CSE/EE 461 – Lecture 22

Naming and the DNS

David Wetherall
djw@cs.washington.edu

Last Time

• Network support for QOS

• Focus
  – What network mechanisms provide which kinds of quality assurances?

• Topics
  – Scheduling and Buffer management
  – Fair Queuing
  – Intserv
  – Diffserv
This Lecture

- Naming

- Focus
  - How do we name hosts etc.?

- Topics
  - Domain Name System (DNS)
  - Email/URLs

Names and Addresses

- Names are identifiers for objects/services (high level)
- Addresses are locators for objects/services (low level)
- Resolution is the process of mapping name to address
- But, addresses are really lower-level names; many levels used
Naming in Systems

• Ubiquitous
  – Files in filesystem, processes in OS, pages on the web, ...

• Decouple identifier for object/service from location
  – Hostnames provide a level of indirection for IP addresses

• Naming greatly impacts system capabilities and performance
  – Ethernet addresses are a flat 48 bits
    • flat ➔ any address anywhere but large forwarding tables
  – IP addresses are hierarchical 32/128 bits
    • hierarchy ➔ smaller routing tables but constrained locations

Internet Hostnames

• Hostnames are human-readable identifiers for end-systems based on an administrative hierarchy
  – galah.cs.washington.edu is my desktop machine

• IP addresses are a fixed-length binary encoding for end-systems based on their position in the network
  – 128.95.2.106 is galah’s IP address

• Original name resolution: HOSTS.TXT
• Current name resolution: Domain Name System
• Future name resolution: ?
Original Hostname System

- When the Internet was really young …

- Flat namespace
  - Simple (host, address) pairs

- Centralized management
  - Updates via a single master file called HOSTS.TXT
  - Manually coordinated by the Network Information Center (NIC)

- Resolution process
  - Look up hostname in the HOSTS.TXT file

Scaling Problems

- Coordination
  - Between all users to avoid conflicts

- Inconsistencies
  - Between update and distribution of new version

- Reliability
  - Single point of failure

- Performance
  - Competition for centralized resources
Domain Name System (DNS)

- Designed by Mockapetris and Dunlap in the mid 80s

- Namespace is hierarchical
  - Allows much better scaling of data structures
  - e.g., galah.cs.washington.edu

- Namespace is distributed
  - Decentralized administration and access
  - e.g., galah managed by CSE

- Resolution is by query/response
  - With replicated servers for redundancy
  - With heavy use of caching for performance

DNS Hierarchy

- “dot” is the root of the hierarchy
- Top levels now controlled by ICANN
- Lower level control is delegated
- Usage governed by conventions
- FQDN = Fully Qualified Domain Name
**DNS Distribution**

- Data managed by **zones** that contain resource records
  - Zone is a complete description of a portion of the namespace
  - e.g., all hosts and addresses for machines in washington.edu with pointers to subdomains like cs.washington.edu

- One or more nameservers manage each zone
  - Zone transfers performed between nameservers for consistency
  - Multiple nameservers provide redundancy

- Client resolvers query nameservers for specified records
  - Multiple messages may be exchanged per DNS lookup to navigate the name hierarchy (coming soon)

---

**DNS Lookups/Resolution**

- DNS queries/responses carried on UDP port 53

---
Caching

- Servers and clients cache results of DNS lookups
  - Cache partial results too (e.g., server for princeton.edu)
  - Greatly improves system performance; lookups the rare case

- Cache using time-to-live (TTL) value from provider
  - higher TTL means less traffic, lower TTL means less stale info

- Negative caching is used too!
  - errors can cause repeated queries for non-existent data
DNS Bootstrapping

- Need to know IP addresses of root servers before we can make any queries

- Addresses for 13 root servers ([a-m].root-servers.net) handled via initial configuration (named.ca file)

Building on the DNS

- Other naming designs leverage the DNS

  - Email:
    - e.g., [djw@cs.washington.edu](mailto:djw@cs.washington.edu) is djw in the domain cs.washington.edu

  - Uniform Resource Locators (URLs) name for Web pages
    - e.g., [www.cs.washington.edu/homes/djw](http://www.cs.washington.edu/homes/djw)
    - Use domain name to identify a Web server
    - Use “/” separated string to name path to page (like files)
Future Evolution of the DNS

- Design constrains us in two major ways that are increasingly less appropriate
  - Static host to IP mapping
    - What about mobility (Mobile IP) and dynamic address assignment (DHCP)
  - Location-insensitive queries
    - What if I don’t care what server a Web page comes from, as long as it’s the right page?
    - e.g., a yahoo page might be replicated

Akamai

- Use the DNS to effect selection of a nearby Web cache
  - Leverage separation of static/dynamic content
  - Beware DNS caching
**Key Concepts**

- The design of names, addresses and resolution has a significant impact on system capabilities

- Hierarchy, decentralization and caching allow the DNS to scale
  - These are general techniques!