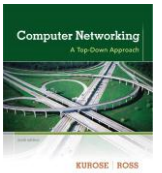
Chapter 2
Application Layer



Computer Networking: A Top Down Approach
6th edition
Jim Kurose, Keith Ross
Addison-Wesley
March 2012

A note on the use of these ppt slides:
We're making these slides freely available to all (faculty, students, readers). They're in PowerPoint form so you see the animations; and can add, modify, and delete slides (including this one) and slide content to suit your needs. They obviously represent a *lot* of work on our part. In return for use, we only ask the following:
- If you use these slides (e.g., in a class) that you mention their source (after all, we'd like people to use our book!)
- If you post any slides on a www site, that you note that they are adapted from (or perhaps identical to) our slides, and note our copyright of this material.

Thanks and enjoy! JFK/KWR

The course notes are adapted for Bucknell's CSCI 363
Xiannong Meng
Spring 2016

Chapter 2: outline

2.1 principles of network applications
- app architectures
- app requirements

2.2 Web and HTTP
2.3 FTP
2.4 electronic mail
- SMTP, POP3, IMAP

2.5 DNS
2.6 P2P applications
2.7 socket programming with UDP and TCP

DNS: domain name system

people: many identifiers:
- SSN, name, passport #

Internet hosts, routers:
- IP address (IPv4: 32 bit, IPv6: 128 bit) - used for addressing datagrams
- "name", e.g., www.yahoo.com - used by humans

Q: how to map between IP address and name, and vice versa ?

Domain Name System:
- *distributed database* implemented in hierarchy of many *name servers*
- *application-layer protocol:* hosts, name servers communicate to *resolve* names (address/name translation)
  - note: core Internet function, implemented as application-layer protocol
  - complexity at network's "edge"

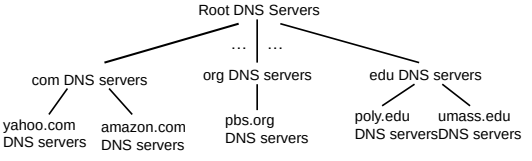
DNS: services, structure

DNS services
- hostname to IP address translation
- host aliasing
  - canonical, alias names
- mail server aliasing
- load distribution
  - replicated Web servers: many IP addresses correspond to one name

why not centralize DNS?
- single point of failure
- traffic volume
- distant centralized database
- maintenance

A: doesn't scale!

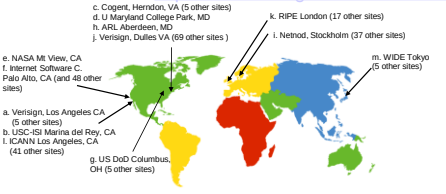
DNS: a distributed, hierarchical database



client wants IP for www.amazon.com; 1st approx:
- client queries 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
  - a total of 13 root name servers, each of which may have many more physical servers
  - official info: https://www.iana.org/domains/root/servers



Further DNS Information

- Internet Assigned Number Authority (IANA) http://www.iana.org/
- Wikipedia: http://en.wikipedia.org/wiki/Root_name_server
- As of Jul 2015, there are 1,058 top-level domains (TLD), including 301 country code top-level domains (ccTLD) and 730 generic top level domains (gTLD) worldwide, according to Wikipedia.

TLD, authoritative servers

top-level domain (TLD) servers:

- responsible for com, org, net, edu, aero, jobs, museums, and all top-level country domains, e.g.: uk, fr, ca, jp
- Network Solutions maintains servers for [.com], [.net], and [.org] TLDs
- Educause for .edu TLD

authoritative DNS servers:

- organization's own DNS server(s), providing authoritative hostname to IP mappings for organization's named hosts
- can be maintained by organization or service provider

Local DNS name server

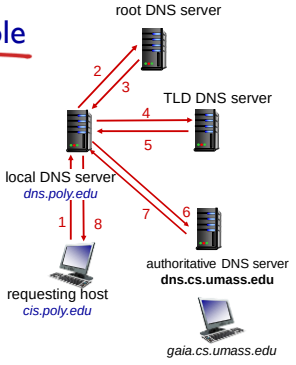
- does not strictly belong to hierarchy
- each ISP (residential ISP, company, university) has one or more
  - also called "default name server"
- when host makes DNS query, query is sent to its local DNS server
  - has local cache of recent name-to-address translation pairs (but may be out of date!)
  - acts as proxy, forwards query into hierarchy

DNS name resolution example

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

iterated query:

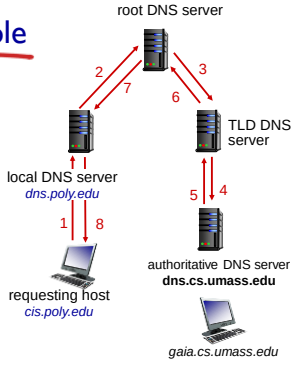
- contacted server replies with name of server to contact
- "I don't know this name, but ask this server"



DNS name resolution example

recursive query:

- puts burden of name resolution on contacted name server
- heavy load at upper levels of hierarchy?



DNS: caching, updating records

- once (any) name server learns mapping, it *caches* mapping
  - cache entries timeout (disappear) after some time (TTL)
  - TLD servers typically cached in local name servers
    - thus root name servers not often visited
- cached entries may be *out-of-date* (best effort name-to-address translation!)
  - if name host changes IP address, may not be known Internet-wide until all TTLs expire
- update/notify mechanisms proposed IETF standard
  - RFC 2136 see Wikipedia article

DNS records

DNS: distributed db storing resource records (RR)

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

type=A
- **name** is hostname
- **value** is IP address

type=NS
- **name** is domain (e.g., foo.com)
- **value** is hostname of authoritative name server for this domain

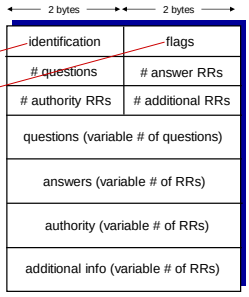
type=CNAME
- **name** is alias name for some "canonical" (the real) name
- `www.ibm.com` is really `servereast.backup2.ibm.com`
- **value** is canonical name

type=MX
- **value** is name of mailserver associated with **name**

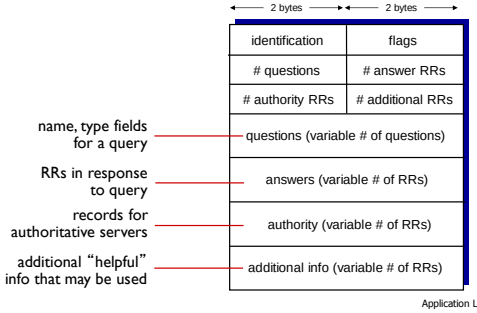
DNS protocol, messages

- *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
  - reply is authoritative



DNS protocol, messages



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 NS record for networkutopia.com

General file structure of DNS information

- On Linux, "resolv.conf" specifies the DNS server for local machines; on Windows, "ipconfig" shows the DNS.
- If a host is a DNS server, the following general file structures are used
  - named.conf: specifies current file structure of the DNS server, it also specifies "forward" to send unknown names or IPs, domain it serves, "zones" it is responsible, and for each zone, where the mapping resides.
  - named.hosts: specifies "authority" including domain name and email server name, IP addresses of the server
  - named.ca: specifies the known root DNS servers

DNS information sources

- named.conf: http://www.zytrax.com/books/dns/ch7/
- named.hosts: https://docs.oracle.com/cd/E19683-01/817-4843/dnsintro-94/index.html

How to examine the DNS information

- On Linux, "resolv.conf" specifies the DNS server for local machines; on Windows, "ipconfig" shows the DNS.
- Windows DNS can be configured or examined as follows
  - Computer->Control panel
  - Network and Internet > Network and Sharing Center > Change adapter settings
  - Select the device (e.g., Wireless network connection)
  - Select Properties, then select IPv4 or IPv6, properties
  - One can specify the IP address and DNS, among others
- Most, if not all, today's computers use DHCP, or Dynamic Host Configuration Protocol, so we don't have configure computers manually. We'll study DHCP later.

Attacking DNS

DDoS attacks

- Bombard root servers with traffic
  - Not successful to date
  - Traffic Filtering
  - Local DNS servers cache IPs of TLD servers, allowing root server bypass
- Bombard TLD servers
  - Potentially more dangerous

Redirect attacks

- Man-in-middle
  - Intercept queries
- DNS poisoning
  - Send bogus relies to DNS server, which caches

Exploit DNS for DDoS

- Send queries with spoofed source address: target IP
- Requires amplification

DHCP and DNS

- DNS provides name look-up service
- How does each computer on the internet establish a name-address map?
  - Manually configuration: such as named.conf and related files
  - Automatic assignment: DHCP (Dynamic Host Configuration Protocol) which we will study in detail in the network layer
  - The popular DHCP protocol is related to Bucknell!!!
  - See http://www.ietf.org/rfc/rfc2131.txt
  - And http://www.youtube.com/watch?v=_DkG4ur8A7g
    - Ralph Droms gave a talk about DHCP at the IETF and Internet Hall of Fame gathering in August 2013