Internet: Naming and Addressing

- Names, addresses and routes:

  According to Shoch (1979)

  - name: identifies what you want
  - address: identifies where it is
  - route: identifies a way to get there

- Internet names and addresses

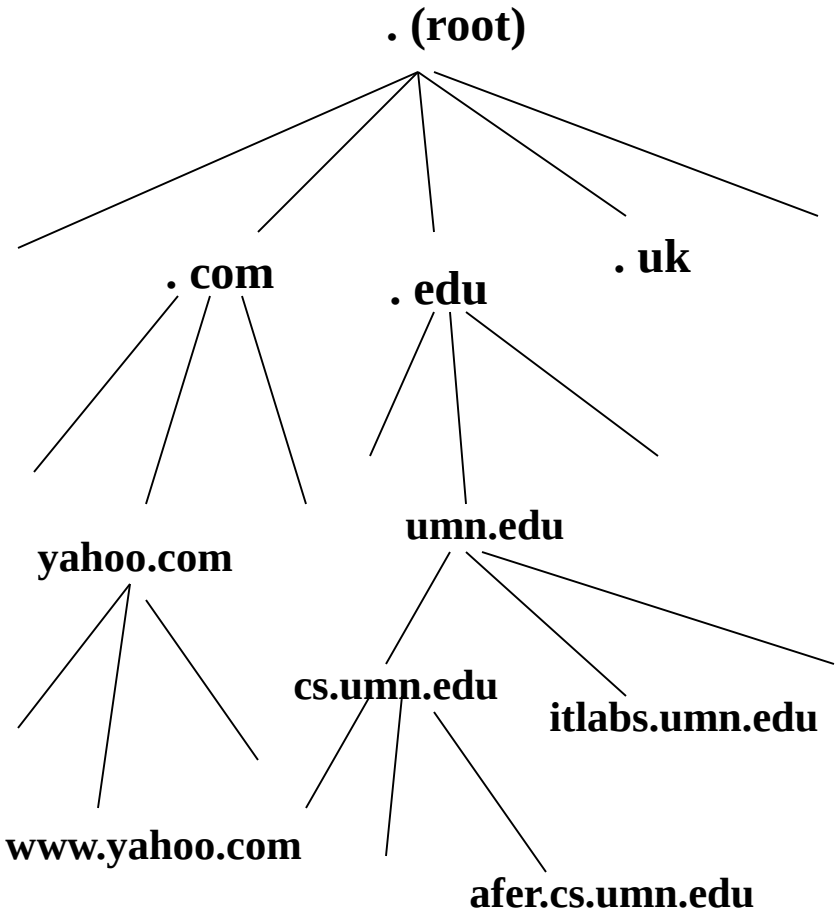
	Example	Organization
MAC address		flat, permanent
IP address	128.101.35.34	2-level
Host name	afer.cs.umn.edu	hierarchical

IP addresses

- Two-level hierarchy: network id. + host id.
  - (or rather 3-level, subnetwork id.)
  - 32 bits long usually written in dotted decimal notation
    e.g., 128.101.35.34
- No two hosts have the same IP address
  - host's IP address may change, e.g., dial-in hosts
  - a host may have multiple IP addresses
  - IP address identifies *host interface*
- Mapping of IP address to MAC (physical) IP done using IP ARP (this is called *address resolution*)
  - one-to-one mapping
- Mapping between IP address and host name done using Domain Name Servers (DNS)
  - many-to-many mapping

Internet Domain Names

- Hierarchical: anywhere from two to possibly infinity
- Examples: afer.cs.umn.edu, lupus.fokus.gmd.*de*
  - *edu, de:* organization type or country (a "domain")
  - *umn, fokus:* organization administering the "sub-domain"
  - *cs, fokus:* organization administering the host
  - *afer, lupus: host name (have IP address)*



Domain Name Resolution and DNS

DNS: Domain Name System:

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

- hierarchy of redundant servers with time-limited cache
- 13 root servers, each knowing the *global top-level domains* (e.g., edu, gov, com) , refer queries to them
- each server knows the 13 root servers
- each domain has at least 2 servers (often widely distributed) for fault distributed
- DNS has info about other resources, e.g., mail servers

DNS name servers

Why not centralize DNS?

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

doesn't *scale!*

- no server has all name-to-IP address mappings

local name servers:

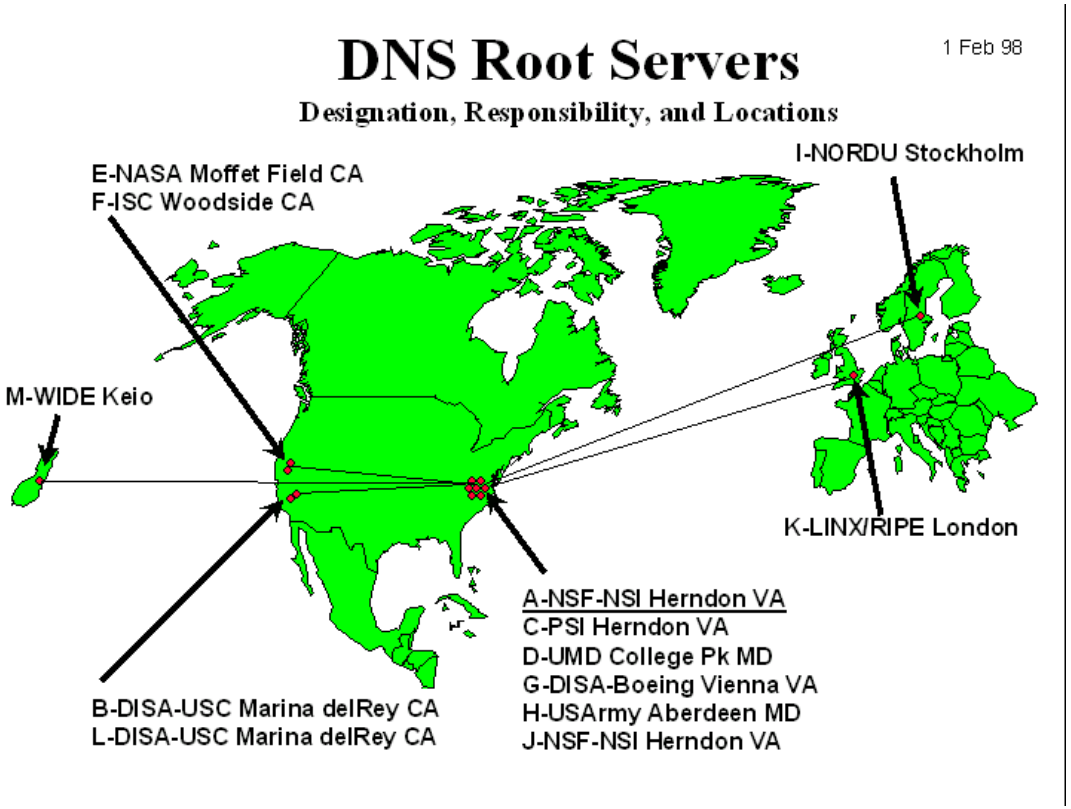
- each ISP, company has *local (default) name server*
- host DNS query first goes to local name server

authoritative name server:

- for a host: stores that host's IP address, name
- can perform name/address translation for that host's name

DNS: Root name servers

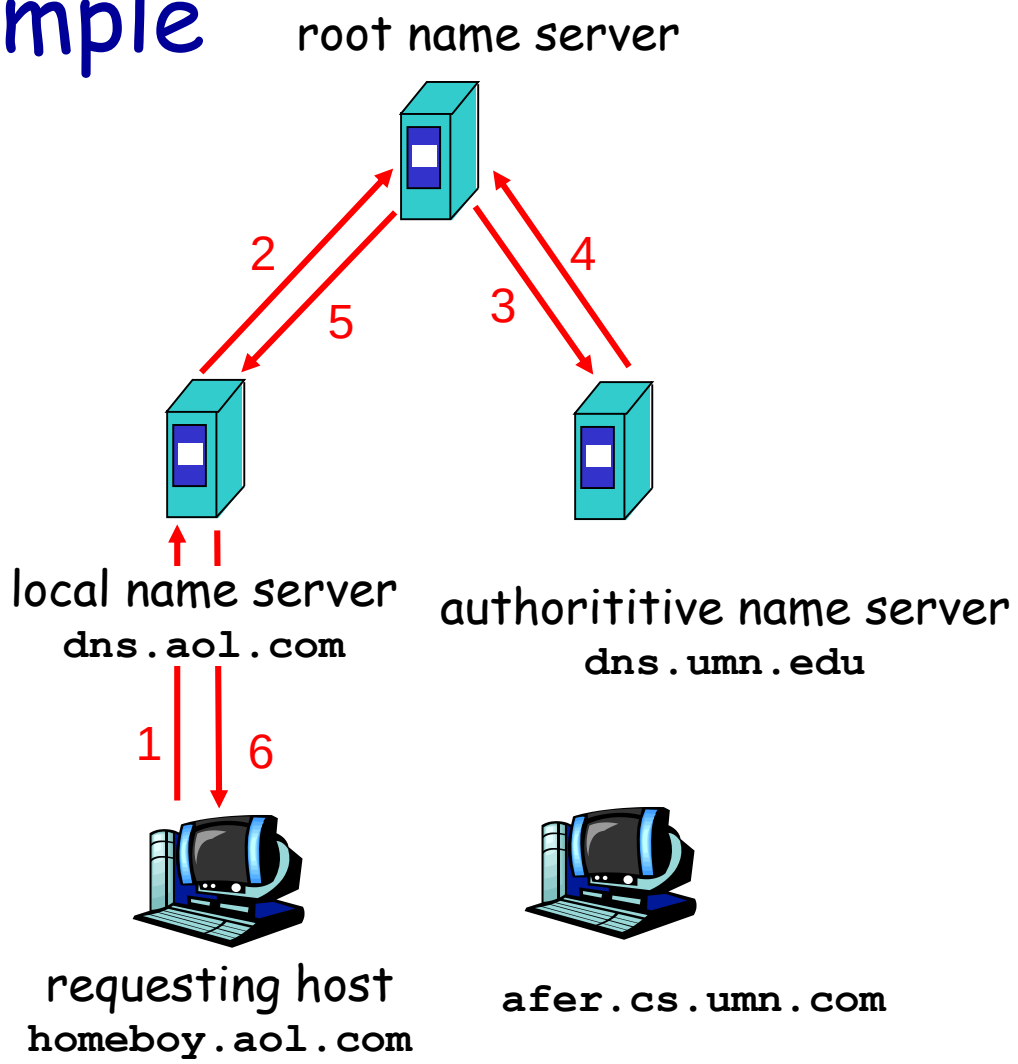
- contacted by local name server that can not resolve name
- root name server:
  - contacts authoritative name server if name mapping not known
  - gets mapping
  - returns mapping to local name server
- ~ dozen root name servers worldwide



Simple DNS example

host **homeboy.aol.com** wants IP address of **afer.cs.umn.edu**

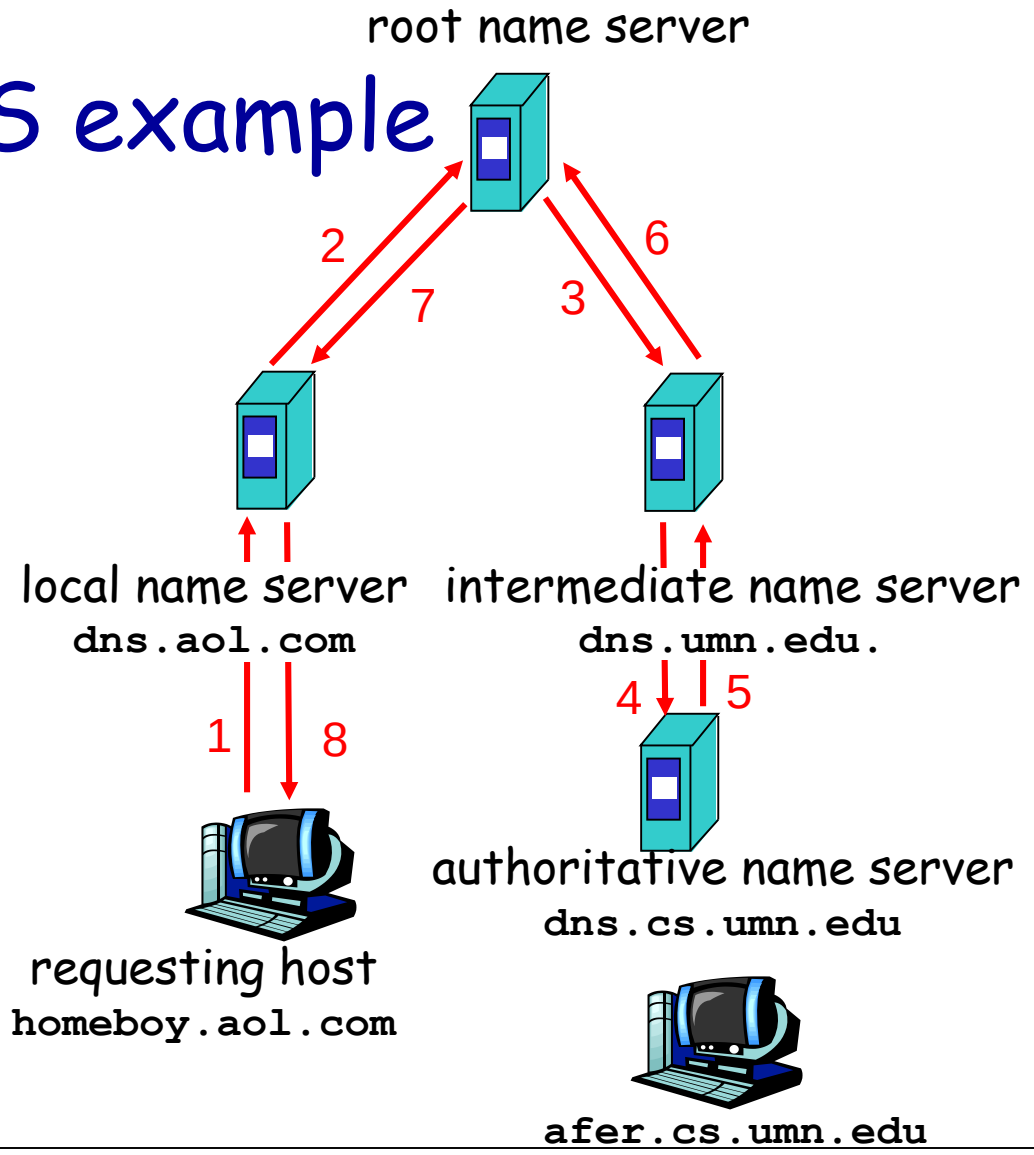
1. Contacts its local DNS server, **dns.aol.com**
2. **dns.aol.com** contacts root name server, if necessary
3. root name server contacts authoritative name server, **dns.umn.edu**, if necessary



DNS example

Root name server:

- may not know authoritative name server
- may know *intermediate name server:* who to contact to find authoritative name server



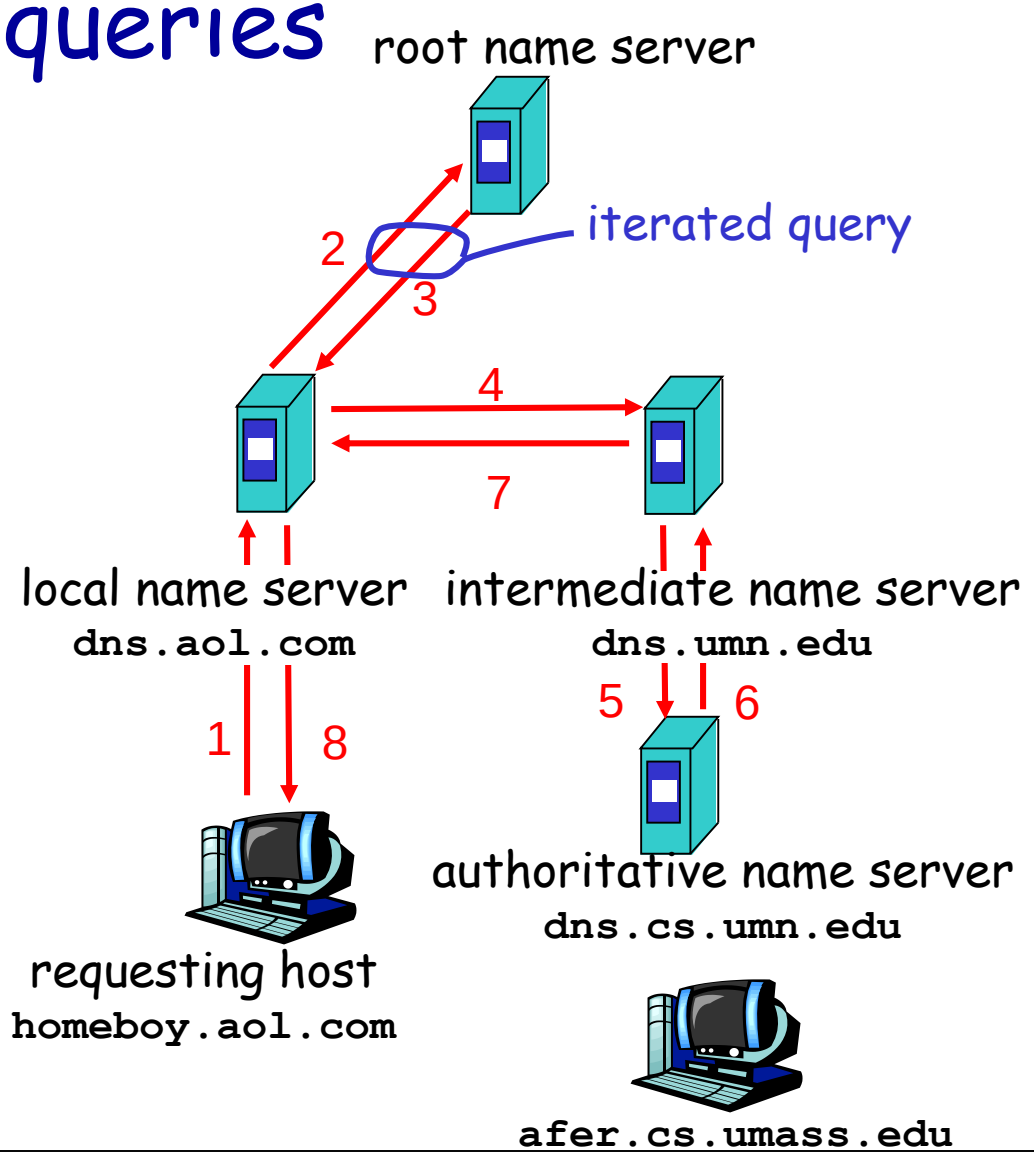
DNS: iterated queries

recursive query:

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

iterated query:

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



DNS: caching and updating records

- once (any) name server learns mapping, it *caches* mapping
  - cache entries timeout (disappear) after some time
- update/notify mechanisms under design by IETF
  - RFC 2136
  - http://www.ietf.org/html.charters/dnsind-charter.html

DNS records

DNS: distributed db storing resource records (RR)

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

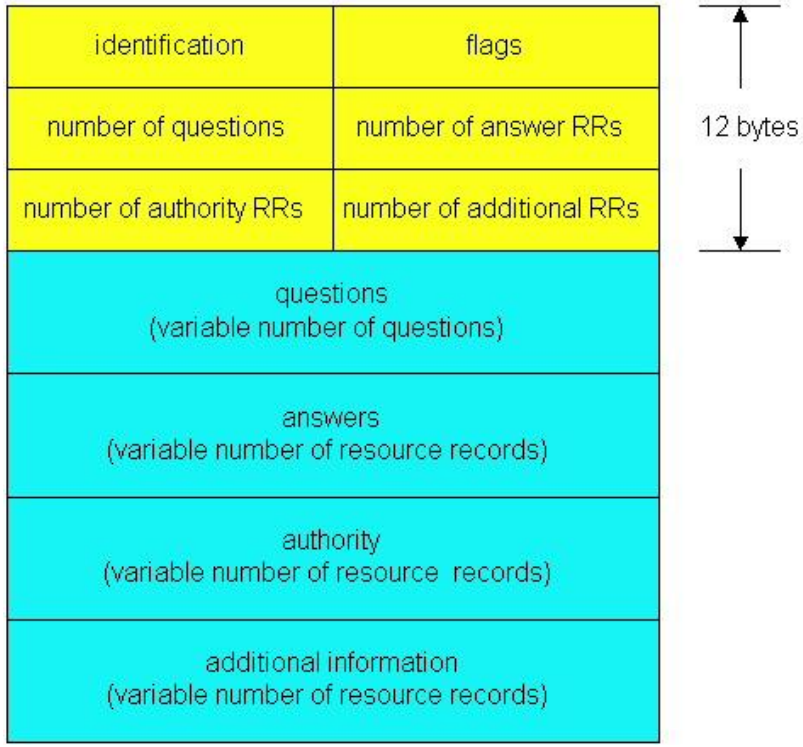
- Type=A (or AAAA)
  - **name** is hostname
  - **value** is IPv4 (IPv6) address
- Type=NS
  - **name** is domain (e.g. foo.com)
  - **value** is IP address of authoritative name server for this domain
- Type=CNAME
  - **name** is an alias name for some "canonical" (the real) name
  - **value** is canonical name
- Type=MX
  - **value** is hostname of mailserver associated with **name**

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
  - reply is authoritative



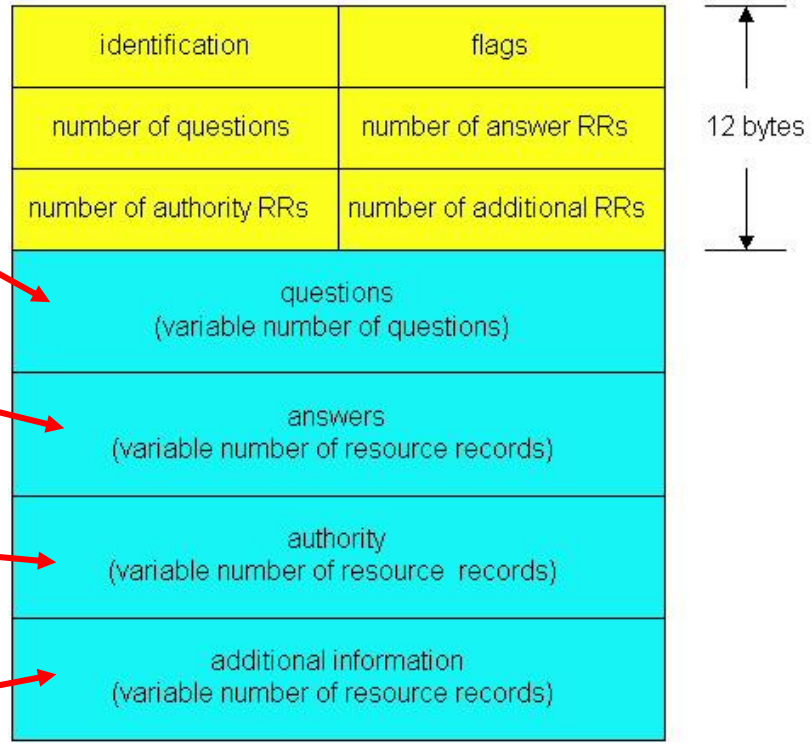
DNS protocol, messages

Name, type fields for a query

RRs in reponse to query

records for authoritative servers

additional "helpful" info that may be used



DNS Protocol

- Query/Reply: use UDP, port 53
- Transfer of DNS Records between authoritative and replicated servers: use TCP