National University of Computer & Emerging Sciences

CS 3001 - COMPUTER NETWORKS

Lecture 09

Chapter 2

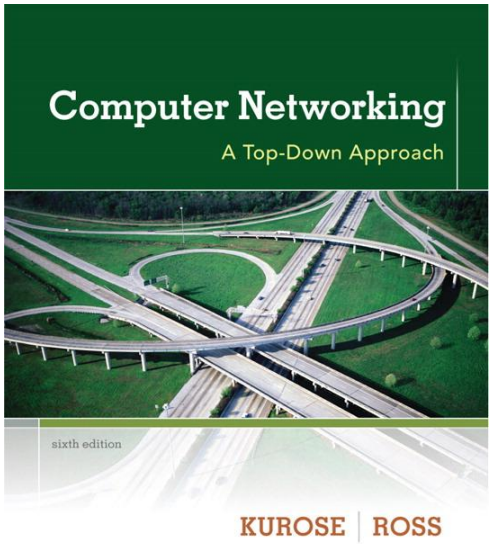
20th September, 2022

Nauman Moazzam Hayat

nauman.moazzam@lhr.nu.edu.pk

Office Hours: 02:30 pm till 06:00 pm (Every Tuesday & Thursday)

Chapter 2
Application Layer



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

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

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 (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"

DNS: services, structure

DNS services

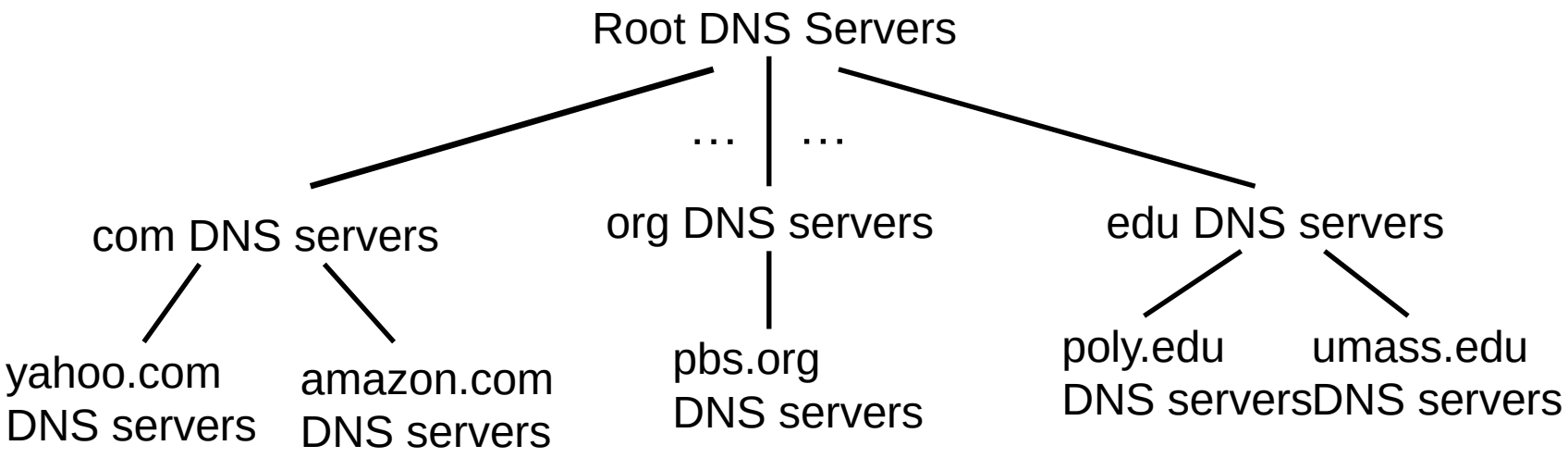
- hostname to IP address translation
- host aliasing
  - canonical, alias names
- mail server aliasing
- load distribution
  - replicated Web servers: many IP addresses correspond to one name

why not centralize DNS?

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

A: doesn't scale!

DNS: a distributed, hierarchical database

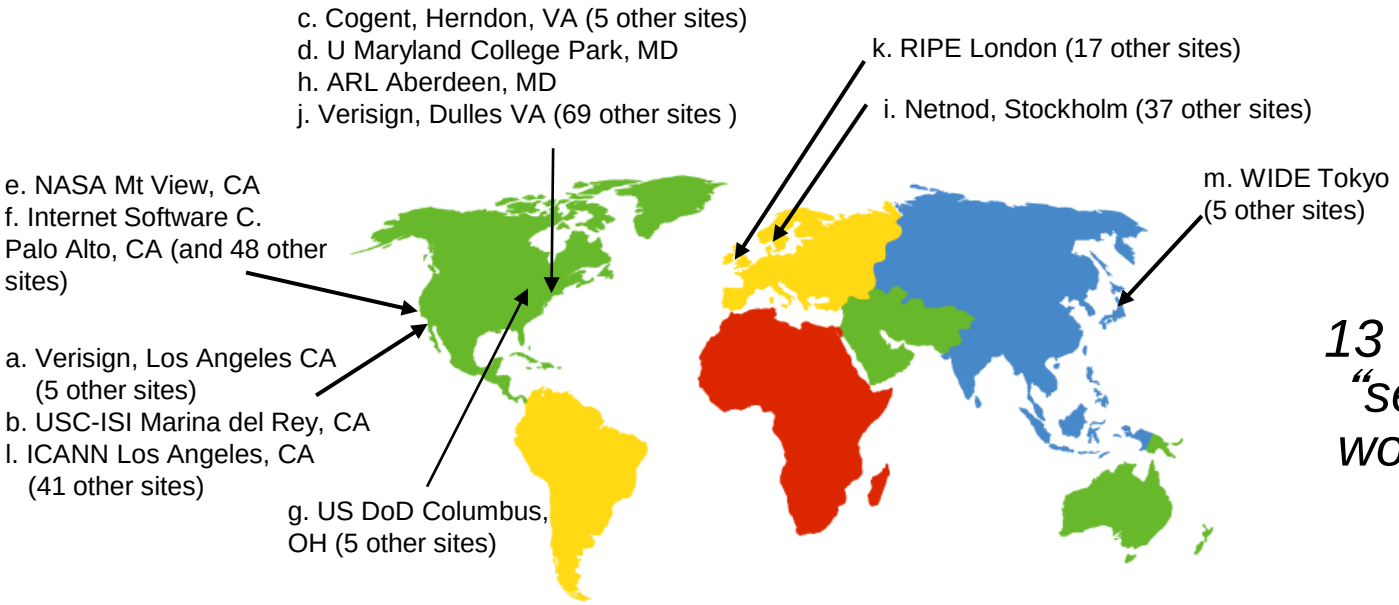


client wants IP for www.amazon.com; 1st approx:

- 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



13 root name "servers" worldwide

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 (Default Name Server)

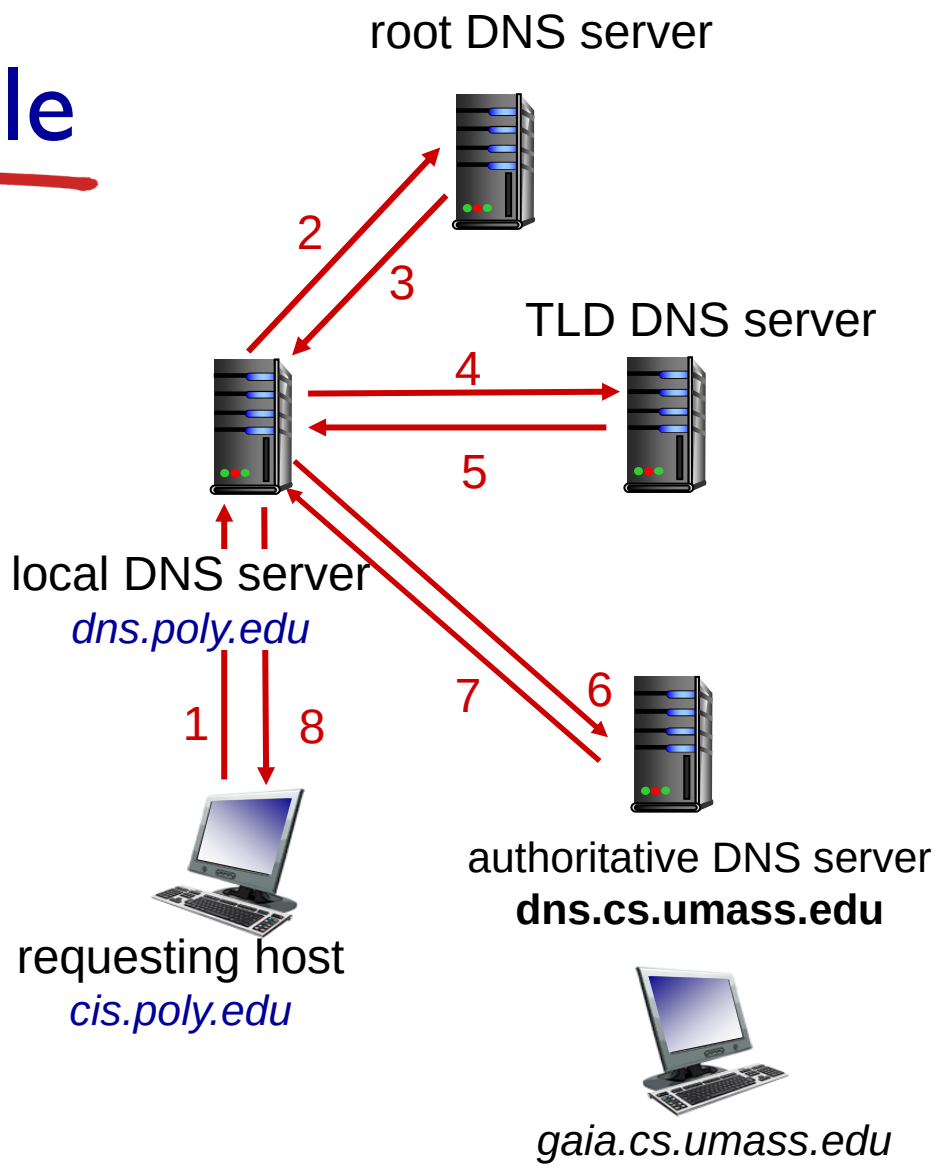
- does not strictly belong to hierarchy
- each ISP (residential ISP, company, university) has one
  - also called "default name server"
- when host makes DNS query, query is sent to its local DNS server
  - has local cache of recent name-to-address translation pairs (but may be out of date!)
  - acts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

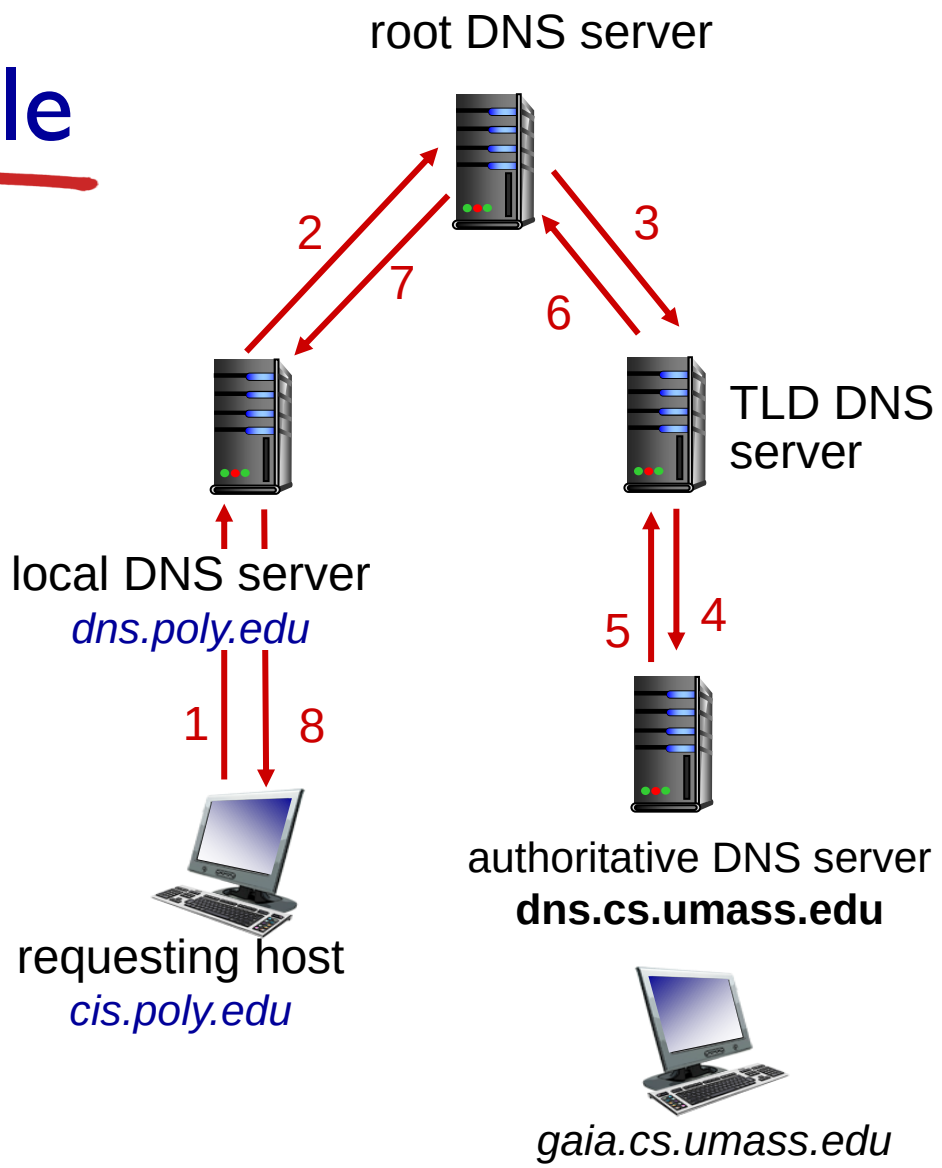
- contacted server replies with name of server to contact
- "I don't know this name, but ask this server"



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 db storing resource records (RR) (RFC 1035)

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

if type=A (Address Mapping Record, RFC 1035)

- **`name`** is hostname
- **`value`** is IP address
- Used to map (point) a domain name to an IP address
- E.g. (relay1.bar.foo.com, 145.37.93.126,A)

if type=CNAME (Canonical Name Record, RFC 1035)

- **`name`** is alias (mnemonic) name for some "canonical" (the real) name
- **`value`** is canonical name
- **`www.ibm.com`** is really **`servereast.backup2.ibm.com`**
- Used to map (point) a domain name to another domain name (for example your website is example.com, but you have also registered examples.com, thus examples.com can be redirected towards example.com via this record
- E.g. (foo.com, relay1.bar.foo.com, CNAME)

if type=NS (Name Server Record, RFC 1035)

- **`name`** is domain
- **`value`** is hostname of authoritative name server for this domain
- NS records specifies which DNS server is authoritative for this domain
- E.g. (foo.com, dns.foo.com, NS)

if type=MX (Mail Exchange Record, RFC 1035)

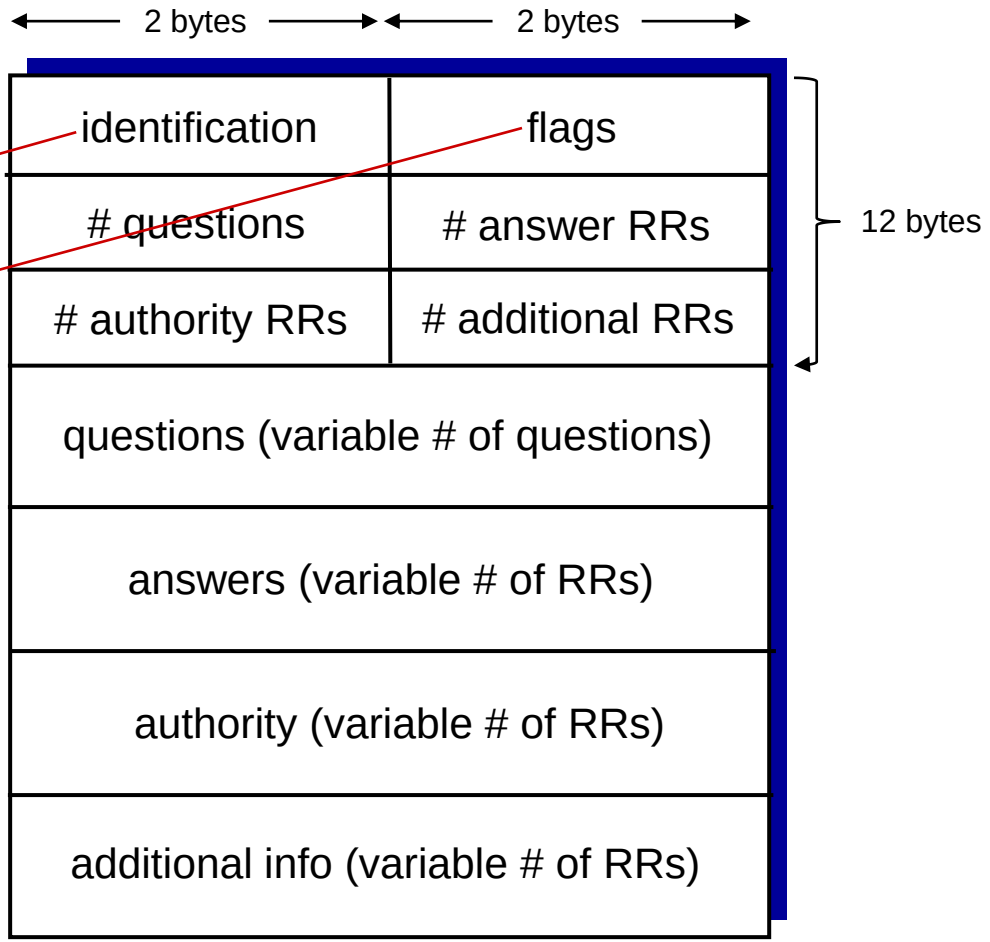
- **`name`** is alias name for some "canonical" (the real) name
- **`value`** is canonical name of mailserver associated with alias **`name`**
- Same as CNAME but for mailserver
- Used by SMTP to locate mail server name for that domain (thus mail server name must also have a Type A record.)
- E.g. (foo.com, mail.bar.foo.com, MX)

DNS protocol, messages

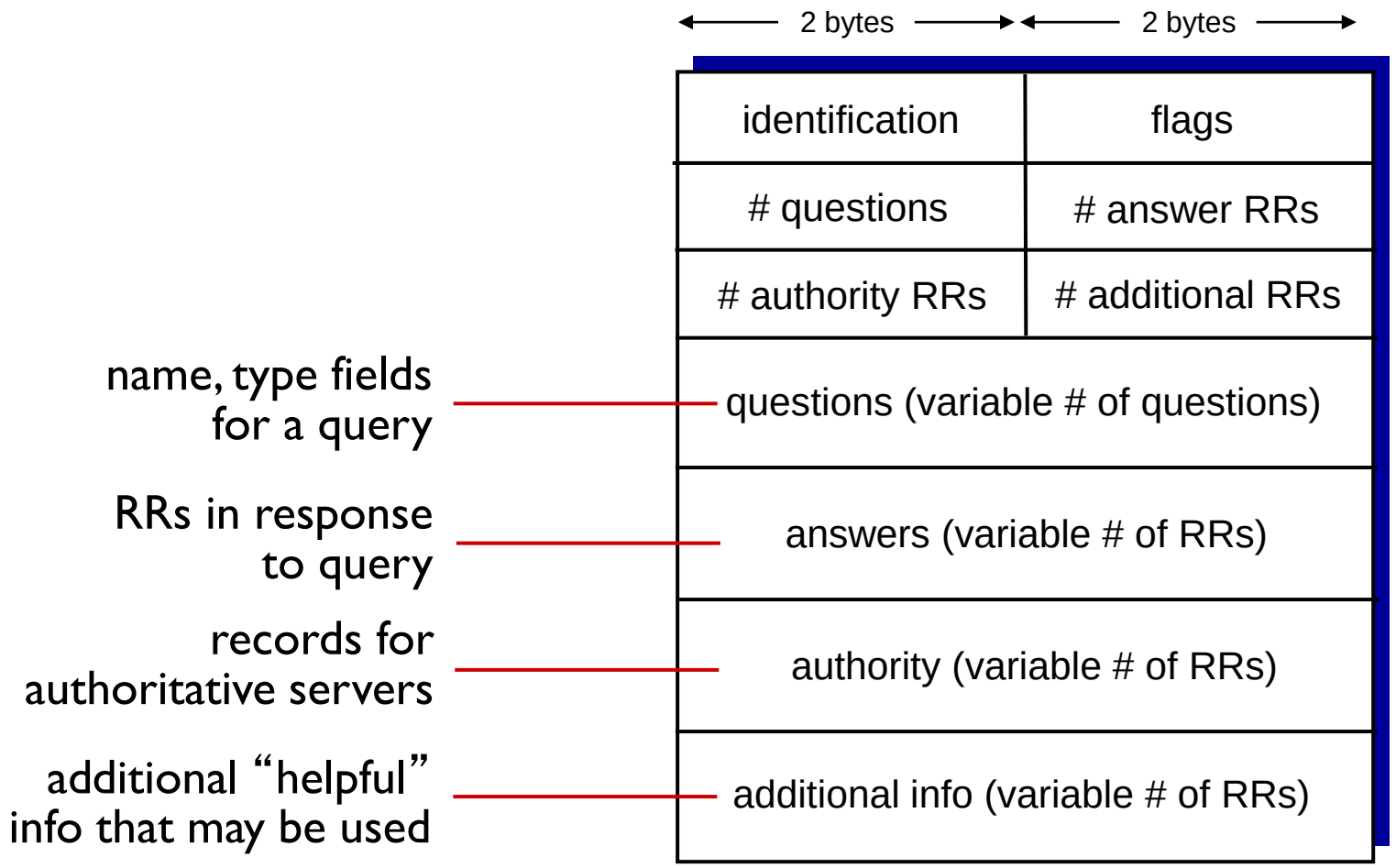
- *query* and *reply* messages, both with same *message format*

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



DNS Messages (Header Section)

- **Header Section:** The first 12 bytes is the header section, which has a number of fields:

  - The first field is a 16-bit number that identifies the query. This identifier is copied into the reply message to a query, allowing the client to match received replies with sent queries.

  - Flags: There are a number of flags in the flag field. A one-bit query/reply flag indicates whether the message is a query (0) or a reply (1). A one-bit authoritative flag is set in a reply message when a DNS server is an authoritative server for a queried name. A one-bit recursion-desired flag is set when a client (host or DNS server) desires that the DNS server perform recursion when it doesn't have the record. A one-bit recursion available flag is set in a reply if the DNS server supports recursion.

- **# Fields (Four number-of fields):** These fields indicate the number of occurrences of the four types of data sections that follow the header.

  - **Question Section:** contains information about the query that is being made. This section includes (i) a name field that contains the name that is being queried, and (ii) a type field that indicates the type of question being asked about the name

  - **Answer Section:** In a reply from a DNS server, the answer section contains the resource records for the name that was originally queried.

  - **Authority Section:** contains records of other authoritative servers.

  - **Additional Section:** contains other helpful records

Inserting records into DNS

- example: you start new startup "Network Utopia"
- register name networkuptopia.com at *DNS registrar* (e.g., Network Solutions, GoDaddy)
  - You 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

Quiz # 2 (Chapter – 2)

- *Quiz # 2 for Chapter 2 to be taken in the class on Thursday, 22nd September, 2022 during the lecture time*
- *Quiz to be taken for own section only*

No Retake

Be on time