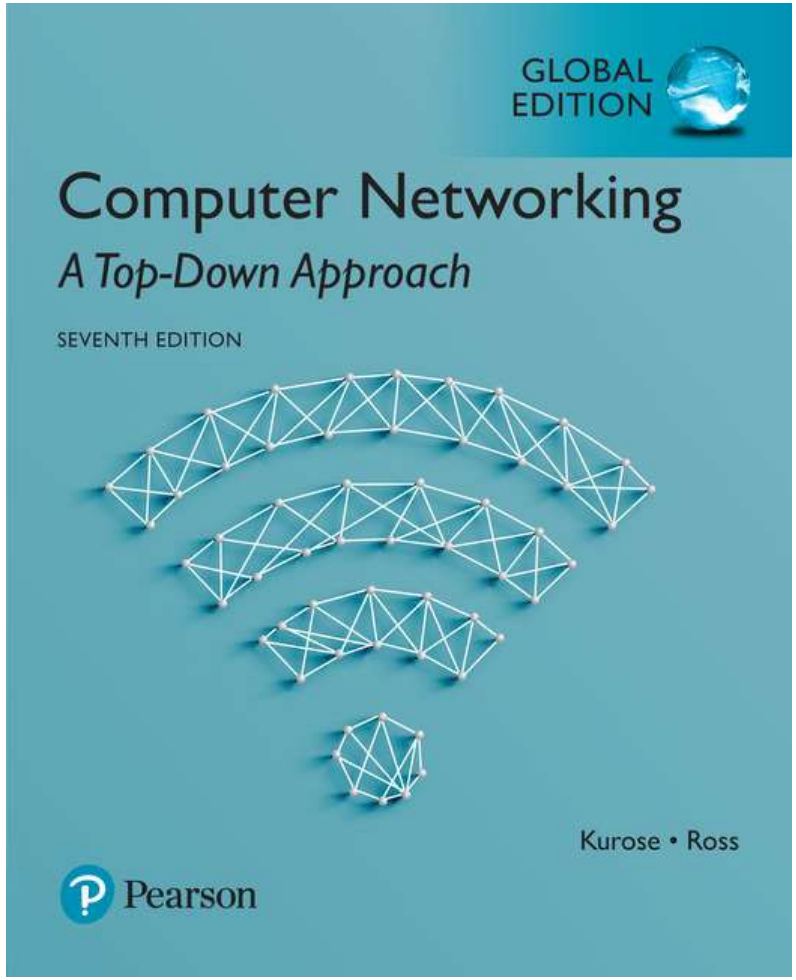
제10강 DNS

Computer Networking: A Top Down Approach

컴퓨터 네트워크
(2019년 1학기)

박승철교수

한국기술교육대학교
컴퓨터공학부



Pre-study Test :

1) 다음 중 도메인 이름에 대한 설명 중 틀린 것은?

① IP 주소에 비해 사용하기 쉽다.
② IP 주소에 비해 변경이 적다.
③ 응용 프로토콜에서 주로 사용한다.
④ TCP가 주로 사용한다.

2) 다음 중 DNS가 제공하는 역할이 아닌 것은?

① 도메인 이름을 IP 주소로 변환
② 별칭 이름(nick name)을 정규 이름으로 변환
③ 프록시 서버
④ 서버 부하 분산

3) 다음 중 중앙 집중형 DNS의 단점이 아닌 것은?

① 단일 고장 지점
② 부하 분산
③ 트래픽 집중
④ 유지보수

4) 다음 중 최상위 이름 서버는 ?

① Top-level name server
② Root name server
③ Authoritative name server
④ Local name server

5) 다음 중 특정 도메인 이름에 대한 정보를 유지하는 이름 서버는 ?

① Top-level name server
② Root name server
③ Authoritative name server
④ Local name server

6) 다음 중 가장 먼저 접근하는 이름 서버는 ?

① Top-level name server
② Root name server
③ Authoritative name server
④ Local name server

7) 다음 중 루트 서버에게 질의를 하고 결과를 기다리는 DNS 해석 기법은?

① Iterative 기법
② Recursive 기법
③ Integrated 기법
④ Proxy 기법

8) 다음 중 도메인 이름에 대한 IP 주소를 저장하는 DNS 레코드 유형은?

① A 유형
② NS 유형
③ CNAME 유형
④ MX 유형

Chapter 2: outline

2.1 principles of network applications

2.2 Web and HTTP

2.3 electronic mail

- SMTP, POP3, IMAP

2.4 DNS

2.5 P2P applications

2.6 video streaming and content distribution networks

2.7 socket programming with UDP and TCP

DNS: domain name system

people: many identifiers:

- SSN, name, passport #

Internet hosts, routers:

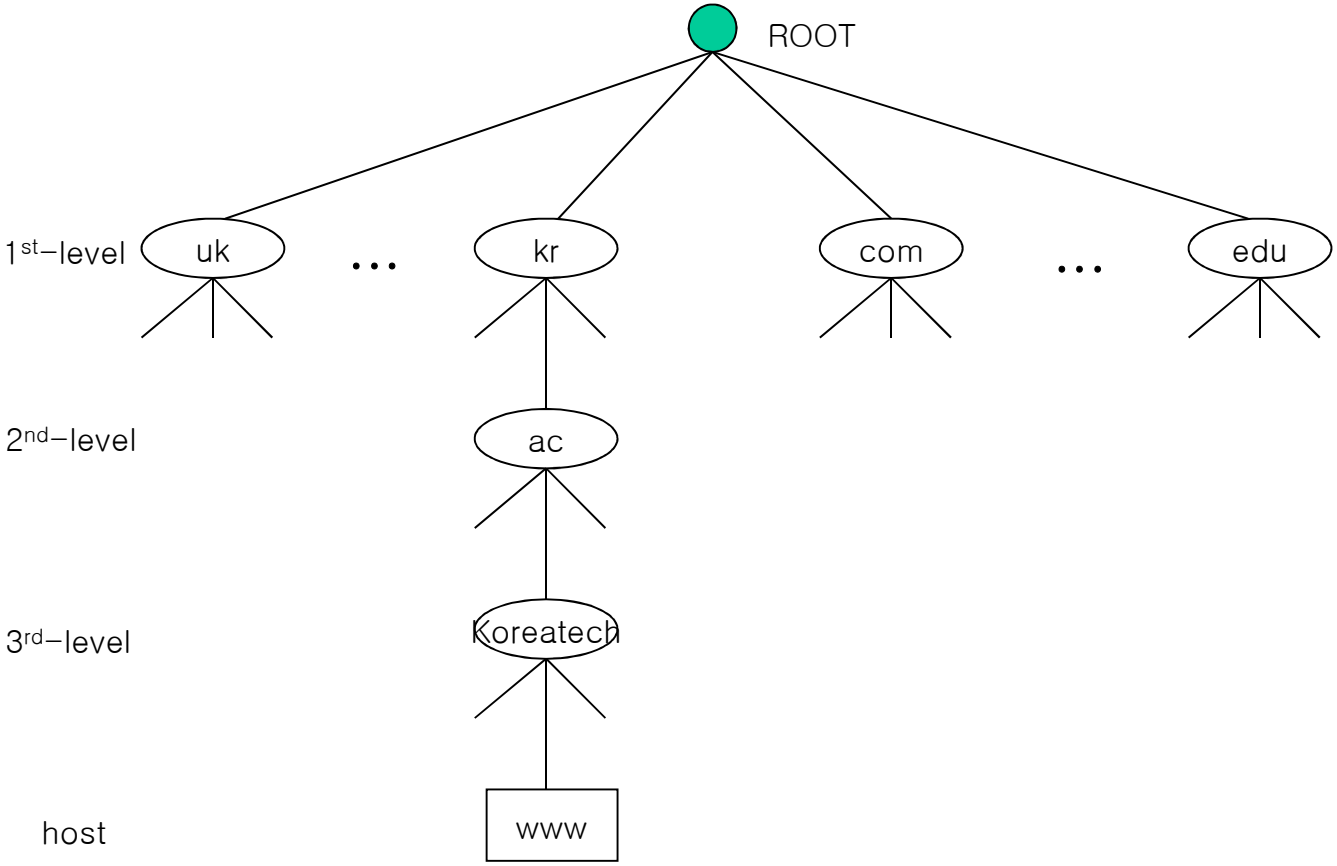
- IP address (32 bit) - used for addressing datagrams
- "name", e.g., www.yahoo.com - used by humans

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"

트리 구조의 도메인 이름 체계



트리 구조의 도메인 이름 체계

- 트리 구조의 인터넷 이름 체계에서 각 단계의 이름을 도메인 이름(Domain Name)이라 함
- 하위 단계의 도메인 이름은 항상 점(.)으로 구분되는 상위 단계의 도메인 이름 포함
- 트리 구조에서 호스트는 마지막 단계에 위치하므로 호스트 이름은 마지막 단계의 도메인 이름
- 상위 단계 도메인 이름을 공유하는 하위 단계 도메인 이름은 반드시 유일하게 부여되도록 관리

DNS: services, structure

DNS services

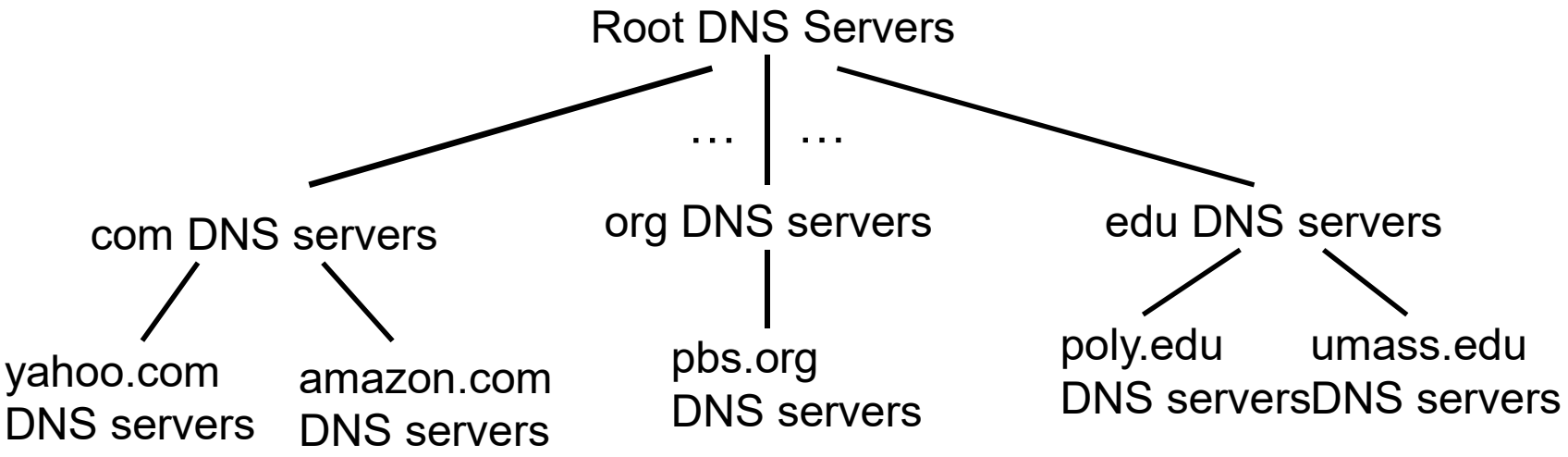
- hostname to IP address translation
- host aliasing
  - canonical, alias names
- mail server aliasing
- load distribution
  - replicated Web servers: many IP addresses correspond to one name

why not centralize DNS?

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

A: doesn't scale!

DNS: a distributed, hierarchical database

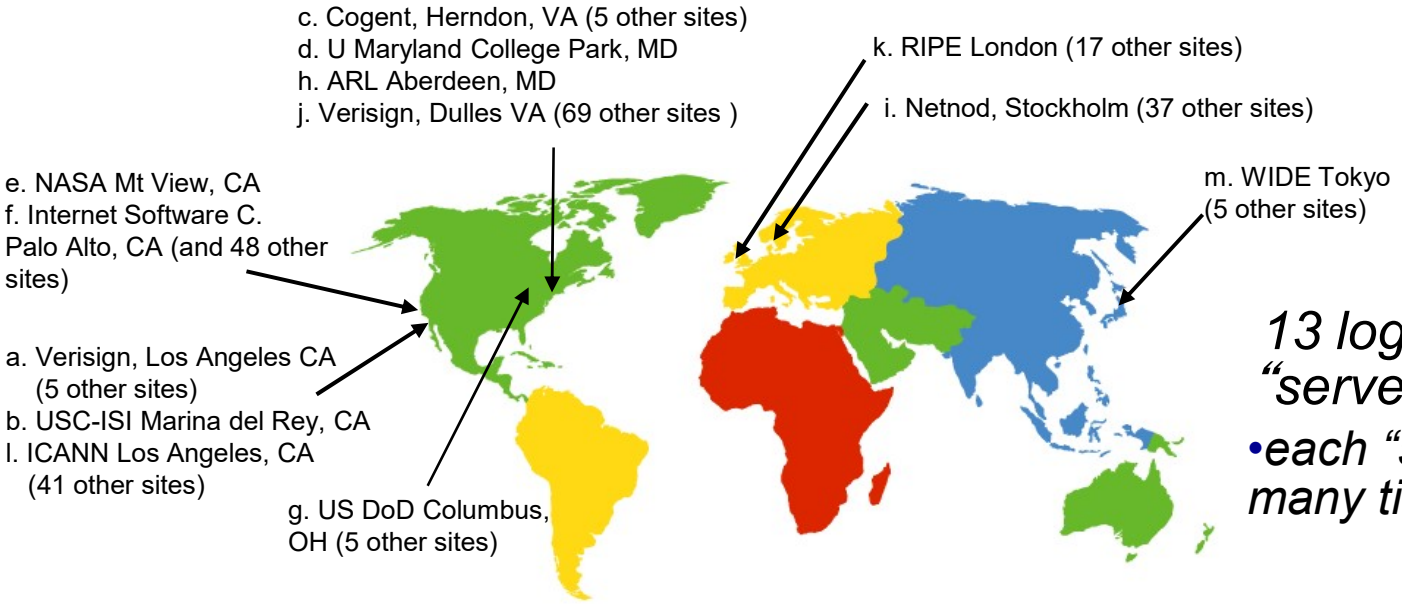


client wants IP for www.amazon.com; 1st approximation:

- 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



13 logical root name "servers" worldwide

- *each "server" replicated many times*

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 TLD
- 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

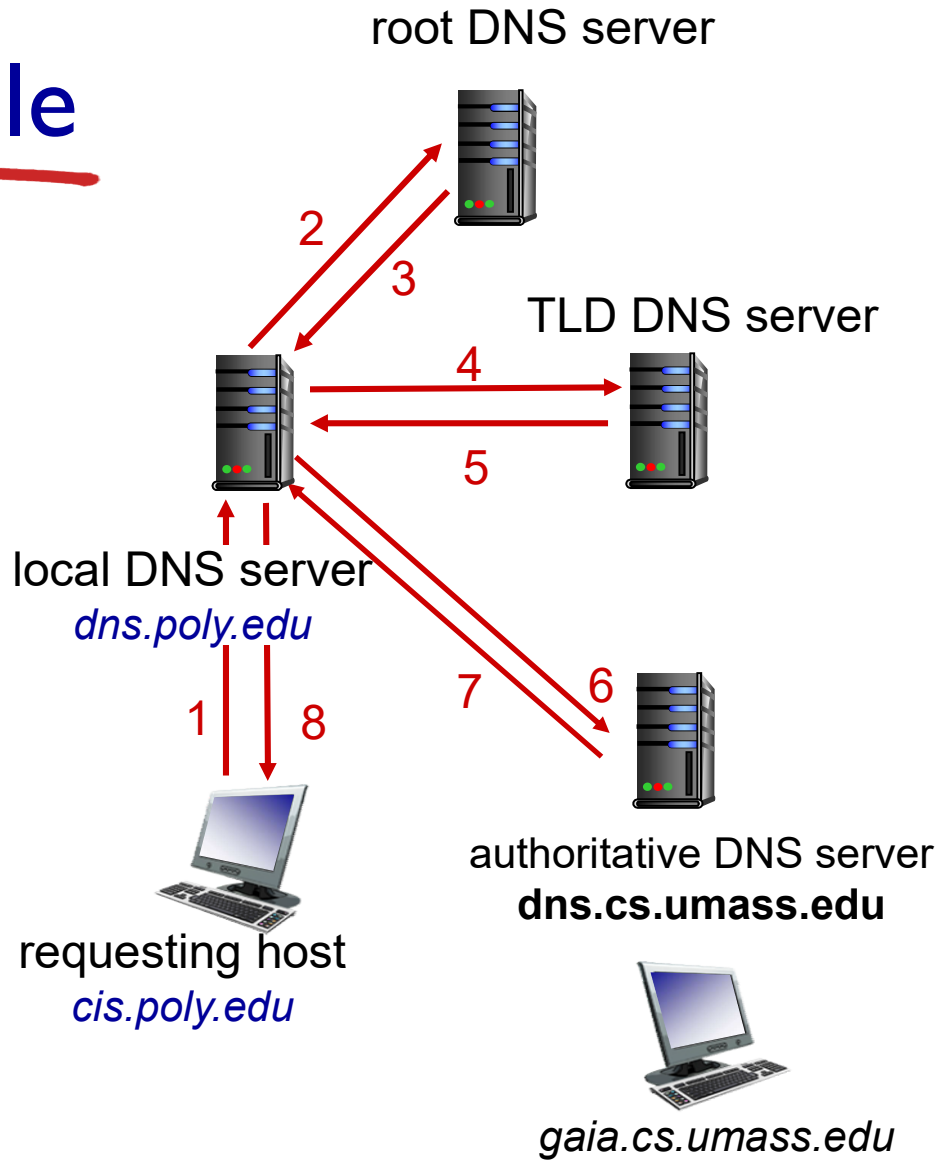
- does not strictly belong to hierarchy
- each ISP (residential ISP, company, university) has one
  - also called "default name server"
- when host makes DNS query, query is sent to its local DNS server
  - has local cache of recent name-to-address translation pairs (but may be out of date!)
  - acts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

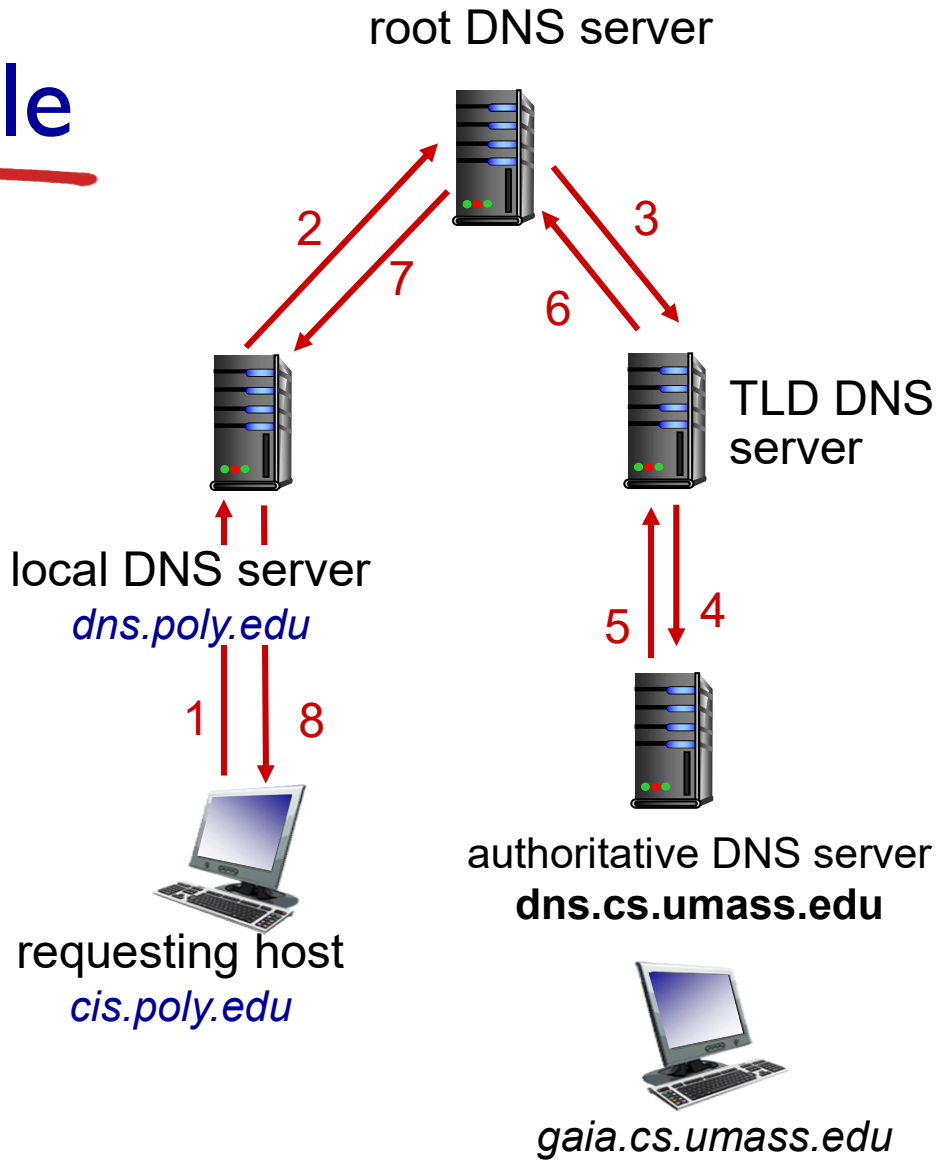
- contacted server replies with name of server to contact
- "I don't know this name, but ask this server"



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

DNS records

DNS: distributed database storing resource records (RR)

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

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

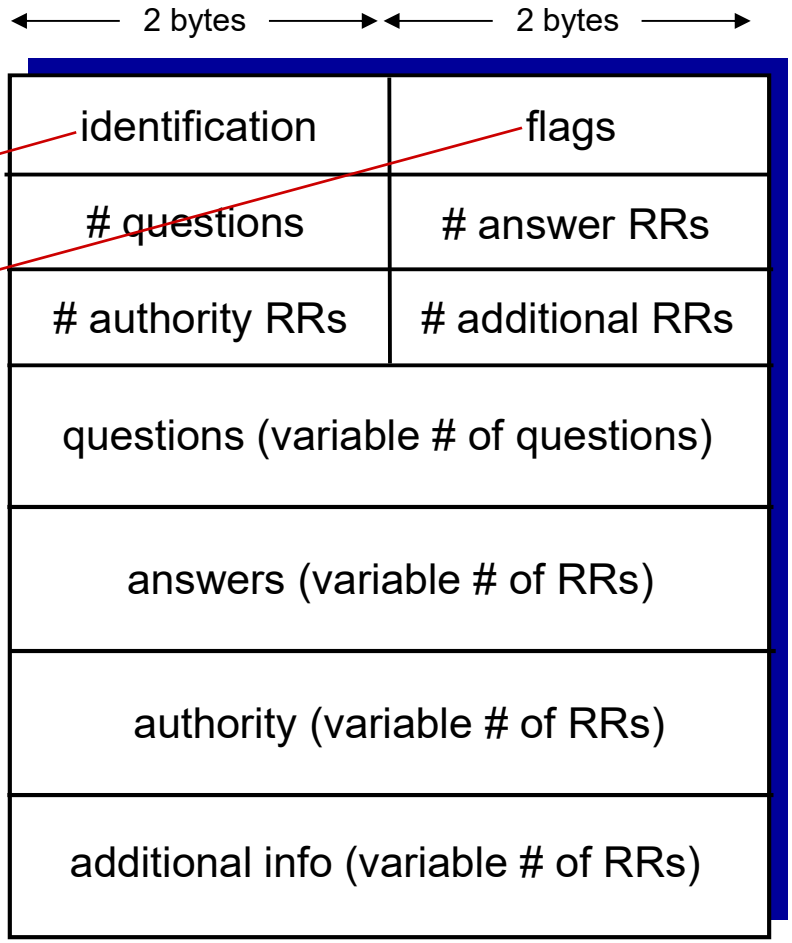
- **`value`** is name of mailserver associated with **`name`**

DNS protocol, messages

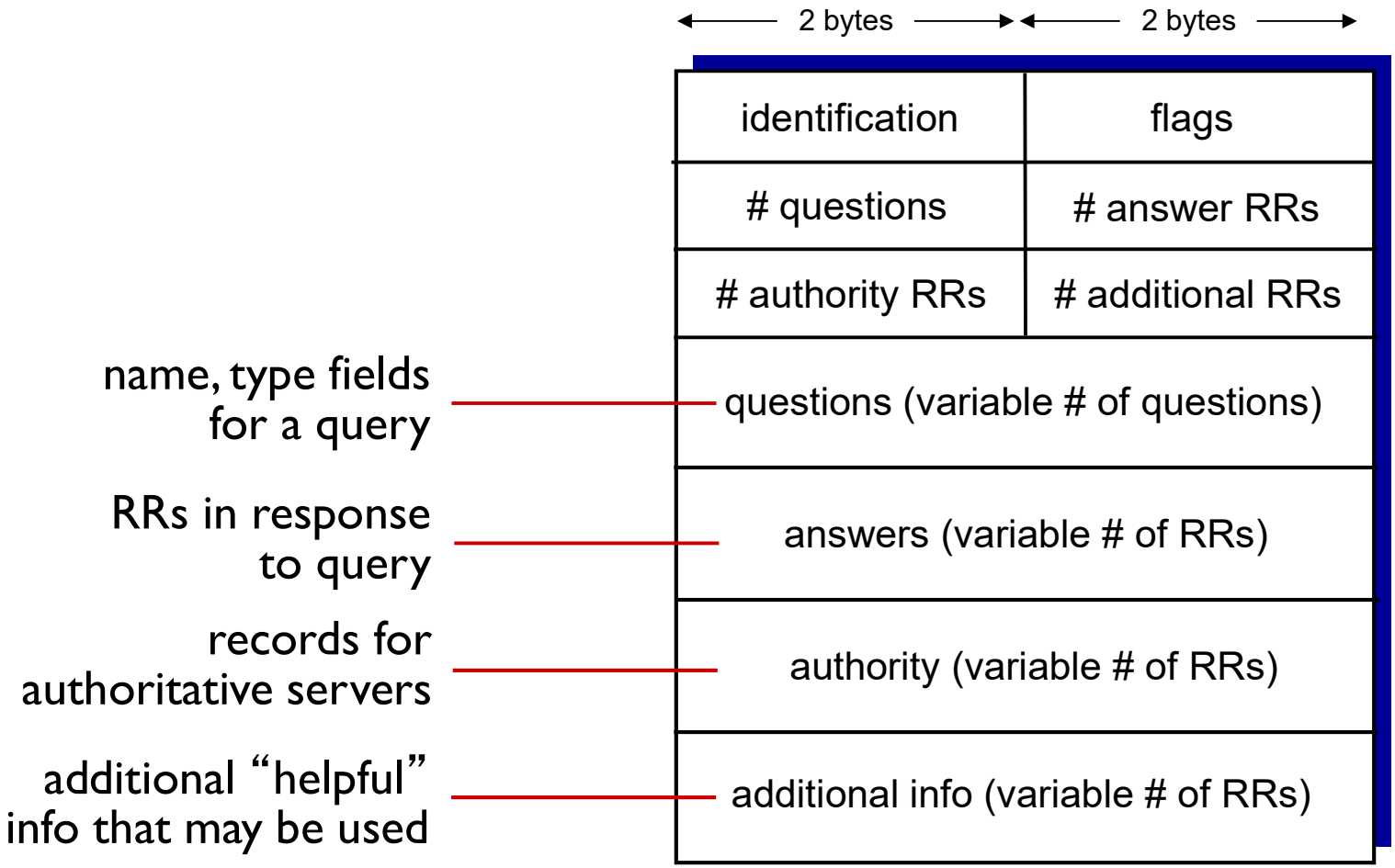
- *query* and *reply* messages, both with same *message format*

message 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 MX record for networkutopia.com

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

After-study Test :

1) 다음 중 도메인 이름에 대한 설명 중 틀린 것은?

① IP 주소에 비해 사용하기 쉽다.
② IP 주소에 비해 변경이 적다.
③ 응용 프로토콜에서 주로 사용한다.
④ TCP가 주로 사용한다.

2) 다음 중 DNS가 제공하는 역할이 아닌 것은?

① 도메인 이름을 IP 주소로 변환
② 별칭 이름(nick name)을 정규 이름으로 변환
③ 프록시 서버
④ 서버 부하 분산

3) 다음 중 중앙 집중형 DNS의 단점이 아닌 것은?

① 단일 고장 지점
② 부하 분산
③ 트래픽 집중
④ 유지보수

4) 다음 중 최상위 이름 서버는 ?

① Top-level name server
② Root name server
③ Authoritative name server
④ Local name server

5) 다음 중 특정 도메인 이름에 대한 정보를 유지하는 이름 서버는 ?

① Top-level name server
② Root name server
③ Authoritative name server
④ Local name server

6) 다음 중 가장 먼저 접근하는 이름 서버는 ?

① Top-level name server
② Root name server
③ Authoritative name server
④ Local name server

7) 다음 중 루트 서버에게 질의를 하고 결과를 기다리는 DNS 해석 기법은?

① Iterative 기법
② Recursive 기법
③ Integrated 기법
④ Proxy 기법

8) 다음 중 도메인 이름에 대한 IP 주소를 저장하는 DNS 레코드 유형은?

① A 유형
② NS 유형
③ CNAME 유형
④ MX 유형