Domain Name Service

Agenda

- What is DNS?
- What services does it provide?
- How does it operate?
- Message format
- Types of messages

What is DNS?

- DNS is a host name to IP address translation service
- DNS is
  - a distributed database implemented in a hierarchy of name servers
  - an application level protocol for message exchange between clients and servers

Why DNS?

- It is easier to remember a host name than it is to remember an IP address.
- An name has more meaning to a user than a 4 byte number.
- Applications such as FTP, HTTP, email, etc., all require the user to input a destination.
- The user generally enters a host name.
- The application takes the host name supplied by the user and forwards it to DNS for translation to an IP address.

DNS Services

Besides the address translation service, DNS also provides the following services:

- Host aliasing: a host with a complicated name can have one or more aliases that are simpler to remember,e.g., relay1.west-coast.media.com -> media.com. The longer name is the canonical hostname, the shorter the alias hostname.

DNS Services (cont'd)

- Mail server aliasing: same as above, aliases can exist for long canonical host names.
- Load Balancing: a set of servers can have one name mapped onto several machines. DNS provides the full list of names to the end user's application which generally takes the first one in the list. DNS rotates the names on the list.

How does it work?

- DNS works by exchanging messages between client and server machines.
- A client application will pass the destination host name to the DNS process (in Unix referred to as the gethostbyname() routine) to get the IP address.
- The application then sits and waits for the response to return.

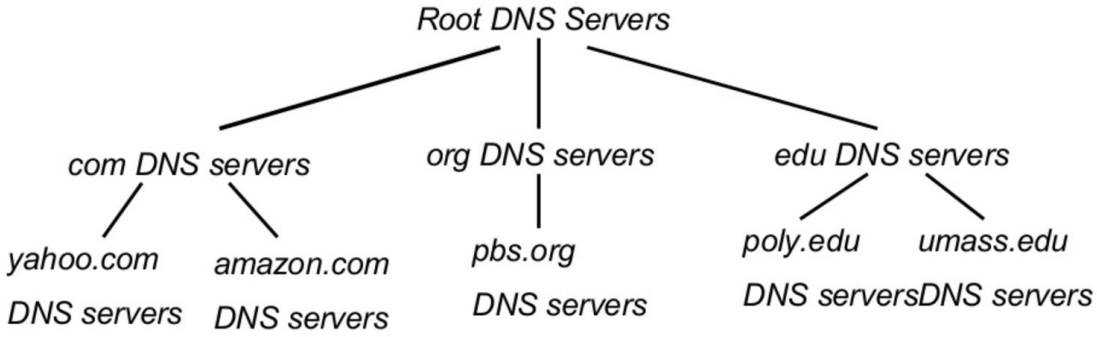
DNS

Why not centralize DNS?

- single point of failure
- traffic volume
- distant centralized database
- maintenance

doesn’t scale!

Distributed, Hierarchical Database



Client wants IP for www.amazon.com; 1st approx:

- client queries a root server to find com DNS server
- client queries com DNS server to get amazon.com DNS server
- client queries amazon.com DNS server to get IP address for www.amazon.com

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

- 13 root name servers worldwide

TLD and Authoritative Servers

- Top-level domain (TLD) servers:
  - responsible for com, org, net, edu, etc, and all top-level country domains uk, fr, ca, jp.
  - Network Solutions maintains servers for com TLD
  - Educause for edu TLD
- Authoritative DNS servers:
  - organization’s DNS servers, providing authoritative hostname to IP mappings for organization’s servers (e.g., Web, mail).
  - can be maintained by organization or service provider

Local Name Server

- does not strictly belong to hierarchy
- each ISP (residential ISP, company, university) has one.
  - also called “default name server”
- when host makes DNS query, query is sent to its local DNS server
  - acts as proxy, forwards query into hierarchy

DNS Queries

Recursive:

- The client machine sends a request to the local name server, which, if it does not find the address in its database, sends a request to the root name server, which, in turn, will route the query to an intermediate or authoritative name server. Note that the root name server can contain some hostname to IP address mappings. The intermediate name server always knows who the authoritative name server is.

DNS Queries (cont'd)

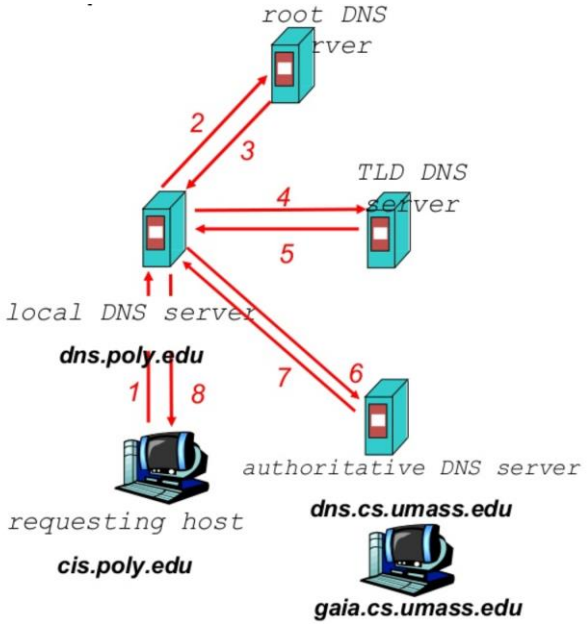
Iterative:

- The local server queries the root server. If address not in its database, will have the name/address of an intermediate or authoritative name server and forward that information to the local name server so that it can directly communicate with the intermediate or authoritative name server. This is to prevent the overloading of the root servers that handle millions of requests. •10/24/15 •14

DNS name resolution example

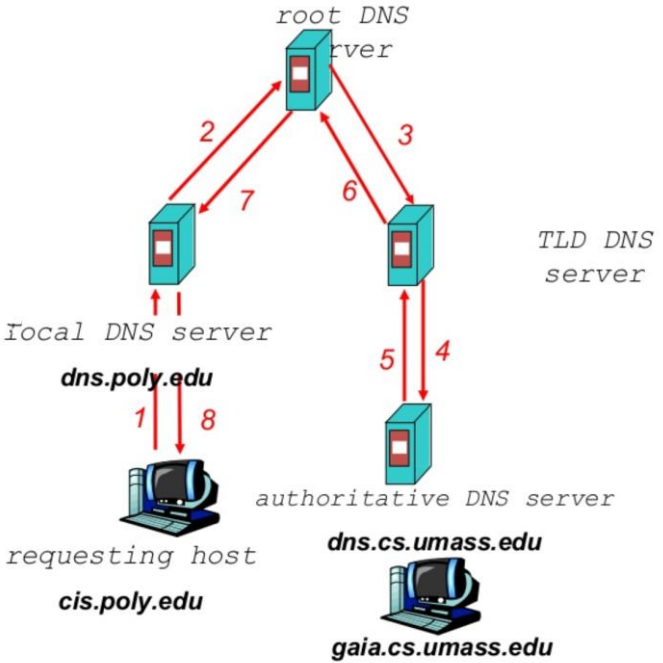
Host at cis.poly.edu wants IP address for gaia.cs.umass.edu

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



DNS name resolution example

- recursive query:
  - puts burden of name resolution on contacted name server
  - heavy load?



DNS: caching and updating records

- once (any) name server learns mapping, it caches mapping
  - cache entries timeout (disappear) after some time
  - TLD servers typically cached in local name servers
    - Thus root name servers not often visited
- update/notify mechanisms under design by IETF
  - RFC 2136
  - http://www.ietf.org/html.charters/dnsind-charter.html

Operation of DNS

- DNS uses caching to increase the speed with which it does the translation.
- The DNS data is stored in the database in the form of resource records (RR). The RRs are directly inserted in the DNS messages.
- The RRs are a 4 tuple that consist of: {name, value, type, TTL}.

RRs

- TTL: time to live, used to indicate when an RR can be removed from the DNS cache.
- Type =
  - A - then NAME is a hostname and Value its IP address
  - NS - then NAME is a domain name and Value is the IP address of an authoritative name server
  - CNAME - then NAME is an alias for a host and Value is the canonical name for the host
  - MX - then NAME is an alias for an email host and Value is the the canonical name for the email server

DNS records

DNS: distributed db storing resource records (RR)

RR format: (name, value, type, ttl)

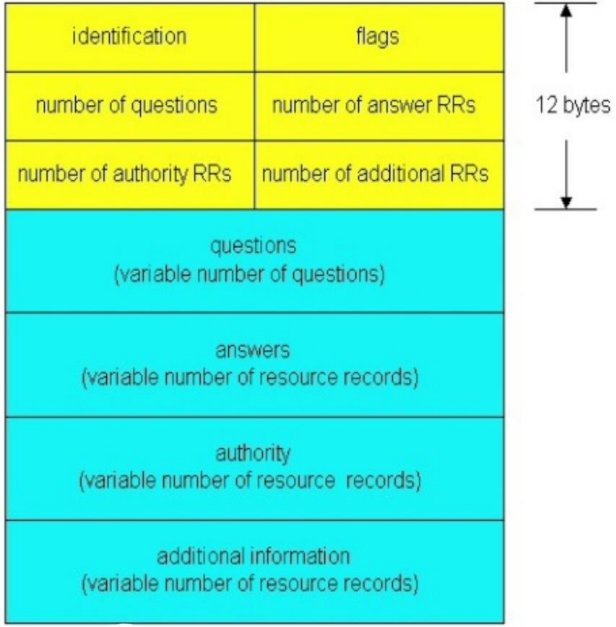
- *Type=A*
  - ***name** is hostname*
  - ***value** is IP address*
- *Type=NS*
  - ***name** is domain (eg., foo.com)*
  - ***value** is hostname of authoritative name server for this domain*
- *Type=CNAME*
  - ***name** is alias name for some "canonical" (the real) name, eg., www.ibm.com is really servereast.backup2.ibm.com*
  - ***value** is canonical name*
- *Type=MX*
  - ***value** is name of mailserver associated*

DNS protocol, messages

DNS protocol: query and reply messages, both with same message format

msg header

- identification: 16 bit # for query, reply to query uses same #
- flags:
  - query or reply
  - recursion desired
  - recursion available



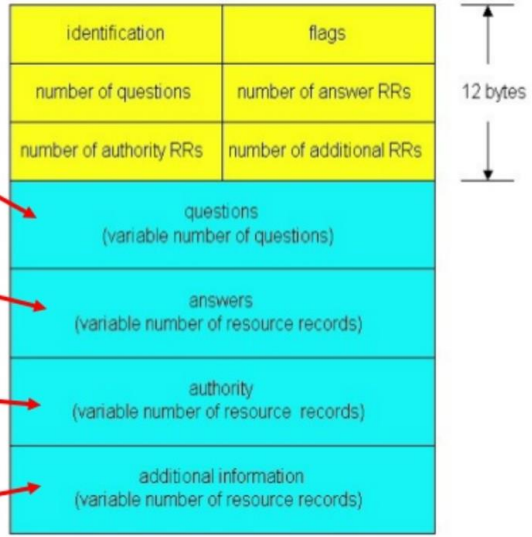
DNS protocol, messages

Name, type fields for a query

RRs in response to query

records for authoritative servers

additional "helpful" info that may be used



Message Fields

- Identification - identifies a query and is copied in the reply message to match it to the query at the client side.
- Flags - one bit flag set to indicate whether the message is a query or a reply. Another bit to identify if reply is from an authoritative sender or not. A third bit is used to indicate that recursion method is desired.

Fields cont’d

- Questions - contains the name that is being queried and the type, ie type A or MX.
- Answers - contains the RRs for the name(s) that were requested
- Authority - contains records of authoritative servers
- Additional Info - e.g., if type of query is MX, then this info can be a Type - A RR containing the IP address of the canonical hostname

Inserting records into DNS

- example: new startup “Network Utopia”
- register name networkuptopia.com at DNS **registrar** (e.g., Network Solutions)
  - provide names, IP addresses of authoritative name server (primary and secondary)
  - registrar inserts two RRs into com TLD server:

(networkutopia.com, dns1.networkutopia.com, NS) (dns1.networkutopia.com, 212.212.212.1, A)

- create authoritative server Type A record for www.networkuptopia.com; Type MX record for networkutopia.com

Summary

- DNS provides a mechanism for maintaining the user friendliness of the Internet by hiding some of the operational details.
- DNS servers have to be created manually. Recently an update protocol was introduced that allows DNS to exchange data for additions and deletions.

Q&A