DNS

Mobile Computing

Prof. Jongwon Yoon

Intelligent Machines Lab.

Host names vs. IP addresses

- Host names
  - Memonic name appreciated by humans
  - Variable length, full alphabet of characters
  - Provide little (if any) information about location
  - Examples: www.cnn.com and bbc.co.uk

- IP addresses
  - Numerical address appreciated by routers
  - Fixed length, binary number
  - Hierarchical, related to host location
  - Examples: 64.236.16.20 and 212.58.224.131

Separating naming and addressing

- Names are easier to remember
  - www.cnn.com vs. 64.236.16.20
- Addresses can change underneath
  - Move www.cnn.com to 4.125.91.21
  - E.g., renumbering when changing providers
- Name could map to multiple IP addresses
  - www.cnn.com to multiple (8) replicas of the Web site
  - Enables
    - Load-balancing
    - Reducing latency by picking nearby servers
    - Tailoring content based on requester's location/identity
- Multiple names for the same address
  - E.g., aliases like www.cnn.com and cnn.com

Before there was DNS ….

…. there was the HOSTS.TXT file

- Before DNS (until 1985), the name-to-IP address was done by downloading a single file (hosts.txt) from a central server with FTP.
  - Names in hosts.txt are not structured.
  - The hosts.txt file still works on most operating systems.
  - It can be used to define local names.

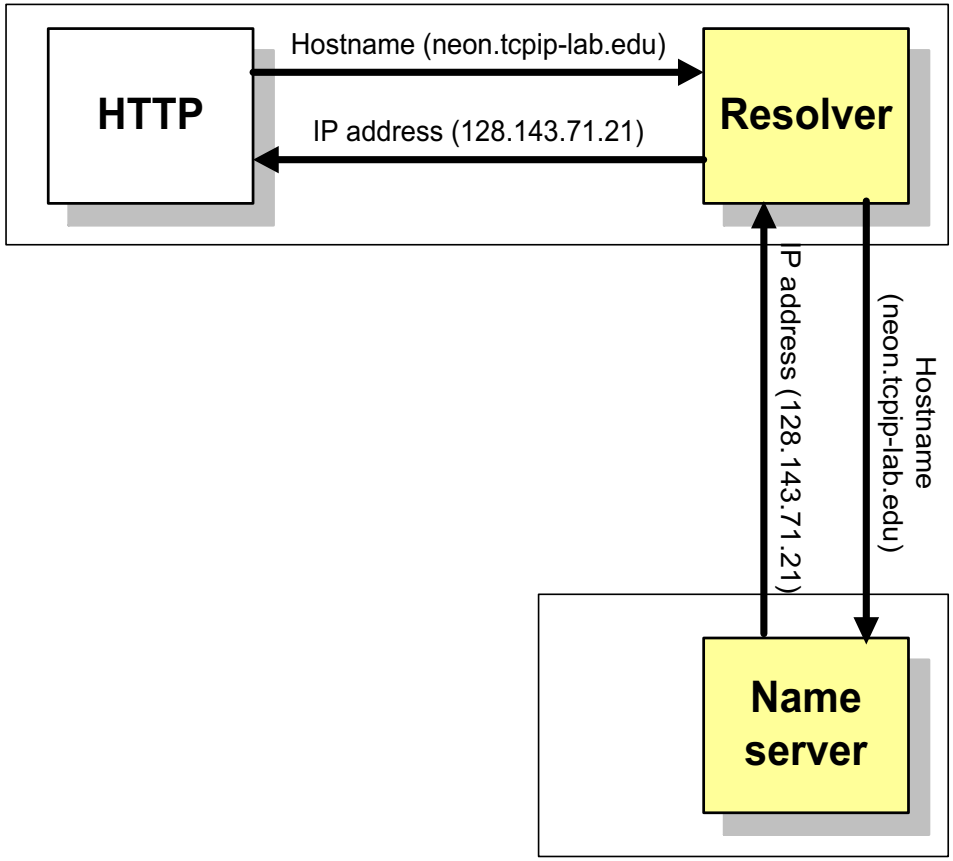
Domain Name System (DNS)

- Properties of DNS
  - Hierarchical name space divided into zones
  - Zones distributed over collection of DNS servers
- Hierarchy of DNS servers
  - Root (hardwired into other servers)
  - Top-level domain (TLD) servers
  - Authoritative DNS servers
- Performing the translations
  - Local DNS servers
  - Resolver software

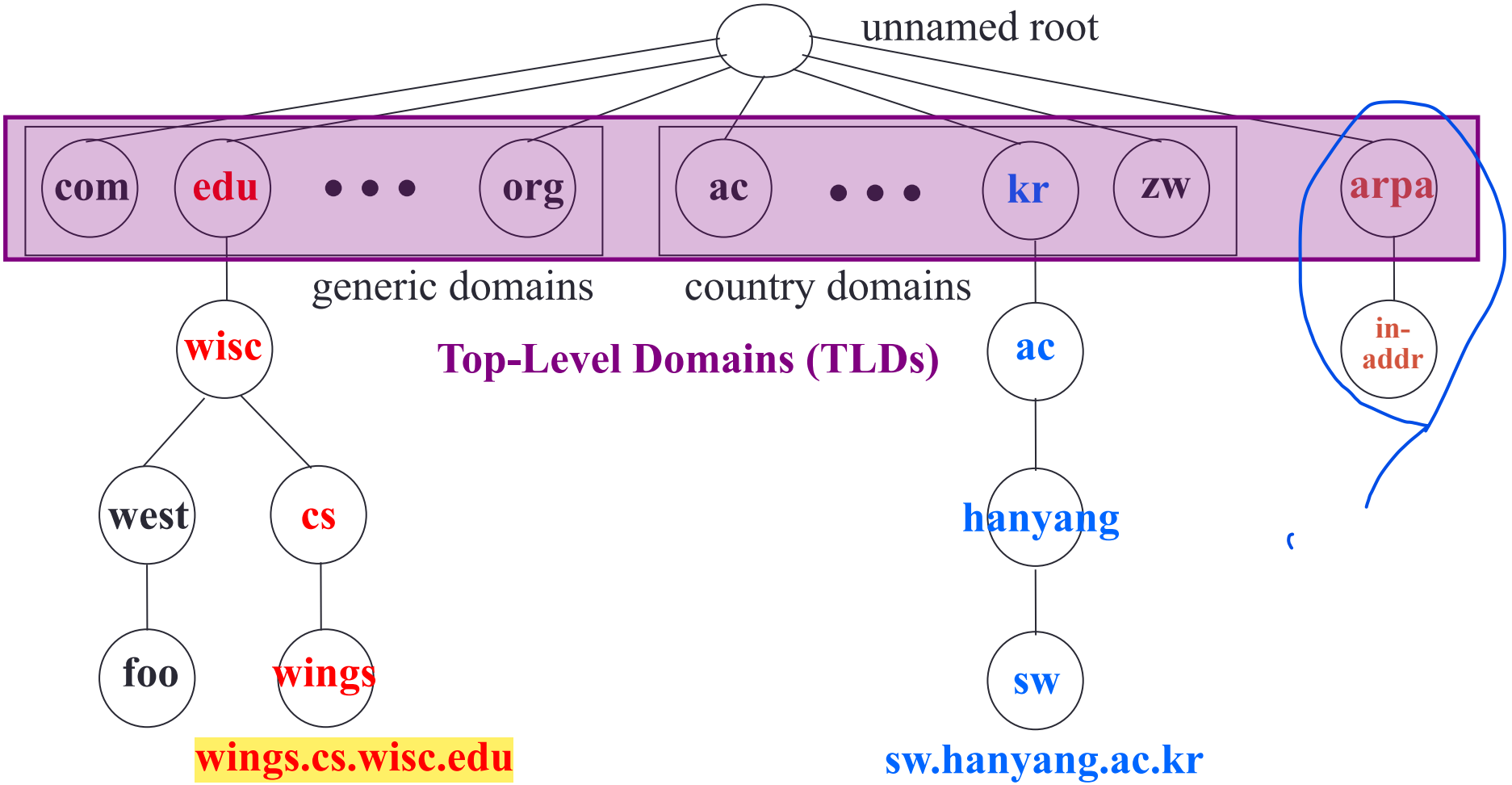
Resolver and name server

1. An application program on a host accesses the domain system through a DNS client, called the **resolver**
2. Resolver contacts DNS server, called name server
3. DNS server returns IP address to resolver which passes the IP address to application

- Reverse lookups are also possible, i.e., find the hostname given an IP address



Distributed Hierarchical Database



Top-level domains

- Three types of top-level domains:
  - Generic Top Level Domains (gTLD): 3-character code indicates the function of the organization
    - Used primarily within the US
    - Examples: gov, mil, edu, org, com, net
  - Country Code Top Level Domain (ccTLD): 2-character country or region code
    - Examples: us, va, jp, de
  - Reverse domains: A special domain (in-addr.arpa) used for IP address-to-name mapping

There are more than 200 top-level domains.

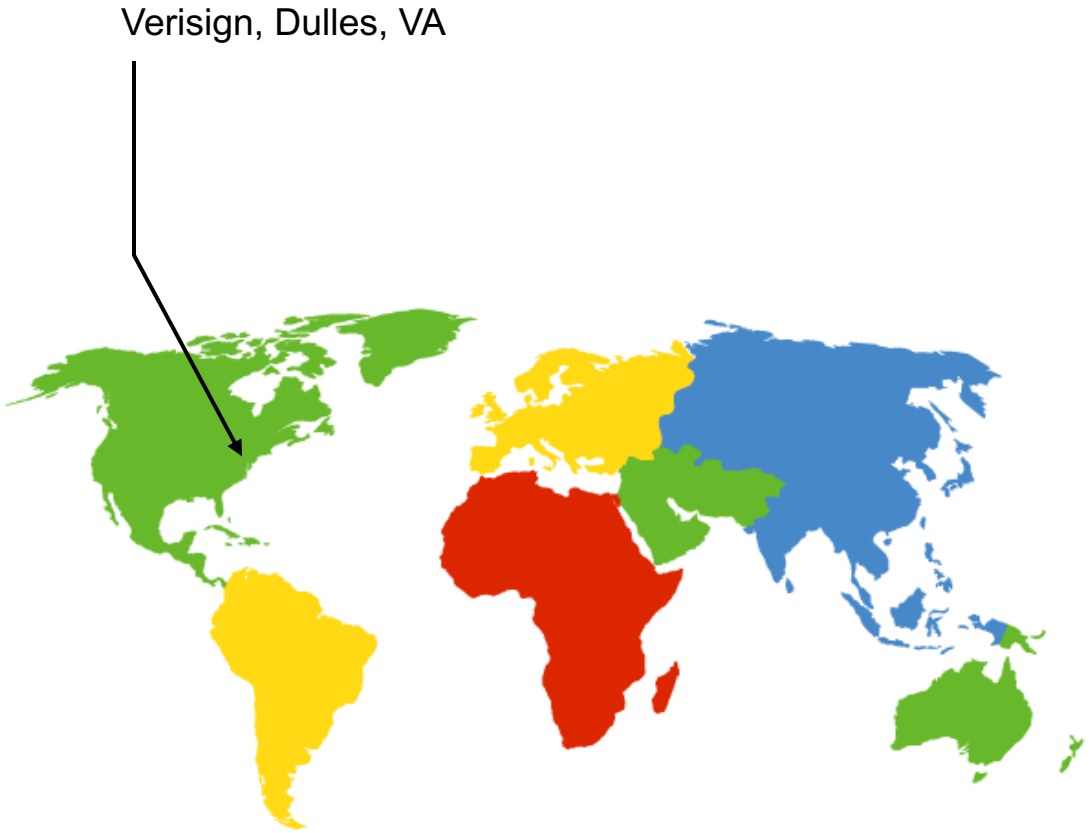
Generic Top Level Domains (gTLD)

com	Commercial organizations
edu	Educational institutions
gov	Government institutions
int	International organizations
mil	U.S. military institutions
net	Networking organizations
org	Non-profit organizations

- gTLDs are authoritatively administered by the Internet central name registration authority ICANN
- FQDN (fully qualified domain name): host name + domain name
  e.g., cse.hanyang.ac.kr

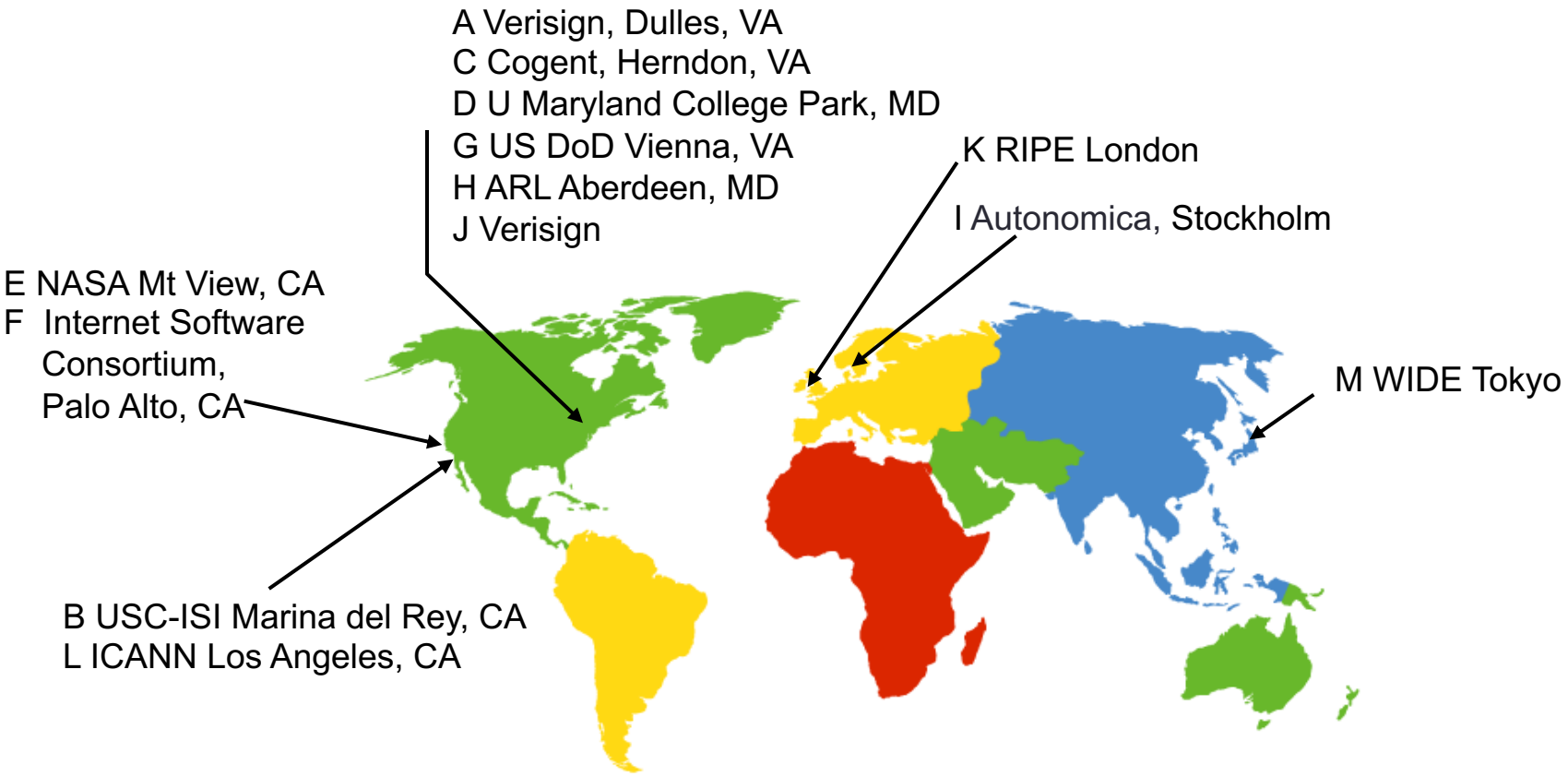
DNS Root

- Located in Virginia, USA
- How do we make the root scale?



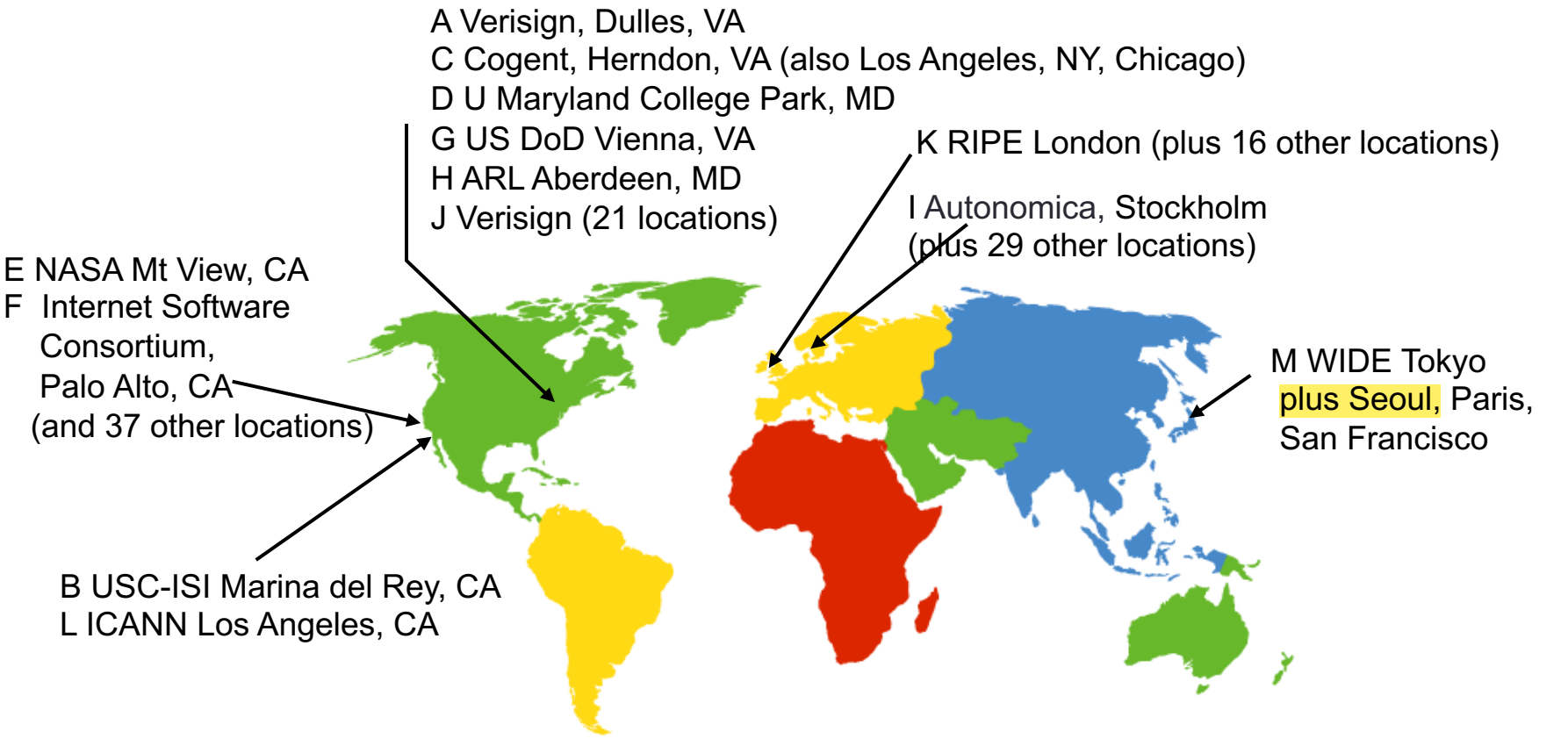
DNS Root Servers

- 13 root servers (see http://www.root-servers.org/)
  - Labeled A through M
- Does this scale?



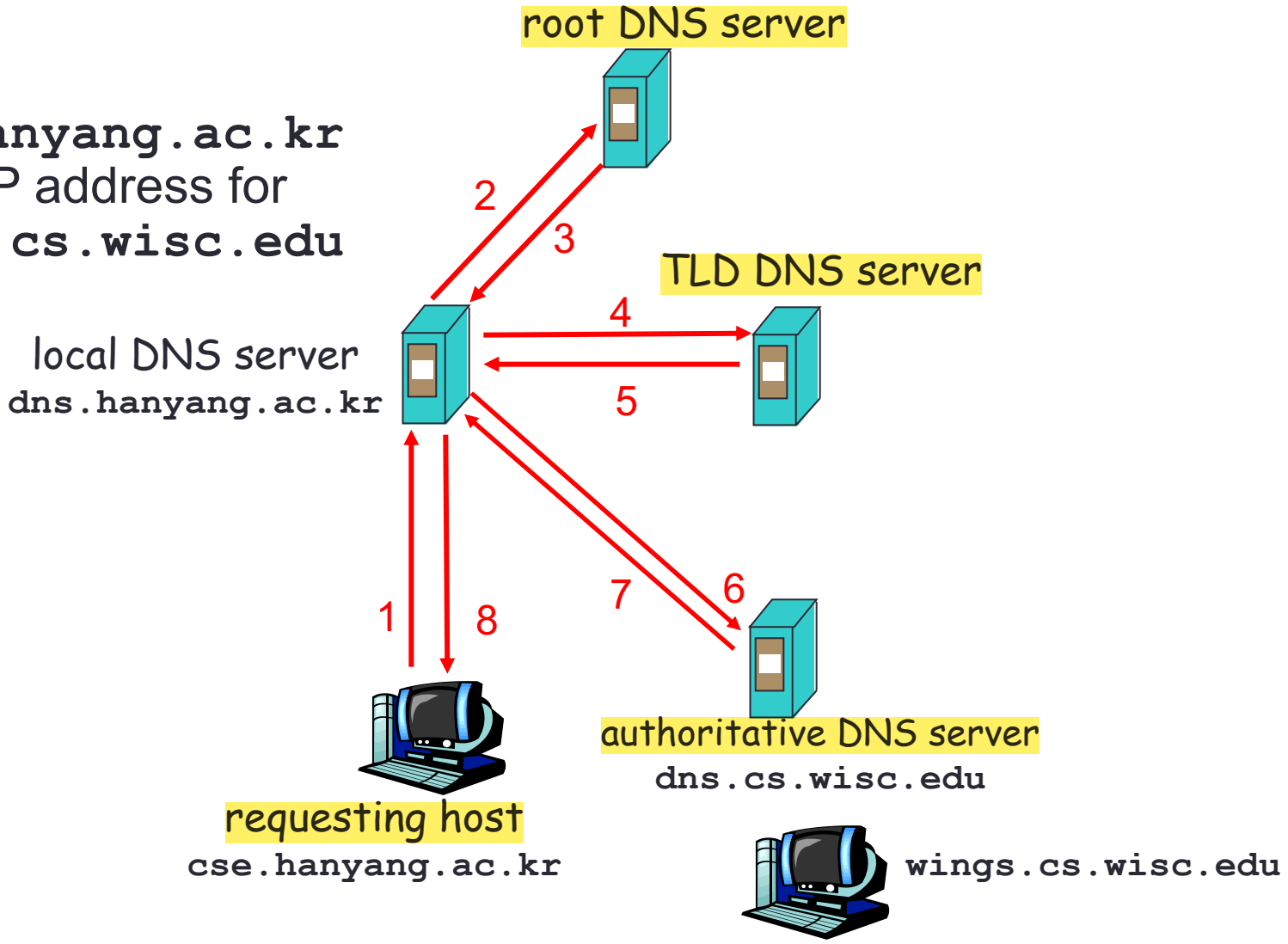
DNS Root Servers

- 13 root servers (see http://www.root-servers.org/)
  - Labeled A through M
- Replication via any-casting (localized routing for addresses)



Example

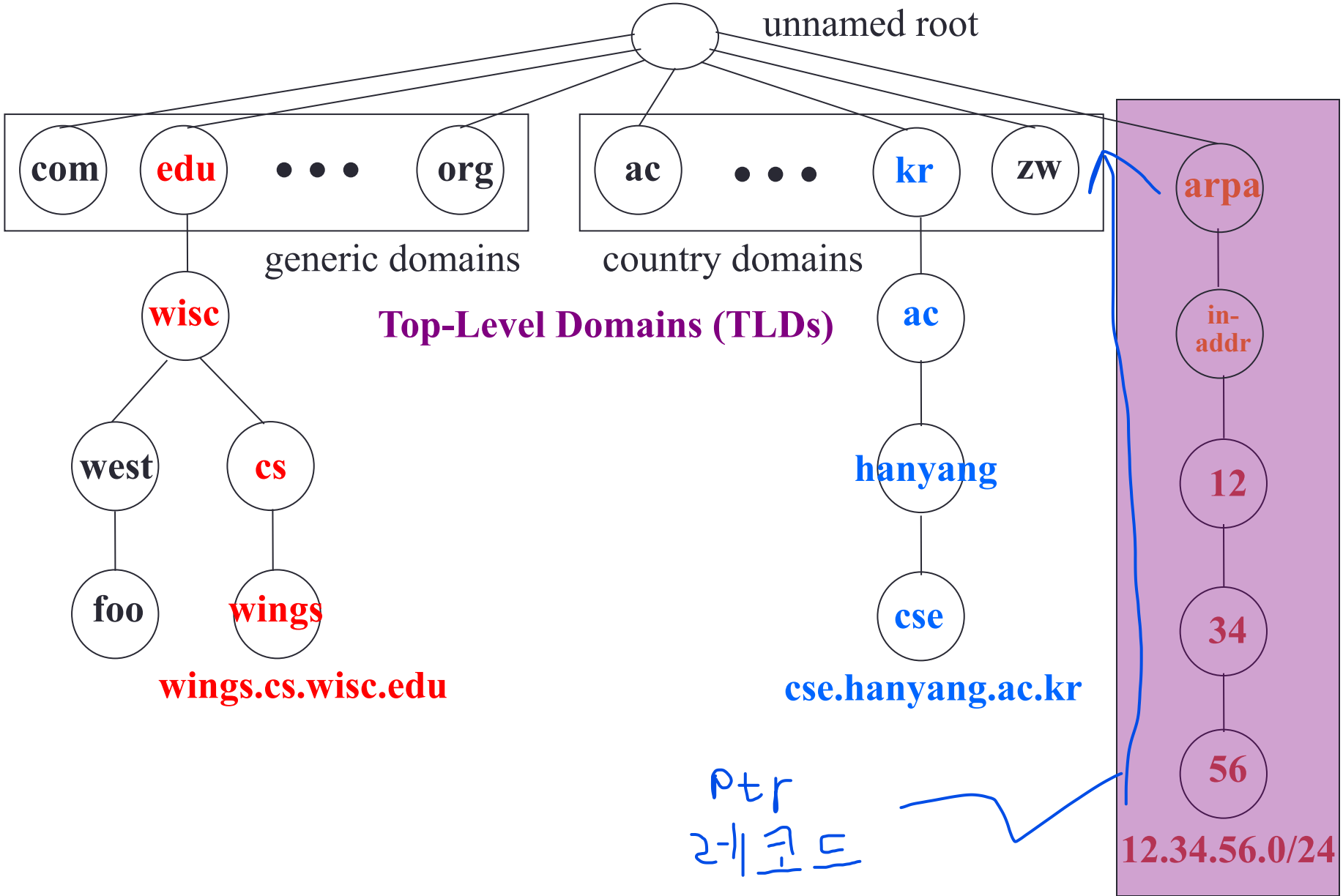
Host at
`cse.hanyang.ac.kr`
wants IP address for
`wings.cs.wisc.edu`



Reverse Mapping (Address -> Host)

- How do we go the other direction, from an IP address to the corresponding hostname?
- Addresses already have natural "quad" hierarchy:
  - 12.34.56.78
- But: quad notation has most-sig. hierarchy element on left, while www.cnn.com has it on the right
- Idea: reverse the quads = 78.56.34.12 …
  - … and look that up in the DNS
- Under what TLD?
  - Convention: **in-addr.arpa**
  - So lookup is for **`78.56.34.12.in-addr.arpa`**

Distributed Hierarchical Database



DNS Caching

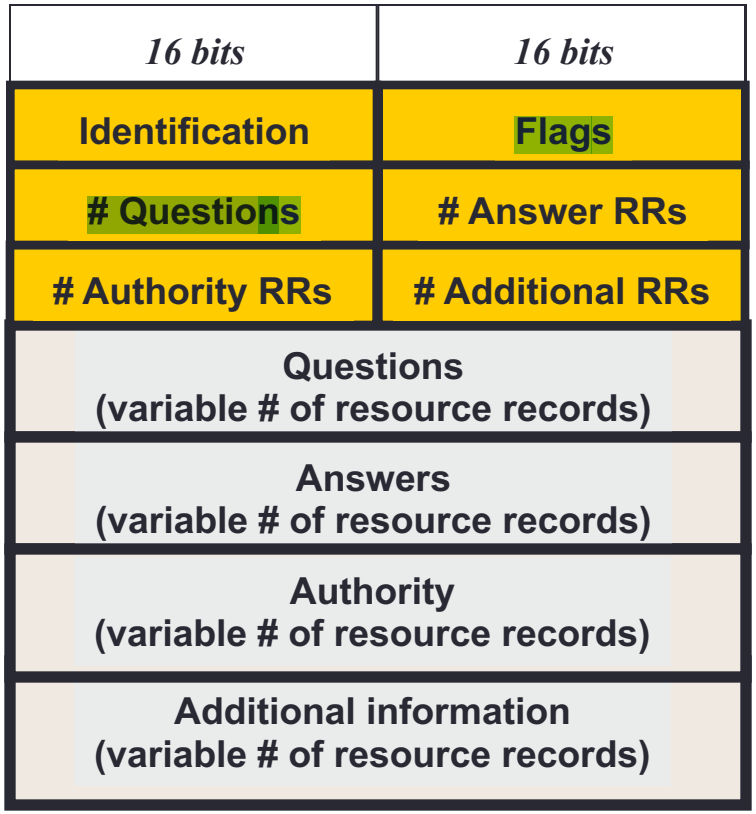
- Performing all these queries takes time
  - And all this before actual communication takes place
  - E.g., 1-second latency before starting Web download
- Caching can greatly reduce overhead
  - The top-level servers very rarely change
  - Popular sites (e.g., www.cnn.com) visited often
  - Local DNS server often has the information cached
- Note: If an entry is sent from a cache, the reply from the server is marked as "unauthoritative"

DNS Protocol

DNS protocol: *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
- Plus fields indicating size (0 or more) of optional header elements



DNS message format

MAC header	
Protocol=17 (UDP), =6 (TCP)	
Port 53 (DNS)	
Identification	2
Flags	2
Numbers of questions	2
Numbers of answer RRs	2
Numbers of authority RRs	2
Numbers of additional RRs	2
Questions	n
Answer RRs	n
Authority RRs	n
Additional information RRs	n

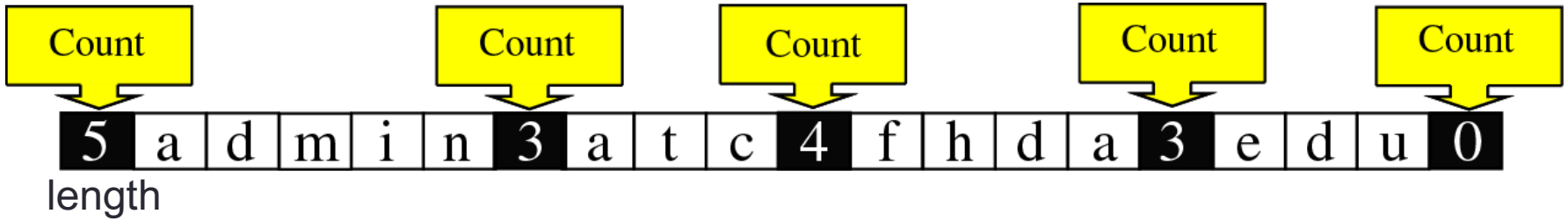
- Identification : DNS 메시지 순서 번호.
resolver에 의해 설정, 서버는 동일한 번호 사용
- Flags
(1) Q/R : Query (0), response (1)
(2) opcode : 질의나 응답의 종류
0 : 표준 query (기본값)
(3) AA : authoritative answer.
RR의 내용이 공인된 네임서버가 작성한 것임
(4) TC : truncated 표시, 512 바이트 초과된
응답인 경우에 내용이 잘려있음을 표시
(5) RD (recursion desired)
클라이언트로 부터 질의 메시지를 수신한 네임
서버는 자신의 DB 에 질의 받은 내용이 없을때
다른 서버에 다시 질의하여 얻어지는 결과를 전
(6) RA (recursion available)
RA=1 네임서버가 recursive방식 응답 가능함
(7) rcode (return code)
요청에 대한 처리 결과를 표시,
0 -> Ack 의미, 다른 값은 모두 오류를 의미

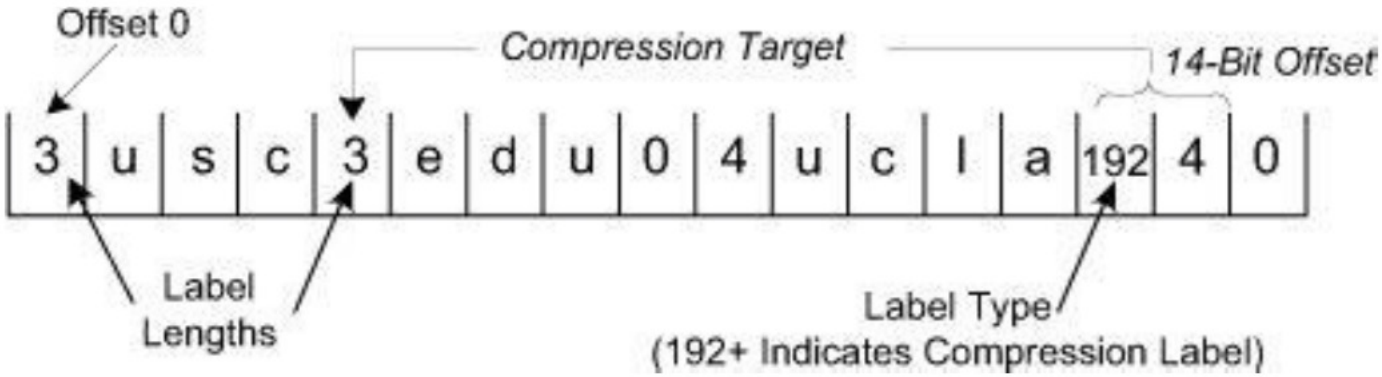
DNS message format

Question format

name string ends up with 0x00	n
Query type	2
Query class	2

admin.atc.fhda.edu





usc.edu ucla.edu

DNS message format

Query Type

1	A	IP address (IPv4)
2	NS	Name server record
5	CNAME	Canonical name
6	SOA	Mark of the start of a zone
0xC	PTR	Authority pointer record
0xD	HINFO	Host info
0xF	MX	Mail exchange record request
0xFC	AFXR	Zone transfer request for all records
0xFF	ANY	request for all records
	AAAA	IP address (IPv6)

DNS resource records

DNS: distributed DB storing resource records (RR)

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

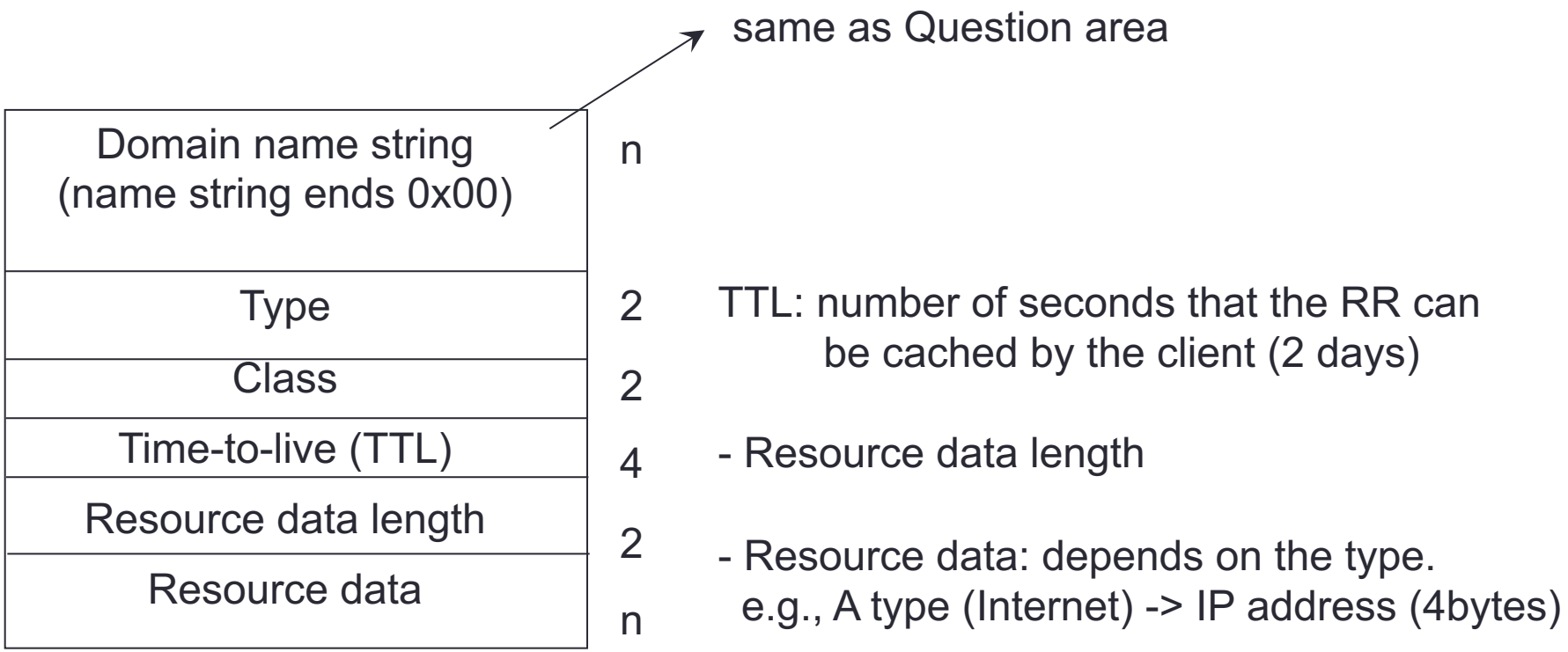
- Type=A
  - name is hostname
  - value is IP address
- Type=NS
  - **`name`** is domain (e.g. foo.com)
  - **`value`** is hostname of authoritative name server for this domain
- Type=PTR
  - **`name`** is reversed IP quads
    - E.g. 78.56.34.12.in-addr.arpa
  - **`value`** is corresponding hostname
- Type=CNAME
  - name is alias name for some "canonical" name

    E.g., www.cs.mit.edu is really eecsweb.mit.edu
  - value is canonical name
- Type=MX
  - value is name of mailserver associated with name
  - Also includes a weight/preference

DNS message format

Resource record (RR) format

(answer, authority, additional information fields)



Summary

- Domain Name System (DNS)
  - Distributed, hierarchical database
  - Distributed collection of servers
  - Caching to improve performance

- DNS lacks authentication
  - Can’t tell if reply comes from the correct source
  - Can’t tell if correct source tells the truth
  - Malicious source can insert extra (mis)information
  - Malicious bystander can spoof (mis)information
  - Playing with caching lifetimes adds extra power to attacks