



# COMPUTER NETWORKS

---

## TEAM NETWORKS

Department of Computer Science and Engineering

# COMPUTER NETWORKS

---

## Application Layer

Department of Computer Science and Engineering

## Unit – 2 Application Layer

### 2.3 The 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”

### 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

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

www.example.com -> Alias Name

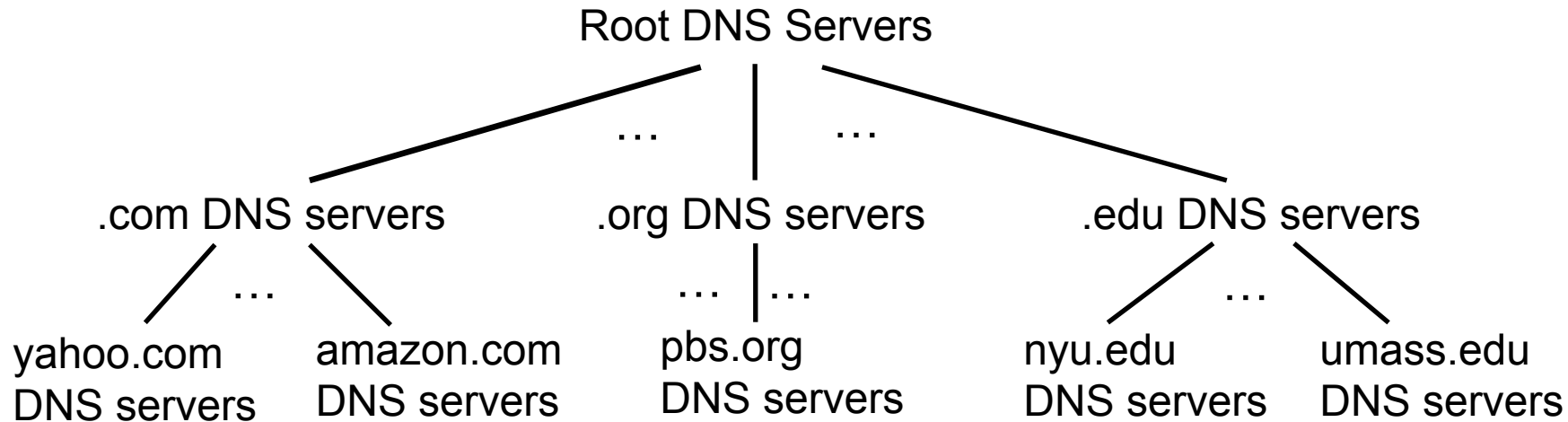
### *Q: Why not centralize DNS?*

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

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

### *A: doesn't scale!*

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



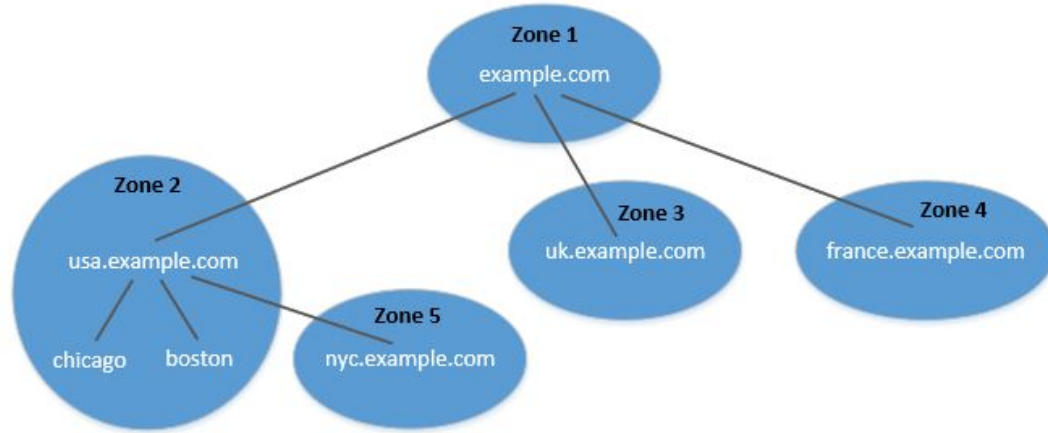
*Root*

*Top Level Domain*

*Authoritative*

Client wants IP address for `www.amazon.com`; 1<sup>st</sup> approximation:

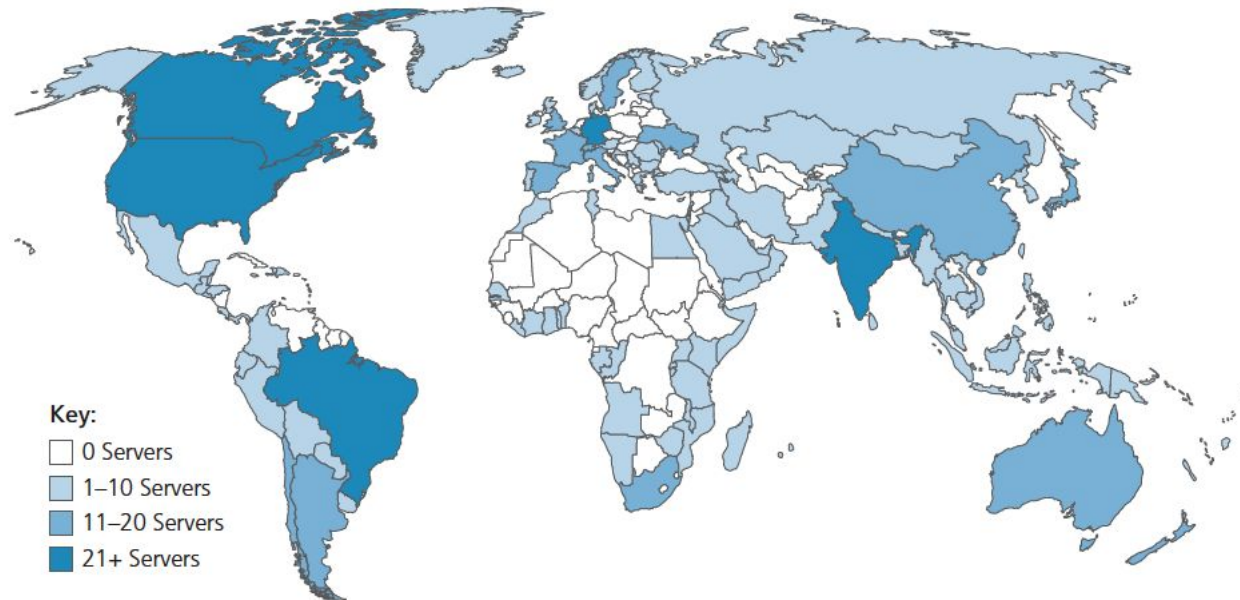
- client queries root server to find .com DNS server
- client queries .com DNS server to get amazon.com DNS server
- client queries amazon.com DNS server to get IP address for `www.amazon.com`



- DNS 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

- 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)





### 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:

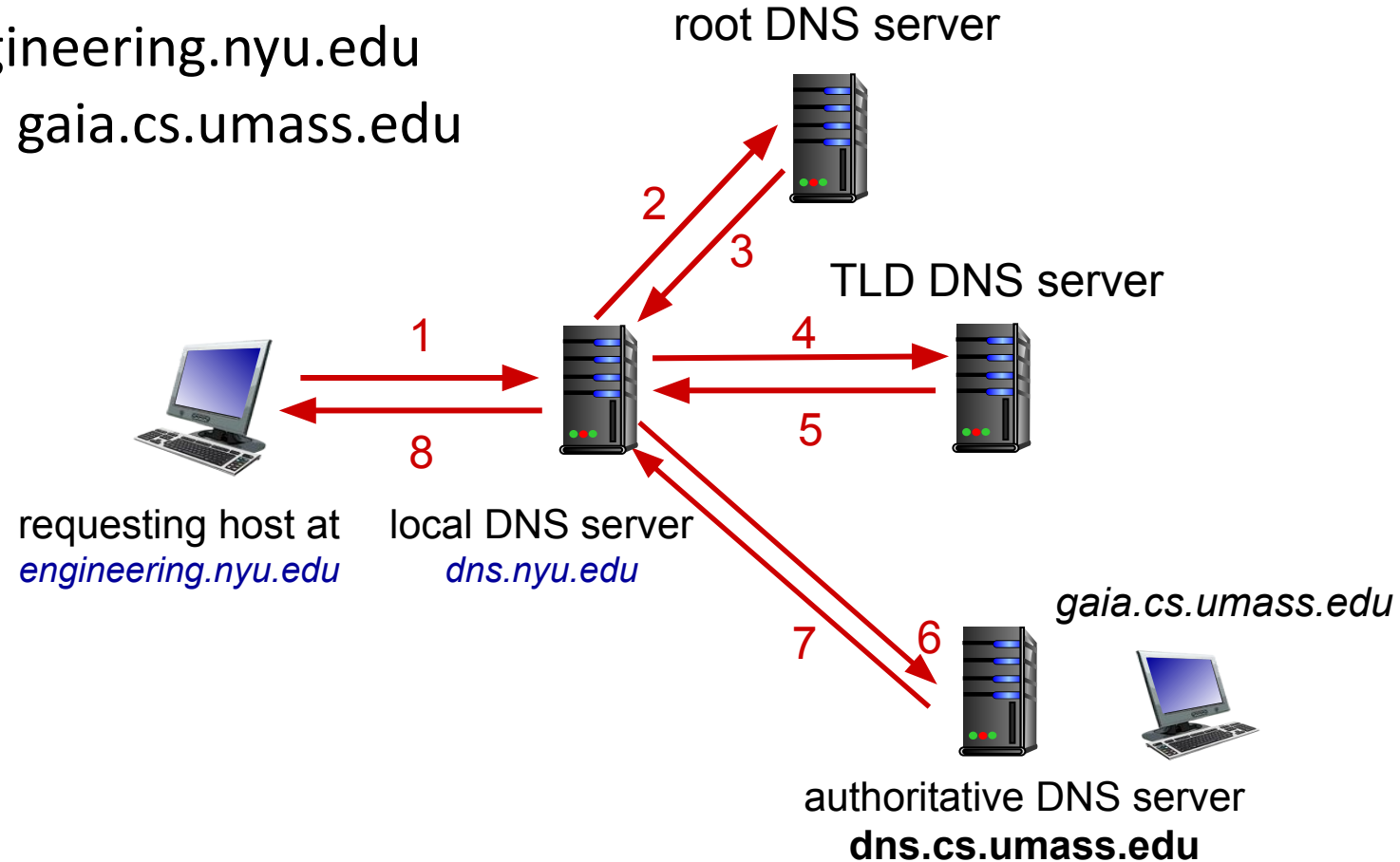
- organization's own DNS server(s), providing authoritative hostname to IP mappings for organization's named hosts
- can be maintained by organization or service provider

- 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

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

### Iterated query:

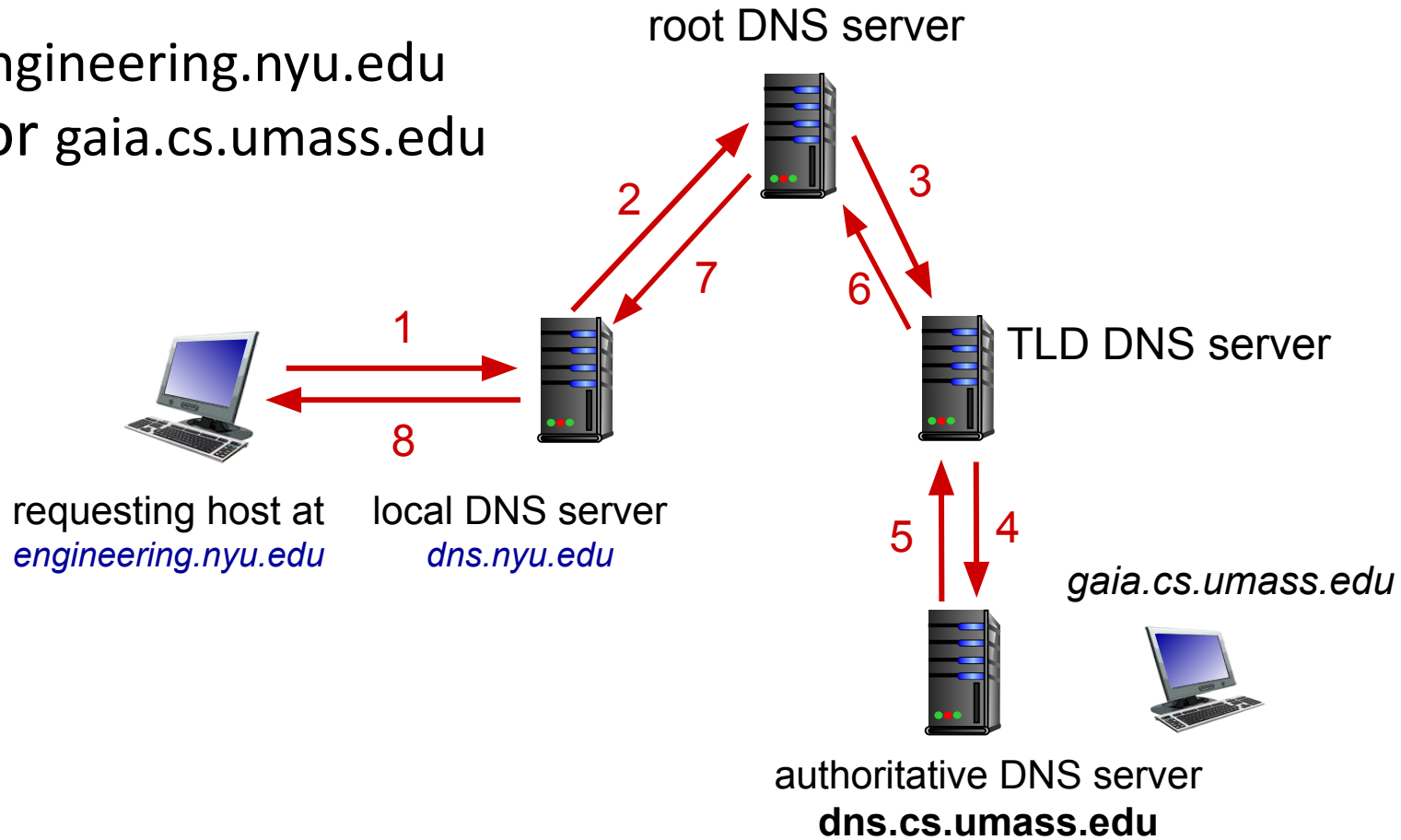
- contacted server replies with name of server to contact
- “I don’t know this name, but ask this server”



**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?



- 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:** 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

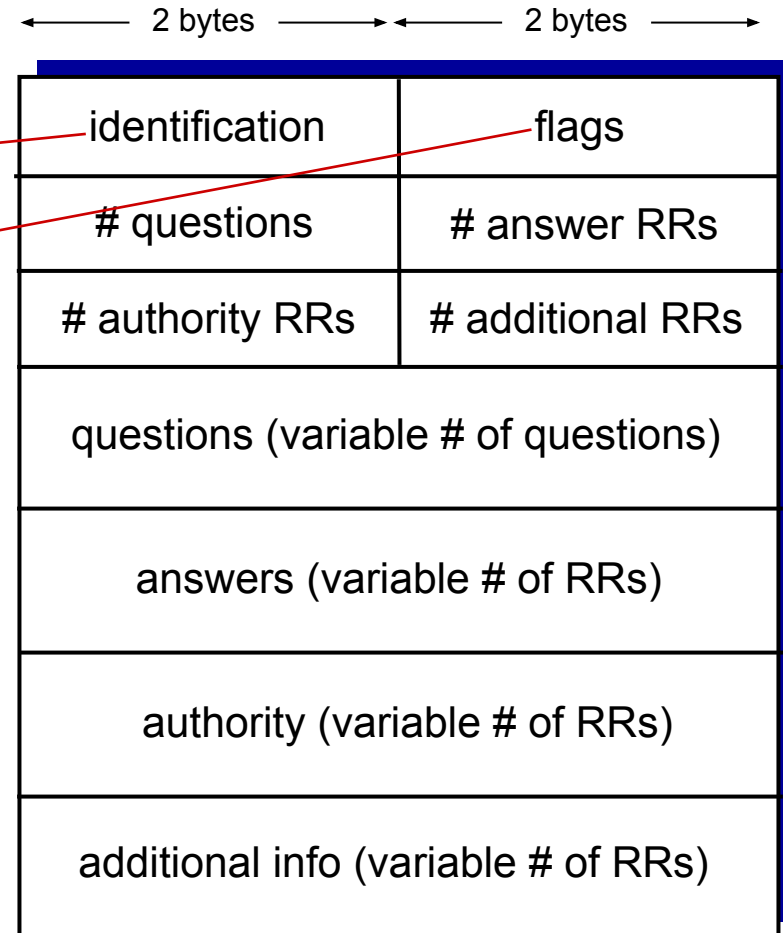
# COMPUTER NETWORKS

## DNS Protocol Messages

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



**12 bytes**

Name, type fields for a query

RRs in response to query

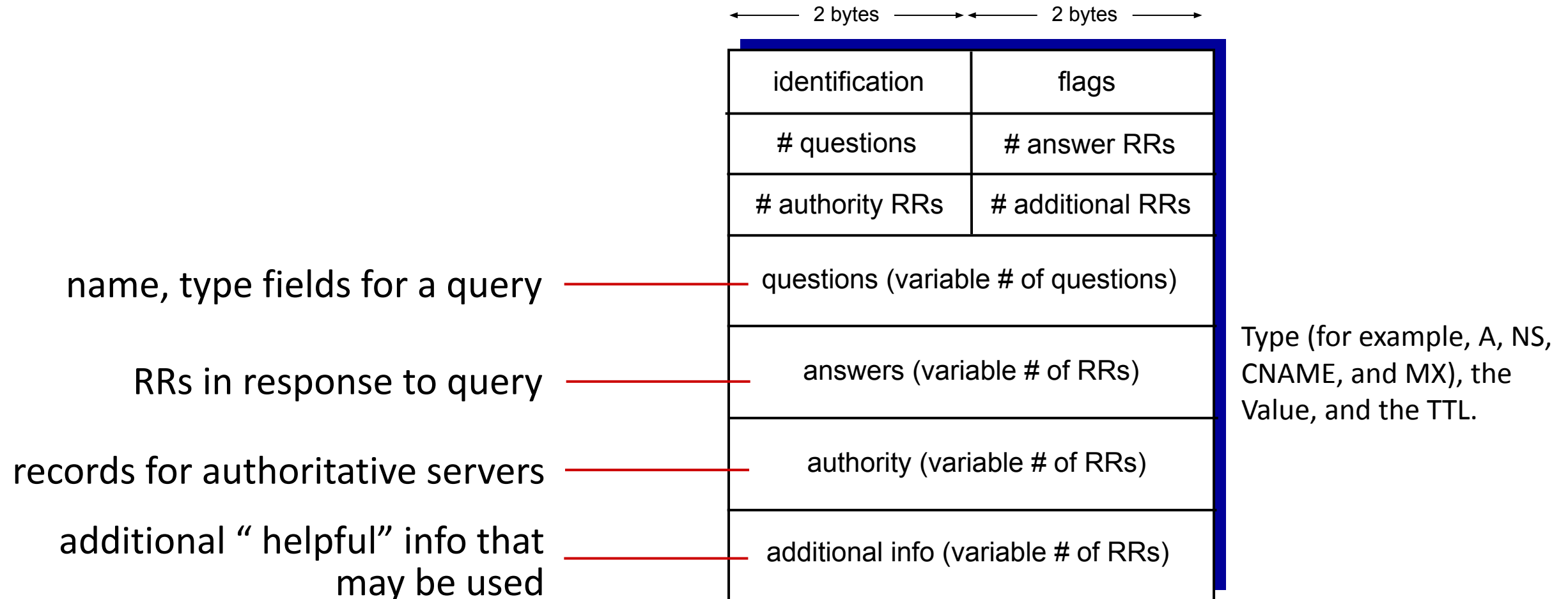
Records for authoritative servers

Additional “helpful” info that may be used

# COMPUTER NETWORKS

## DNS Protocol Messages

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





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!

# COMPUTER NETWORKS

## 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

Example: new startup “Network Utopia”

- register name **networkutopia.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.networkutopia.com
  - type MX record for networkutopia.com



# COMPUTER NETWORKS

## DNS Request - Wireshark Packet Capture

Microsoft: \Device\NPF\_{483C83F4-DCBA-4863-B523-3C4E1B03D06F} [Wireshark 1.8.5 (SVN Rev 47350 from /trunk-1.8)]

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

Filter: ip.addr == 10.36.41.43 Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
13	13:51:23.477657000	173.194.43.37	10.36.41.43	TCP	54	https > 62364 [FIN, ACK] Seq=103 Ack=2 win=63784 Len=0
14	13:51:23.477694000	10.36.41.43	173.194.43.37	TCP	54	62364 > https [ACK] Seq=3 Ack=104 win=16478 Len=0
15	13:51:23.491240000	173.194.43.37	10.36.41.43	TCP	54	https > 62364 [ACK] Seq=104 Ack=3 win=63784 Len=0
16	13:51:27.041610000	10.36.41.43	10.40.4.44	DNS	72	standard query 0x9f7d A www.ietf.org
17	13:51:27.160178000	10.40.4.44	10.36.41.43	DNS	473	standard query response 0x9f7d A 64.170.98.30
18	13:51:27.166692000	10.36.41.43	10.40.4.44	DNS	88	standard query 0x6028 A tunnel.cfw.trustedsource.org
19	13:51:27.167744000	10.40.4.44	10.36.41.43	DNS	104	standard query response 0x6028 A 8.21.161.7
20	13:51:27.180583000	10.36.41.43	8.21.161.7	TCP	62	62382 > https [SYN] Seq=0 win=8192 Len=0 MSS=1460 SACK_PERM=1
21	13:51:27.258985000	8.21.161.7	10.36.41.43	TCP	62	https > 62382 [SYN, ACK] Seq=0 Ack=1 win=5840 Len=0 MSS=1460 SACK
22	13:51:27.259111000	10.36.41.43	8.21.161.7	TCP	54	62382 > https [ACK] Seq=1 Ack=1 win=17520 Len=0
23	13:51:27.259472000	10.36.41.43	8.21.161.7	TLSv1	149	Client Hello
24	13:51:27.336962000	8.21.161.7	10.36.41.43	TCP	54	https > 62382 [ACK] Seq=1 Ack=96 win=5840 Len=0
25	13:51:27.337735000	8.21.161.7	10.36.41.43	TLSv1	1446	Server Hello, Certificate, Certificate Request, Server Hello Done
26	13:51:27.340425000	10.36.41.43	8.21.161.7	TLSv1	1005	Certificate, Client Key Exchange, Certificate verify, Change Ciph
27	13:51:27.422036000	8.21.161.7	10.36.41.43	TLSv1	113	Change cipher spec, Encrypted Handshake Message
28	13:51:27.425726000	10.36.41.43	8.21.161.7	TLSv1	395	Application Data
29	13:51:27.502692000	8.21.161.7	10.36.41.43	TLSv1	192	Application Data, Application Data

Domain Name System (query)  
[Response In: 17]  
Transaction ID: 0x9f7d

Flags: 0x0100 Standard query  
Questions: 1  
Answer RRs: 0  
Authority RRs: 0  
Additional RRs: 0

Queries  
www.ietf.org: type A, class IN

0000 00 1e 17 4c 01 3f cc a1 78 0a de 0b 08 00 45 00 ...La?... x..k..E.  
0010 00 3a 47 7d 00 00 80 11 b1 93 0a 24 29 2b 0a 28 ...G}.....\$)+.C

# COMPUTER NETWORKS

## DNS Response - Wireshark Packet Capture

Filter: `ip.addr == 10.36.41.43` Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
13	13:51:23.477657000	173.194.43.37	10.36.41.43	TCP	54	https > 62364 [FIN, ACK] Seq=103 Ack=2 win=63784 Len=0
14	13:51:23.477694000	10.36.41.43	173.194.43.37	TCP	54	62364 > https [ACK] Seq=3 Ack=104 win=16478 Len=0
15	13:51:23.491240000	173.194.43.37	10.36.41.43	TCP	54	https > 62364 [ACK] Seq=104 Ack=3 win=63784 Len=0
16	13:51:27.041610000	10.36.41.43	10.40.4.44	DNS	72	Standard query 0x9f7d A www.ietf.org
17	13:51:27.160178000	10.40.4.44	10.36.41.43	DNS	473	Standard query response 0x9f7d A 64.170.98.30
18	13:51:27.166692000	10.36.41.43	10.40.4.44	DNS	88	Standard query 0x6028 A tunnel.cfw.trustedsource.org
19	13:51:27.167744000	10.40.4.44	10.36.41.43	DNS	104	Standard query response 0x6028 A 8.21.161.7
20	13:51:27.180583000	10.36.41.43	8.21.161.7	TCP	62	62382 > https [SYN] Seq=0 win=8192 Len=0 MSS=1460 SACK_PER
21	13:51:27.258985000	8.21.161.7	10.36.41.43	TCP	62	https > 62382 [SYN, ACK] Seq=0 Ack=1 win=5840 Len=0 MSS=14

Flags: 0x8180 Standard query response, No error

Questions: 1

Answer RRs: 1

Authority RRs: 6

Additional RRs: 11

Queries

www.ietf.org: type A, class IN

Name: www.ietf.org

Type: A (Host address)

Answers

www.ietf.org: type A, class IN, addr 64.170.98.30

Authoritative nameservers

ietf.org: type NS, class IN, ns ns1.yyz1.afiliast.net

ietf.org: type NS, class IN, ns ns0.ietf.org

ietf.org: type NS, class IN, ns ns1.sea1.afiliast.net

ietf.org: type NS, class IN, ns ns1.ams1.afiliast.net

ietf.org: type NS, class IN, ns ns1.mia1.afiliast.net

```
000  cc af 78 0a de 6b 00 1e f7 4c 61 3f 08 00 45 00  ..x..k.. .La?..E.
010  01 cb 63 b4 40 00 7e 11 55 cb 0a 28 04 2c 0a 24  ..c.@.~. U..(..$.
020  29 2b 00 35 c3 d5 01 b7 1a 58 9f 7d 81 80 00 01  )+.5.... .X.}....
030  00 01 00 06 00 0b 03 77 77 77 04 69 65 74 66 03  .....w ww.ietf.
040  6f 72 67 00 00 01 00 01 c0 0c 00 01 00 01 00 00  org.....
050  07 08 00 04 40 aa 62 1e c0 10 00 02 00 01 00 00  ....@.b. ....
060  07 08 00 04 40 aa 62 1e c0 10 00 02 00 01 00 00  ....@.b. ....
070  69 6c 69 61 73 2d 6e 73 74 04 69 6e 66 6f 00 c0  ilias-ns t'info
```



- DNS (Domain Name System) – Explained – <https://youtu.be/JkEYOt08-rU>
- How a DNS Server (Domain Name System) works – <https://youtu.be/rdVPfIECed8>
- Wireshark Lab: DNS v7.0 – [http://www-net.cs.umass.edu/wireshark-labs/Wireshark\\_DNS\\_v7.0.pdf](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