedia
  - iTLD: infrastructure (.arpa)
- may be 127 levels deep, 63 characters per label, 255 characters per name

DNS Hierarchy



- root, top-level domain (TLD), and local name servers
- each level represents a *zone*
- what zone is BYU in charge of?

Root Name Servers



- can be contacted by any local name server that can not resolve a name
- refers the local name server to another server down the hierarchy
- only 13 of them worldwide

Top-level Domain (TLD) Name Servers

- responsible for .com, .org, .net, .edu, .name, .info, etc, as well as all country domains (.uk, .fr, .jp, .us, etc)
- refer name queries to local name servers
- .com is run by Verisign
- .net is run by Verisign
- .edu is run by Educause (operated by Educause)
- .org is run by Public Interest Registry (operated by Afilias)

Local and Authoritative Name Servers

- local name server
  - run by a given organization, for its domain
  - resolve queries from hosts in the domain, forwarding them up the hierarchy as needed
- authoritative name server
  - provides answers to queries from hosts outside the domain for the zone
  - often the same as the local name server
- a local name server can be a caching-only name server – it is not authoritative for any domain, it simply makes queries for hosts and caches DNS responses

Using DNS

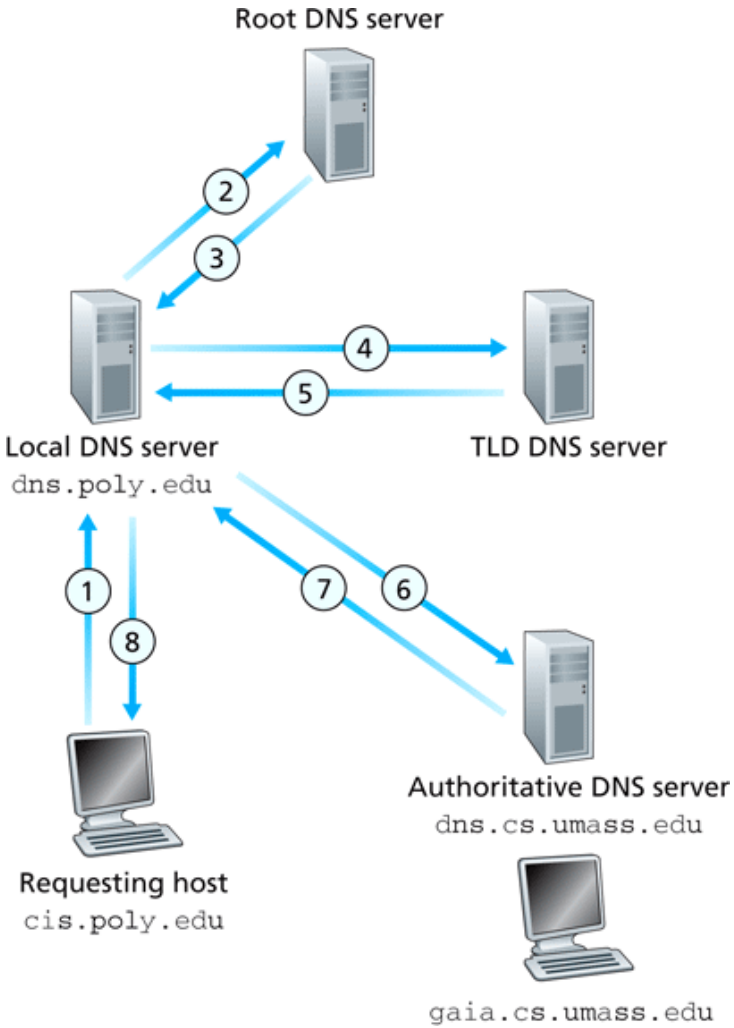
```
struct hostent *gethostbyname(const char *name);
```

- name = host name
- on success returns a hostent structure
- on error returns NULL

```
struct hostent {
  char  *h_name;          /* official name of host */
  char **h_aliases;       /* alias list */
  int    h_addrtype;      /* host address type */
  int    h_length;        /* length of address */
  char **h_addr_list;     /* list of addresses */
}
```

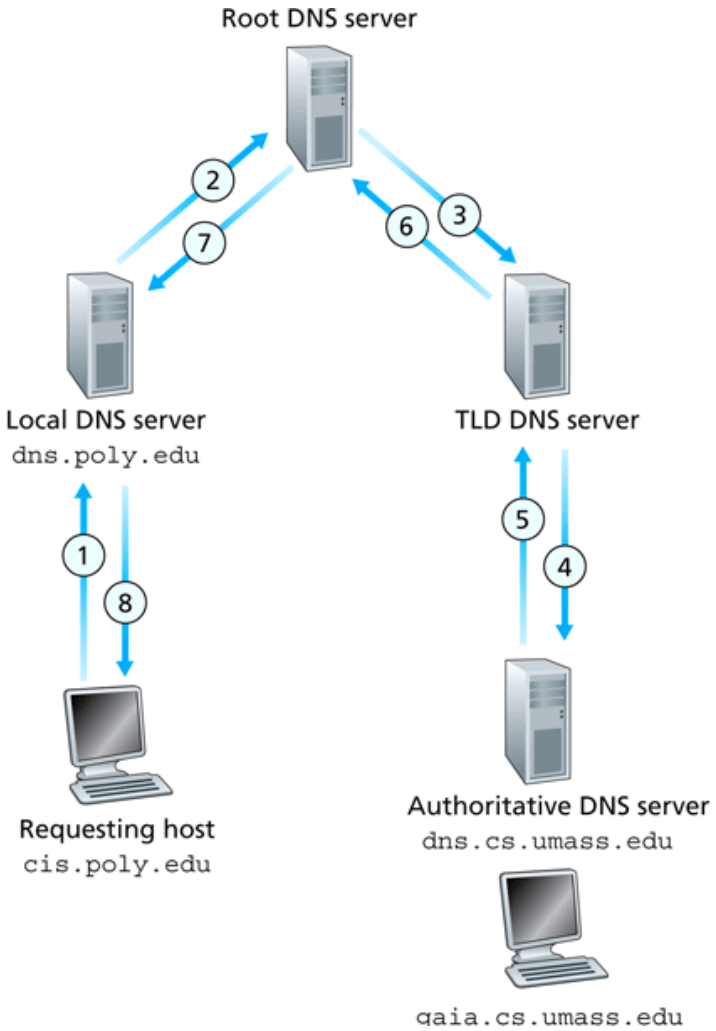
See DNS example code on class web site.

Iterated Query



- local name server contacts root name server if it doesn't have the mapping
- iterated query: each server that doesn't know the mapping tells the local name server the identity of the next server in the hierarchy that can answer the query

Recursive Query



- recursive query: root name server (and other servers) may forward the query for the local name server and return the reply when done
- puts a heavier load on the root name server
- query type indicates whether it is recursive or iterative, name servers are *not* required to support recursive queries

Reverse Mappings

- what if you want to lookup the name associated with an IP address?
- this is very useful for authenticating that someone comes from an authorized domain, e.g. check that they can send email through your server
- addresses turned into a name by reversing dotted-decimal notation and appending IN-ADDR.ARPA
  - 128.187.22.132 $\Rightarrow$ 132.22.187.128.IN-ADDR.ARPA
- TLD server in charge of .ARPA
- when IP addresses are assigned, the authoritative name server is also assigned a prefix from the reverse mapping space

Caching

- any name server that learns a mapping caches it
- cache entries time out after some time – timeout value set by the authoritative name server
- TLD servers usually cached in a local name server, so root name server not visited often

Replication

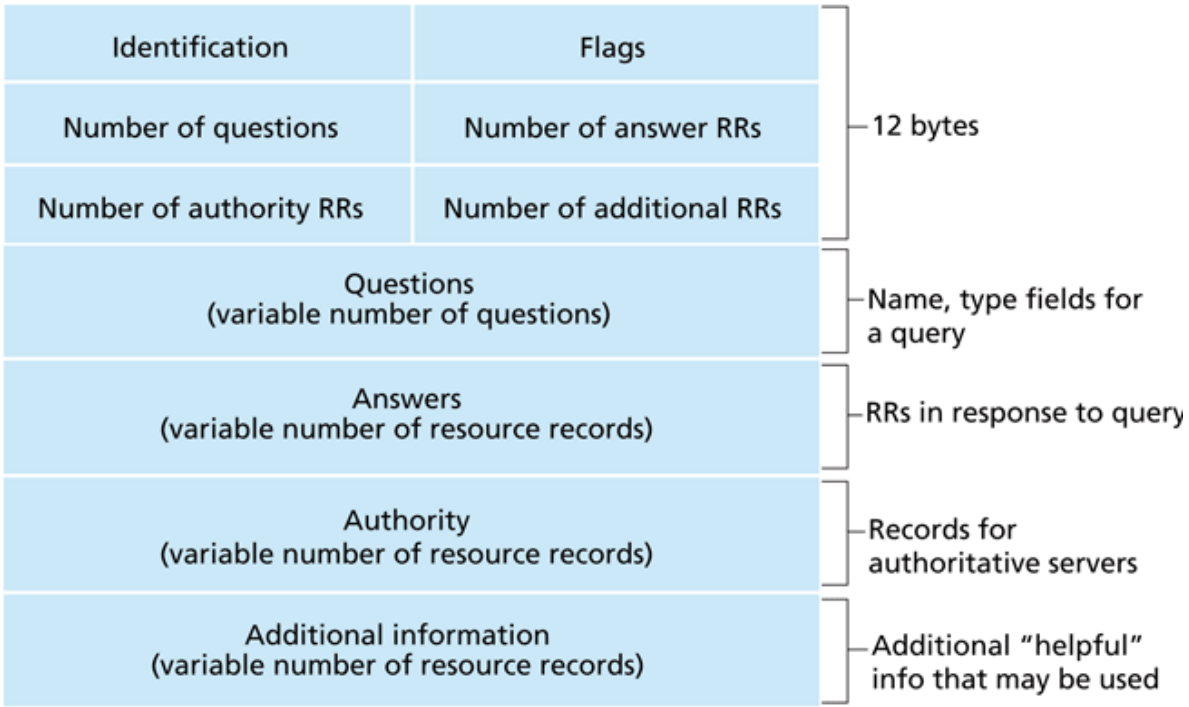
- an organization typically wants to replicate its authoritative DNS server in case it fails or needs maintenance
- define a master and various secondary servers for the zone
- secondary servers must poll master for updates to the zone and perform a "zone transfer"
- RFC 2136 specifies mechanisms for dynamically updating DNS entries (e.g. for hosts using DHCP, mobile hosts)

DNS Records

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

- type=A
  - name is a host name
  - value is an IP address
- type=CNAME
  - name is an alias for the real name
  - value is the canonical name
  - e.g. ilab.cs.byu.edu is really carmelo.cs.byu.edu
- type=MX
  - name is a host name
  - value is the name of the mail server associated with the name
- type=NS
  - name is a domain
  - value is IP address of authoritative domain server for this domain

DNS Messages



- query and reply messages use same format
- identification is 16 bits, unique to the query, reply uses the same number
- flags
  - query or reply
  - recursion desired
  - recursion available
  - reply is authoritative (vs cached)

Adding DNS Records

- example: register the new name *zappala.org*
- register the name at a registrar (e.g., GoDaddy)
- provide registrar with names and IP addresses of your authoritative name server (primary and secondary - comes from hosting service)
- registrar inserts two RRs into the .com TLD server
  - (zappala.org, ns1.phpwebhosting.com, NS)
  - (ns1.phpwebhosting.com, 64.65.1.112, A)
- authoritative name server adds a Type A record and a Type MX record for zappala.org

ARPAnet

- use a text file to map names to addresses: hosts.txt
- to update an address
  - email your changes to the NIC
  - the NIC updates the hosts.txt file every few days
  - download the hosts.txt file via FTP
- problems
  - single point of failure
  - consistency
  - traffic volume
  - delay
  - maintenance

IANA

- Internet Addressing and Naming Authority
- assigns globally-unique names, addresses, ports, character encodings, and other parameters that require central administration
- run by Jon Postel at the Information Sciences Institute, which is affiliated with USC
  - wrote the original RFCs for IP, ICMP, TCP
  - wrote or edited 200+ RFCs
  - Postel's Law: be conservative in what you do, be liberal in what you accept from others
- funded by the Department of Defense

DNS

- 1984: Paul Mockapetris (ISI) defined the Domain Name System in 1984, RFCs 882 and 883 (later superseded by RFCs 1034 and 1035)
  - distributed database of name servers
  - application-layer protocol to query name servers
  - *end-to-end principle – keep the core of the network as simple as possible, put complex functionality at the edges*
- 1992: NSF awards a contract to Network Solutions for maintenance of gTLDs (.com, .org, .net, .edu): $100 to register a name
- 1998: government decides to privatize DNS
- 2000: transition to ICANN

ICANN

- about
  - formed to privatize functions of IANA
  - originally intended to have Jon Postel as CTO, but he died in 1998
  - California non-profit run out of ISI
- manages IANA functions
- establishes domain name policy
  - which gTLDs should be created (.biz, .info, .aero, .jobs, .travel) and which should not be allowed (.xxx)
  - settle domain name disputes for gTLDs
- criticism
  - governance - how board members are chosen, how meetings are held
  - policy - $50,000 fee to become a registrar, dispute resolution policies, more free market control of gTLDs
  - too much control by the US and its laws

Alternatives

- anyone can setup an alternative DNS root system
  - run a set of root name servers
  - establish a set of TLDs
  - 24/7 reliable operation
- examples
  - http://www.openic.unrated.net (OpenNIC): democratically governed
  - http://www.unifiedroot.com (UnifiedRoot): free market
  - http://www.new.net (New.Net): profit

The Growth of DNS and the Internet

- how can we measure the size of the Internet?
  - can't count number of users who are on the net; must estimate
  - some hosts have multiple domain names and IP addresses
  - can't tell if some hosts are missing
- can't determine the exact size of the Internet or the number of users
- approximations
  - count domain names with IP addresses (old)
  - count IP addresses with domain names (new)

Internet Growth (1981-2010)