CPSC 317 COMPUTER NETWORKING

Module 3: Application Layer Protocols - Day 3 – DNS

READING

- Reading: 2.4

LEARNING GOALS

Domain Name Service (DNS)

- Define the purpose and major design goals of DNS
- For each design goal, describe the strategy/technology/design approach used to address the problem
- Define the purpose of the various servers in the hierarchy of name servers
- Trace how DNS resolves a name to an IP address
- Describe the different resource records returned by a DNS server
- Interpret the information returned by dig or displayed by wireshark.
- Apply the information returned by dig to determine the next server to contact or to determine the final answer

NAMING AND NETWORK STRUCTURE

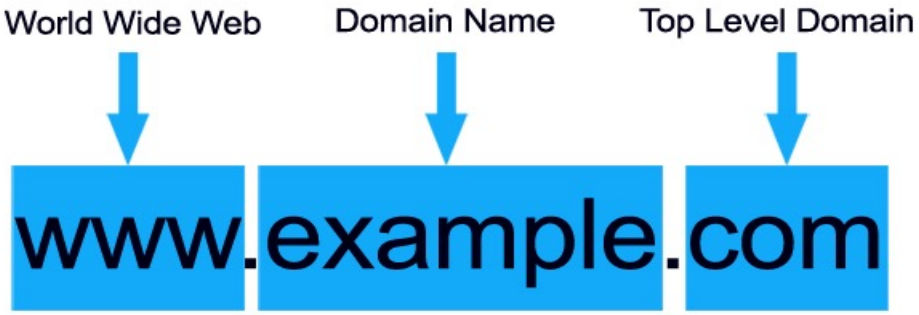
- How do we know which destination IP address to use?
- Problems:
  - Humans have a hard time remembering numbers
  - Addresses can change
- Solution: Map user-friendly names to IP addresses
  - Names are easier to remember
  - Names can mask address changes
- Domain Name System (DNS)

www.students.cs.ubc.ca:80

198.162.1.10:80

DOMAIN NAME

- A **domain name** is an identification string that defines a realm of administrative autonomy, authority or control within the Internet.
- Domain names are formed by the rules and procedures of the Domain Name System (DNS).



WHAT IS DNS?

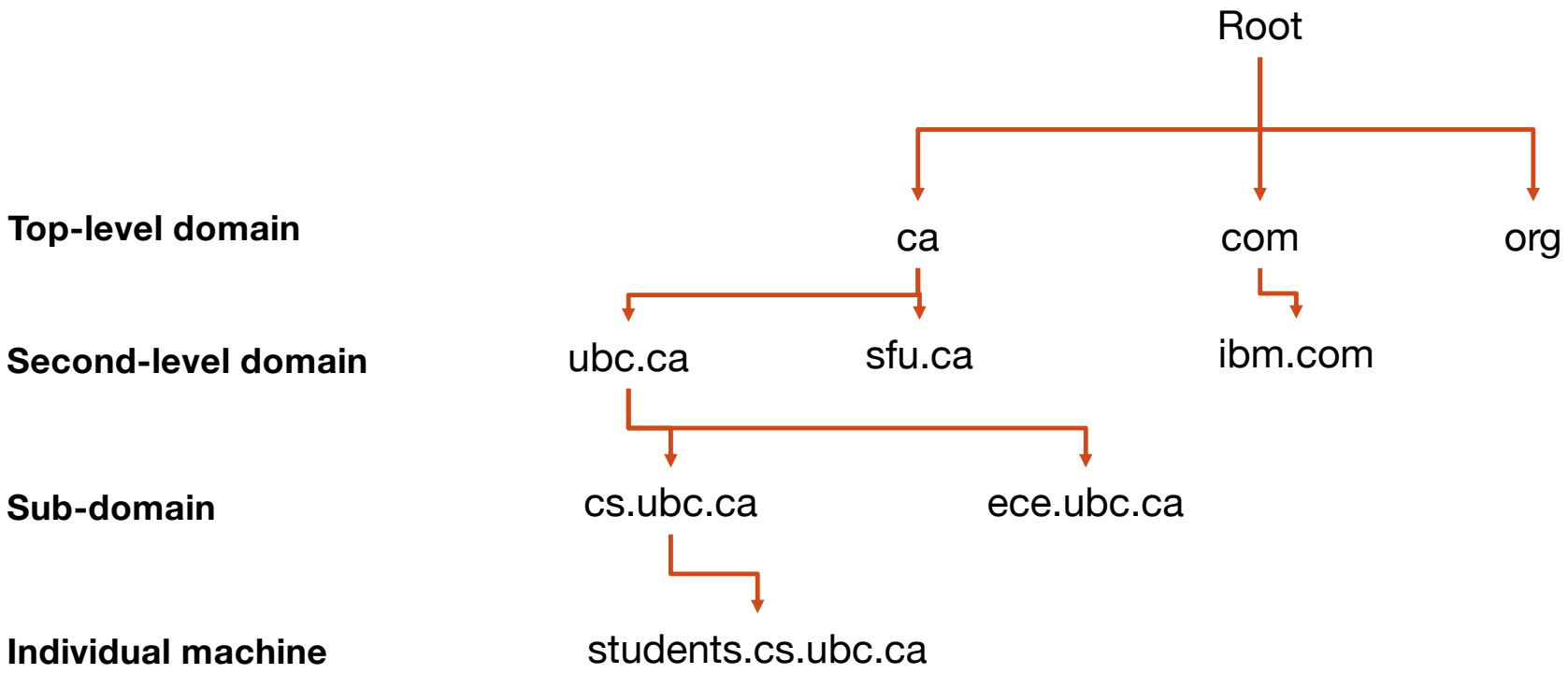
- A **distributed database** implemented by a hierarchy of many name servers
- An **application-layer protocol** used by hosts to communicate with name servers to resolve names (translate names to addresses)

DNS GOALS (DESIGN CHALLENGES)

- Scale (names, users, updates, etc.)
  - 364.6 million domain name registrations (September 2021)
- Ease of management (uniqueness of names, etc.)
  - Who decides if cs.ubc.ca can name a host “students”?
- Availability and consistency and security
  - Is there only one answer for the question: “What is the IP address of www.cs.ubc.ca?”? How do we ensure this?
- Performance
  - OpenDNS, Cloudfare each serve ~100 billion requests per day (>1 million per second)

Solution:
- Hierarchical design
- Caching
- Replication

A HIERARCHY OF NAMES



A HIERARCHY OF NAME SERVERS

Name servers

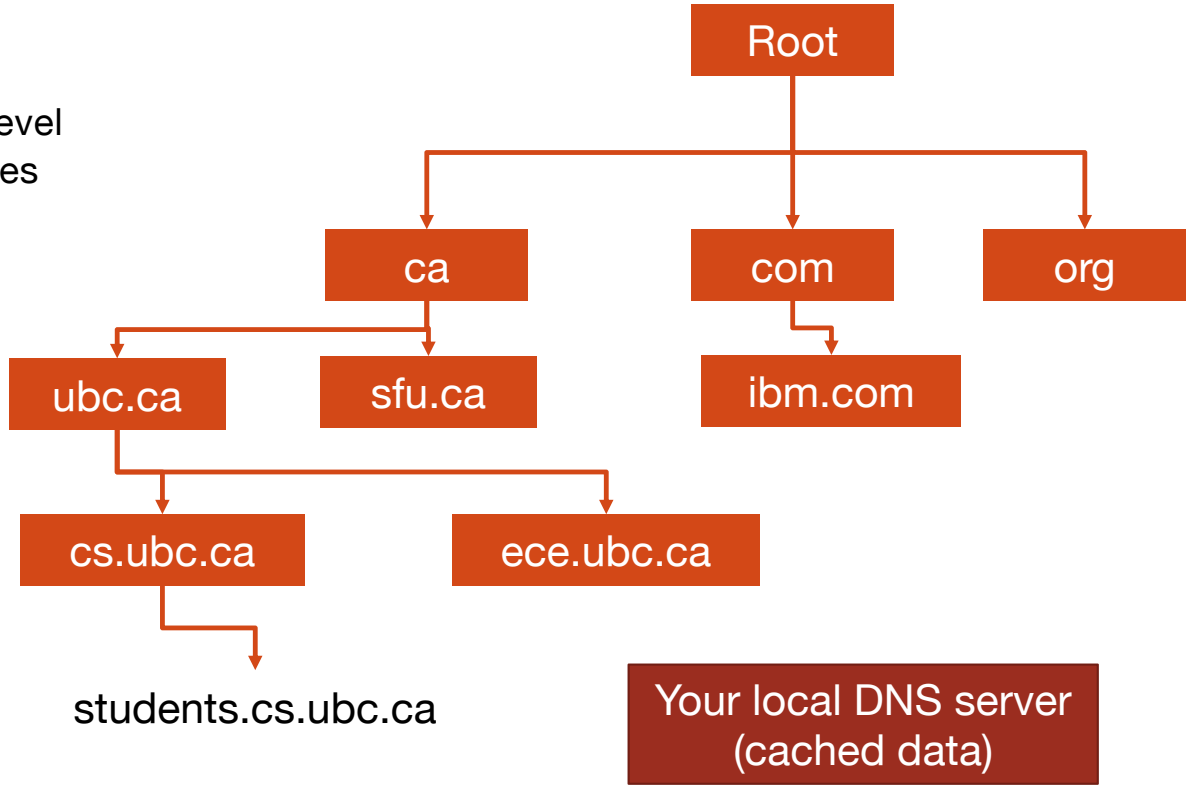
stores mapping of next level domain and IP addresses

Top-level domain server

Authoritative name server

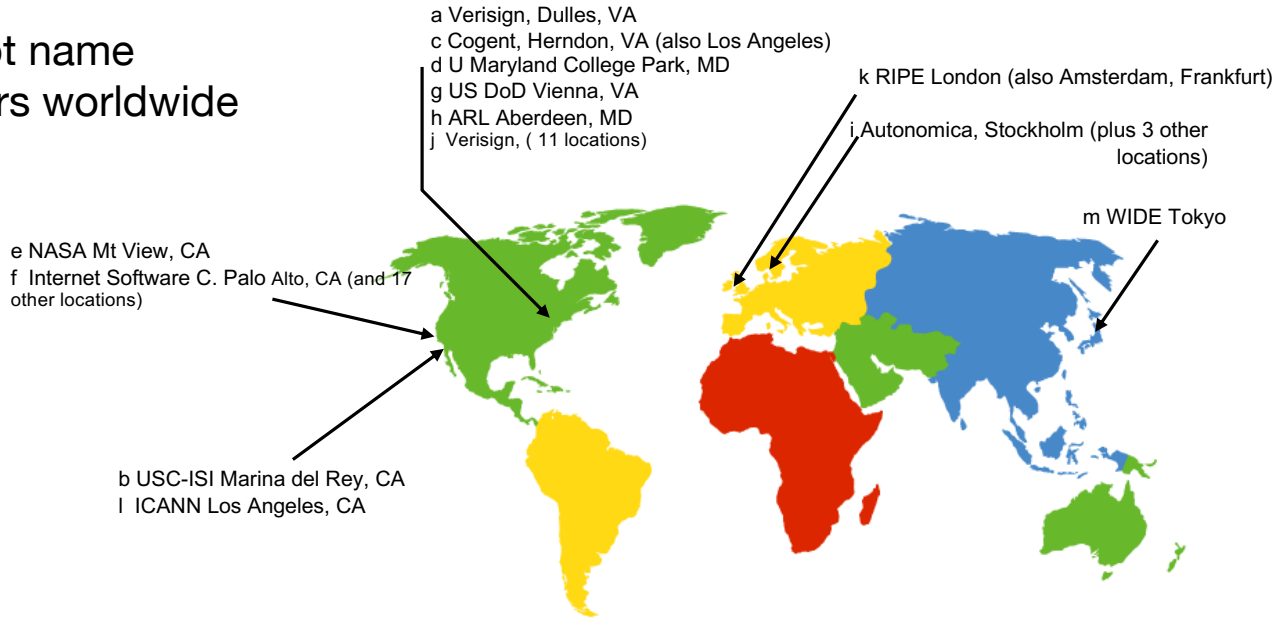
Authoritative name server

Individual machine



DNS: ROOT NAME SERVERS

13 root name servers worldwide



TLD AND AUTHORITATIVE SERVERS

- Top-level domain (TLD) servers: responsible for domains ending in
  - .com, .org, .net, .edu, etc. (e.g., Network Solutions maintains servers for .com TLD, Educause for .edu TLD), and
  - all top-level country domains .uk, .fr, .ca, .jp, etc.
- Authoritative DNS servers: provide authoritative hostname to IP mappings for an organization's subdomains and servers (e.g., Web and mail)
  - Can be maintained by organization or service provider

WHO KNOWS WHAT?

- Every server knows the address of the root name servers
- Root servers know the address of all TLD servers
- Every node knows the addresses of its children
- An ***authoritative*** DNS server stores name-to-address mappings (resource records) for all names in the domain that it has authority for
- Therefore, each server:
  - Stores only a subset of the total DNS database (scalable!)

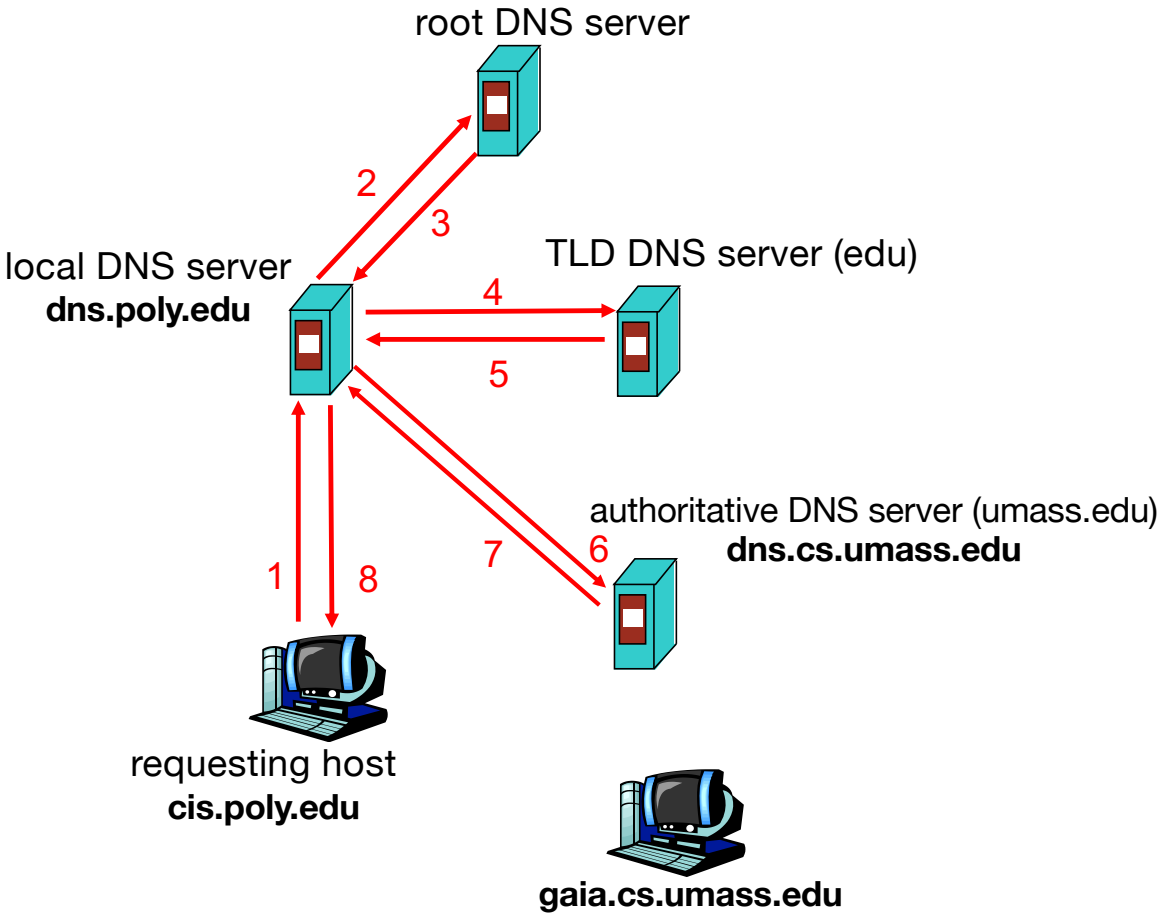
LOCAL NAME SERVER

- Does not belong to the hierarchy
- Each ISP (residential ISP, company, university) has one
  - Also called “default name server”
- When a host makes a DNS query, the query is sent to the local DNS server
  - Acts as a proxy, forwards query into hierarchy of DNS servers
  - Local DNS server resolves queries **iteratively**
- You will build a local name server in PA2

DNS LOOKUPS

Host at cis.poly.edu wants IP address for the domain gaia.cs.umass.edu.

- Host contacts local DNS server.
- Local DNS server recursively contacts the root server and subsequent DNS servers iteratively.
- Each contacted server replies with name of next server to contact.
- Authoritative server returns the IP address of the requested domain.
- Local DNS server returns IP address to the host.



DNS CACHING

- Every DNS request starting at root servers → root servers will be a bottleneck
- Solution: caching
  - Local DNS servers cache responses to queries
  - Responses include a “time to live” (TTL) field
  - Server deletes cached entry after TTL expires
- Caching is effective because
  - The top-level name servers very rarely change
  - Popular sites are visited often → the local DNS server often has the information cached

INSERTING RECORDS INTO DNS

- Example: I just created the startup "Network Utopia"
- Register name networkutopia.com at a registrar (e.g., Network Solutions)
  - I provide registrar with names and IP addresses of my authoritative name servers (primary and secondary)
  - Registrar inserts two RRs into the com TLD server for each authoritative name server:

```
(networkutopia.com, dns1.networkutopia.com, NS)
(dns1.networkutopia.com, 212.212.212.1, A)
(networkutopia.com, dns2.networkutopia.com, NS)
(dns2.networkutopia.com, 212.212.212.2, A)
```

- In my authoritative servers, add a Type A record for www.networkutopia.com
  - `(networkutopia.com, 212.212.212.100, A)`
- How do people get the IP address of my Web site?

DNS GOALS

- Scaling (names, users, updates, etc.) – Yes
  - Can add TLDs just by changing root database or new domains just by changing a TLD server
- Ease of management (uniqueness of names, etc.) – Yes
  - Each autonomous administration manages own names and servers, and can further delegate
  - Easily ensures uniqueness of names
  - And consistency of databases
- Availability and consistency and security – Yes
  - Domains replicate independently
- Fast lookups – Yes
  - Caching is a key, locality is very high

DETAILS

DNS RECORDS

DNS servers store resource records (RRs)

- RR is (name, type, value, TTL)

type	name	value	example
A (Address)	hostname	IPv4 address	(www.cs.ubc.ca, A, 142.103.6.5, TTL
AAAA (Address x 4)	hostname	IPv6 address	(www.google.com, AAAA, 2607:f8b0:400a:80b::2004, TTL)
NS (Name Server)	domain	name of DNS server for domain	(cs.ubc.ca, NS, fs1.ugrad.cs.ubc.ca, TTL)
CNAME (Canonical NAME)	alias	canonical name	(foo.com, CNAME, relay1.bar.foo.com, TTL)

DNS PROTOCOL

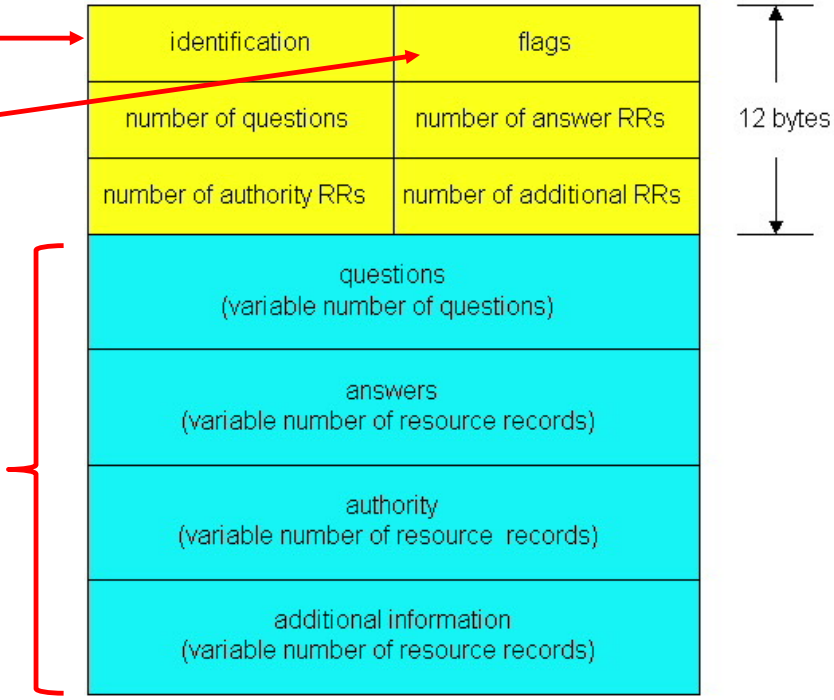
- Client-Server interaction on UDP Port 53
  - Message size limited by max UDP segment size (512 bytes)
  - For larger DNS messages, could use EDNS
  - Spec supports TCP too, but not always implemented
  - Reliability via repeating requests if the client times out
- Query and Reply messages
  - Both with the same message format
- Resolution is almost always “iterative”

DNS PROTOCOL MESSAGES

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

identification: 16 bit # set in query, reply uses same #

flags: query or reply, reply is authoritative

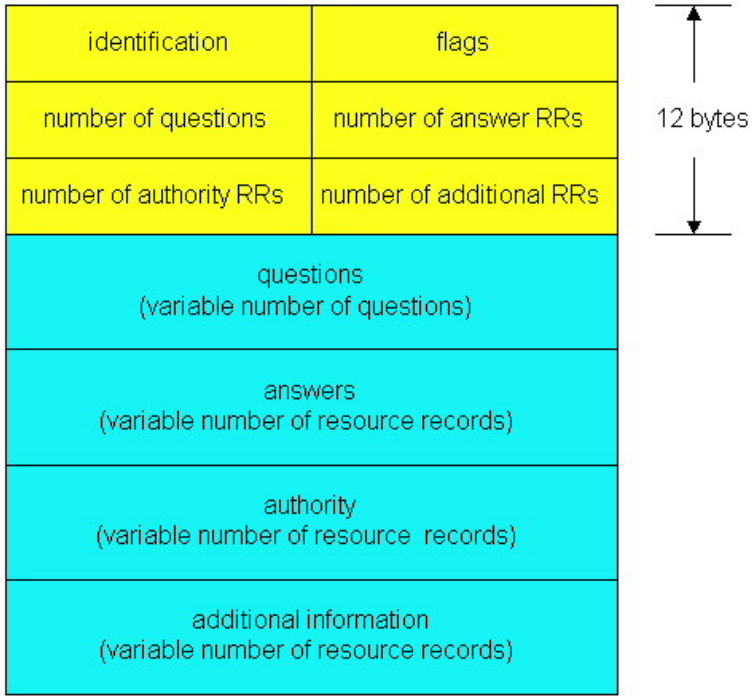


Resource records (RRs):
<Name, Type, Class, TTL, RDLENGTH, RDATA>

- Name: a fully qualified domain name
- Type: a valid RR type (e.g., A, AAAA, ...)
- Class: mostly IN for internet application

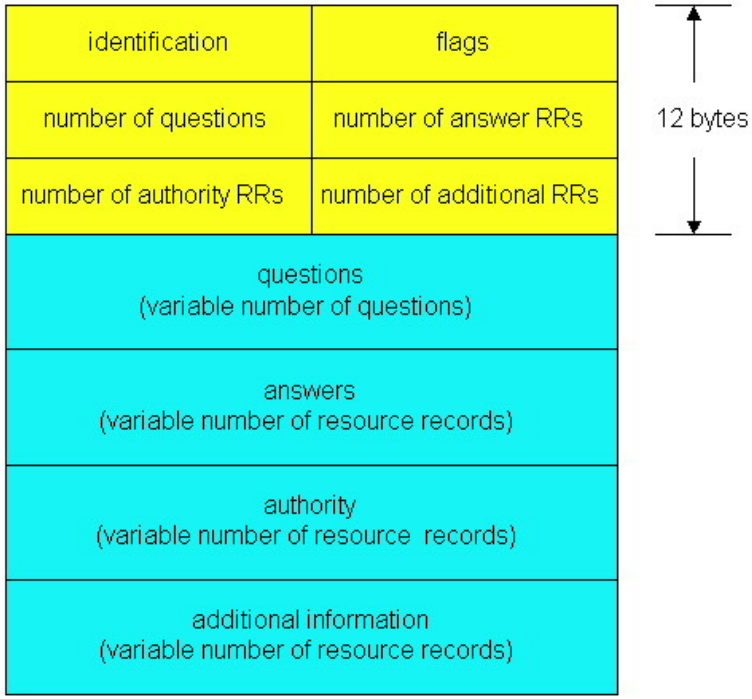
RESOURCE RECORDS (RR) IN A QUERY

- **Questions** are always <Name, Type, Class> tuples
- The question is the only section included in a query



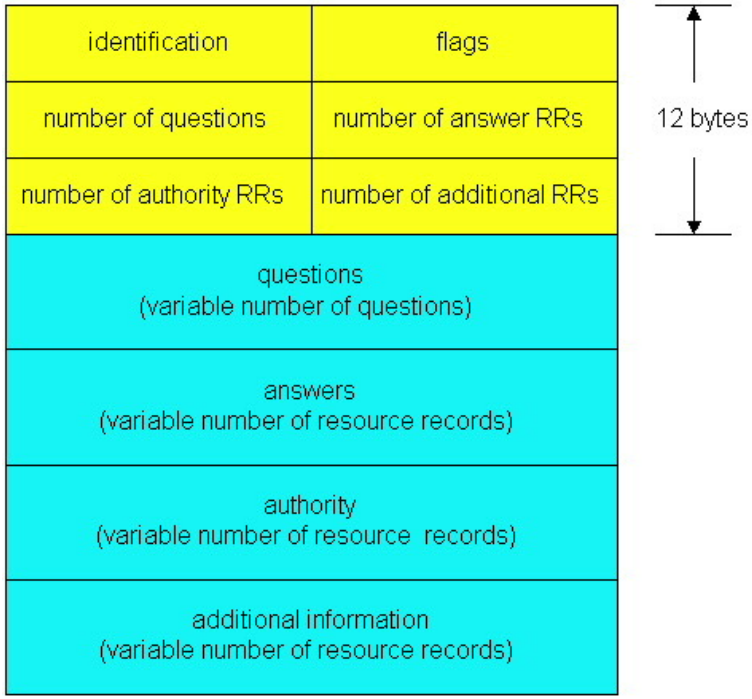
RESOURCE RECORDS (RR) IN A REPLY

- **Answers** are RRs that match the Name, Type, Class from the question
- If a DNS server has CNAME pointers for the requested query with same class, returns CNAME records in the answer
- There may be multiple answers, since there may be multiple RRs with the same labels



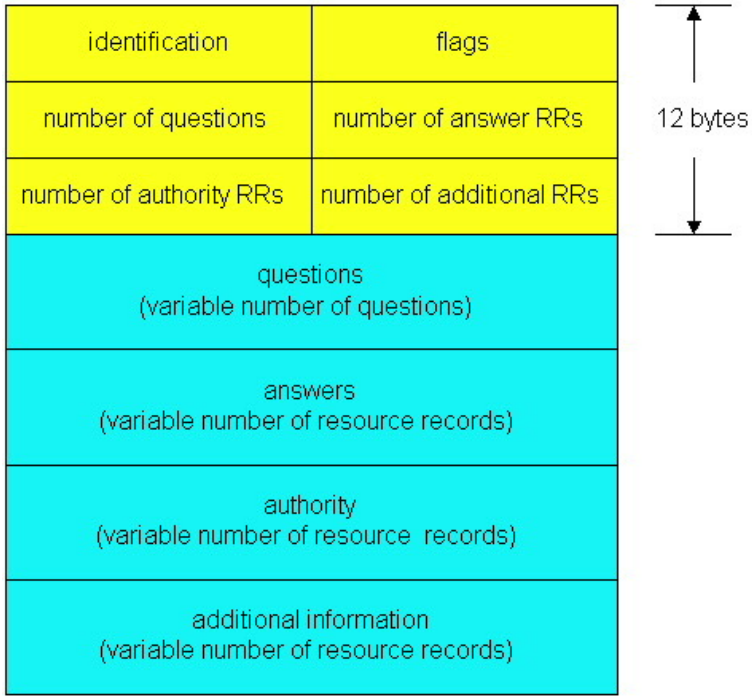
RESOURCE RECORDS (RR) IN A REPLY

- **Authority** RRs are type NS records pointing to name servers closer to the target name in the naming hierarchy
- Used to redirect the client to a "better" server
- This field is optional



RESOURCE RECORDS (RR) IN A REPLY

- **Additional** RRs are records that the name server believes may be useful to the client
- Most commonly used to supply A or AAAA (address) records for the name servers listed in the Authority section



QUERY AT A ROOT NAME SERVER

```
$ dig +noedns @a.root-servers.net www.cs.ubc.ca

; <<>> DiG 9.10.6 <<>> +noedns @a.root-servers.net www.cs.ubc.ca
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 13410
;; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 4, ADDITIONAL: 8
;; WARNING: recursion requested but not available

;; QUESTION SECTION:
;www.cs.ubc.ca.                 IN      A

;; AUTHORITY SECTION:
ca.                     172800  IN      NS      any.ca-servers.ca.
ca.                     172800  IN      NS      x.ca-servers.ca.
ca.                     172800  IN      NS      c.ca-servers.ca.
ca.                     172800  IN      NS      j.ca-servers.ca.

;; ADDITIONAL SECTION:
any.ca-servers.ca.      172800  IN      A       199.4.144.2
any.ca-servers.ca.      172800  IN      AAAA    2001:500:a7::2
x.ca-servers.ca.        172800  IN      A       199.253.250.68
x.ca-servers.ca.        172800  IN      AAAA    2620:10a:80ba::68
c.ca-servers.ca.        172800  IN      A       185.159.196.2
c.ca-servers.ca.        172800  IN      AAAA    2620:10a:8053::2
j.ca-servers.ca.        172800  IN      A       198.182.167.1
j.ca-servers.ca.        172800  IN      AAAA    2001:500:83::1

;; Query time: 7 msec
;; SERVER: 198.41.0.4#53(198.41.0.4)
;; WHEN: Fri Feb 11 09:50:52 PST 2022
;; MSG SIZE  rcvd: 284
```

NEXT STEP

```
$ dig +noedns @any.ca-servers.ca www.cs.ubc.ca

; <<>> DiG 9.10.6 <<>> +noedns @any.ca-servers.ca www.cs.ubc.ca
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 10677
;; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 3, ADDITIONAL: 3
;; WARNING: recursion requested but not available

;; QUESTION SECTION:
;www.cs.ubc.ca.                 IN      A

;; AUTHORITY SECTION:
ubc.ca.                 86400   IN      NS      hub.ubc.ca.
ubc.ca.                 86400   IN      NS      nightbird.eis.utoronto.ca.
ubc.ca.                 86400   IN      NS      dns3.ubc.ca.

;; ADDITIONAL SECTION:
nightbird.eis.utoronto.ca. 86400 IN     A       128.100.72.90
dns3.ubc.ca.            86400   IN      A       142.103.1.1
hub.ubc.ca.             86400   IN      A       137.82.1.1

;; Query time: 37 msec
;; SERVER: 199.4.144.2#53(199.4.144.2)
;; WHEN: Fri Feb 11 09:51:43 PST 2022
;; MSG SIZE  rcvd: 161
```

AND THE STEP AFTER THAT?

```
$ dig +noedns @nightbird.eis.utoronto.ca www.cs.ubc.ca

; <<>> DiG 9.10.6 <<>> +noedns @nightbird.eis.utoronto.ca www.cs.ubc.ca
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 60085
;; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 3, ADDITIONAL: 3
;; WARNING: recursion requested but not available

;; QUESTION SECTION:
;www.cs.ubc.ca.                 IN      A

;; AUTHORITY SECTION:
cs.ubc.ca.              86400   IN      NS      fs1.ugrad.cs.ubc.ca.
cs.ubc.ca.              86400   IN      NS      temp120.cs.ubc.ca.
cs.ubc.ca.              86400   IN      NS      ns1.cs.ubc.ca.

;; ADDITIONAL SECTION:
ns1.cs.ubc.ca.          86400   IN      A       142.103.6.6
fs1.ugrad.cs.ubc.ca.    86400   IN      A       198.162.35.1
temp120.cs.ubc.ca.      86400   IN      A       137.82.61.120

;; Query time: 61 msec
;; SERVER: 128.100.72.90#53(128.100.72.90)
;; WHEN: Fri Feb 11 09:53:07 PST 2022
;; MSG SIZE  rcvd: 143
```

AND THE FINAL STEP?

```
$ dig +noedns @temp120.cs.ubc.ca www.cs.ubc.ca

; <<>> DiG 9.10.6 <<>> +noedns @temp120.cs.ubc.ca www.cs.ubc.ca
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 22393
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0
;; WARNING: recursion requested but not available

;; QUESTION SECTION:
;www.cs.ubc.ca.                 IN      A

;; ANSWER SECTION:
www.cs.ubc.ca.          3600    IN      A       142.103.6.5

;; Query time: 8 msec
;; SERVER: 137.82.61.120#53(137.82.61.120)
;; WHEN: Fri Feb 11 09:53:48 PST 2022
;; MSG SIZE  rcvd: 47
```

IN-CLASS ACTIVITY

- ICA33