COMP 431

Internet Services & Protocols

# Applications & Application-Layer Protocols: The Domain Name System

*Jasleen Kaur*

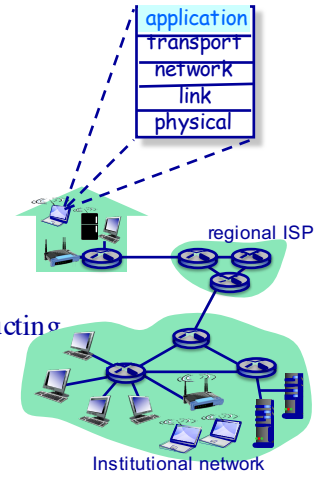
February 6, 2020



## Application-Layer Protocols

### Outline

- The architecture of distributed systems
  - » Client/Server computing
- 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)
- The programming model used in constructing distributed systems
  - » Socket programming



# 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

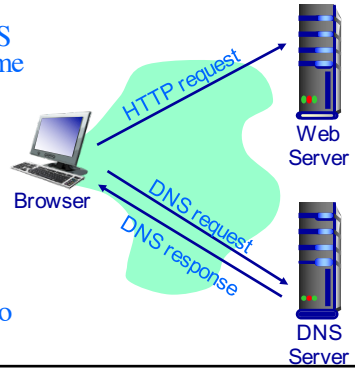


32 bit numbers are not easy to remember

# The Domain Name System

## Web browsing (HTTP) example

- The DNS is mainly used by applications, not end-users
  - » And virtually all applications use the DNS for every request they generate
- Web browsing: User enters URL *www.someSchool.edu*
  - » In order to create the socket to *www.someSchool.edu*, the OS (TCP) must resolve the hostname to an IP address
  - » The OS contacts a DNS name server to learn the web server's IP address
  - » The IP address is then used by TCP to create the socket to the server
  - » All this happens transparently to the user and the browser!






DNS - domain name system

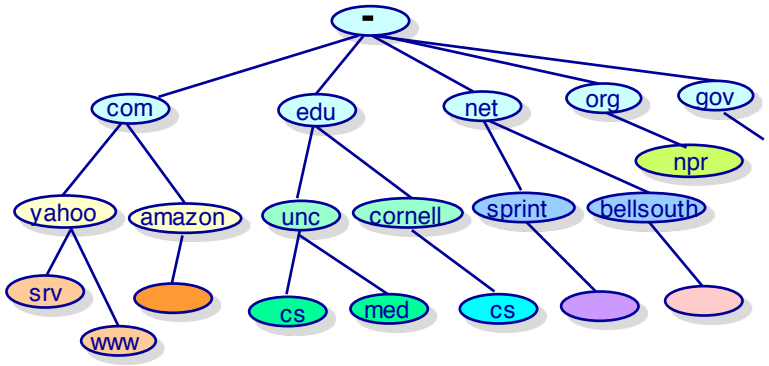
used to map domain names to ip addresses

decoupling allows the ip address and server to be changed whenever

whenever it is changed UNC or the entity would only have to go to its dns database and change the mappings

# The Domain Name System

## Name Hierarchy in DNS

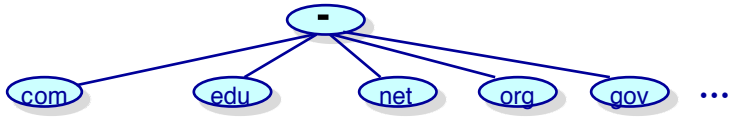



- *hostname* = "dot" separated concatenation of domain names along path toward the root
  - » *unc.edu*
  - » *cs.unc.edu*
  - » *classroom.cs.unc.edu*
- There are *name servers* associated with every domain



# Name Hierarchy in the DNS

## Top level domains




- Generic domains:
  - » (1980) *.com*, *.org*, *.net*, *.edu*, *.gov*, *.mil*, *.int*
  - » (2000) *.biz*, *.info*, *.name*, *.pro*
- Special sponsored names
  - » (2000) *.aero*, *.coop*, *.museum*
  - » (2003) *.asia*, *.cat*, *.jobs*, *.mobi*, *.tel*, *.travel*
- Country code domains
  - » *.uk*, *.de*, *.jp*, *.us*, … (250 more!)



Can be the same machine but just two wualitatively differen dervers

# Names Are Valuable

## And prices are "more" rational

GreatDomains℠
The Trusted Marketplace for Domain Names
A VeriSign® Company

| | |
|---|---|
| smoking.com | $500,000 |
| beef.com | $250,000 |
| sample.com | $90,000 |
| upscale.com | $80,000 |
| clerical.com | $75,000 |
| snake.com | $50,000 |
| barbecues.com | $30,000 |
| geeky.com | $25,000 |
| mime.com | SOLD! |
| dinner.com | $20,000 |

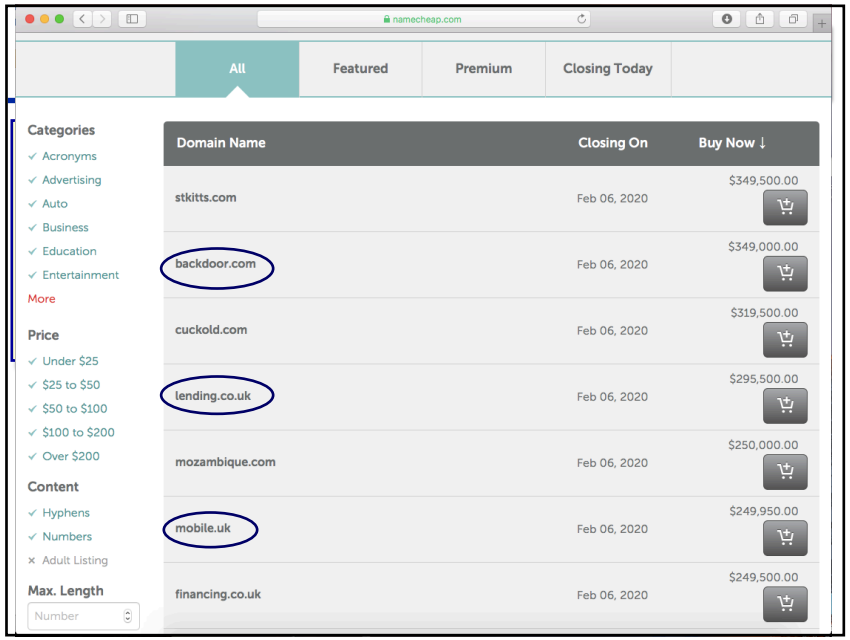
| | |
|---|---|
| now.tv | $150,000 |
| science.tv | $100,000 |
| british.tv | $25,000 |
| dancing.tv | $20,000 |
| cafes.tv | $10,000 |
| performer.tv | $10,000 |
| bowling.tv | $10,000 |
| merger.tv | $10,000 |
| article.tv | $5,000 |
| grandma.tv | $5,000 |

| | |
|---|---|
| career.net | $150,000 |
| invest.net | $75,000 |
| dunk.net | $50,000 |
| pornos.net | $35,000 |
| wholesale.net | $30,000 |
| exploring.net | $20,000 |
| bonded.net | $15,000 |
| worked.net | $10,000 |
| wealthy.net | $8,000 |
| russians.net | $7,000 |

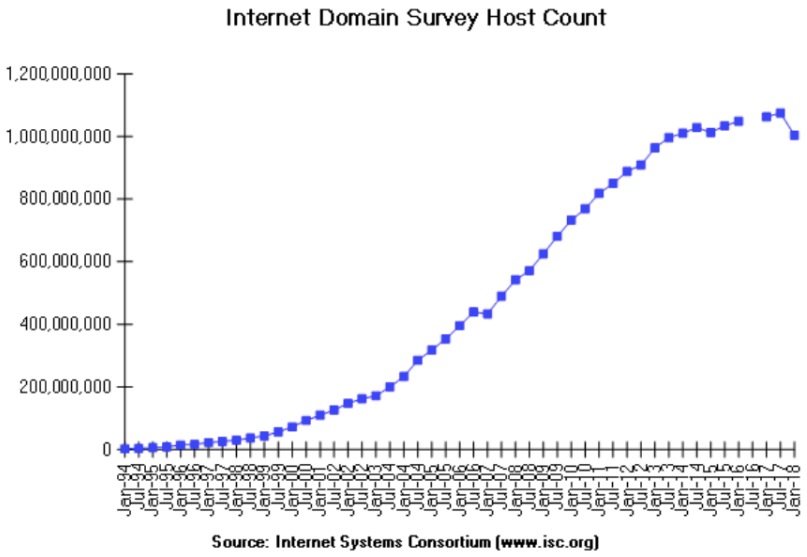
"New"

| | |
|---|---|
| mr.com | $350,000 |
| teeth.net | $55,000 |
| saving.net | $45,000 |
| pen.net | $35,000 |
| gems.org | $20,000 |
| train.net | $15,000 |
| equestrians.com | $10,000 |
| motorcars.tv | $10,000 |
| storms.us | $8,000 |
| burnable.net | $6,000 |

7

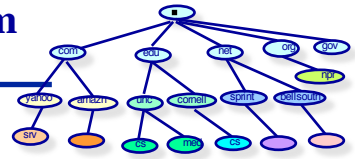


# Growth of DNS Registrations




# The Domain Name System

## Designing a distributed service

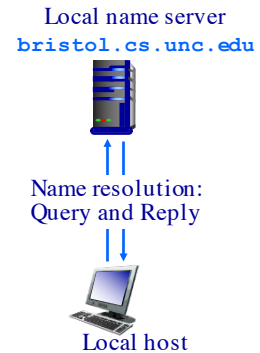



- Why not centralize the DNS
  - » A server process on a big, well connected supercomputer?
- Centralized systems do not scale!
  - » Poor reliability: centralized = single point of failure
  - » Poor performance: centralized = "remote access" for most users
  - » Difficult to manage: centralized = all customer 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

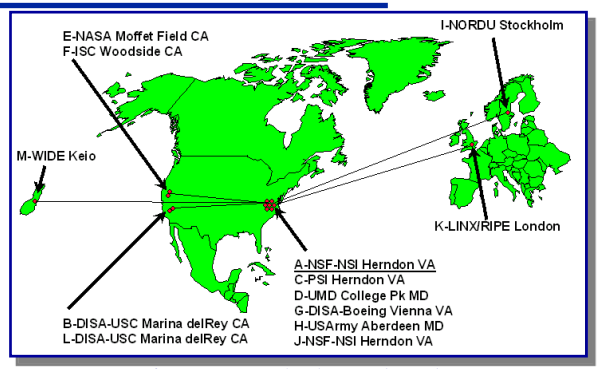



What if the name is not a local host (*e.g.*, `www.yahoo.com`)?



# DNS Name Servers

## Root name servers




- A root name server is contacted when a local name server that can't resolve a name
  - » The root server either resolves the name or provides pointers to authoritative servers at lower level of name hierarchy
- In 1998, there were a dozen root name servers worldwide

# DNS Name Servers

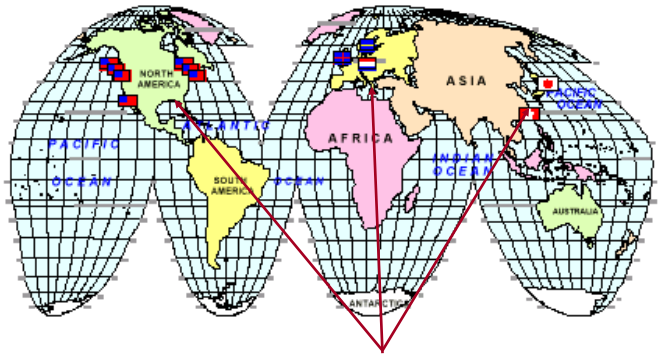
## 2011 Root name servers


- In 2011 there were a few more servers…



# DNS Name Servers

## Generic TLD servers (Verisign Corp.)

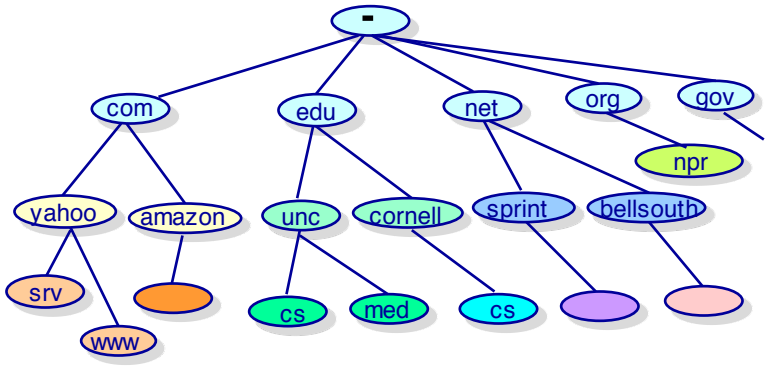



13 independent sites

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

# The Domain Name System

## Name Hierarchy in DNS



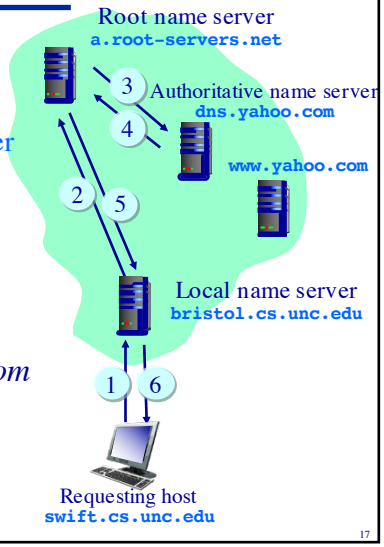

- *hostname* = "dot" separated concatenation of domain names along path toward the root
  - » *unc.edu*
  - » *cs.unc.edu*
  - » *classroom.cs.unc.edu*
- There are *name servers* associated with every domain



# DNS Name Servers

## Using a server hierarchy for resolving names

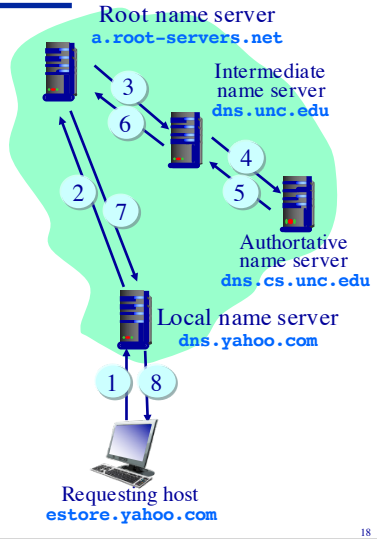
- Example: Host *swift.cs.unc.edu* wants to know the IP address of *www.yahoo.com*
  - » *Swift* contacts its local DNS server *bristol.cs.unc.edu*
- To resolve a non-local name the local name server queries the root server (if necessary)
- The root server contacts the authoritative server *dns.yahoo.com* (if necessary)
- Results propagate back to *swift*



# DNS Name Servers

## Using a server hierarchy for resolving names

- It's possible that the root name server may not know the authoritative name server for a domain
- The root server contacts an *intermediate* name server that knows the authoritative name server
- The intermediate name server contacts the authoritative name server
- Results propagate back to the requesting host

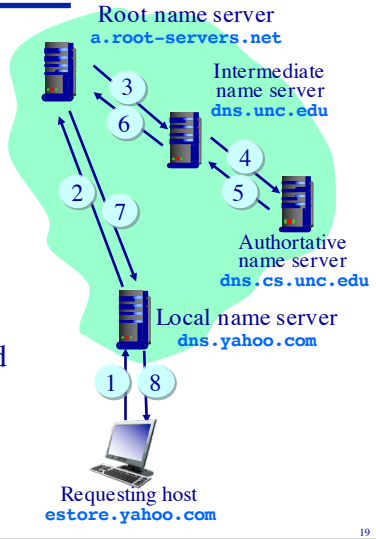





# DNS Name Servers

## Using a server hierarchy for resolving names

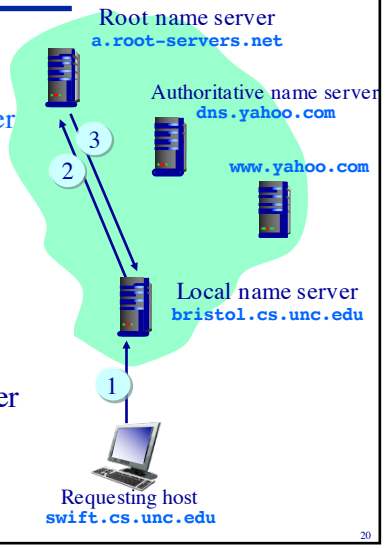
- The DNS supports two forms of queries:
  - » Recursive queries
  - » Iterative queries
- Recursive queries place the burden of name resolution (recursively) on the contacted server
- In an iterated query the contacted server simply replies with the name of the server to contact
  - » "I don't know;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



# DNS Name Servers

## Iterated queries

- *Swift* wants to know the IP address of *www.yahoo.com*
  - » *Swift* contacts its local DNS server *bristol.cs.unc.edu*
- If necessary, the local name server queries the root server
  - » "What server is the authority for *www.yahoo.com*?"
- The root server returns the name and IP address of the server it knows is the closest match to the query
  - » "Try **dns.yahoo.com**"

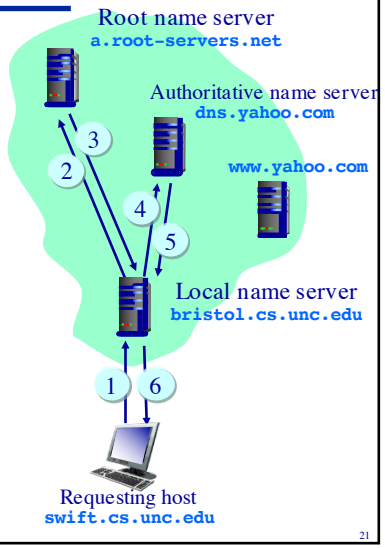





# DNS Name Servers

## Iterated queries

- The local DNS server sends the same query to the closest match server
  - » "What server is the authority for *www.yahoo.com*?"
- The process can be iterated until the local authoritative name server is found and responds
- (And iterated and recursive queries can be combined!)






More load on local server

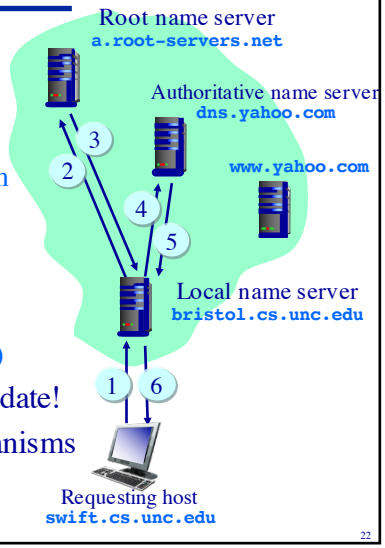
Local server can cache

response time includes network latency and server load

# DNS Name Servers

## Caching and updating DNS entries

- Every server caches all the mappings it learns
  - » TLD servers typically cached in local name servers
  - » (Thus root name servers not often queried)
- Cache entries are "soft state"
  - » They timeout (are deleted) after some time period
  - » Called the "time to live" ("TTL")
- So cached entries can be out of date!
- 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`

# The Domain Name System

## Inserting records into the DNS

www.networkutopia.com

- Example: New startup "Network Utopia"
- Register name *networkuptopia.com* at DNS registrar (*e.g.*, Network Solutions)
  - » You provide names & IP addresses of authoritative name server (primary and secondary)
  - » The registrar inserts two RRs into *.com* TLD server:
    - ❖ *networkutopia.com*, *dns1.networkutopia.com*, NS
    - ❖ *dns1.networkutopia.com*, 212.212.212.1, A
- You stand up *dns1.networkutopia.com* running BIND and create:
  - » Authoritative server type A record for *www.networkuptopia.com*
  - » MX record for *networkutopia.com*

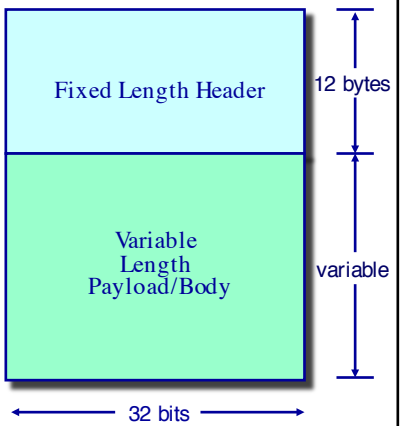
dns1.networkutopia.com



# The Domain Name System

## The DNS protocol

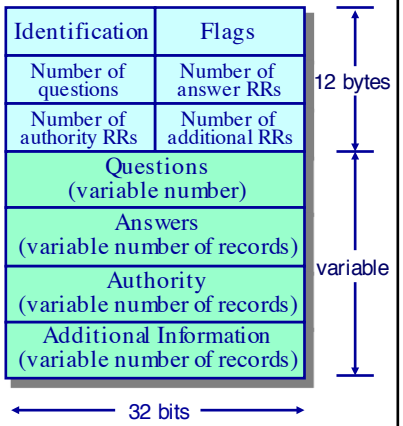
- The DNS service is implemented by the DNS protocol
- A request/response protocol run on top of UDP
  - » Uses port 53
- Why UDP?!
  - » Doesn't reliability matter?!



# 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 replies to queries
  - » Flags:
    - ❖ Query/Reply bit
    - ❖ "Reply is authoritative" bit
    - ❖ "Recursion desired" bit
    - ❖ ....

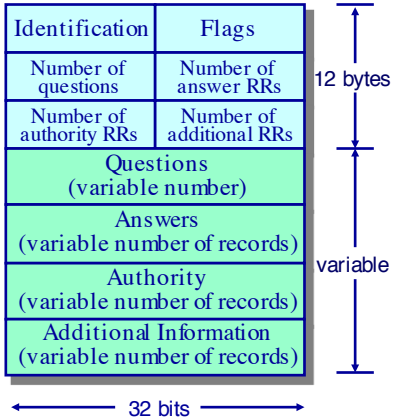





# DNS Protocol

## 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

## nslookup query/reply message example

```
(parris) 101> nslookup
> set debug
> www.yahoo.com
Server:  bristol.cs.unc.edu
Address:  152.2.131.228

QUESTIONS:
    www.yahoo.com,  type = A, class = IN
```



# DNS Resource Records

## nslookup query/reply message example

```
ANSWERS:
->  www.yahoo.com
    canonical name = www.yahoo-ht3.akadns.net
->  www.yahoo-ht3.akadns.net
    internet address = 69.147.114.210
AUTHORITY RECORDS:
->  akadns.net
    nameserver = zc.akadns.org.
->  akadns.net
    nameserver = zd.akadns.org.
->  akadns.net
    nameserver = eur1.akadns.net.
->  akadns.net
    nameserver = use3.akadns.net.
->  akadns.net
    nameserver = use4.akadns.net.
->  akadns.net
    nameserver = usw2.akadns.net.
->  akadns.net
    nameserver = asia9.akadns.net.
->  akadns.net
    nameserver = za.akadns.org.
->  akadns.net
    nameserver = zb.akadns.org.
```

# DNS Resource Records

## nslookup query/reply message example

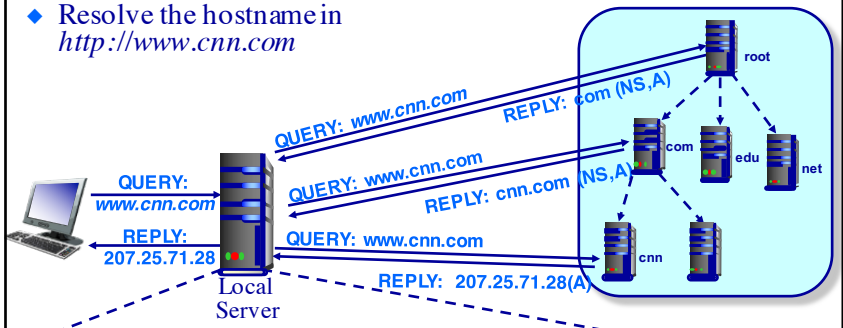
```
ADDITIONAL RECORDS:
->  za.akadns.org
    internet address = 195.219.3.169
->  zb.akadns.org
    internet address = 206.132.100.105
->  zc.akadns.org
    internet address = 124.211.40.4
->  zd.akadns.org
    internet address = 63.209.3.132
->  eur1.akadns.net
    internet address = 213.254.204.197
->  use3.akadns.net
    internet address = 204.2.178.133
->  use4.akadns.net
    internet address = 208.44.108.137
->  usw2.akadns.net
    internet address = 63.209.3.132
->  asia9.akadns.net
    internet address = 220.73.220.4
          Non-authoritative answer:
          www.yahoo.com    canonical name = www.yahoo-ht3.akadns.net.
          Name:    www.yahoo-ht3.akadns.net
          Address: 69.147.114.210
```



# DNS Example

## DNS processing for an iterated query

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

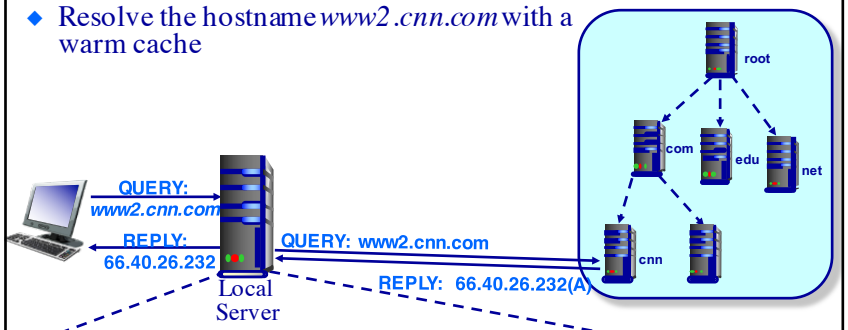



| | | | |
|---|---|---|---|
| *com.* | NS | A.GTLD-SERVERS.NET | (A) 198.41.0.1 |
| *cnn.com* | NS | DNS.CNN.COM | (A) 128.2.232.1 |
| *www.cnn.com* | A | | 207.25.71.28 |
| | | | |

# DNS Example

## DNS processing for an iterated query

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



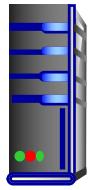

| | | | |
|---|---|---|---|
| *com.* | NS | A.GTLD-SERVERS.NET | (A) 198.41.0.1 |
| *cnn.com* | NS | DNS.CNN.COM | (A) 128.2.232.1 |
| *www.cnn.com* | A | | 207.25.71.28 |
| *www2.cnn.com* | A | | 66.40.26.232 |



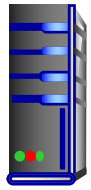
# The Domain Name System



## Attacking the DNS



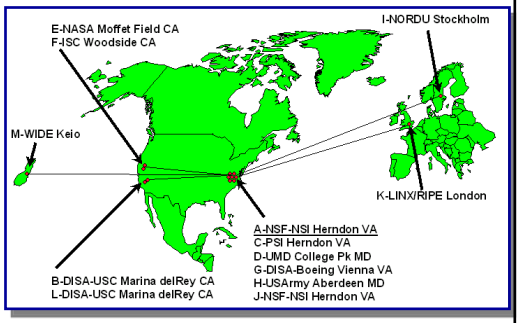
- DDoS attacks: Bombard root servers with requests
  - » Not successful to date(!)
  - » Defeated by traffic filtering
  - » Local DNS servers cache IPs of TLD servers, allowing root server bypass
  - » Bombard TLD servers — Potentially more dangerous
- Redirect attacks
  - » "Man-in-middle" (Intercept queries)
  - » DNS poisoning: Send bogus replies to a DNS server, which will cache them & return to others
- Exploit DNS for DDoS
  - » Send queries with spoofed source address!
  - » (Requires amplification)



# 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!