THE DOMAIN NAME SYSTEM DNS

Internet Protocols

CSC / ECE 573

Fall, 2005

N. C. State University

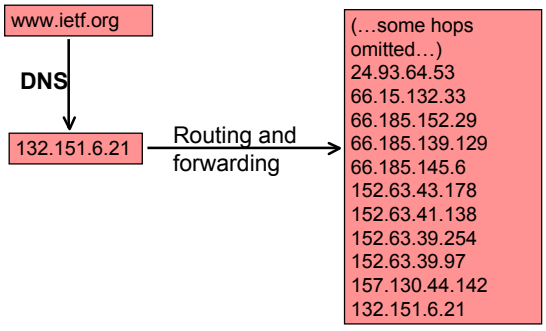
Announcements

Today's Lecture

I. Names vs. Addresses
II. The Namespace Hierarchy
III. Management of the Namespace
IV. Contents of the DNS Database
V. DNS Queries
VI. DNS Messages
VII. DNS Security

NAMES vs. ADDRESSES

How Do I Get to 132.152.6.21?



Mapping of Names to Addresses

- Users refer to hosts by names, while IP addresses are binary numbers
  - need a mapping of name(s) ↔ IP address(es)
- Approaches that will not scale to Internet size…
  - a single file per host, containing all mappings
  - a "flat" name space with no structure
  - centralized management of the entire name space

The Domain Name System (RFCs 1034, 1035)

- A hierarchical naming scheme
  - name syntax
  - rules for delegating authority over names
- A distributed database system
- A protocol for requesting address bindings
- Main function is mapping names to IP addresses, but can be used for many other purposes as well

Mapping Names to Addresses

An application calls a library function, the *resolver*

↓

The resolver sends a UDP packet to a local DNS server

↓

The DNS server looks up the name (may need to contact other servers to find the mapping)

↓

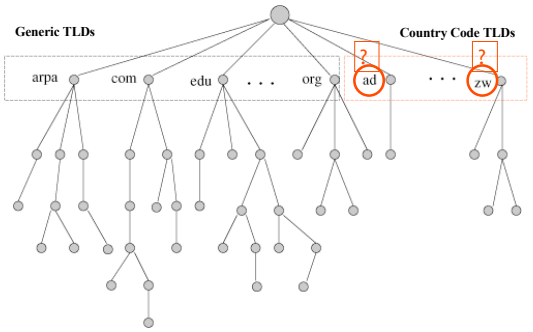
The DNS server returns the IP address to the resolver, which passes it to the application

THE NAMESPACE HIERARCHY

The Naming Hierarchy

- DNS uses a hierarchical delegation of authority
  - the name space is split at the top level
  - topmost level is not concerned with name assignments/changes within a subdivision
  - authority may be further subdivided at each level
- The name space hierarchy does not have to follow the routing hierarchy; they are independent!
  - a domain ≠ a contiguous chunk of address space
- The name space (tree) is broad and flat, usually no more than 4-5 levels deep

A Hierarchy Example

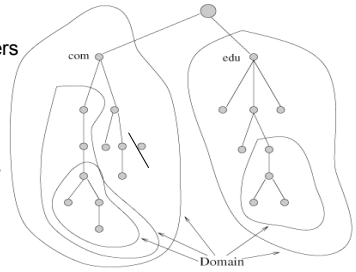


Domain Name Space: Labels

- Each node in the tree has a *label*
  - a string of up to 63 characters
  - no two children of a single node may have the same labels

Domains

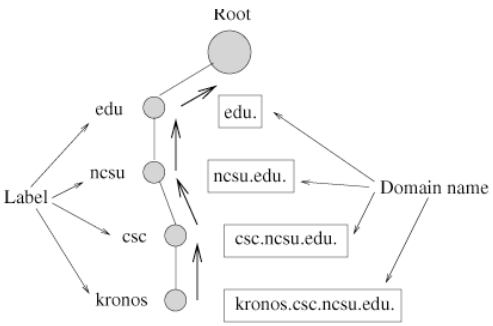
- *Domain*: a subtree of the domain name space
  - each domain covers many hosts
- Top-level domain (TLD): a domain starting just below root



Domain Name Space: Domain Names

- Each node in the tree has a *domain name*
  - a sequence (from the node up to the root) of dot-separated labels (e.g., mulberry.csc.ncsu.edu.)
  - max of 255 characters in a *fully-qualified domain name* (*FQDN*)
- Partially-qualified domain name (*PQDN*) (e.g., mulberry.csc)
  - label not ending in a top-level domain
  - DNS client must provide the missing suffix, e.g., csc.ncsu.edu, ncsu.edu

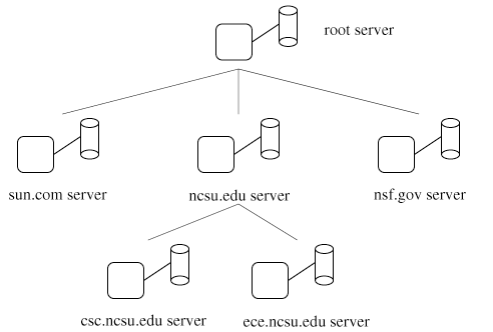
Domain Names and Labels



Hierarchy of Name Servers

- No single entity stores the entire database of resource records, for reasons of…
  - efficiency and scalability
  - reliability
  - security
- Information is distributed among DNS (name) servers
  - servers are organized hierarchically
  - each server is responsible, or *authoritative*, for part of the name space database

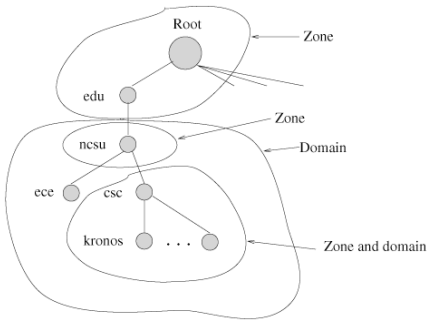
Example Name Server Hierarchy



Zones

- *Zone*: the part of the tree for which a server is responsible (over which it has authority)
  - Zone ≠ Domain
- *Zone file*: database of resource records for each node in zone

Zones and Domains Example



Primary and Secondary Zone Servers

- Primary or authoritative server
  - creates, stores, maintains, and updates zone file for its zone
  - must know IP addresses of all root servers
  - may know parent server (one level up)
- Secondary servers serve as backups / alternates, and should not…
  - run on same host
  - attach to same network
  - obtain electrical power from same source
  - etc.

Zone Servers (cont'd)

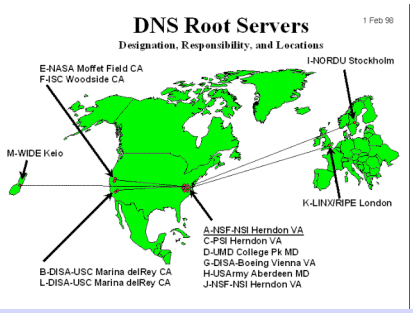
- Secondary server obtains all information from primary (*zone transfer*)
  - query primary every 3 hours
  - secondary server is also authoritative for zone it backs up
- Typical name server is authoritative for hundreds of zones

Root Servers

- *Root server* is a server whose zone contains the root of the DNS hierarchy
- Root server usually does not store any information about domains
  - delegates authority to other servers for all second-level domains
  - keeps pointer (name and IP address) to those servers

Root Servers (cont'd)

- Currently, there are **13** root servers around the world



MANAGEMENT OF THE NAMESPACE

The "Original" Top Level Domains

- *TLD*s come in two flavors
  - one is based on function (*generic*), one is based on geographic location
  - each organization is free to choose how/where to register

The Original Generic TLDs

Domain	Description
.com	Commercial organizations (global)
.net	Network service providers
.edu	US higher education
.mil	**US** military
.org	Nonprofit organizations (global)
.gov	**US** government
.int	International organizations

New Generic TLDs

Domain	Description
.biz	Businesses or firms
.info	Information service providers
.name	Personal nomenclatures
.museum	Museums
.aero	Air transport providers
.coop	Cooperatives
.jobs	Human resources managers
.travel	Travel industry
.int	Organizations established by international treaty
.pro	For the professions

Country Code TLDs (243 of them!)

.ac – Ascension Island
.ad – Andorra
.ae – United Arab Emirates
.af – Afghanistan
.ag – Antigua and Barbuda
.ai – Anguilla
.al – Albania
.am – Armenia
.an – Netherlands Antilles
.ao – Angola
.aq – Antarctica
.ar – Argentina
.as – American Samoa
.at – Austria
.au – Australia
.aw – Aruba
.az – Azerbaijan
.ba – Bosnia and Herzegovina
.bb – Barbados
.bd – Bangladesh
.be – Belgium
.bf – Burkina Faso

• • •

.ua – Ukraine
.ug – Uganda
.uk – United Kingdom
.um – US Minor Outlying Islands
.us – United States
.uy – Uruguay
.uz – Uzbekistan
.va – Holy See (City Vatican State)
.vc – Saint Vincent and the Grenadines
.ve – Venezuela
.vg – Virgin Islands (British)
.vi – Virgin Islands (USA)
.vn – Vietnam
.vu – Vanuatu
.wf – Wallis and Futuna Islands
.ws – Western Samoa
.ye – Yemen
.yt – Mayotte
.yu – Yugoslavia
.za – South Africa
.zm – Zambia
.zw – Zimbabwe

Management of Names and Numbers

- ICANN (Internet Corporation for Assigned Names and Numbers) manages the Domain Name System
  - an international not-for-profit, non-governmental organization
  - also coordinates the allocation of IP address space
- Registrars actually handle requests for name registration and assess fees
  - commercially operated, competitive

Management of Names and Numbers (cont'd)

- IANA (Internet Assigned Numbers Authority) oversees registration for various IP parameters, such as …
  - port numbers
  - protocol and enterprise numbers
  - options
  - codes
  - types

Why is Name Registration so Political?

- Whoever controls names controls access!
  - one root server, or many?
  - who is allowed to provide name registry services?
  - who is allowed to speak on behalf of a country or organization?
  - e.g., does WWF = World Wildlife Foundation or World Wrestling Federation?
- What happens if ICANN loses control and there are independent DNS hierarchies?

CONTENTS OF THE DNS DATABASE

DNS Resource Records

- Domain name database
  - consists of a set of records about each name in the domain
  - distributed among DNS servers by zone
- Resource records
  - specifies one type of information about a domain name
  - users must specify the type of information desired when querying a name

DNS Resource Records (cont'd)

- Contents
  1. Time to live: indication of how long the information is valid
  2. Type: what kind of information this is
  3. Data or information

Examples of Resource Record Types

Type	Meaning	Data
A	Host address	32-bit IPv4 address
AAAA	Host address	128-bit IPv6 address
MX	Mail exchanger	Name of server accepting email for this domain
NS	Name server	Name of authoritative server for zone
CNAME	Canonical name	"Real" domain name (for this alias)
PTR	Pointer	Domain name (for IP address → domain name mapping)
HINFO	Host description	Type of CPU and O.S. for this host
TXT	Text	Uninterpreted ASCII text
SOA	Start of authority	Parameters specifying which part of the naming hierarchy server implements

Some Other RRs

- Service discovery
- Geographic location

Start of Authority (SOA) Parameters

- **`Primary DNS Server`** for the domain
- **`Email Address`** for an administrator for the domain
- Sequence #
- Timers
  - **`Refresh Interval`** = minimum time between successive requests for zone records from the primary server

SOA Parameters (cont'd)

- **`Retry Interval`** = minimum time before retrying the request, if no answer
- **`Validity Interval`** = max time before zone records must be refreshed by the primary server
- **`Cache Interval`** = minimum TTL (validity interval) that should be exported with any RR for this zone

The nslookup, dig, and host commands

- Programs to query Internet name servers interactively
  - available for both Unix and Windows
  - can specify what information you want, for what host, and what server(s) you want to check
  - can turn recursion on or off

Examples

```
> csc.ncsu.edu
Server:  ns4.ncsu.edu
Address:  152.1.1.161
csc.ncsu.edu    MX preference = 10, mail exchanger = celestial-switchboard.csc.ncsu.edu
csc.ncsu.edu
        primary name server = ns1.ncsu.edu
        responsible mail addr = hostmaster.ncsu.edu
        serial  = 8484
        refresh = 28800 (8 hours)
        retry   = 3600 (1 hour)
        expire  = 2678400 (31 days)
        default TTL = 3600 (1 hour)
csc.ncsu.edu    nameserver = ns6.ncsu.edu
csc.ncsu.edu    nameserver = ns1.ncsu.edu
…
celestial-switchboard.csc.ncsu.edu      internet address = 152.1.61.45
ns6.ncsu.edu    internet address = 152.1.2.22
ns1.ncsu.edu    internet address = 152.1.1.22
…
```


DNS QUERIES

Name Resolution

- Features
  - distributed: multiple servers cooperate
  - efficient: most mappings are done locally
  - reliable: no single point of failure
- Search for name executes from bottom upwards
  - most searches are answered locally
- Queries may be recursive or iterative

Name Resolution (cont'd)

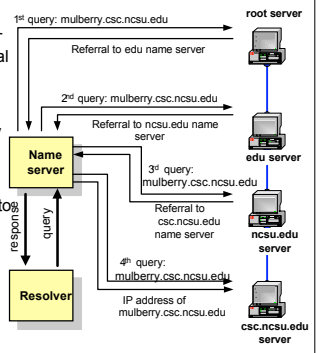
- **`Response`** is either …
  - a complete answer
  - or, the identity of the next server to contact
  - if response comes from authority managing the record, it is marked as *authoritative*

DNS over UDP or TCP?

- Normally, uses UDP as transport protocol
  - if the **`Response`** is truncated, client issues the **`Request`** again, this time using TCP
  - TCP returns the entire **`Response`** without truncation
  - zone transfers (between primary and secondary servers) only use TCP

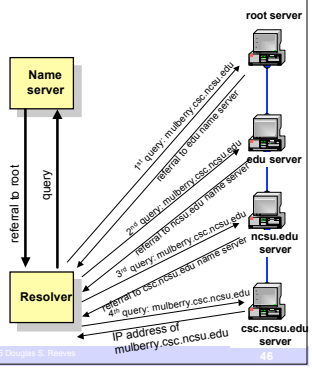
Recursive Resolution (caching not shown)

- In a recursive query, the resolver expects a response from the local name server
- If the local server cannot supply the answer, it will send the query to the "closest known" authoritative name server
- The root server sends a referral to the "edu" server. Querying this server yields a referral to the server of "ncsu.edu"
- etc.



Iterative Resolution (caching not shown)

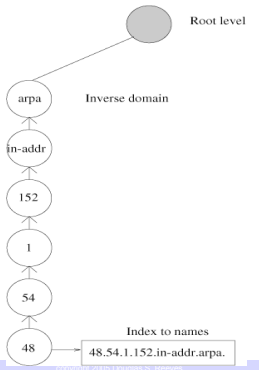
- In an iterative query, the name server sends a "closest known" authoritative name server
- This involves more work for the resolver



Reverse Mappings and Pointer Queries

- How map IP address to domain name?
  - have to start at top of DNS tree and try every top-level domain?!
- in-addr.arpa domain contains domain names which are the inverse of IP addresses of hosts
- Ex.: host with IP address 152.1.54.48 is represented by the domain name 48.54.1.152.in-addr.arpa
  - note reversal of order
  - returns PTR Resource Record (i.e., domain name)

Reverse Mapping



Inverse Queries

- E.g., ask for the IP addresses for domain csc.ncsu.edu
  - Good?
  - Bad?

DNS Caching

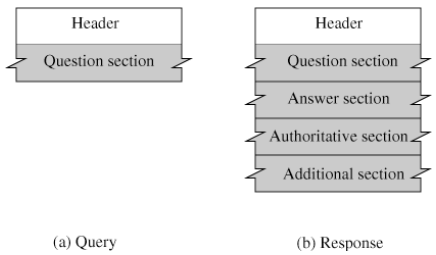
- Not feasible to contact root server for every DNS query
- Each server must maintain a cache of recent bindings
  - hosts may do so
- Benefits
  - less network traffic
  - less load on name servers
  - more robust / better fault tolerance

DNS Caching (cont'd)

- Answers from authoritative server include a **`TTL`** value
  - specifies how long the authority guarantees the binding will remain valid
  - cache entries are purged (timed out) when **`TTL`** expires
- If server has a cached copy of a mapping, it returns…
  - a non-authoritative binding
  - the source of information (the authoritative name server that provided this binding)
  - the IP address of this authoritative name server

DNS MESSAGES

Message Formats



(a) Query (b) Response

DNS Header Format

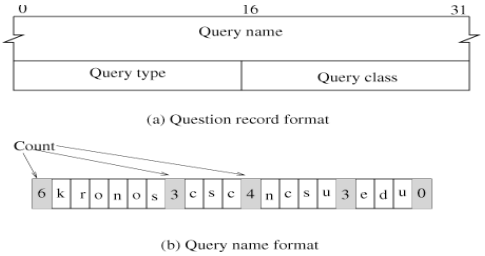
0	16	31
Identification	Flags	
Number of question records	Number of answer records (All 0s in query message)	
Number of authoritative records (All 0s in query message)	Number of additional records (All 0s in query message)	

- **`Identification`** field is set by client
  - for matching **`Responses`** to **`Requests`**

Flags

- **QR**: query or response?
- **OpCode**: standard or reverse query?
- **AA**: authoritative answer?
- **TC**: answer truncated (UDP used, only first 512 bytes returned)
- **RD**: recursion desired?
- **RA**: recursion available?
- **rCode** (return code): no error, query error, server failure, name not found, …

Compressed Name Format



- **Query Name** can be arbitrary length
  - but each part no more than 63 bytes long
- **Query Type** = type of Resource Record desired

Resource Records (in Response Messages)

- Each of **Answer**, **Authority**, and **Additional Info Sections** consist of a set of Resource Records
- **Domain Name**, **Type**, **Class** same as in Query Message
- **TTL** in seconds; usually = 2 days
  - may be 0 to prevent caching

DNS and Security

The KEY Resource Record

- A resource record format defined to associate keys with DNS names
  - permits DNS to be used as a public key distribution mechanism in support of DNS and other protocols
  - a resolver can learn a public key of a zone either by reading it from DNS, or by having it statically configured

The KEY Resource Record

- **Flags**
- **Protocol**
  - 1 = TLS, 2 = email, 3 = DNSSEC, 4 = IPSEC
- **Type**
  - 1 RSA/MD5 [RFC 2537] - recommended
  - 2 Diffie-Hellman [RFC 2539] - optional, key only
  - 3 DSA [RFC 2536] - MANDATORY
  - 4 reserved for elliptic curve cryptography
- **Key Value**

The SIG Resource Record

- Authentication of Resource Records
  - cryptographically-generated digital signatures
- Contents
  - **`Type of RRs`** covered by this signature
  - **`Algorithm Type`**
- At least one SIG Resource Record for each resource type under that name
  - a security aware server will attempt to return with Resource Records the corresponding SIGs

The SIG Resource Record (cont'd)

- **`Inception Date`** and **`Expiration Date`** of signature
- **`Domain Name of the Signer`** generating the signature
- Signature

Summary

- DNS consists of
  - a protocol
  - a distributed database
  - a naming scheme
- The namespace is hierarchical
- Name allocation is profitable and politically charged
- Highly robust

Summary (cont'd)

- DNS uses a distributed database of name→ address mappings
- DNS uses caching to speed up performance
- DNS is vital to the Internet
  - security is critical issue

Next Lecture

- Security and IPSec