Welcome to

CS 3516: Computer Networks

Prof. Yanhua Li

Time: 9:00am –9:50am M, T, R, and F
Location: AK219
Fall 2019 A-term
Updates

- Quiz 3
  - On Friday
  - 1 bonus question
  - Topics: HTTP basics, cookies, RTT

- Project 1
  - Due Next Tuesday

- Extra office hours
  - Prof Li: Friday 9/6 11AM-12PM
  - Lei: Monday 9/9 10:30-11:30AM
Chapter 2: outline

2.1 principles of network applications
  - app architectures
  - app requirements

2.2 Web and HTTP

2.5 DNS
  Service Overview, Structure
  Resolution process
  Data Format
DNS: domain name system

**people:** many identifiers:
- SSN, name, passport #

**Internet hosts, routers:**
- IP address (32 bit) - used for addressing datagrams
- “name”, e.g., www.yahoo.com - used by humans

**Q:** how to map between IP address and name, and vice versa?

**Domain Name System:**
- *distributed database* implemented in hierarchy of many *name servers*
- *application-layer protocol:* hosts, name servers communicate to *resolve* names (address/name translation)
  - note: core Internet function, implemented as application-layer protocol
  - complexity at network’s “edge”
Resolving Name, Locating Service/Object

URL
http://users.wpi.edu/~yli15/courses/CS3516Fall19A/Schedule.html

WPI DNS Server

tcp port 80 121.121.121.121

web server

Network File System Server

Service → 121.121.121.121, tcp port 80
Object → ~yli15/courses/CS4516Fall15B/Schedule.html
DNS: services, structure

DNS services
- hostname to IP address translation
- host aliasing
  - canonical, alias names
- mail server aliasing
- load distribution
  - replicated Web servers: many IP addresses correspond to one name

why not centralize DNS?
- single point of failure
- traffic volume
- distant centralized database
- maintenance

A: doesn’t scale!
DNS: a distributed, hierarchical database

client wants IP for www.amazon.com; 1st approx:
- client queries root server to find com DNS server
- client queries .com DNS server to get amazon.com DNS server
- client queries amazon.com DNS server to get IP address for www.amazon.com

Analogy: Marshalls -> Physical Address
**DNS: root name servers**

- contacted by local name server that cannot resolve name
- root name server:
  - contacts authoritative DNS server if name mapping not known
  - gets mapping
  - returns mapping to local name server

13 root name “servers” worldwide:
TLD, authoritative servers

**top-level domain (TLD) servers:**
- responsible for com, org, net, edu, aero, jobs, museums, and all top-level country domains, e.g.: uk, fr, ca, jp
- **Network Solutions** maintains servers for .com TLD
- **Educause** for .edu TLD

**authoritative DNS servers:**
- organization’s own DNS server(s), providing authoritative hostname to IP mappings for organization’s named hosts
- can be maintained by organization or service provider
DNS: a distributed, hierarchical database

client wants IP for www.amazon.com; 1st approx:
- client queries root server to find com DNS server
- client queries .com DNS server to get amazon.com DNS server
- client queries amazon.com DNS server to get IP address for www.amazon.com

Analogy: Marshalls -> Physical Address
Local DNS name server

- does not strictly belong to hierarchy
- each ISP (residential ISP, company, university) has one
  - also called “default name server”
- when host makes DNS query, query is sent to its local DNS server
  - has local cache of recent name-to-address translation pairs (but may be out of date!)
  - acts as proxy, forwards query into hierarchy
- Difference btw Local DNS and Authoritative DNS server?
  - Given an organization, e.g., WPI, one for its internal users, one for external users
Chapter 2: outline

2.1 principles of network applications
   ▪ app architectures
   ▪ app requirements
2.2 Web and HTTP
2.5 DNS
   Service Overview, Structure
   Resolution process
   Data Format
DNS name resolution example

- host at \textit{cs.wpi.edu} wants IP address for \textit{cs.umass.edu}

**iterated query:**
- contacted server replies with name of server to contact
- “I don’t know this name, but ask this server”
**DNS name resolution example**

**Recursive query:**
- Puts burden of name resolution on contacted name server
- **Cons:** Heavy load at upper levels of hierarchy
DNS queries

**recursive query:**
- puts burden of name resolution on contacted name server
- heavy load?

**iterated query:**
- contacted server replies with name of server to contact
- “I don’t know this name, but ask this server”
DNS: caching, updating records

- once (any) name server learns mapping, it **caches** mapping
  - cache entries **timeout** (disappear) after some time (TTL, Time-to-Live)
  - **TLD servers typically cached** in local name servers
    - thus root name servers not often visited

- cached entries may be **out-of-date** (best effort name-to-address translation!)
  - if name host changes IP address, it may not be known Internet-wide until all TTLs expire
Chapter 2: outline

2.1 principles of network applications
   - app architectures
   - app requirements

2.2 Web and HTTP

2.5 DNS
   - Service Overview, Structure
   - Resolution process
   - Data Format
DNS records

**DNS**: distributed db storing resource records (RR)

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

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

**type=NS**
- *name* is domain (e.g., foo.com)
- *value* is hostname of authoritative name server for this domain

**type=CNAME**
- *name* is alias name for some “canonical” (the real) name
- *value* is canonical name

**type=MX**
- *value* is name of mailserver associated with *name*

- **www.ibm.com** is really servereast.backup2.ibm.com

- example.com is really serverwest.backup2.ibm.com
### DNS protocol, messages

- **query** and **reply** messages, both with same *message format*

**msg header**
- **identification**: 16 bit # for query, reply to query uses same #
- **flags**:  
  - query or reply
  - recursion desired (query)
  - recursion available (reply)
  - reply is authoritative (reply)  
  (DNS is an authoritative DNS to a queried name)

<table>
<thead>
<tr>
<th>Identification</th>
<th>Flags</th>
</tr>
</thead>
<tbody>
<tr>
<td># questions</td>
<td># answer RRs</td>
</tr>
<tr>
<td># authority RRs</td>
<td># additional RRs</td>
</tr>
</tbody>
</table>

- **questions** (variable # of questions)
- **answers** (variable # of RRs)
- **authority** (variable # of RRs)
- **additional info** (variable # of RRs)
## DNS protocol, messages

### Query:
- name, type fields for a query

### Reply:
- RRs in response to query
- records for authoritative servers
- additional “helpful” info that may be used

<table>
<thead>
<tr>
<th>Field</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>identification</td>
<td>2 bytes</td>
</tr>
<tr>
<td>flags</td>
<td>2 bytes</td>
</tr>
<tr>
<td># questions</td>
<td>2 bytes</td>
</tr>
<tr>
<td># answer RRs</td>
<td>2 bytes</td>
</tr>
<tr>
<td># authority RRs</td>
<td>2 bytes</td>
</tr>
<tr>
<td># additional RRs</td>
<td>2 bytes</td>
</tr>
<tr>
<td>questions (variable # of questions)</td>
<td></td>
</tr>
<tr>
<td>answers (variable # of RRs)</td>
<td></td>
</tr>
<tr>
<td>authority (variable # of RRs)</td>
<td></td>
</tr>
<tr>
<td>additional info (variable # of RRs)</td>
<td></td>
</tr>
</tbody>
</table>
Inserting records into DNS

- example: new startup “Networkabc”
- register name networkabc.com at DNS registrar (e.g., Network Solutions) (and pay a fee for it.)
  - provide names, IP addresses of authoritative name server (primary and secondary)
  - registrar inserts two RRs into .com TLD server:
    (networkabc.com, dns1.networkabc.com, NS)
    (dns1.networkabc.com, 212.212.212.1, A)

- Authoritative server
  - create type A record for www.networkabc.com;
  - create type MX record for networkabc.com
Attacking DNS

DDoS attacks

- Bombard root servers with traffic
  - Not successful to date
  - Traffic Filtering
  - Local DNS servers cache IPs of TLD servers, allowing root server bypass

- Bombard TLD servers
  - Potentially more dangerous
Questions?
Quiz 4 and Lab 2

Quiz 4, 9/12, Tuesday
  ▪ Topic: DNS

Lab 2: DNS
  Due 9/15 Friday at 23:59PM
  Link:
  https://users.wpi.edu/~yli15/courses/CS3516Fall17A/Assignments.html