COMPUTER NETWORKS

TEAM NETWORKS

Department of Computer Science and Engineering

PES UNIVERSITY
CELEBRATING 50 YEARS

COMPUTER NETWORKS

Application Layer

Department of Computer Science and Engineering

PES UNIVERSITY
CELEBRATING 50 YEARS

Unit – 2 Application Layer

2.3 The Domain Name System

DNS: Domain Name System

people: many identifiers:

- SSN, name, passport #

Internet hosts, routers:

- IP address (32 bit) - used for addressing datagrams
- "name", e.g., cs.umass.edu - 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"

www.abc.example.com -> Canonical Host Name
www.example.com -> Alias Name

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

Q: Why not centralize DNS?

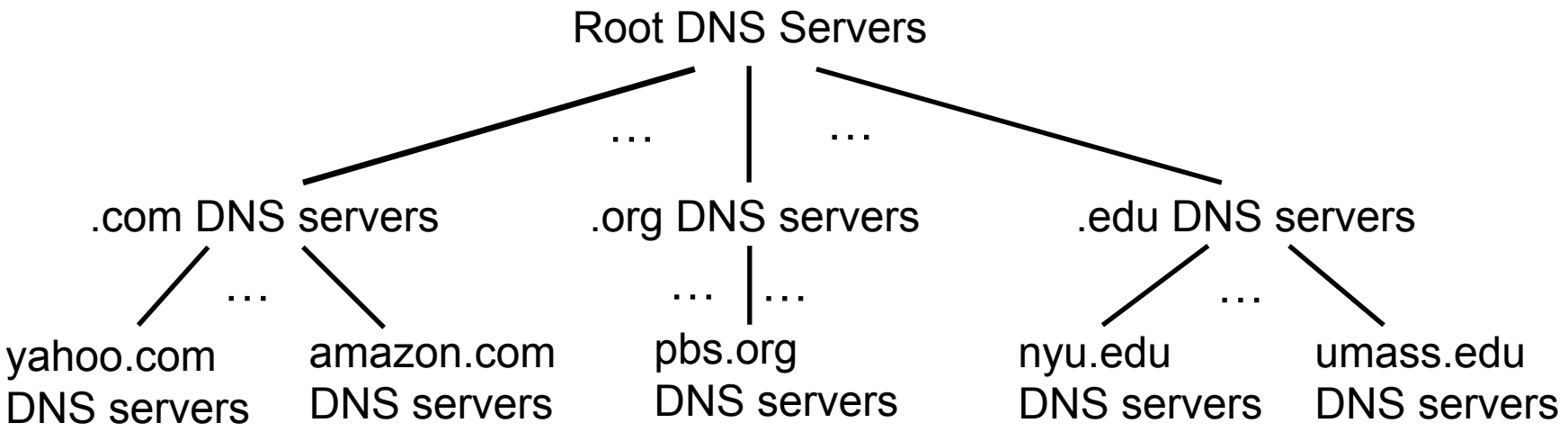
- single point of failure
- traffic volume
- distant centralized database
- maintenance

A: doesn't scale!

- Comcast DNS servers alone: 600B DNS queries per day

www.abc.example.com -> Canonical Host Name
bob@example.com -> Alias Name

DNS: a distributed, hierarchical database



Root

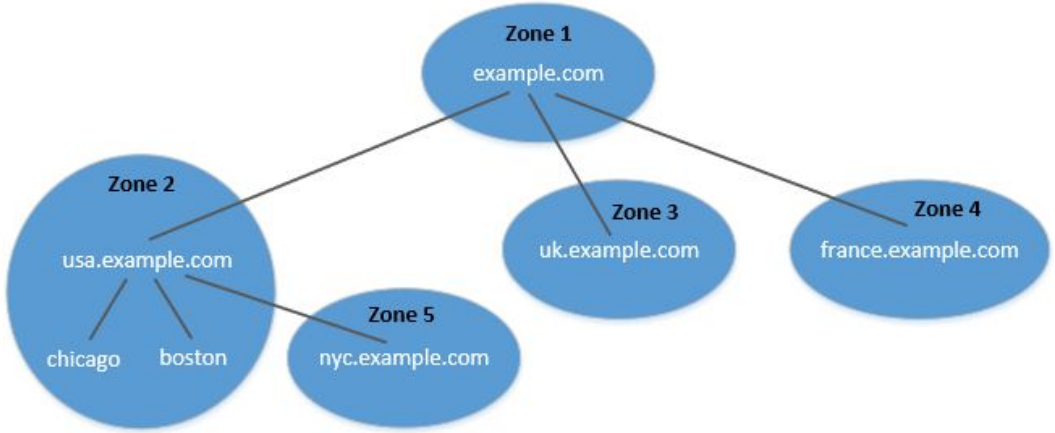
Top Level Domain

Authoritative

Client wants IP address 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 Zone vs Domain



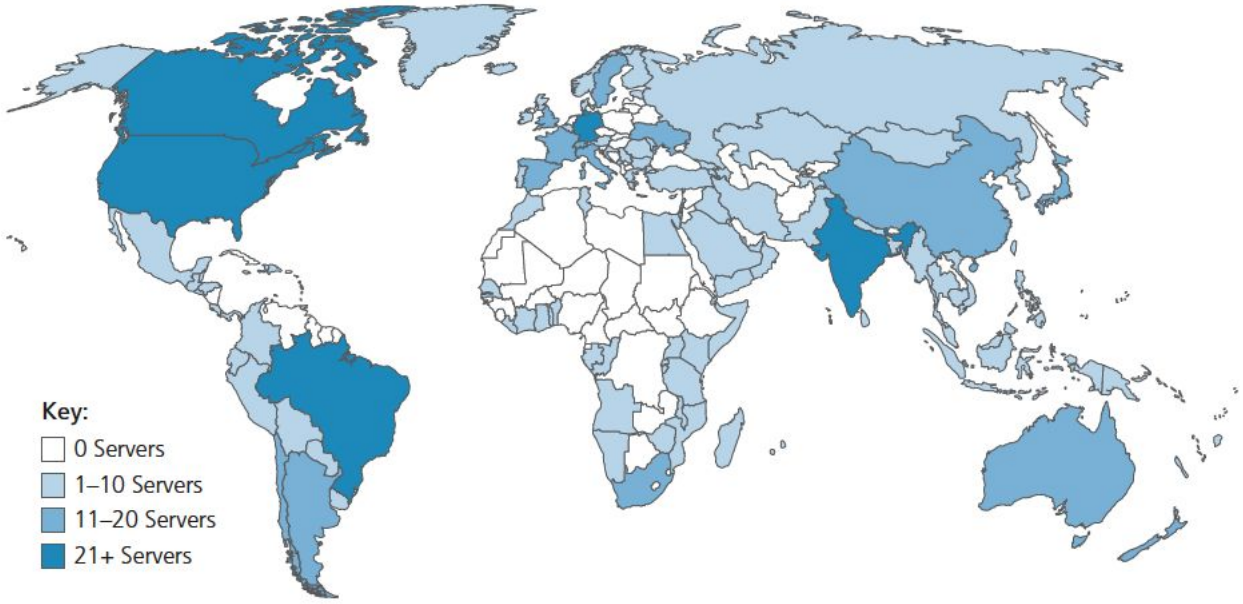
- DNS is organized according to zones.
- A zone groups contiguous domains and subdomains on the domain tree.
- Assign management authority to an entity.

- The tree structure depicts subdomains within example.com domain.
- Multiple DNS zones one for each country. The zone keeps records of who the authority is for each of its subdomains.
- The zone for example.com contains only the DNS records for the hostnames that do not belong to any subdomain like mail.example.com

DNS: root name servers

- official, contact-of-last-resort by name servers that can not resolve name
- *incredibly important* Internet function
  - Internet couldn't function without it!
  - DNSSEC – provides security (authentication and message integrity)
- ICANN (Internet Corporation for Assigned Names and Numbers) manages root DNS domain

13 logical root name "servers" worldwide each "server" replicated many times (~200 servers in US)



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.: .cn, .uk, .fr, .ca, .jp
- Network Solutions: authoritative registry for .com, .net TLD
- Educause: .edu TLD

Authoritative DNS servers:

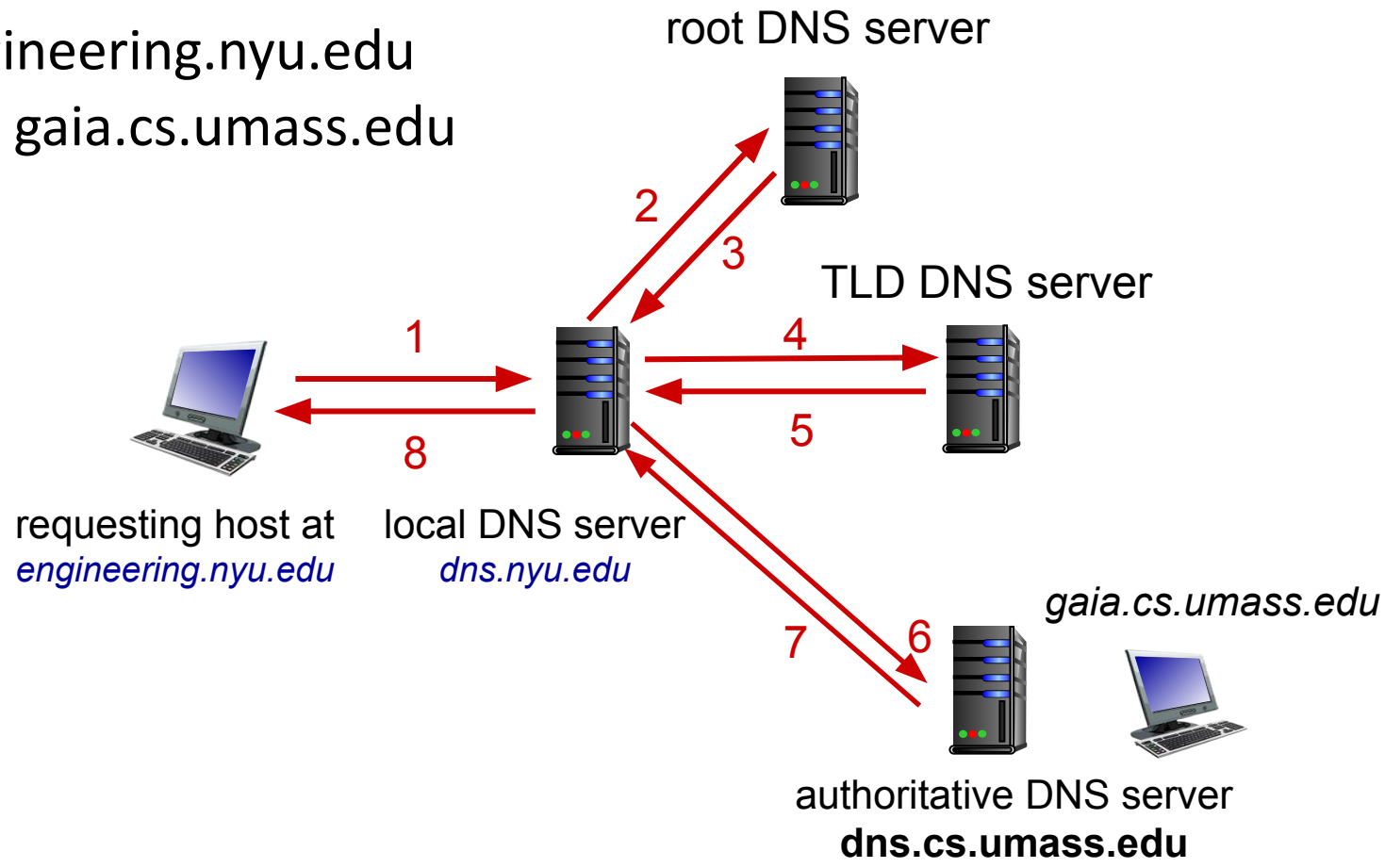
- organization's own DNS server(s), providing authoritative hostname to IP mappings for organization's named hosts
- can be maintained by organization or service provider

Local DNS Name Servers

- 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: iterated query

Example: host at engineering.nyu.edu wants IP address for gaia.cs.umass.edu



Iterated query:

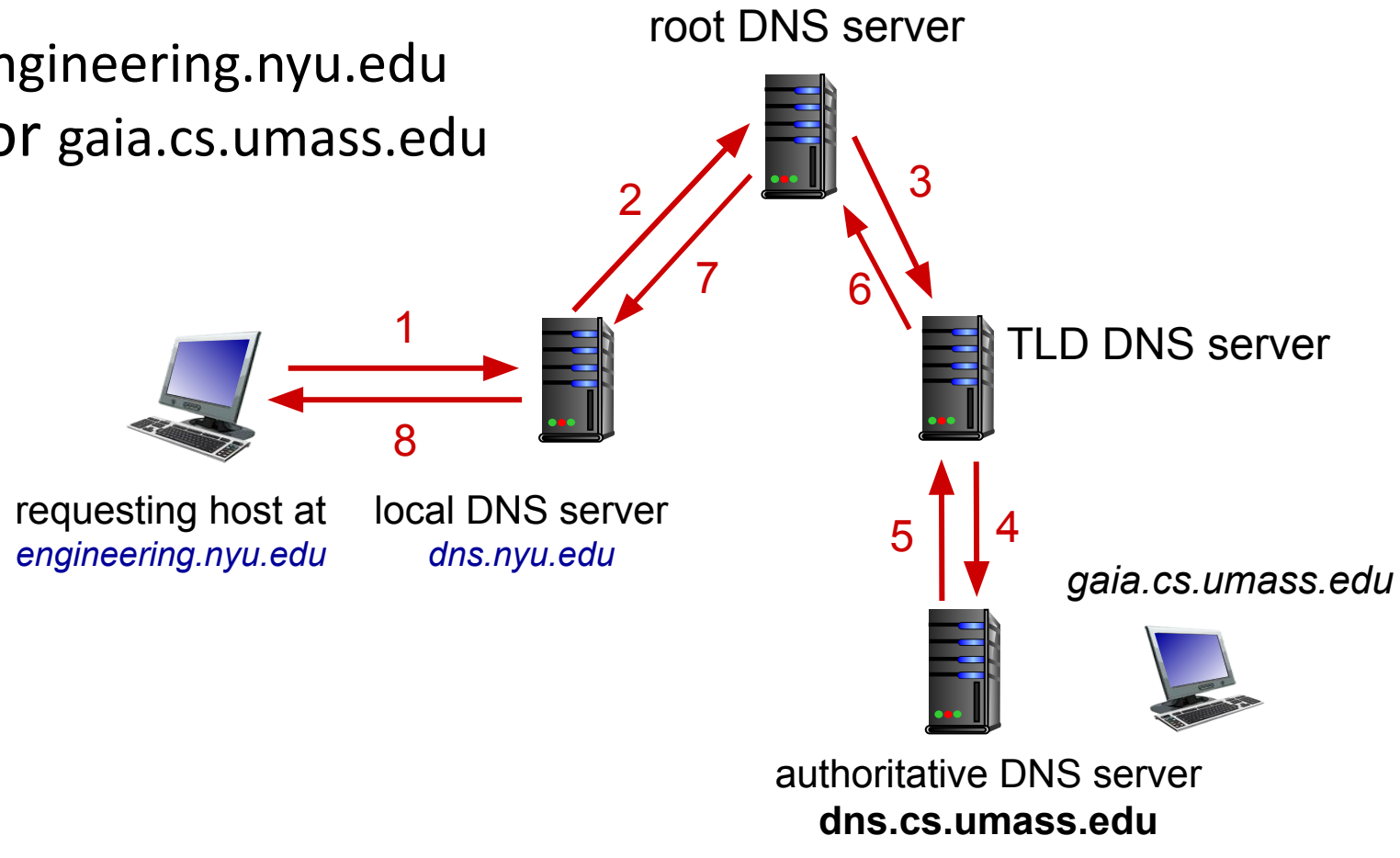
- contacted server replies with name of server to contact
- "I don't know this name, but ask this server"

DNS name resolution: recursive query

Example: host at engineering.nyu.edu wants IP address for gaia.cs.umass.edu

Recursive query:

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



Caching and Updating DNS Records

- Suppose that a host **apricot.nyu.edu** queries **dns.nyu.edu** for the IP address for the hostname **cnn.com**. After an hour later, another NYU host, say, **kiwi.nyu.edu**, also queries **dns.nyu.edu.**
- 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

relayl.bar.foo.com, 145.37.93.126, A

type=NS

- `name` is domain (e.g., foo.com)
- `value` is hostname of authoritative name server for this domain

foo.com, dns.foo.com, NS

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

ibm.com, servereast.backup2.ibm.com, CNAME

type=MX

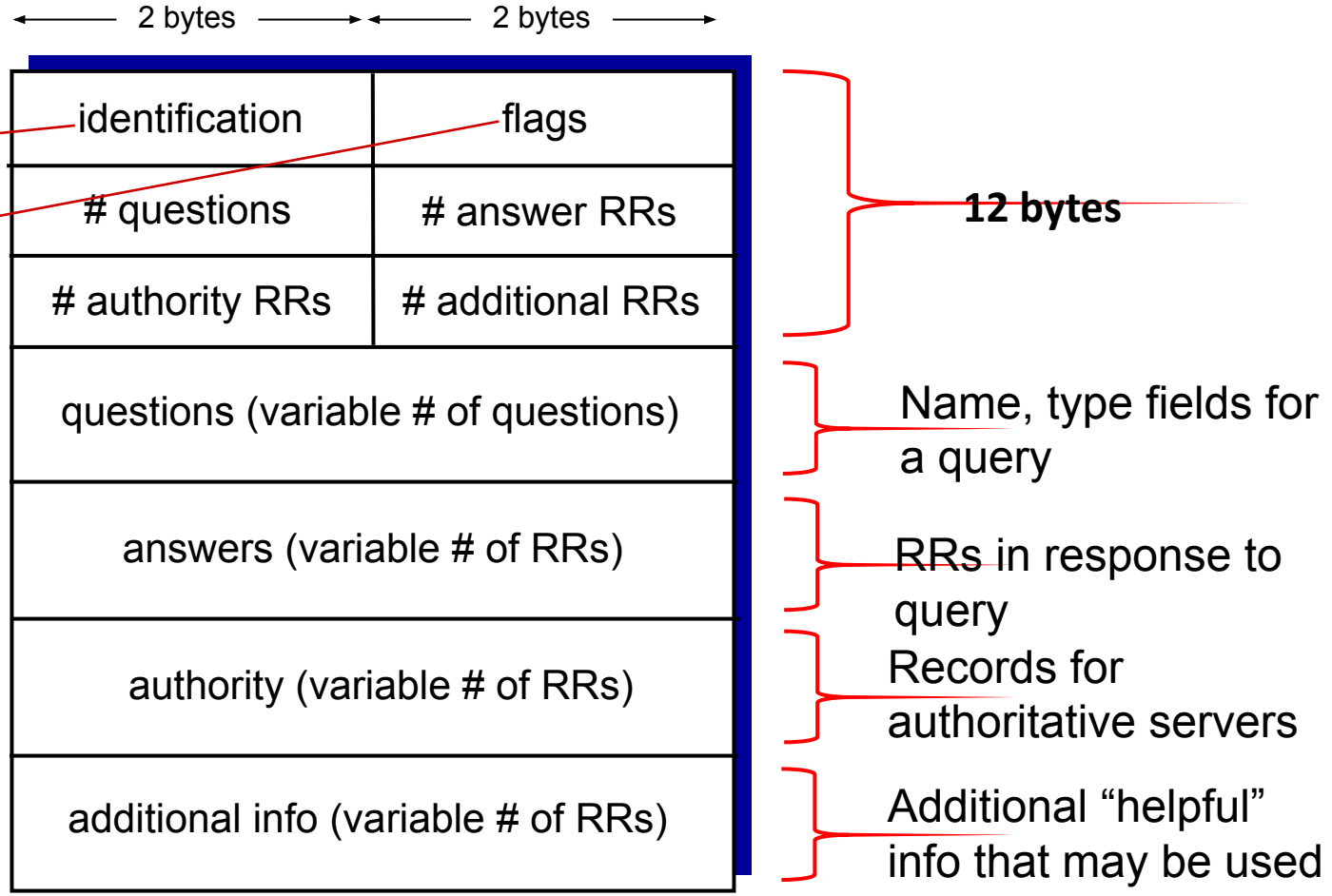
- `value` is canonical name of a mailserver associated with alias hostname `name`

example.com, mail.example.com, MX

DNS *query* and *reply* messages, both have same *format:*

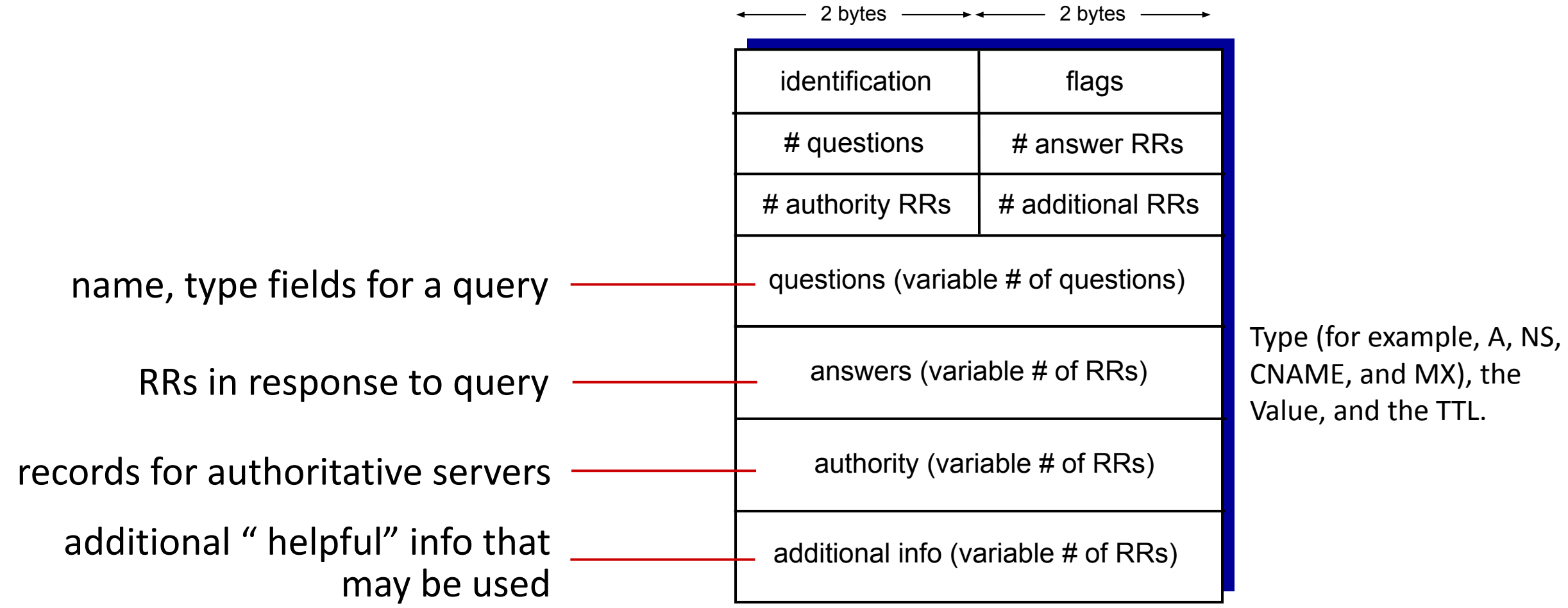
message header:

- identification: 16 bit # for query, reply to query uses same #
- flags:
  - query or reply (1-bit)
  - recursion desired
  - recursion available
  - reply is authoritative



DNS Protocol Messages

DNS *query* and *reply* messages, both have same *format:*



Emulating Local DNS Server (Step 1: Ask Root)

Directly send the query to this server.

```
seed@ubuntu:~$ dig @a.root-servers.net www.example.net

(Only a portion of the reply is shown here)
;; QUESTION SECTION:
;www.example.net.               IN      A

;; AUTHORITY SECTION:
net.                    172800  IN      NS      m.gtld-servers.net.
net.                    172800  IN      NS      l.gtld-servers.net.
net.                    172800  IN      NS      k.gtld-servers.net.

;; ADDITIONAL SECTION:
m.gtld-servers.net.     172800  IN      A       192.55.83.30
l.gtld-servers.net.     172800  IN      A       192.41.162.30
k.gtld-servers.net.     172800  IN      A       192.52.178.30
```

No answer (the root does not know the answer)

Go ask them!

Steps 2-3: Ask .net & example.net servers

```
seed@ubuntu:~$ dig @m.gtld-servers.net www.example.net

;; QUESTION SECTION:
;www.example.net.               IN      A

;; AUTHORITY SECTION:
example.net.            172800  IN      NS      a.iana-servers.net.
example.net.            172800  IN      NS      b.iana-servers.net.

;; ADDITIONAL SECTION:
a.iana-servers.net.     172800  IN      A       199.43.132.53
b.iana-servers.net.     172800  IN      A       199.43.133.53
```

Go ask them!

```
seed@ubuntu:$ dig @a.iana-servers.net www.example.net

;; QUESTION SECTION:
;www.example.net.               IN      A

;; ANSWER SECTION:
www.example.net.        86400   IN      A       93.184.216.34
```

- Ask an example.net nameservers.

Finally got the answer

Inserting records into DNS

Example: new startup “Network Utopia”

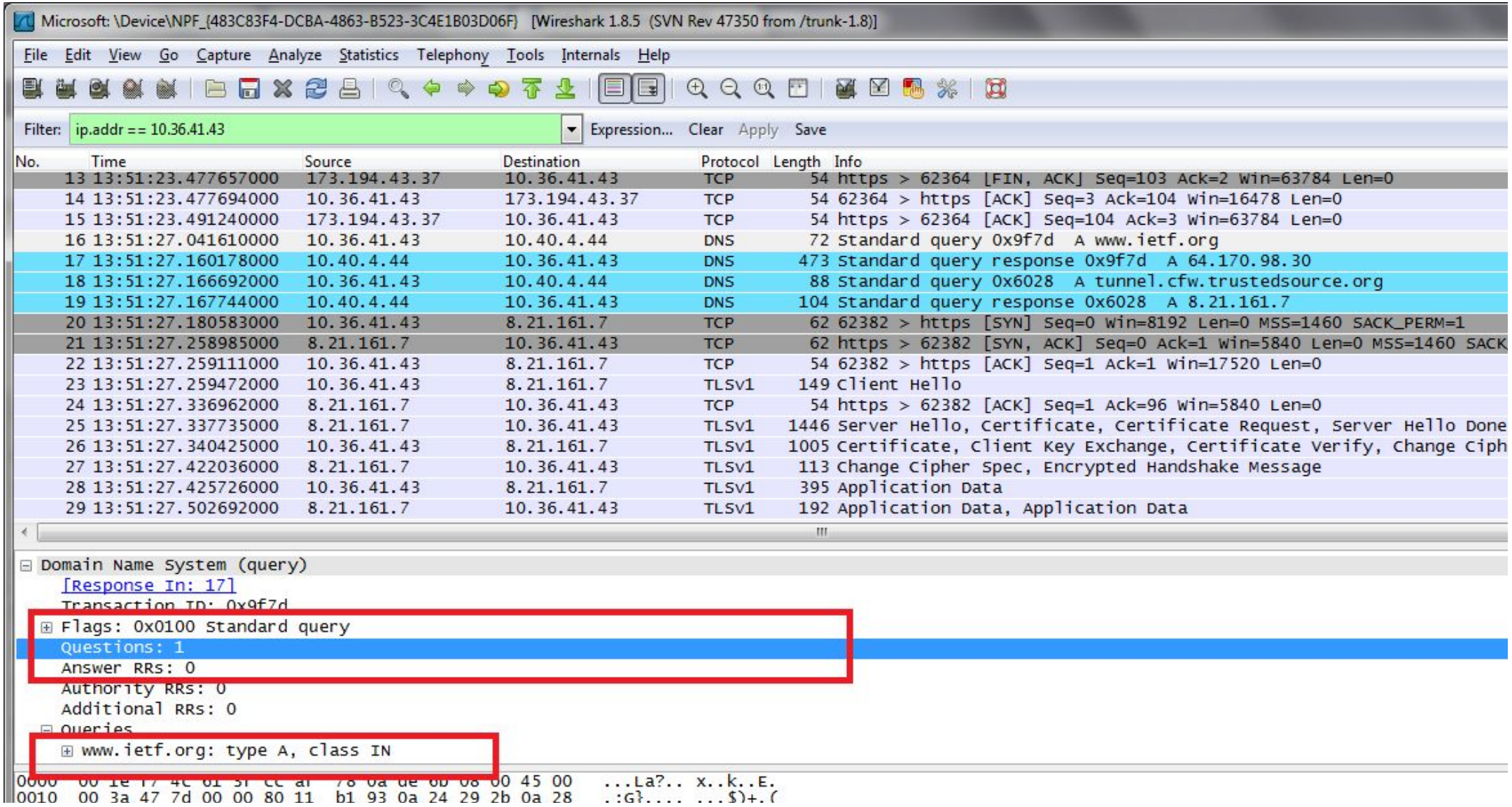
- register name **networkuptopia.com** at *DNS registrar* (e.g., Network Solutions)
  - provide names, IP addresses of authoritative name server (primary and secondary)
  - registrar inserts NS, A RRs into .com TLD server:

    `(networkutopia.com, dns1.networkutopia.com, NS)`

    `(dns1.networkutopia.com, 212.212.212.1, A)`
- create authoritative server locally with IP address `212.212.212.1`
  - type A record for www.networkuptopia.com
  - type MX record for networkutopia.com

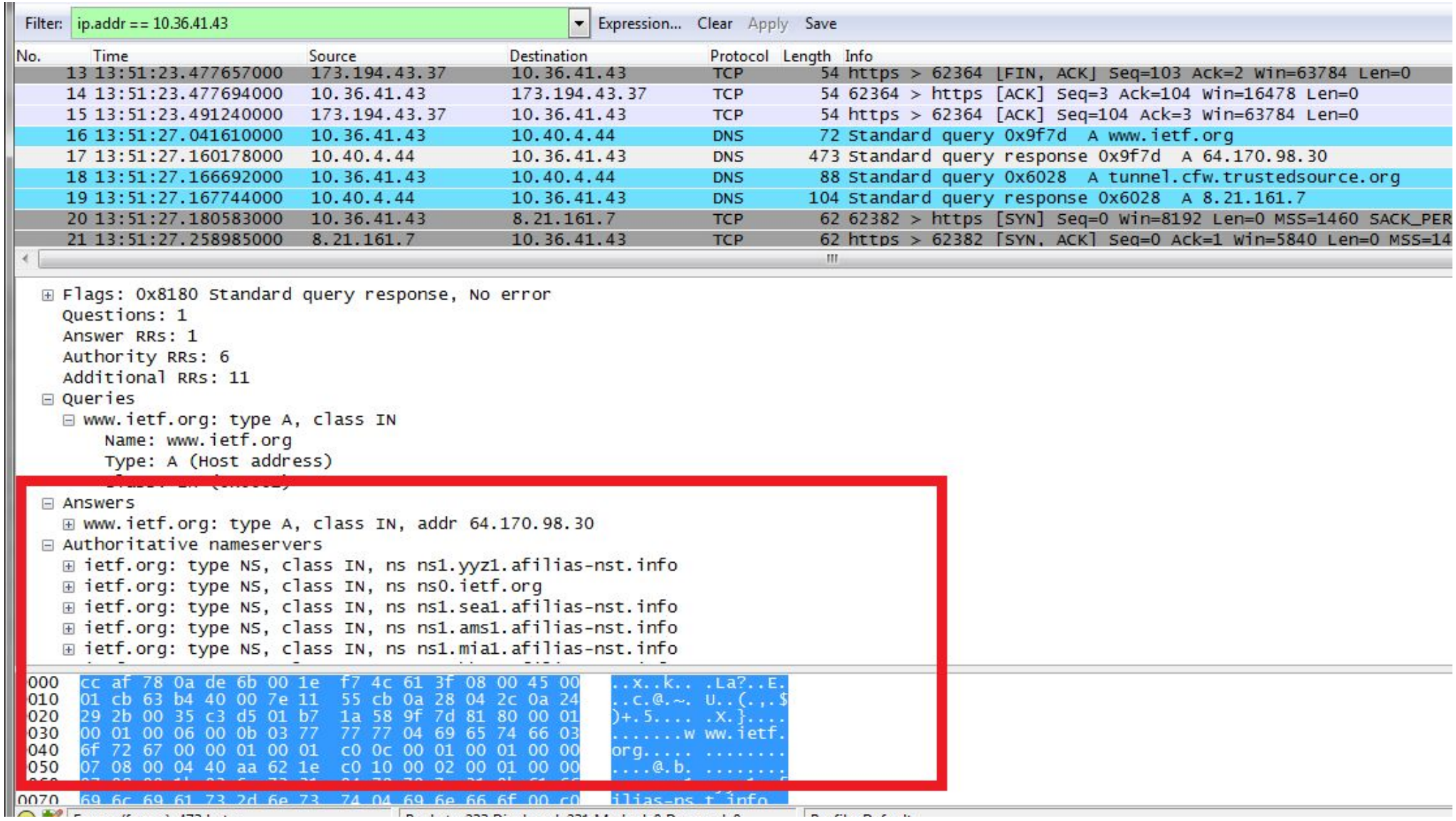
COMPUTER NETWORKS

DNS Request - Wireshark Packet Capture



COMPUTER NETWORKS

DNS Response - Wireshark Packet Capture



Suggested Readings

- DNS (Domain Name System) – Explained – https://youtu.be/JkEYOt08-rU
- How a DNS Server (Domain Name System) works – https://youtu.be/rdVPflECed8
- Wireshark Lab: DNS v7.0 – http://www-net.cs.umass.edu/wireshark-labs/Wireshark_DNS_v7.0.pdf

Thank You
For Your Attention

THANK YOU

TEAM NETWORKS

Department of Computer Science and Engineering