1 EECS 122, Lecture 4

Kevin Fall

kfall@cs.berkeley.edu

2 Domain Name System

- Internet (IP) only understands addresses
- Naming easier for humans (e.g. files)
- Need a way to map names to numbers
- DNS (Domain Name System):
  - hierarchical distributed database
  - Internet *application* layer
  - see RFC 1034 and 1035

3 Naming

- Important theme in systems engineering
  - files in a file system
  - processes in operating system
  - web pages
  - printers and other services
- Name and location decoupling

4 Decoupling

- DNS provides a *level of indirection* between name and its location (solves any CS problem!)
- How to do this?
  - Flat vs hierarchical name space
  - Distributed vs centralized approach

5 Original Name System

- Flat name space:
  - simple (string, address) pairs
  - manual coordination
  - examples: ucbvax, sdcsvax, sri-nic
- Centralized Management
  - HOSTS.TXT file
  - single point of failure/update

6 Scaling Problems

- Name space overlap
  - had to coordinate with other users of name space to avoid overlap
  - greater management time
- Inconsistencies and poor performance
  - centralized updates
  - single point of failure
  - congestion at central point

7 DNS Approach

- Hierarchical, nested domain naming
- Distributed, recursive servers
- Basic ideas:
  - distribute name/address database across network hierarchically
  - implement query/response protocol
  - use caching heavily

8 Naming Hierarchy

9 Naming Conventions

- “Top-Level Domains” (TLD’s)
- Non-geographic
  - .COM, .NET, .ORG, .INT, .EDU, .MIL, .GOV
- Geographic
  - (baed on ISO3166 country codes)
  - .JP, .AU, .UK, .DE, .US, …
- The special “.ARPA” (reverse) domain

10 Naming Example

- www.cs.Berkeley.EDU
  - host: www, subdomain cs, domain: Berkeley.EDU
  - case insensitive
  - a “fully qualified” domain name (FQDN)
- Hierarchy is right-to-left with “.” delimiter
- Not necessarily tied to network topology/geography

11 DNS Components

- (Mockapetris & Dunlap, 1983, pub 1988)
- *Zones* contain *resource records* (RRs)
- *Name server(s)* manage each zone
- Client *resolvers* query name servers

12 Zones

- Complete description of a contiguous section of the total name space, plus some linkage info to other contig zones (separately administered DNS subtrees)
- Associated maintenance (>1 server)
- Zone transfers between redundant servers

13 Caching

- Servers and some clients *cache* data retrieved
- Resource records contain time-to-live (TTL), set by provider
- Higher TTL: less traffic, stale info
- Lower TTL: more traffic, current info

14 DNS Resource Records

- Components: owner (which domain), class (IN is only significant one), type, TTL

- Types:
  - A, CNAME, HINFO, MX, NS, SOA, PTR
- Record Data
  - variable length, specific to type

15 Address-related Types

- **A** type (internet address(es))

```
www.AAD        A 128.32.51.214
```

- **CNAME** type (alias(es))

```
boalt CNAME boalt363-001-d6.Law.Berkeley.EDU
```

- **PTR** type (used for reverse queries)

```
214.51  PTR  www.AAD.Berkeley.EDU
```

16 Authority Record

- **SOA** type (start of authority)
  - current serial number of zone data
  - refresh, retry, and expire info
  - Berkeley.EDU SOA ns1.Berkeley.EDU   dns-roadkill.NAK.Berkeley.EDU

```
   90001481        ; serial (vers)
   3600            ; refresh period
   900             ; retry refresh this often
   3600000         ; expiration period
   86400           ; minimum TTL
```

17 Name Server Records

- **NS** type (name server)
  - indicates authoritative name servers
  - used to construct the hierarchy

```
Berkeley.EDU.  NS   vangogh.CS.Berkeley.EDU
Berkeley.EDU.  NS   cgl.UCSF.EDU
CS             NS   vangogh.CS.Berkeley.EDU
CS             NS    nexus.EECS.Berkeley.EDU
```

18 Other Records

- **MX** type (mail exchanger)
  - indicates e-mail relay host and its preference

```
Berkeley.EDU.  MX   5 mailhost.Berkeley.EDU
```

- **HINFO** type (host info)
  - indicates OS or type of host

```
UCSD.EDU. HINFO      Sun        Unix
```

19 An Example (nslookup):

20 Locally-satisfied DNS query:

- User in domain “foo.com” asks for “bar”

21 Globally-satisfied DNS query:

- User in domain “foo.com” asks for “blah.baz.com

22 Reverse Queries

- Forward queries use domain name
- How to do reverse (addr-to-name) queries?
  - Addresses left-to-right, names right-to-left
  - Idea: REVERSE query
- Reverse network number, add “.IN-ADDR.ARPA” and perform PTR type query

23 Reverse Query Example

- Find the name of the host with address “208.212.172.33”
- This is a class “C” address, network 208.212.172.0
- So, look for the string “33.172.212.208.IN-ADDR.ARPA”:

```
33.172.212.208.IN-ADDR.ARPA  PTR  www.nsa.gov.
```

24 Bootstrapping

- How does local host locate name server?
  - Set up during host configuration
- How do servers locate root servers?
  - Set up during DNS configuration
  - 13 root servers ([a-m].root-servers.net)
  - root servers do not provide recursion

25 Negative Caching

- Caching works well for correct queries
- With many wrong queries, scaling is hurt:
  - cache negative queries also!
  - Covers both nonexistent domain names and nonexistent resource records
- See RFC 2308
  - Set up during host configuration

26 DNS Protocol

- DNS is an Application
- Uses both TCP and UDP for transport
  - UDP: used for most queries
  - TCP: used for zone transfers, and when UDP results indicated message was too big
- Use of UDP requires clients to implement their own reliability

27 DNS Lessons

- Naming was first show-stopping scaling problem
- Scaling problem addressed with:
  - caching
  - decentralization
  - hierarchy