DNS
CS640, 2015-04-07

Naming Hosts
● Thus far we have identified hosts using IP addresses and MAC address
  ○ Hard for humans to remember these identifiers
● Want to assign human readable names to hosts
  ○ Routing still needs IP addresses
  ○ Need a way to define and lookup the mapping between a hostname and an IP address
● Early Internet: a file mapping IP addresses to hostnames was manually updated and manually copied to all hosts in the Internet
  ○ Problem: does not scale
    ■ Still useful for small local networks -- look at the /etc/hosts file on a CS dept machine
● Domain name system (DNS)
  ○ Distributed name resolution system
    ■ Many name servers (NSs) distributed throughout the Internet -- in IPSs, in campus/enterprise networks, in department networks, etc.
  ○ Domain names (DNs) are hierarchical
    ■ A single NS doesn’t need to store the name for every host in the Internet
  ○ DNs can be mapped to IPv4 addresses, IPv6 addresses, and other DNs
    ■ Mapping can be changed over time, or based on other factors (e.g., geo location)
  ○ Queries are issued to a sequence of NSs
    ■ Each knows about a different part of the DN hierarchy
    ■ Answers can be cached to avoid the overhead of frequent lookups

Domain Name Hierarchy
● DNs are processed right to left, with periods as the separator
  ○ Rightmost name is at the top of the hierarchy, and leftmost is at the bottom

● Rightmost name is referred to as the top-level domain (TLD)
  ○ Since DNS was invented in the US, originally TLDs were designed for the US -- edu, com, gov, mil, org, net
  ○ Expanded to include TLDs for countries -- uk, cn, etc.
    ■ Sometimes TLDs from other countries are used such that the full DN is a specific word or phrase -- e.g., bit.ly uses the ly TLD for the country Libya,
  ○ Expanded to address high demand for .com -- .biz, .info, .tv, etc.
  ○ Recently expanded to include arbitrary TLDs
    ■ Lots of contention over who should have the rights to a specific TLD
● Second from right is the second-level domain (SLD)
● DN consisting of 3+ names is often referred to as a subdomain
  ○ E.g., cs.wisc.edu is a subdomain of wisc.edu
• Complete hierarchy only exists conceptually -- no single DNS server stores the entire hierarchy
• How do we divide responsibility for different parts of the hierarchy to different DNS servers?

Zones
• Zone: a portion of the hierarchy that is managed by administrative entity
  ○ Internet Corporation for Assigned Names and Numbers (ICANN) is responsible for zone containing all TLDs
  ○ UW (DoIT) is responsible for zone containing wisc.edu and some subdomains (e.g., doit.wisc.edu)
  ○ CS department is responsible for zone containing cs.wisc.edu and all subdomains (e.g., cardinal.cs.wisc.edu, www.cs.wisc.edu, etc.)
  ○ Google is responsible for zone containing google.com and all subdomains (e.g., maps.google.com, scholar.google.com, mail.google.com, etc.)
• Two or more name servers (NSs) are responsible for each zone
  ○ Multiple NSs per zone to ensure availability in case of a failure
  ○ Each stores information for all domain names in the zone
  ○ Example

DNS Records
• Record has: name, type, value, and TTL
• Name = DN
• Types
  ○ A -- value is an IPv4 address
  ○ AAAA -- value is an IPv6 address
  ○ NS -- value is domain name for a DNS server that is responsible for the zone containing the DN
  ○ CNAME -- value is another domain name for a particular host
  ○ MX -- value is the domain name for a mail server that accepts messages for the DN
• TTL specifies how long another DNS server can cache the record
Example

- Root NS
  - edu, NS, a.edu-servers.net
  - edu, NS, c.edu-servers.net
  - ...
  - a.edu-servers.net, A, 192.5.6.30
  - c.edu-servers.net, A, 192.26.92.30
  - ...
- edu NS
  - wisc.edu, NS, adns1.doit.wisc.edu
  - wisc.edu, NS, adns3.doit.wisc.edu
  - wisc.edu, NS, dns2.itd.umich.edu
  - ...
  - adns1.doit.wisc.edu, A, 144.92.9.21
  - adns3.doit.wisc.edu, A, 144.92.104.21
  - adns3.doit.wisc.edu, AAAA, 2607:f388::a53:3
  - dns2.itd.umich.edu, A, 192.12.80.222
  - ...
  - cmu.edu, NS, NSAUTH1.net.cmu.edu
  - ...
  - NSAUTH1.net.cmu.edu, A, 128.2.1.8
  - NSAUTH1.net.cmu.edu, AAAA, 2607:fb28::4
  - ...
- wisc NS
  - cs.wisc.edu, NS, dns.cs.wisc.edu.
  - cs.wisc.edu, NS, dns2.cs.wisc.edu
  - ...
  - dns.cs.wisc.edu, A, 128.105.2.10
  - dns2.cs.wisc.edu, A, 128.105.6.12
  - ...
- cs NS
  - www.cs.wisc.edu, A, 128.105.7.31
  - cs.wisc.edu, MX, granite.cs.wisc.edu
  - cs.wisc.edu, MX, obsidian.cs.wisc.edu
  - ...
  - granite.cs.wisc.edu, A, 128.105.6.24
  - obsidian.cs.wisc.edu, A, 128.105.6.13
  - ...
Name resolution

- Algorithm
  - Client contacts local NS
    - Local NS is provided to client by DHCP or set in client configuration
    - Local NS contacts root name server
  - Root NS provides NS & A records for NS that can resolve TLD
  - NS for TLD provides NS & A record for NS that can resolve SLD
  - Local NS contacts NS for SLD
  - NS for SLD provides A record for domain, or NS & A for NS that can resolve domain
  - ...

- Local DNS server will cache any records it receives

- Example:
  - Resolve cardinal.cs.wisc.edu
  - Resolve stat.cmu.edu
  - Resolve mail.google.com
    - Google NS
      - mail.google.com, CNAME, googlemail.l.google.com
      - googlemail.l.google.com, A, 74.125.225.53
      - googlemail.l.google.com, A, 74.125.225.54
      - ...
      - maps.google.com, A, 64.15.120.20
      - maps.google.com, A, 64.15.120.21
      - ...
  - Resolve email server for cs.wisc.edu
    - Assume query for cardinal.cs.wisc.edu has already happened and local NS has cached some records

- May be fewer/more interactions with NSs depending on
  - How many parts there are to the DN (e.g., wisc.edu vs. cardinal.cs.wisc.edu)
  - How many levels in the name hierarchy are in the same zone (e.g., wisc.edu and doit.wisc.edu are in the same zone, while wisc.edu and cs.wisc.edu are in different zones)
  - Whether there are CNAMEs that require contacting a different NS
  - What records the local NS has already cached

Advanced Name Resolution

- NS can return different sets of records for different queries
  - Use for load balancing or geo-based server selection

- Load balancing
  - Assign short TTL to records
  - Return different A records for each query for a DN -- cycle through A records in weighted round-robin order; weight is based on server load

- Geo-based server selection
  - Used by content distribution networks (CDNs)
  - CDN’s NS is configured with approximate geo-location of certain IP blocks
  - Address of NS that issued query is compared against IP blocks to determine rough location of client that issued query
  - Based on location, CDN’s NS returns CNAME record whose value is DN for nearby server
Example: *google.com*
- Local NS contacts root NS
- Root NS provides NS & A records for com NS
- Local NS contacts com NS
- com NS provides NS & A record for Google NS -- e.g., *ns1.google.com, 216.239.32.10*
- Local NS contacts Google NS
- Google NS looks at source IP for query and provides different address based on estimated location of source IP
  - from home (Charter): 64.15.120.52
  - from Milwaukee (AT&T DSL): 74.125.225.32
  - from Los Angeles: 74.125.239.161

**Content Centric Networking**
- Emerging research focuses on naming content rather than naming hosts
- Same content (e.g. a specific research paper) could reside in multiple different locations (e.g., on a conference website, on the author’s website, and on a digital library website)
- Any host with the content can provide it to the user
- Challenges
  - How do we address the content?
  - How do we determine where the content is located?
  - How do we determine if two pieces of data have the same content? -- e.g., is a video on YouTube and a video on Vimeo the same, but just encoded differently?
  - Can we performing routing based on content identifiers instead of based on IP addresses?