CIS 457 - Data Communications

Nathan Bowman

Images taken from Kurose and Ross book

DNS Structure

We know the purpose(s) of DNS, but how does it work?

Study overall structure first, leave details of messages for another lecture

First want to know where information is stored and how client requests it

DNS is client-server architecture

Client makes DNS query ("what is IP address of `google.com`?")

From client perspective, DNS is black box that returns IP address

In reality, DNS is hierarchy of servers

Single server could not handle entire DNS

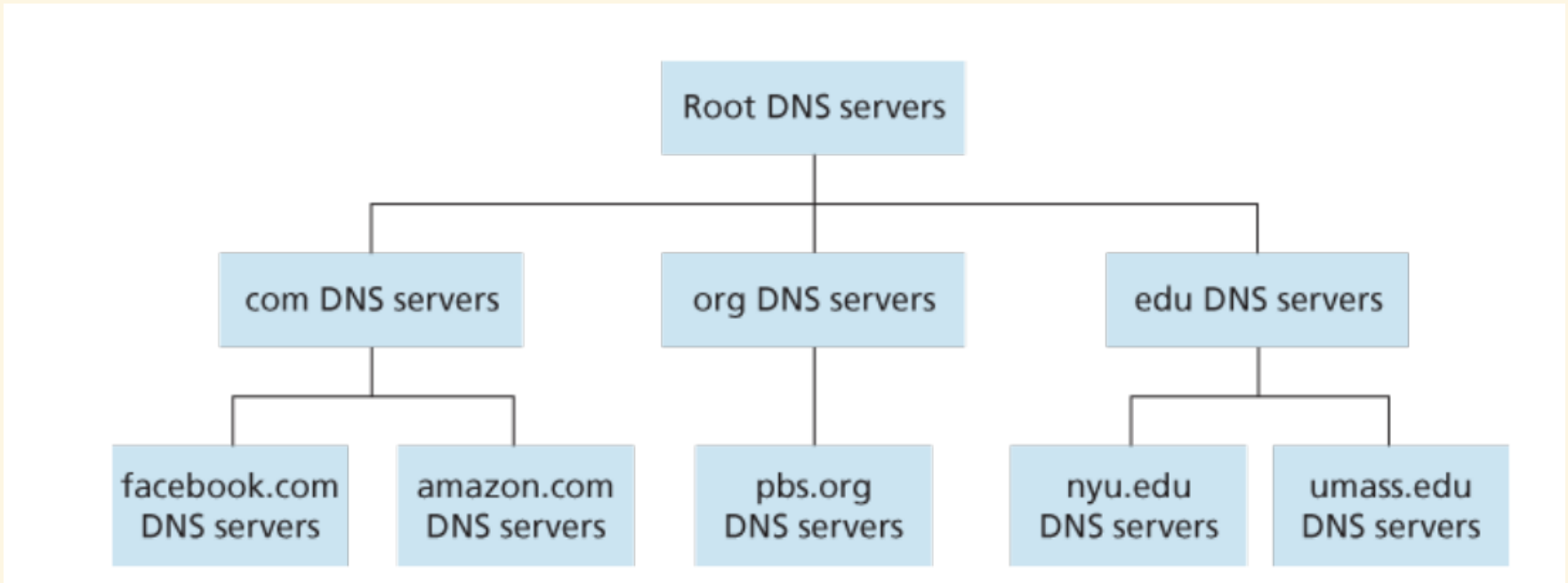
- too many queries
- server would be physically distant from some clients
- single point of failure
- too many records to store and maintain

Hierarchical structure split into three main levels, though there can be more

Root, top-level domain (TLD), and authoritative

These correspond to various parts of hostname

Hostname, such as `www.google.com`, is hierarchical from right to left


Root DNS servers
com DNS servers
org DNS servers
edu DNS servers
facebook.com DNS servers
amazon.com DNS servers
pbs.org DNS servers
nyu.edu DNS servers
umass.edu DNS servers

For IP address of `www.google.com`, first ask root server "Who knows about names ending in `com`?"

Root server replies with name and location of TLD server for `com`

Note that client needs to know address of root server or it cannot begin the process

In reality, we will see that client does not directly contact root server, but rather has another server do so on client's behalf

Armed with IP address of `com` TLD server, client asks "Who is responsible for names on `google.com`?"

TLD server returns name and IP address of Google's authoratative DNS server

TLD server and authoritative server are DNS servers, just like root server but with smaller domains and more specific information

Separate TLD servers exist for each top-level domain: `.com`, `.net`, `.edu`, `.uk`, `.jp`, ...

Next, client requests address of `www.google.com` from `google.com` authoratative server

Authoratative means server will know IP address corresponding to hostname

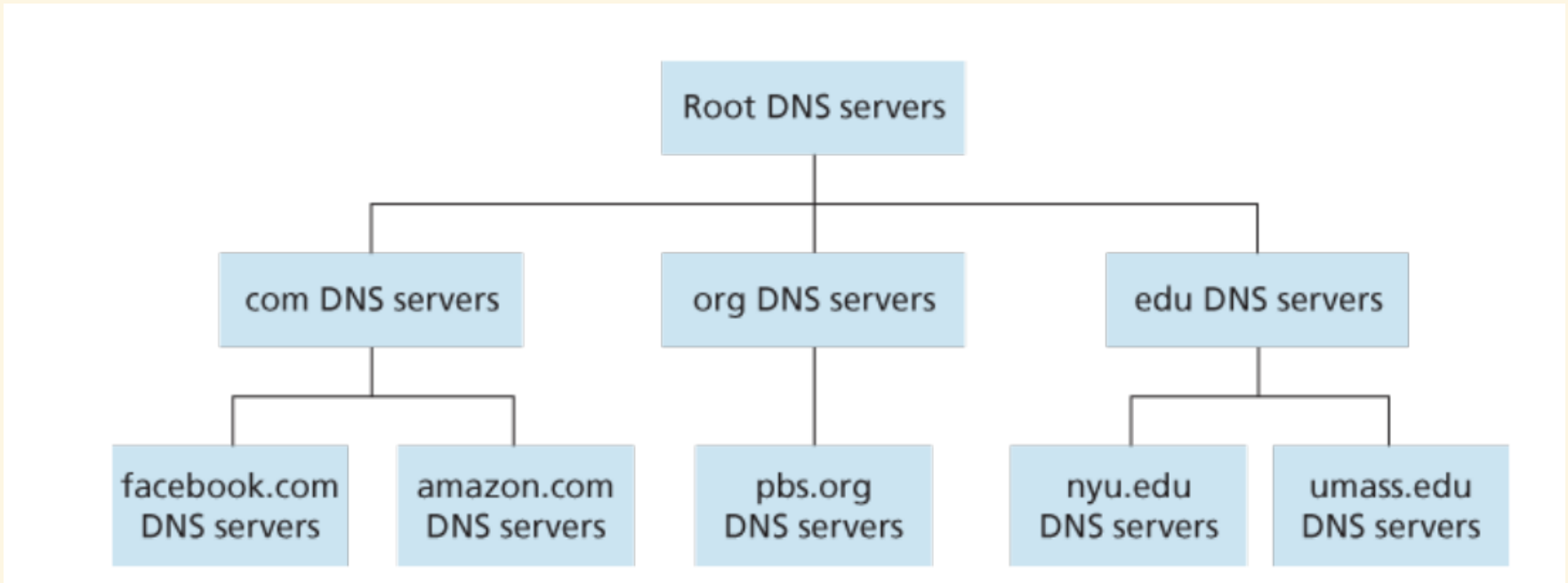
Client finally has hostname it needs

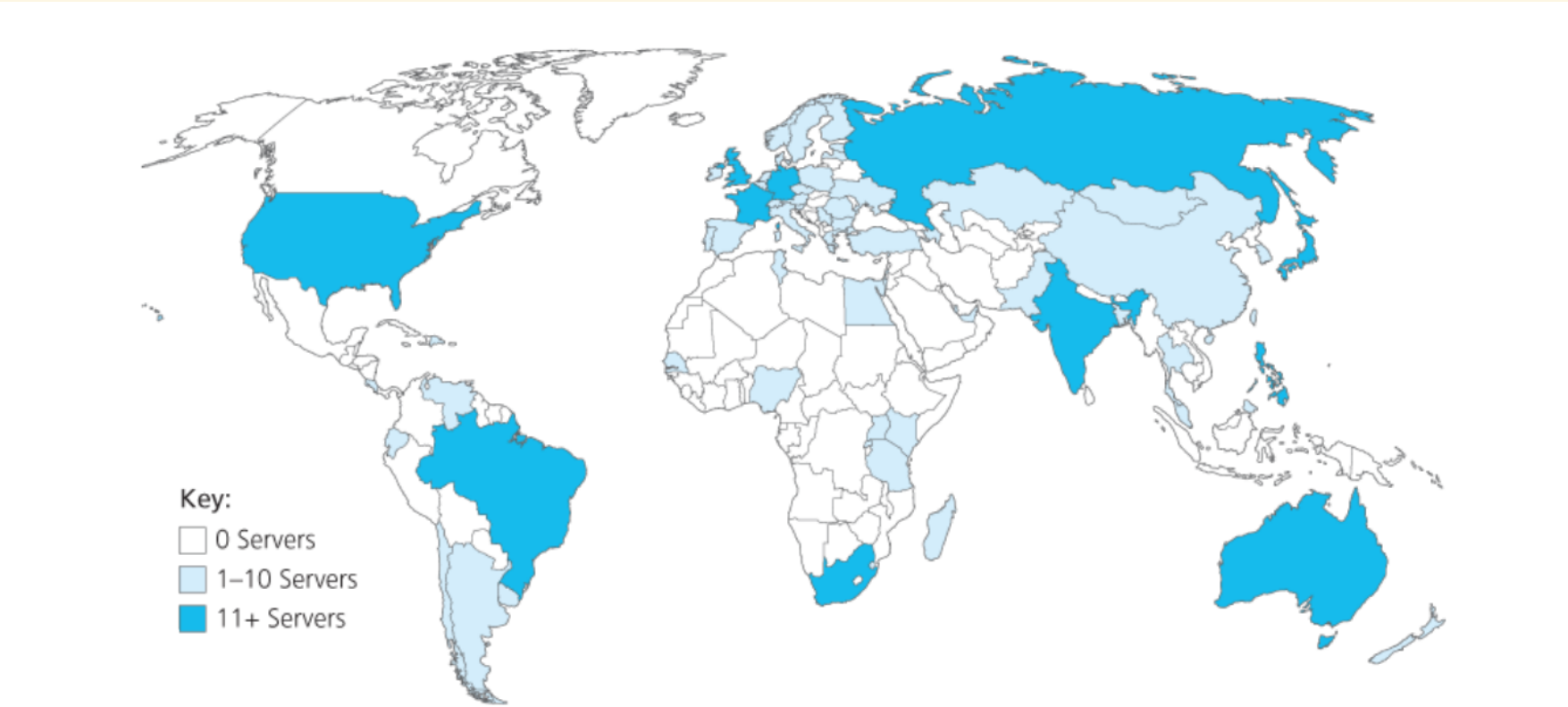
Sometimes, hierarchy can go further

For example, different departments at university might have their own DNS servers

Accessing `gaia.cs.umass.edu` might be slightly different from previous process:

- same steps as above to get DNS server for `umass.edu`
- DNS server for `umass.edu` is not authoritative for `gaia.cs.umass.edu`
- instead, sends back another name and IP address for DNS server managing `cs.umass.edu`
- that fourth-level DNS server is queried, finally resulting in IP address of `gaia.cs.umass.edu`


Root DNS servers
com DNS servers
org DNS servers
edu DNS servers
facebook.com DNS servers
amazon.com DNS servers
pbs.org DNS servers
nyu.edu DNS servers
umass.edu DNS servers



There are more than 400 root DNS servers worldwide

Note than any publicly accessible host on the internet requires a publicly accessible DNS record

Running public web server, email server, etc. requires also providing authoratative nameservers

Possible to pay another company to manage nameservers, though large institutions often manage nameservers themselves

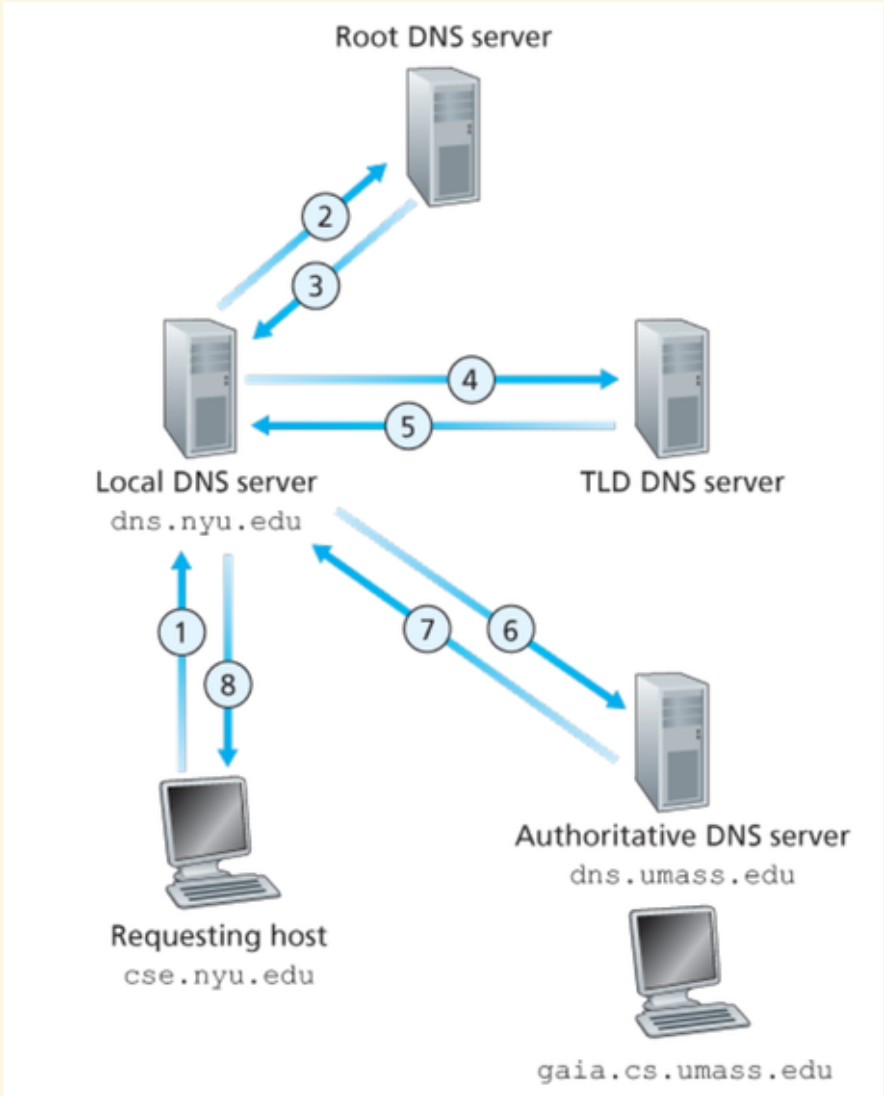
Like any other protocol, clients and servers need to know how to contact one another

DNS operates over UDP on port 53

Another type of nameserver exists outside hierarchy previously described but is also important for efficient operation of DNS

Rather than query DNS hierarchy themselves as described, clients typically contact **local nameserver** to obtain IP addresses on their behalf

Local nameserver generally managed by ISP


Root DNS server
2
3
4
5
Local DNS server
dns.nyu.edu
TLD DNS server
1
8
7
6
Authoritative DNS server
dns.umass.edu
Requesting host
cse.nyu.edu
gaia.cs.umass.edu

Note the number of messages sent just to acquire an IP address

Eight messages in "standard" query, could be even more if additional levels of nameservers used with an organization

Two main drawbacks:

- long delay for client
- excessive traffic on internet

We saw similar problem with web pages, and same solution applies

Local DNS server will also act as DNS cache

When contacted with DNS query, local server will follow DNS hierarchy until it determines IP address

In addition to sending IP address back to client, local server will store this name/address mapping for later use

Future queries for IP address of same hostname can be responded to immediately with stored information -- no need for further DNS messages

Information in cache may eventually become out of date, so entries dropped periodically (often every two days)

Not just hostname -> IP mappings cached -- local server also saves results of intermediate DNS queries, such as location of `com` TLD server

Because of caching, only very small percent of DNS queries actually involve message to root server

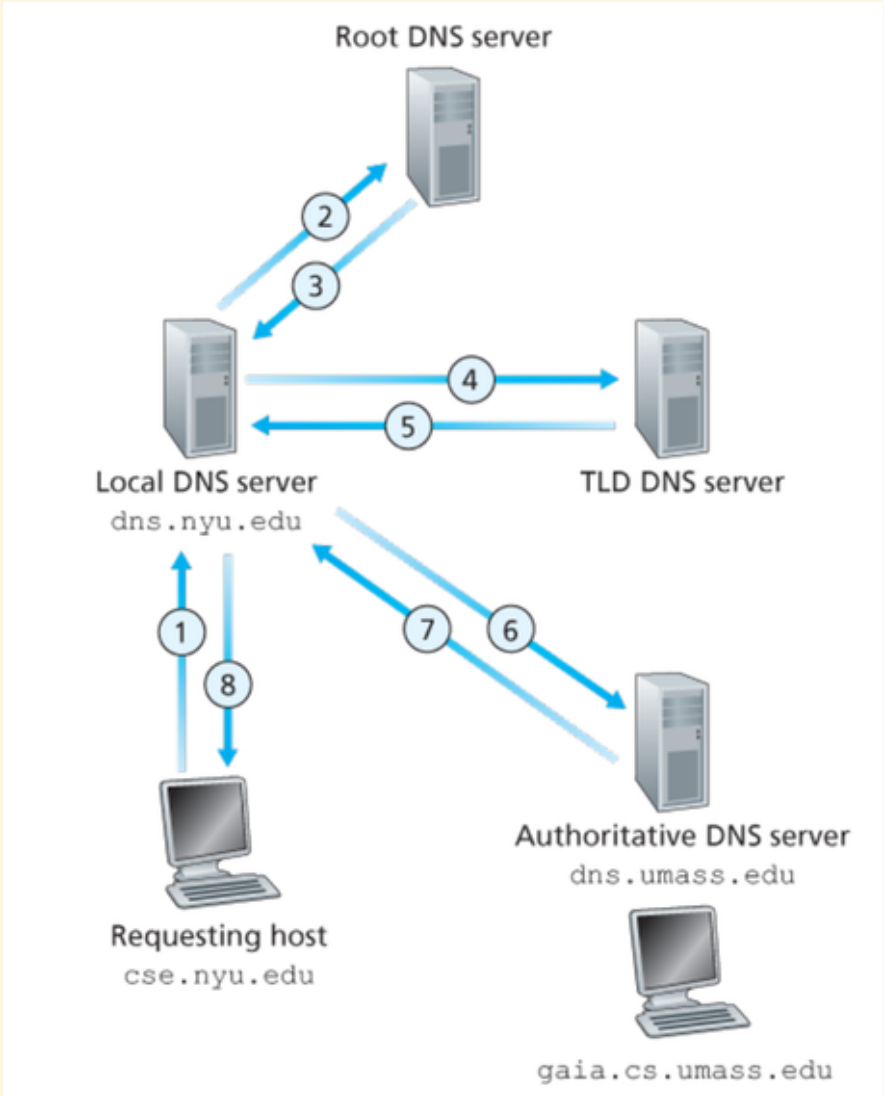

Root DNS server
2
3
4
5
Local DNS server
dns.nyu.edu
TLD DNS server
1
8
7
6
Authoritative DNS server
dns.umass.edu
Requesting host
cse.nyu.edu
gaia.cs.umass.edu

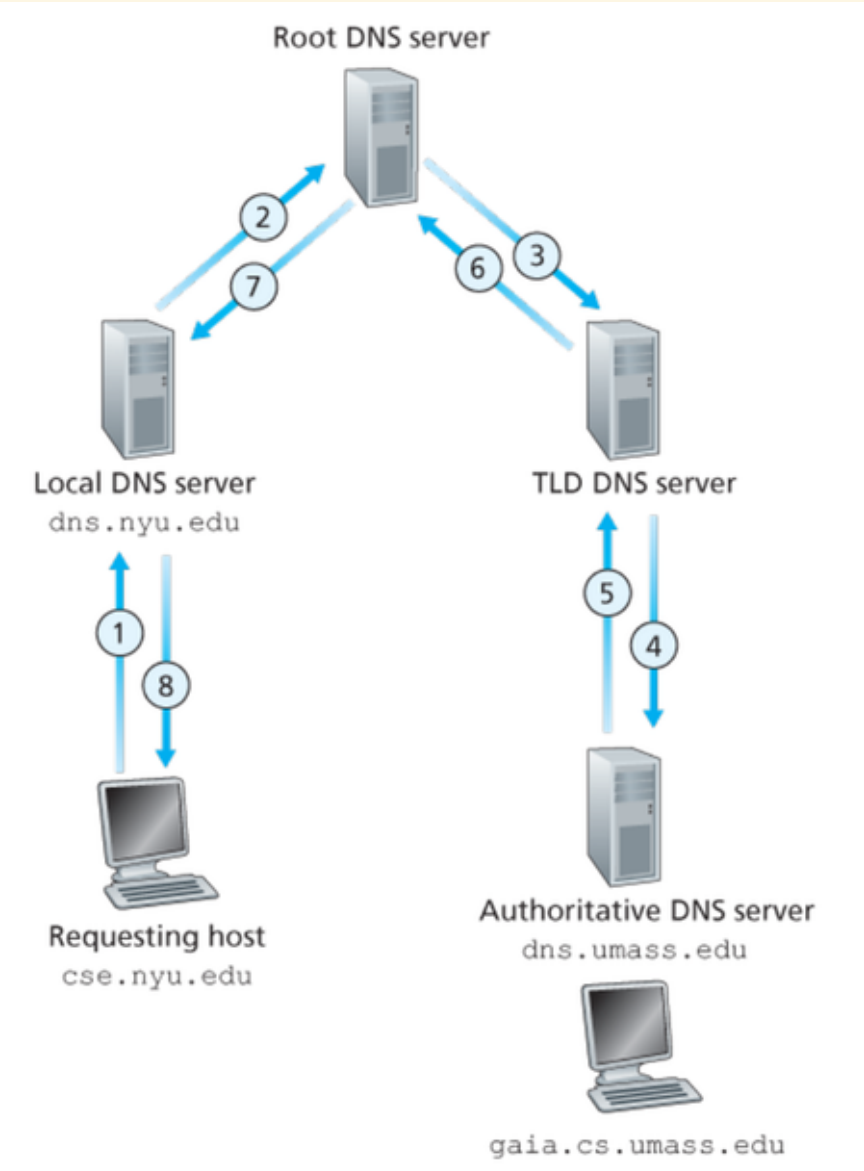
Figure also shows difference between two types of queries: **iterative** and **recursive**

Iterative query returns information needed to proceed to next step of process, but not necessarily final result

Requests sent from local server are iterative

Recursive query requests that server find eventual goal rather than returning only what it knows

Requests sent from client to local server are recursive


Root DNS server
2
7
6
3
Local DNS server
dns.nyu.edu
TLD DNS server
1
8
5
4
Requesting host
cse.nyu.edu
Authoritative DNS server
dns.umass.edu
gaia.cs.umass.edu

Queries generally follow pattern seen previously -- requests to local server are recursive, and requests from local server to other servers are iterative