Massachusetts Institute of Technology
Department of Electrical Engineering and Computer Science

6.892 Fall 1999 **Domain Name SYstem** November 24, 1999

Overview. DNS architecture.

1 Naming

- Provides a level of indirection, a handle to something—the mapping from name to location can change without affecting an application that wants to locate the entity.
- Human convenience.
- Namespace: set of names in the system typically organized as a database. Can be *flat* or *hierarchical.*
- A flat namespace is a flat database of (name, address) pairs typically managed by a central authority (e.g., SRI-NIC in the old Internet that managed HOSTS.TXT). Doesn't scale.
- A hierarchical namespace defines nested, hierarchical domains with decentralized administration. Scales a lot better.

2 DNS

- Paper describes system designed in 1983/84.
- Primary design goals: scalability, decentralized maintenance, robustness.
- Traditional database semantics like atomicity, consistency etc. are relaxed.
- Small degree of replication, aggressive use of caching (contrast with Grapevine system, which also had a fixed hierarchy).
- Architecture:
  - Hierarchical with "." as separator.
  - Inverted path names, read from leaf to root.
  - Top-level domains like edu, gov, com, ca, us, in, etc.
  - *Sub-domains* are sub-trees.
  - A *zone* is a contiguous segment of the hierarchy; one or more name servers manage a zone.
  - Extension of hierarchy and zone creation done by negotiation with parent.
  - In each zone, *primary* and *secondary* name servers. Secondaries updated by *zone transfers.*

- Bootstrap: well-known *root servers.* Today, there are 13—$\{a\text{-}m\}$*.root-server.net.* Root zone contains all top-level domains.
- DB contains tuples called *resource records.* Contains names, values and attributes.
- Resource records are broken into *classes.* Classes are defined for Internet (IN), Chaosnet (CH), Hesiod (HS) and the obsolete CSNet (CS). Notice the MIT influence on this!
- Each class can define its own type attributes. IN defines several, including A (address), NS (authoritative name server for name), CNAME (canonical name, useful for aliasing), HINFO (CPU and OS of machine), MX (mail exchanger for processing and forwarding mail from/to host), PTR (for reverse mappings from address to name).
- Hosts run a *resolver* to resolve name to information. This is usually a library that apps can link.
- The highest-level idea in DNS resolution is to walk from root to sought sub-domain that has information about a name.
- Steps in a typical resolution (if no caching were present)
  - App calles resolver function, e.g., gethostbyname()
  - Resolver library contacts local name server. This is hand-configured in a file like /etc/resolv.conf or obtained via DHCP, etc.
  - Local server queries root server S_1 for name (e.g., daedalus.cs.berkeley.edu)
  - S_1 refers to next-level server S_2, gives us NS record for berkeley.edu
  - Local server queries S_2 for daedalus.cs.berkeley.edu
  - S_2 refers local server to S_3, server for cs.berkeley.edu
  - Local server queries S_3 for daedalus.cs.berkeley.edu
  - S_4 returns required RR info (e.g., A record, MX record, etc.)
- The above shows two types of resolution: *iterative* and *recursive.* The resolver library is invoking a recursive resolution. The local name server invokes several iterative resolutions on S_1, S_2, etc.
- A recursive resolution returns an answer or a "not found." An iterative one can return a "best" referral.
- Address to name resolution: in-addr.arpa special domain (e.g., 82.0.31.18.in-addr.arpa. PTR wind.lcs.mit.edu).
- Negative caching: for malformed queries. Occur because of mistyping (less common today than before?) and search rules.
- Lessons and critique
  - Bad state can get stuck in the system for a while.
  - Plan to include "users" failed (needed today!)
  - Quick updates problematic.
  - Protocol itself doesn't have good upgrade path (e.g., no version number).

- Sample config file had bad defaults!
- Blocking call is an annoyance in many contexts.
- MX records treated specially (ftp, web, etc.?)
- DNSSEC and DNS Notify done more recently.