CS4700/CS5700
Fundamentals of Computer Networks

Lecture 17: Domain Name System

Slides used with permissions from Edward W. Knightly,
T. S. Eugene Ng, Ion Stoica, Hui Zhang
Human Involvement

• Just like your friend needs to tell you his phone number for you to call him
• Somehow, an application needs to know the IP address of the communication peer
• There is no magic, some out-of-band mechanism is needed
  – Word of mouth
  – Read it in the advertisement in the paper
  – Etc.
• But IP addresses are bad for humans to remember and tell each other
• So need names that makes some sense to humans
Internet Names & Addresses

• Names: e.g. www.rice.edu
  – human-usable labels for machines
  – conforms to “organizational” structure

• Addresses: e.g. 128.42.247.150
  – router-usable labels for machines
  – conforms to “network” structure

• How do you map from one to another?
  – Domain Name System (DNS)
DNS: History

• Initially all host-address mappings were in a file called hosts.txt (in /etc/hosts)
  – Changes were submitted to SRI by email
  – New versions of hosts.txt ftp’d periodically from SRI
  – An administrator could pick names at their discretion
  – Any name is allowed: eugenesdesktoppatrice

• As the Internet grew this system broke down because:
  – SRI couldn’t handled the load
  – Hard to enforce uniqueness of names
  – Many hosts had inaccurate copies of hosts.txt

• Domain Name System (DNS) was born
Basic DNS Features

• Hierarchical namespace
  – as opposed to original flat namespace

• Distributed storage architecture
  – as opposed to centralized storage (plus replication)

• Client--server interaction on UDP Port 53
  – but can use TCP if desired
Naming Hierarchy

- "Top Level Domains" are at the top
- Depth of tree is arbitrary (limit 128)
- Domains are subtrees
  - E.g: .edu, rice.edu, ece.rice.edu
- Name collisions avoided
  - E.g. rice.edu and rice.com can coexist, but uniqueness is job of domain
Host names are administered hierarchically.

A **zone** corresponds to an administrative authority that is responsible for that portion of the hierarchy.

E.g. Eugene controls names: x.cs.rice.edu and y.ece.rice.edu

E.g. The President controls names: x.rice.edu and y.owlnet.rice.edu
Server Hierarchy

• Each server has authority over a portion of the hierarchy
  – A server maintains only a subset of all names

• Each server contains all the records for the hosts or domains in its zone
  – might be replicated for robustness

• Every server knows the root

• Root server knows about all top-level domains
DNS Name Servers

- Local name servers:
  - Each ISP (company) has local default name server
  - Host DNS query first goes to local name server
  - Local DNS server IP address usually learned from DHCP
  - Frequently cache query results

- Authoritative name servers:
  - For a host: stores that host’s (name, IP address)
  - Can perform name/address translation for that host’s name
DNS: Root Name Servers

- Contacted by local name server that can not resolve name
- Root name server:
  - Contacts authoritative name server if name mapping not known
  - Gets mapping
  - Returns mapping to local name server
- ~ Dozen root name servers worldwide
Basic Domain Name Resolution

• Every host knows a local DNS server
  – Through DHCP, for example
  – Sends all queries to a local DNS server

• Every local DNS server knows the ROOT servers
  – When no locally cached information exists about the query, talk to a root server, and go down the name hierarchy from the root
  – If we lookup www.rice.edu, and we have a cached entry for the .edu name server, then we can go directly to the .edu name server and bypass the root server
Example of Recursive DNS Query

Root name server:
• May not know authoritative name server
• May know intermediate name server: who to contact to find authoritative name server?

Recursive query:
• Puts burden of name resolution on contacted name server
• Heavy load?

How does moe knows reply #7 is for frosty?
Example of Recursive DNS Query

Root name server:
- May not know authoritative name server
- May know intermediate name server: who to contact to find authoritative name server?

Recursive query:
- Puts burden of name resolution on contacted name server
- Heavy load?

How does moe knows reply #7 is for frosty?
Randomly choose an ID and use in all related messages to match replies to original queries
Example of Iterated DNS Query

Iterated query:
- Contacted server replies with name of server to contact
- “I don’t know this name, but ask this server”

This is how today’s DNS system behaves
DNS Resource Records

- DNS Query:
  - Two fields: (name, type)

- Resource record is the response to a query
  - Four fields: (name, value, type, TTL)
  - There can be multiple valid responses to a query

- Type = A:
  - name = hostname
  - value = IP address
DNS Resource Records (cont’d)

- **Type = NS:**
  - name = domain
  - value = name of dns server for domain

- **Type = CNAME:**
  - name = hostname
  - value = canonical name

- **Type = MX:**
  - name = domain in email address
  - value = canonical name of mail server and priority
DNS as Indirection Service

• Can refer to machines by name, not address
  – Not only easier for humans
  – Also allows machines to change IP addresses without having to change way you refer to machine

• Can refer to machines by alias
  – www.rice.edu can be generic web server
  – But DNS can point this to particular machine that can change over time

• Can refer to a set of machines by alias
  – Return IP address of least-loaded server
DNS Security

• Man-in-the-middle

• Spoofing responses
  – eavesdrop on requests and race the real DNS server to respond

• Name Chaining
  – AKA cache poisoning
  – Attacker waits for a query, injects possibly unrelated response resource record (frequently NS or CNAME record)
  – Bogus record is cached for later use
Cache Poisoning Attack

- Until the TTL expires, queries to moe.rice.edu for ebay.com's nameserver will return the poison entry from the cache.
Solution: DNSSEC

- Cryptographically sign critical resource records
  - Resolver can verify the cryptographic signature

- Two New resource record type:

  - Type = KEY:
    - name = Zone domain name
    - value = Public key for the zone

  - Type = SIG:
    - name = (type, name) tuple (i.e. a query)
    - value = Cryptographic signature of query result
Security 2: Denial of Service

- Caching can mitigate the effect of a large-scale denial of service attack

- October 2002: root name servers subjected to massive DoS attack
  - By and large, users didn’t notice
  - Locally cached records used

- More directed denial of service can be effective
  - DoS local name server → cannot access DNS
  - DoS authoritative names server → cannot access domain
Special Topics

• DNS caching
  – Improve performance by saving results of previous lookups
  – E.g. results of address records and name server records
    (e.g. if .edu name server is cached, then can bypass root
    server the second time looking for a .edu host)

• DNS “hacks”
  – Return records based on requesting IP address
  – Round-robin DNS

• Dynamic DNS
  – Allows remote updating of IP address for mobile hosts

• DNS politics (ICANN) and branding battles
More Special Topics

- Suppose you want your own top-level domain…
- AlterNIC, OpenNIC et al.
  - “Alternative” DNS hierarchies, with their own top-level domains
    - (.glue, .geek, .fur, .indy, …)
  - Can return results from the “regular” DNS hierarchy where there is no collision
    - (.biz)