CS 455/555
Intro to Networks and Communications

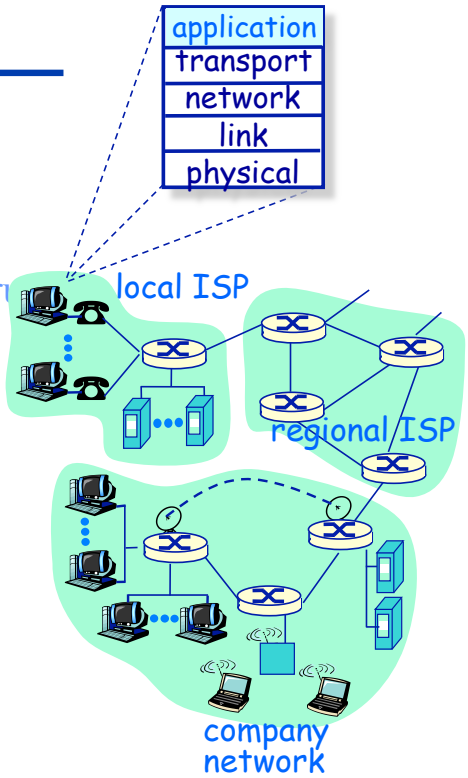
Applications & Application-Layer Protocols:
The Domain Name System

Dr. Michele Weigle
Department of Computer Science
Old Dominion University
mweigle@cs.odu.edu

http://www.cs.odu.edu/~mweigle/CS455-S13

Application-Layer Protocols
Outline

- The architecture of distributed systems
  - » Client/Server computing
  - » P2P and Hybrid computing
- The programming model used in constru distributed systems
  - » Socket programming
- Example client/server systems and their application-layer protocols
  - » The World-Wide Web (HTTP)
  - » Reliable file transfer (FTP)
  - » E-mail (SMTP & POP)
  - » Internet Domain Name System (DNS)



Application-Layer Protocols
The Domain Name System (DNS)

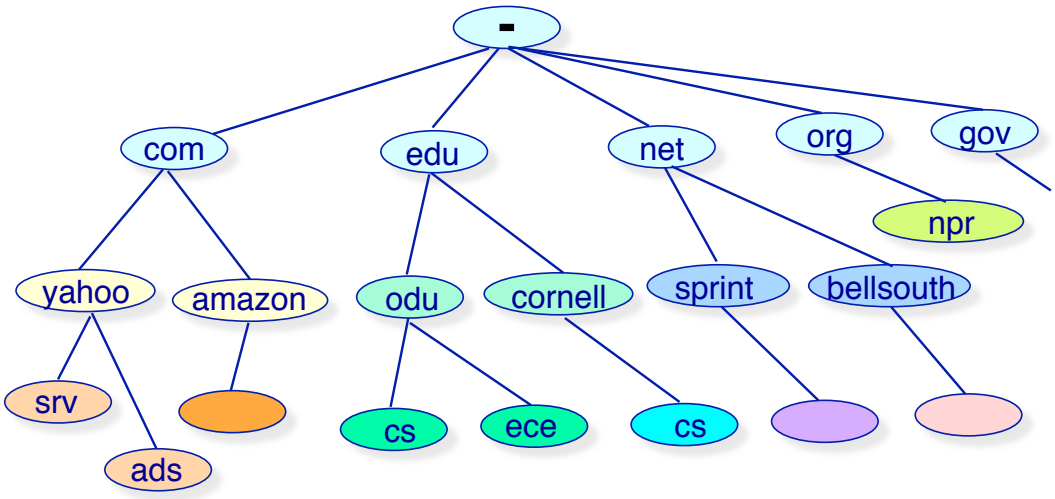
- Computers (hosts, routers) connected to the Internet have two forms of names:
  - » ***IP address*** — a 32 bit identifier used for addressing hosts and routing data to them
  - » ***Hostname*** — an ASCII string used by applications
- The DNS is an Internet-wide *service* that provides mappings between IP addresses and hostnames
  - » The DNS is a distributed database implemented in a hierarchy of name servers
  - » The DNS is also an application-layer protocol
- Hosts and routers use name servers to *resolve* names (address/name translation)
  - » Name resolution is an *essential* Internet function implemented as application-layer protocol

The Domain Name System
Services

- Host Aliasing
  - » *canonical hostname:* `relay1.west-coast.enterprise.com`
  - » *aliases:* `enterprise.com, www.enterprise.com`
- Mail Server Aliasing
  - » *email address:* `bob@hotmail.com`
  - » *mail server:* `relay1.west-coast.hotmail.com`
- Load Distribution
  - » set of IP addresses associated with 1 canonical hostname (e.g., `cnn.com`)
  - » server response with whole set, but rotates ordering

The Domain Name System

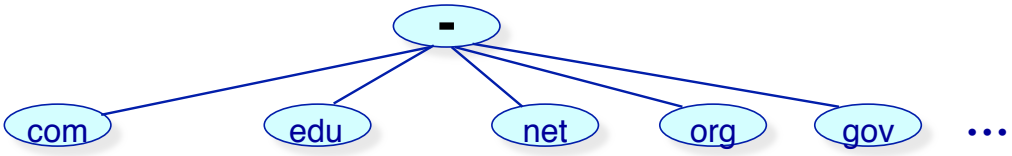
Name Hierarchy in DNS



- *hostname* = "dot" separated concatenation of domain names along path toward the root
  - » *odu.edu*
  - » *cs.odu.edu*
  - » *atria.cs.odu.edu*

Name Hierarchy in the DNS

Top level domains



- Generic domains:
  - » *.com*, *.org*, *.net*, *.edu*, *.gov*, *.mil*, *.int*
  - » *.biz*, *.info*, *.name*, *.pro*
- Special sponsored names
  - » *.aero*, *.coop*, *.museum*
- Country code domains
  - » *.uk*, *.de*, *.jp*, *.us*, *etc.*

DNS
Overview

- Applications need IP address to open connection
- Use DNS to find the IP address given a hostname
- Steps:
  1. Application invokes DNS (`gethostbyname()` in C)
  2. DNS application in host sends query into network (UDP port 53)
  3. DNS application in host receives reply with IP address (after some delay)
  4. IP address passed up to the application

DNS is a black box as far as the application is concerned.

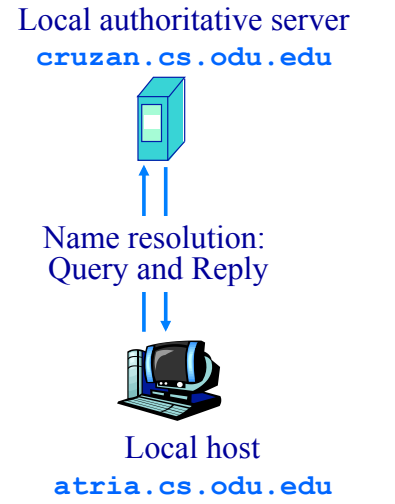
The Domain Name System
Designing a distributed service

- Why not centralize the DNS
  - » A server process on a big, well connected supercomputer?
- Centralized systems do not scale!
  - » Poor reliability: centralized = single point of failure
  - » Poor performance: centralized = "remote access" for most users
  - » Difficult to manage: centralized = all traffic goes to one location, a large staff has to be present to handle registrations
- A centralized system is not politically feasible in an international network

Designing a Distributed Service

DNS Name Servers

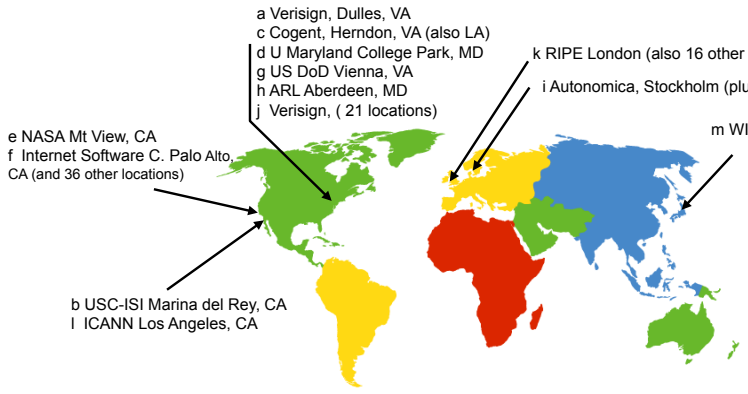
- No server has every hostname-to-IP address mapping
- Authoritative name server:
  - » Every host is registered with at least one authoritative server that stores that host's IP address and name
  - » The authoritative name server can perform name/address translation for that host's name/address
- *Local* authoritative name servers:
  - » Each ISP, university, company, has a *local (default) name server* authoritative for its own hosts
  - » *Resolvers* always query a name server local to it to resolve *any* host name

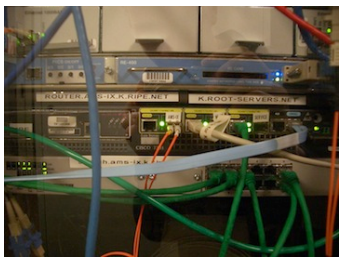


What if the name is not a local host (*e.g.*, **www.yahoo.com**)?

DNS Name Servers

Root name servers



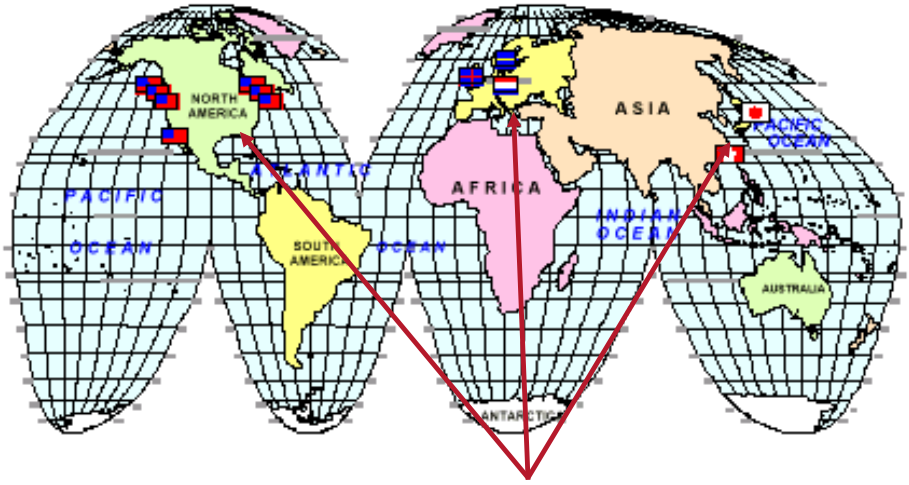


- A root name server is contacted when a local name server can't resolve a name
  - » The root server either resolves the name or provides pointers to authoritative servers at lower level of name hierarchy
- There are 13 root name servers worldwide
  - » a.root-servers.org – m.root-servers.org

http://root-servers.org/

DNS Name Servers

Generic TLD servers (Verisign Corp.)



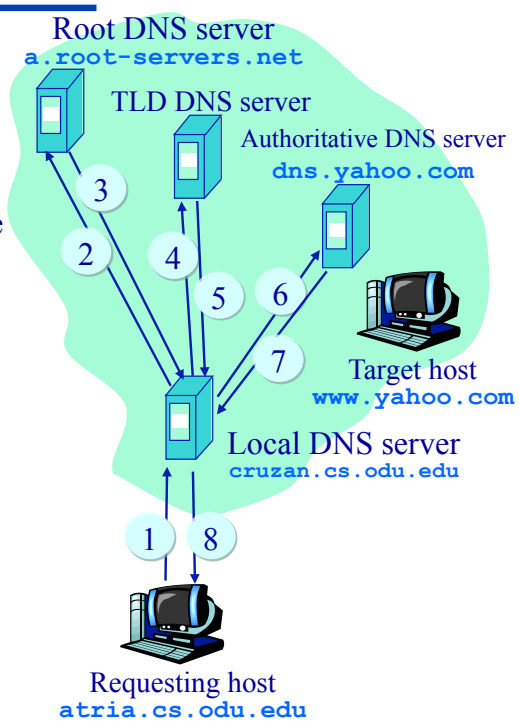
12 independent sites

- *.com*, *org*, *.net* server locations (separated from root servers)

DNS Name Servers

Using a server hierarchy for resolving names

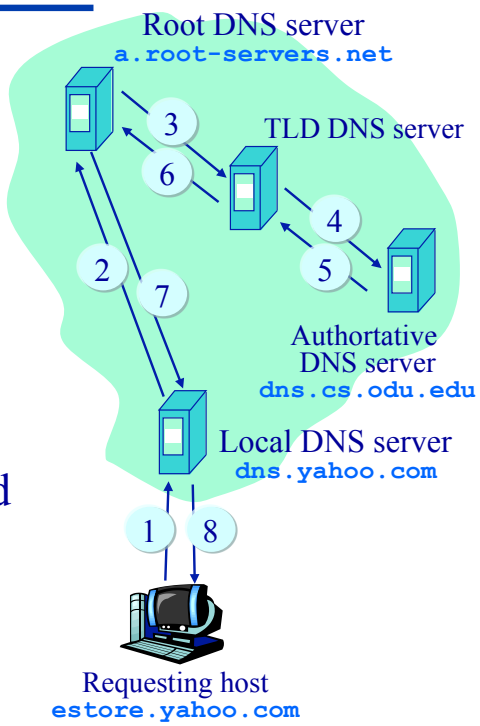
- Host *atria.cs.odu.*edu wants to know the IP address of *www.yahoo.com*
  - » *atria* contacts its local DNS server *cruzan.cs.odu.edu*
- To resolve a non-local name, the local name server queries the root server
- The root server responds with the TLD for .com
- The local DNS server contacts the TLD server
- The local DNS server contacts the authoritative server *dns.yahoo.com*
- Results feed back to *atria*
- *atria* can now use the IP address of *www.yahoo.com* to make a connection



DNS Name Servers

Recursive vs. Iterative Queries

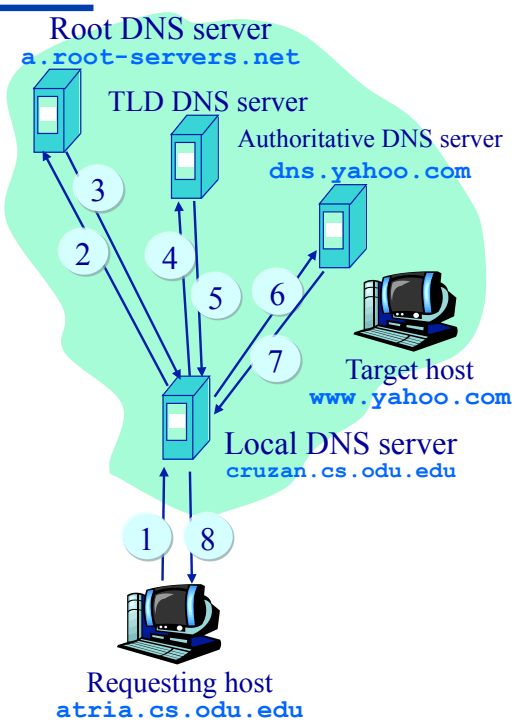
- The DNS supports two paradigms of queries:
  - » Recursive queries
  - » Iterative queries
- Recursive queries place the burden of name resolution (recursively) on the contacted server
- In an iterated query the contacted server simply replies with the name of the server to contact
  - » "I don't know; trying asking X"



DNS Name Servers

Recursive vs. Iterative Queries

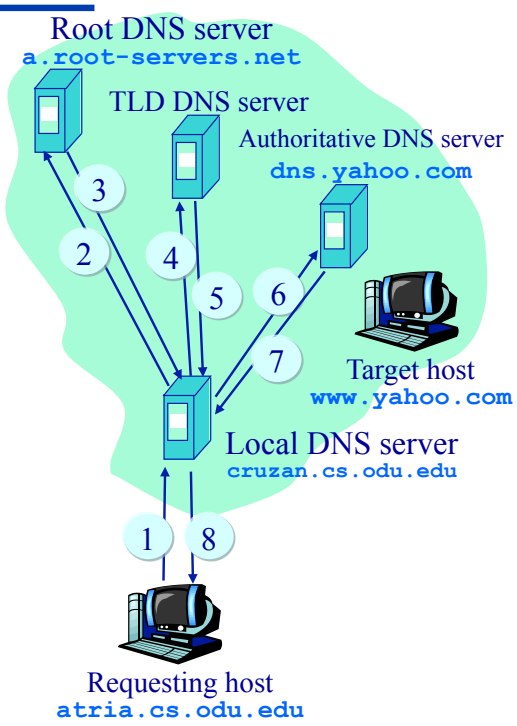
- Any query can be recursive or iterative
- Iterative and recursive queries can be combined
- Typically, the query from the requesting host to the local DNS server is recursive and the remaining queries are iterative



DNS Name Servers

Caching and updating DNS entries

- Every server caches all the mappings it learns
  - » Cache entries are "soft state"
  - » They timeout (are deleted) after some time period
- DNS cache update/notify mechanisms under design by the IETF
  - » See RFC 2136



DNS Name Servers

DNS resource records

RR format: `<name, value, type, time_to_live>`

- The DNS is a distributed database storing *resource records* (RRs)
- Type = A
  - » `name` is a hostname
  - » `value` is hostname's IP address
- Type = NS
  - » `name` is a domain
  - » `value` is name of authoritative name server for this domain
- Type = CNAME
  - » `name` is an alias name for some "canonical" (the real) name
  - » `value` is canonical name
- Type = MX
  - » `value` is name of mail server host associated with `name`

DNS Name Servers

DNS resource records / Examples

- A record
  - » `(relay1.west-coast.yahoo.com, 145.137.93.126, A)`
- NS record
  - » `(yahoo.com, dns.yahoo.com, NS)`
- CNAME record
  - » `(yahoo.com, relay1.west-coast.yahoo.com, CNAME)`
- MX record
  - » `(yahoo.com, mail.yahoo.com, MX)`

DNS

Inserting Records into DNS

- Example: new startup "Network Utopia"
- Register name networkuptopia.com at *DNS registrar* (e.g., Network Solutions)
  - » provide names, IP addresses of authoritative name server (primary and secondary)
  - » registrar inserts two RRs into com TLD server:

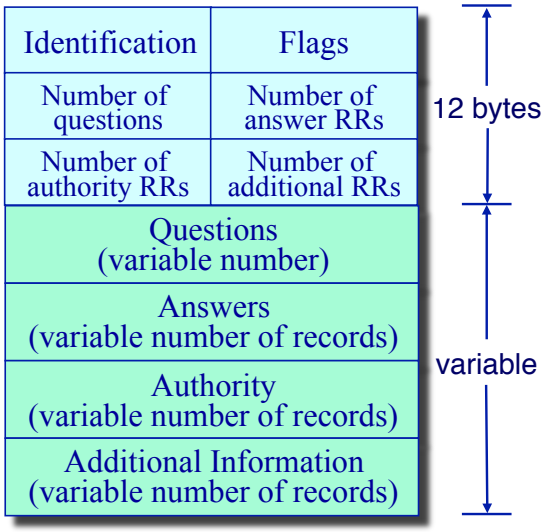
```
(networkutopia.com, dns1.networkutopia.com, NS)
(dns1.networkutopia.com, 212.212.212.1, A)
```

- Create authoritative server Type A record for www.networkuptopia.com; Type MX record for networkutopia.com
- *How do people get IP address of your Web site?*

DNS Protocol

DNS *query* and *reply* messages

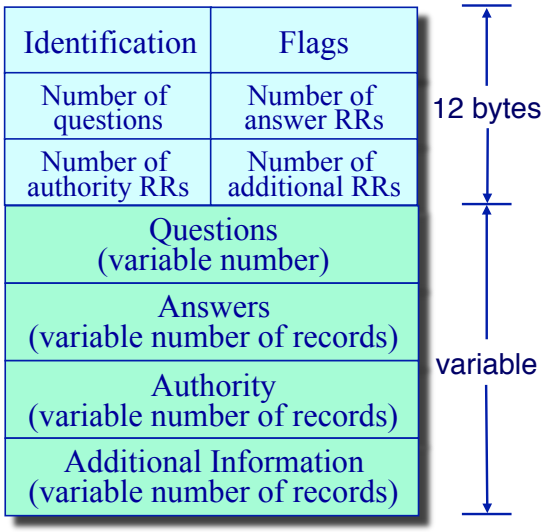
- DNS *query* and *reply* messages both have the same message format
- Messages have a fixed length message header
  - » Identification — 16 bit query/reply identifier used to match relies to queries
  - » Flags:
    - ❖ Query/Reply bit
    - ❖ "Reply is authoritative" bit
    - ❖ "Recursion desired" bit
    - ❖ ....



DNS Protocol

DNS *query* and *reply* messages

- Messages have a variable-length "question & answer" body
- Questions:
  - » The name and type fields (type A or MX) for a query — `hotmail.com MX`
- Answers:
  - » One RR for each IP address answering query
- Authority:
  - » Resource records of other authoritative servers



DNS Resource Records

dig query/reply message example

```
atria:[~]% /usr/bin/dig www.google.com

; <<>> DiG 9.8.1-P1 <<>> www.google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 65309
;; flags: qr rd ra; QUERY: 1, ANSWER: 6, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;www.google.com.                        IN      A

;; ANSWER SECTION:
www.google.com.         33      IN      A       173.194.73.105
www.google.com.         33      IN      A       173.194.73.147
www.google.com.         33      IN      A       173.194.73.106
www.google.com.         33      IN      A       173.194.73.103
www.google.com.         33      IN      A       173.194.73.104
www.google.com.         33      IN      A       173.194.73.99
```

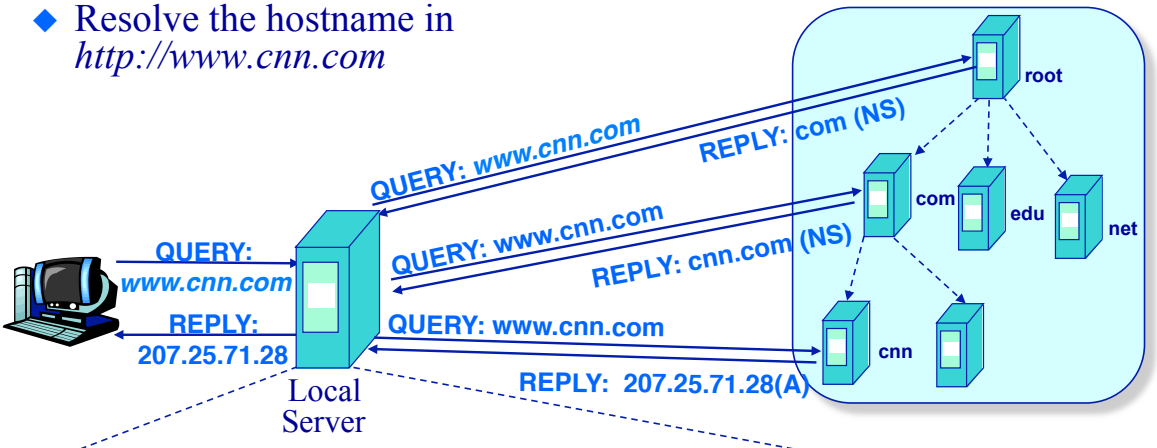

dig

/usr/bin/dig on CS Linux machines

- man dig
  - » at Unix command line
  - » or see http://linux.die.net/man/1/dig
- +norecurse
  - » run an iterative query
- *@servername*
  - » ask the specified server (instead of your local authoritative server)
- +trace
  - » make iterative queries and show results each time
- MX, CNAME, NS
  - » indicate type of query (default is A)

DNS Example
Empty Cache

- Resolve the hostname in *http://www.cnn.com*



com.	NS	A.GTLD-SERVERS.NET	192.5.6.30
cnn.com	NS	DNS.CNN.COM	128.2.232.1
www.cnn.com	A		207.25.71.28

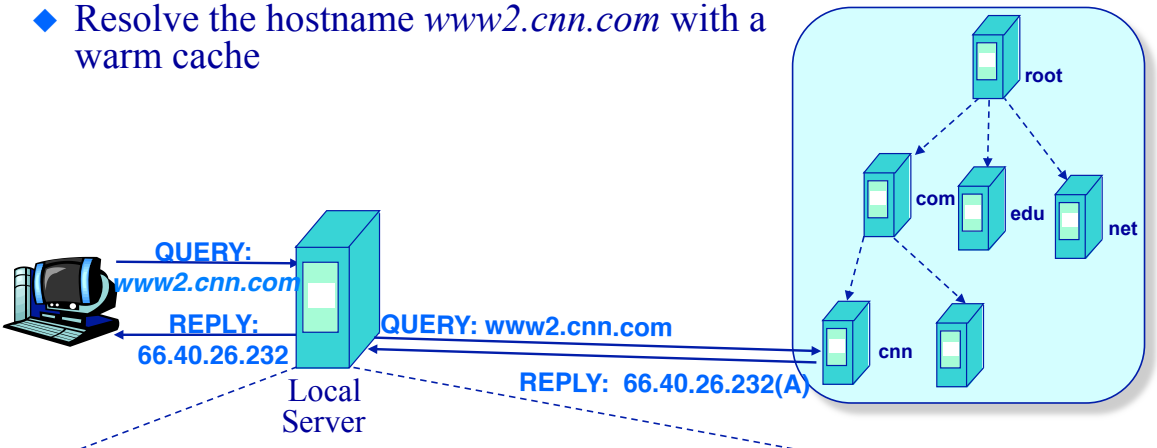
DNS Example
Empty Cache - Steps

- Client sends query www.cnn.com to local server
- Local server looks in cache, cache is empty
- Local server contacts root server
  - » root server responds with NS of .com domain
- Local server contacts .com NS
  - » .com NS responds with NS of cnn.com domain
- Local server contacts cnn.com NS
  - » cnn.com NS responds with A of www.cnn.com

- At every step, the information sent to the local server is stored in its cache

DNS Example
Warm Cache

- Resolve the hostname *www2.cnn.com* with a warm cache



com.	NS	A.GTLD-SERVERS.NET	192.5.6.30
cnn.com	NS	DNS.CNN.COM	128.2.232.1
www.cnn.com	A		207.25.71.28
www2.cnn.com	A		66.40.26.232

DNS Example
Warm Cache - Steps

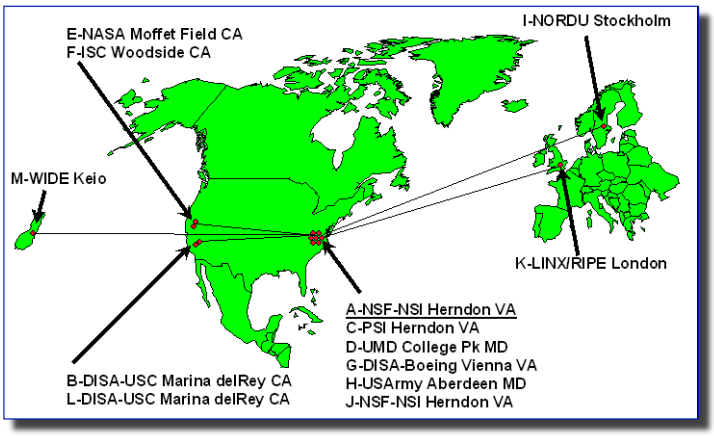
- Client sends query www2.cnn.com to local server
- Local server looks in cache
  - » doesn't find www2.cnn.com, but does find cnn.com NS
- Local server contacts cnn.com NS
  - » cnn.com NS responds with A of www2.cnn.com

- At every step, the information sent to the local server is stored in its cache

The Domain Name System

Summary

- F gets 270,000,000+ hits per day
  - » Other servers have comparable load
- The Verisign TLD servers answer 5,000,000,000 queries per day



- Clearly the DNS would collapse without:
  - » Hierarchy
  - » Distributed processing
  - » Caching
- If DNS fails, Internet services stop working!

The Domain Name System

For More Info

- DNS Specification
  - » RFC 1034, RFC 1035
- DNS Caching
  - » RFC 2136
- DNS Attacks
  - » CAIDA, Nameserver DoS Attack October 2002 analysis
    http://www.caida.org/projects/dns/dns-root-gtld/oct02dos.xml

Application-Layer Protocols
Outline

- The architecture of distributed systems
  - » Client/Server computing
  - » P2P computing
  - » Hybrid (Client/Server and P2P) systems
- The programming model used in constructing distributed systems
  - » Socket programming
- Example client/server systems and their application-level protocols
  - » The World-Wide Web (HTTP)
  - » Reliable file transfer (FTP)
  - » E-mail (SMTP & POP)
  - » Internet Domain Name System (DNS)