Domain Name System (DNS) (Contd.)

CPSC 433/533, Spring 2021

Anurag Khandelwal

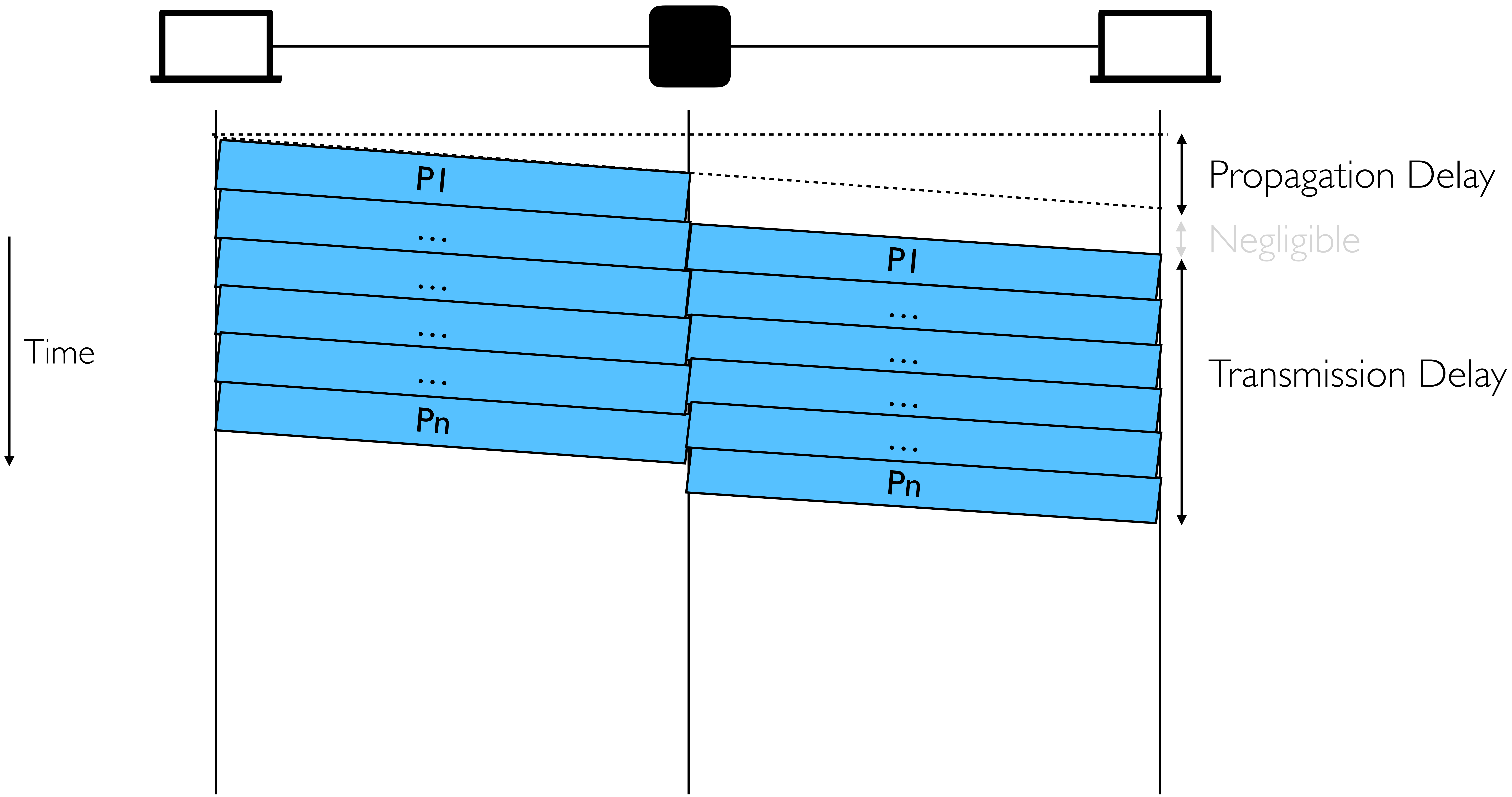
Evaluating Midterm Performance

- **The exam was tough & long**
  - This was intentional — we didn't expect you to get a perfect score!
  - But the class did really well!! :')
- **Post-mortem: re-evaluate your understanding**
  - Try to understand why you lost points; unsure about concepts? **Clarify them!**
  - Can create a class to recap concepts you didn't follow: look out for a piazza poll
- **Observation: 45+ scorers attend class regularly, and turn up to office hours!**
  - Correlation is not causation, but it does make you wonder…
  - Attend class (encourage your friends to do the same!)
  - Participate (ask questions, post on Piazza, show up for OH, …)

Addressing Midterm Q2

- (2) Suppose Yale and Harvard are connected via one switch and two links, each of length 150 km and bandwidth 10^7 bits per second, respectively. Alice wants to send a 10^4 byte file to Bob on this dedicated link. Which of the following is closest to the end-to-end delay (ignore queuing and processing delays)? (Speed of light: 300,000 km/s)
- What we did wrong:
  - Did not specify the MTU (implicit)
  - Options were calibrated for a 0.0001s propagation delay, but inadvertently introduced a bug (0.001s propagation delay) in transitioning question to Canvas
  - We messed up! How do we fix it?
    - We will award everyone 1 point for the question

Understand the concept tested in Q2



Moving forward

- **If you feel you didn't do great on the midterm, don't fret!**
  - First, calibrate: the absolute scores don't matter. This was a hard exam, adjust your expectations based on class average.
  - Second, the midterm only counts for a fraction of your grade: you can still make up points on hw3, project2 and finals!
  - Third, you can submit regrade requests if you feel you should have received points
- **Please fill out the mid-semester survey!**
  - Survey link: https://www.surveymonkey.com/r/XGYCGXD
  - Will help us understand where you are struggling, and what we can do to fix it!
  - If you don't speak up, nothing will change…
    - … but if you do, I promise I will do everything I can to help you learn!

Feedback so far…

- **Lecture pacing & exceeding 1h 15m**
  - Will introduce breaks within lecture for questions & will stick to 1h 15m :)
  - Ask questions! My teaching pace = TCP slow start :(Questions = ECN!
- **Exam difficulty & time constraint**
  - This one is tricky: conceptual questions are important, but end up seeming hard
  - The online format makes it trickier (and takes longer to solve)
  - Will recalibrate time for finals, but will still have deep conceptual questions
- **For god's sake, stop with the 300 page lecture slide pdfs!**
  - 😅… will post pdfs both with & without animations

Administriva: Project 2

- **Project 2 will be released tonight, due in 3 weeks**
    - Will let Jonathan tell you more about it…

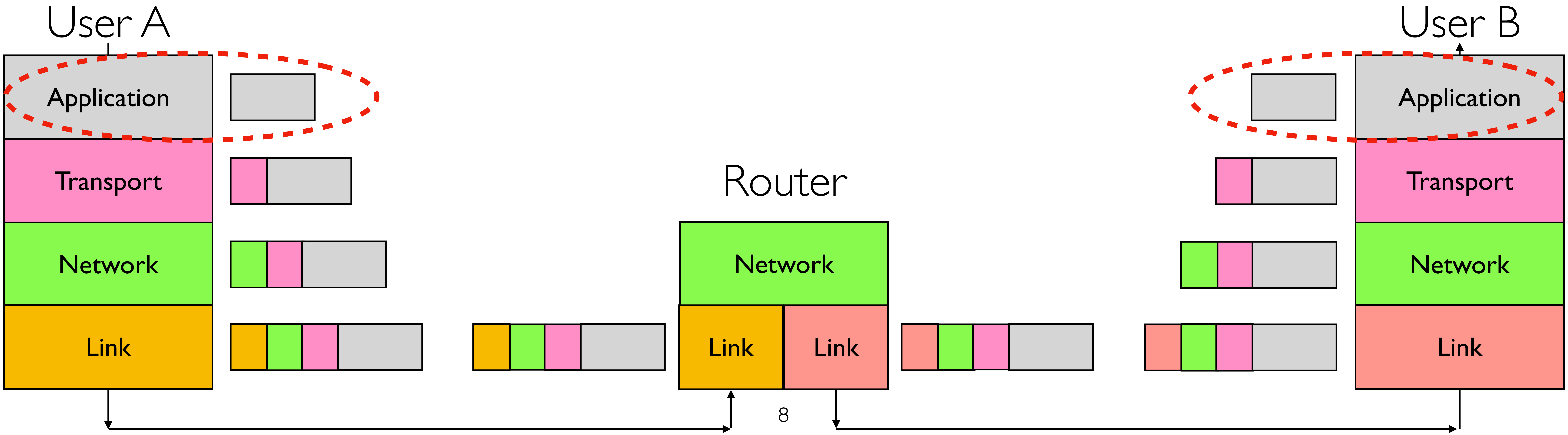
Where we are in the course…

Course so far:

- **Concepts,** *Links, delays, switches*
- **Overall Architecture,** *Layers, protocols principles*
- **Network Layer,** *Best-effort global delivery of packets*
- **Transport Layer,** *Reliable (or unreliable) delivery of data*

What's left?

- **Application Layer,** *DNS, HTTP (today)*
- **Lower Layers,** *Ethernet, Wireless*
- **Advanced Topics,** *Datacenters, SDN*



Back to DNS!

Recap: What is DNS?

Domain name service maps host names (e.g., www.google.com) to host addresses (e.g., 172.217.8.174)

- Why bother?
- Convenience
  - Easier to remember www.google.com than 172.217.8.174
- Provides a level of indirection!
  - Decoupled names from addresses
  - Many uses beyond just naming a specific host

Recap: DNS Goals

- Scalable
  - Many *names*
  - Many *updates*
  - Many *users* creating names
  - Many *users* looking up names
- Highly available
- Correct
  - No naming conflicts (uniqueness)
  - Consistency → observe the latest update
- Lookups are fast

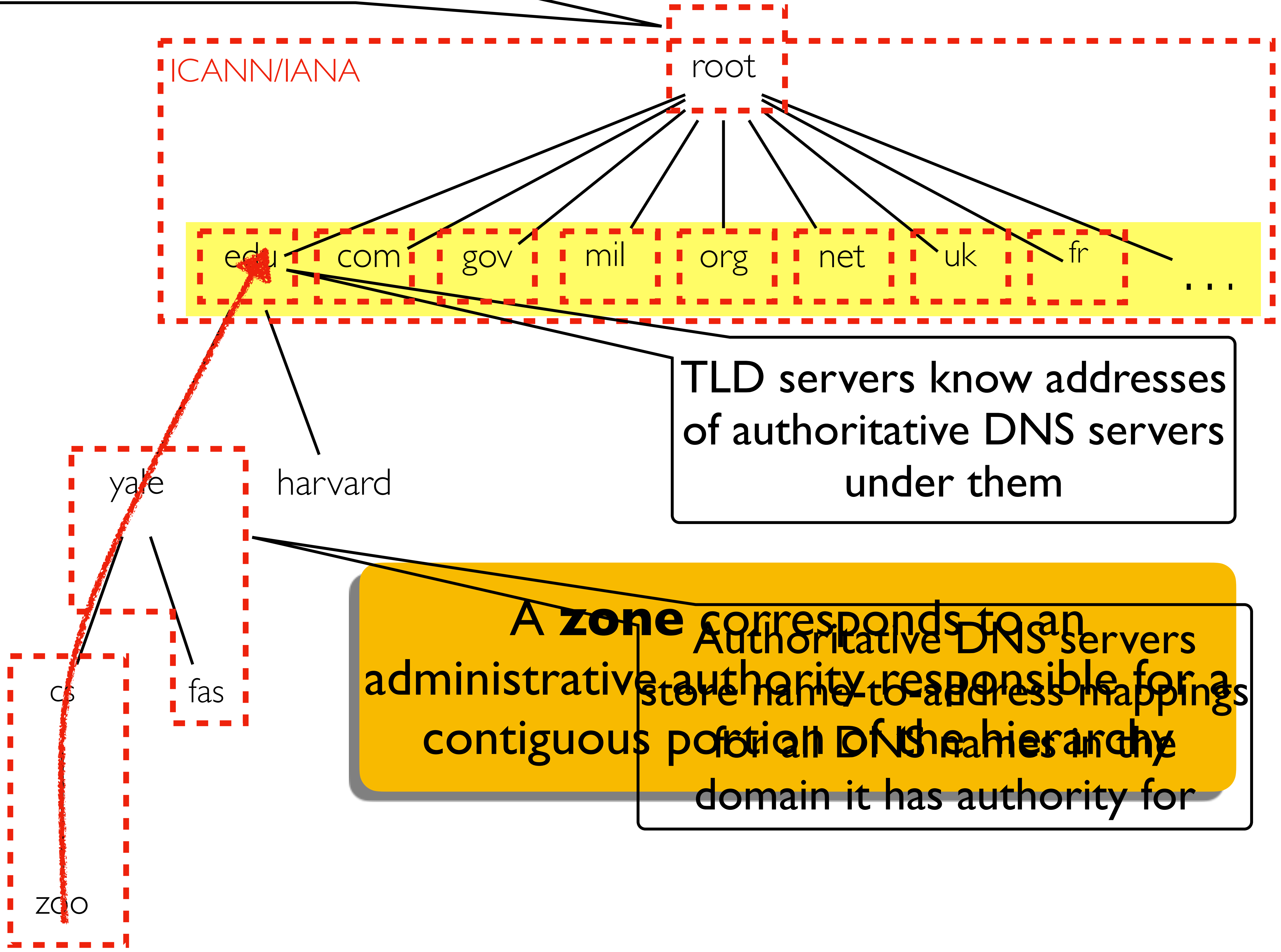
Recap: Scaling with Hierarchical Distribution

Three intertwined hierarchies

- Hierarchical naming
  - As opposed to flat namespace
- Hierarchical administration
  - As opposed to centralized administration
- Hierarchical storage
  - As opposed to centralized storage

Root servers located via anycast on universally known IP addresses

Root servers knows addresses of all TLD servers



TLD servers know addresses of authoritative DNS servers under them

A **zone** corresponds to an administrative authority responsible for a contiguous portion of the hierarchy

Authoritative DNS servers store name-to-address mappings for all DNS names in the domain it has authority for

Recap: DNS Records

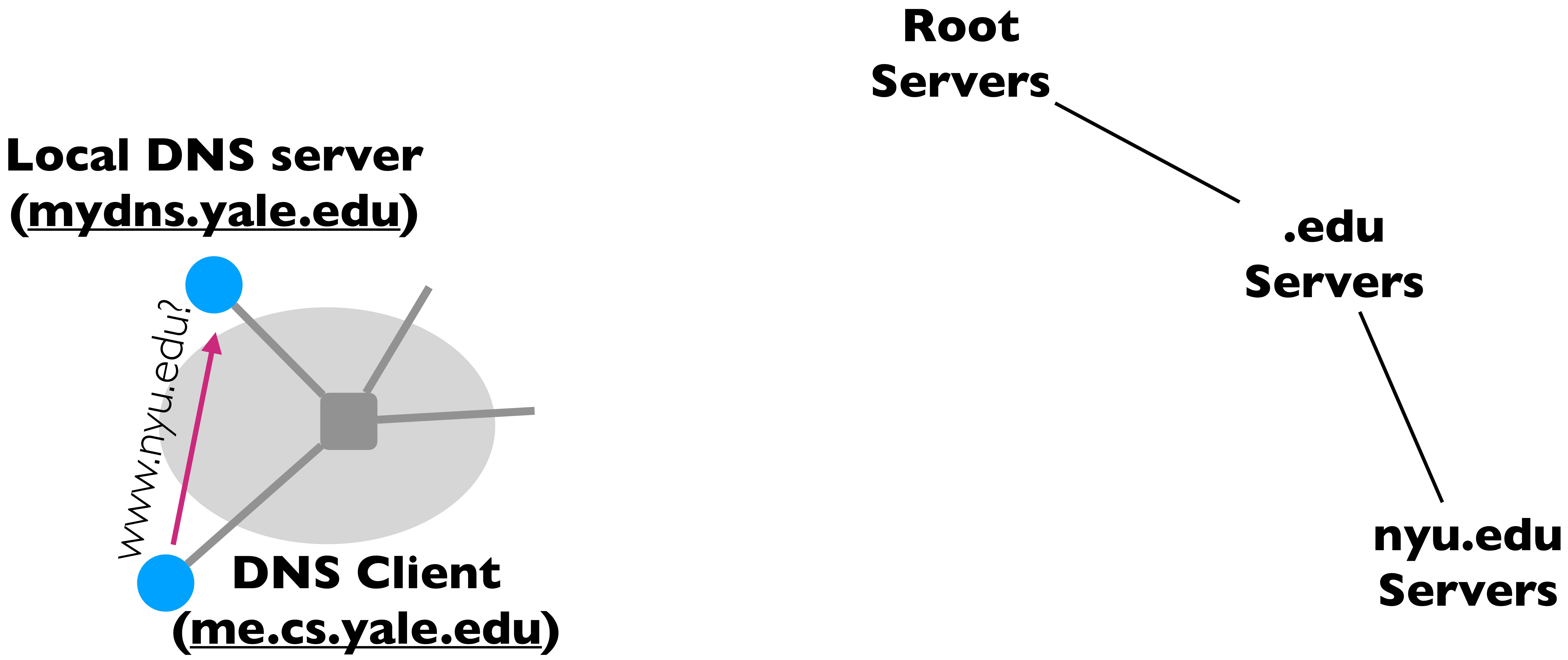
- DNS Servers store **resource records (RRs)**
  - RR is (name, value, type, TTL)

- Type = A: (→*Address*)
  - Name = hostname
  - Value = IP address
- Type = NS: (→*Name Server*)
  - Name = domain
  - Value = name of DNS server for domain
- Type = MX: (→*Mail eXchanger*)
  - Name = domain in email address
  - Value = name(s) of mail server(s)

Recap: Inserting Resource Records into DNS

- Example: you just created company “FooBar”
- You get a block of IP addresses from your ISP
  - Say 212.44.9.128/25
- Register foobar.com at registrar (e.g., GoDaddy)
  - Provide registrar with names and IP addresses of your authoritative name server(s)
  - Registrar inserts RR pairs into the .com TLD server
    - (foobar.com, dns1.foobar.com, NS)
    - (dns1.foobar.com, 212.44.9.129, A)
- Store resource records in your server dns1.foobar.com
  - e.g., type A records: (foobar.com, 212.44.9.130, A), (social.foobar.com, 212.44.9.131, A), etc.
  - e.g., type MX records for foobar.com

Recap: Using DNS (Client/Application view)



Root Servers

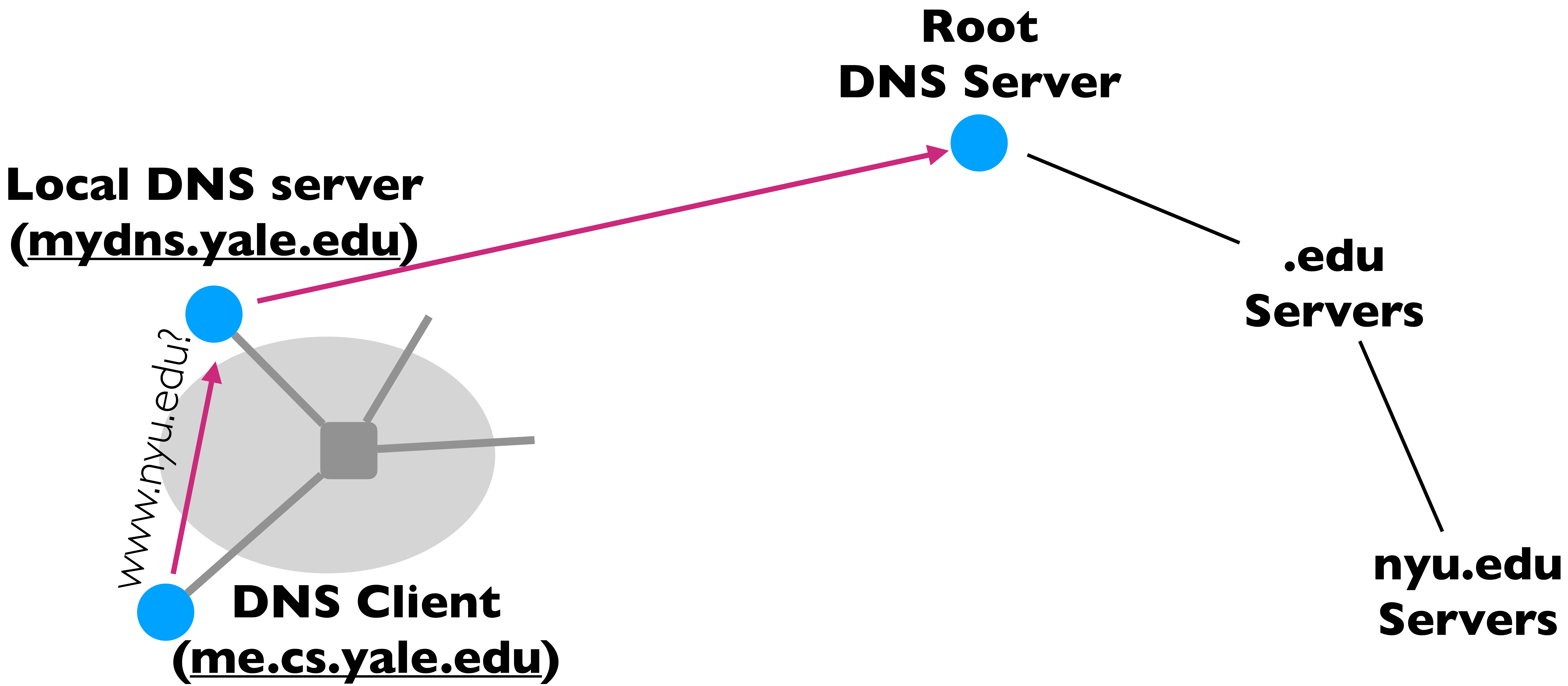
.edu Servers

nyu.edu Servers

Local DNS server (mydns.yale.edu)

www.nyu.edu?

DNS Client (me.cs.yale.edu)



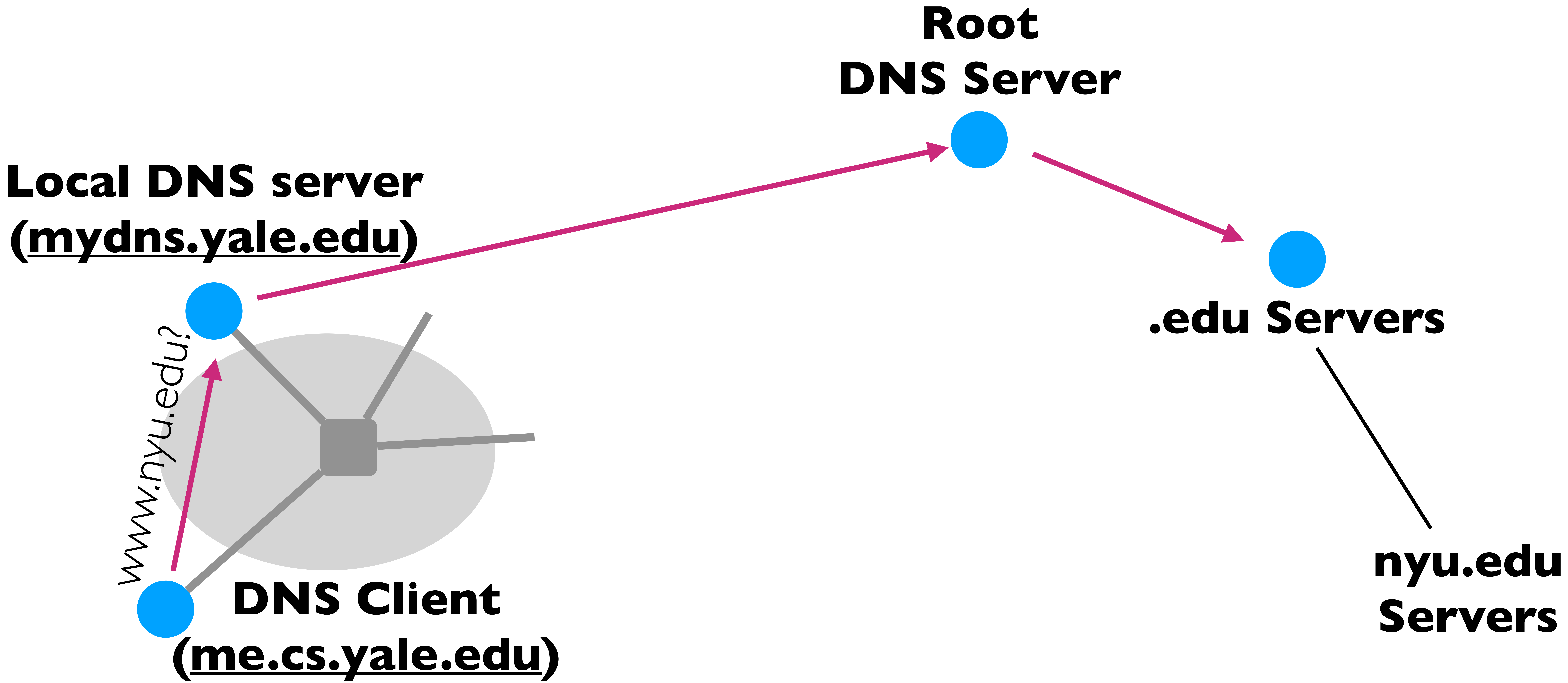
Root
DNS Server

Local DNS server
(mydns.yale.edu)

.edu
Servers

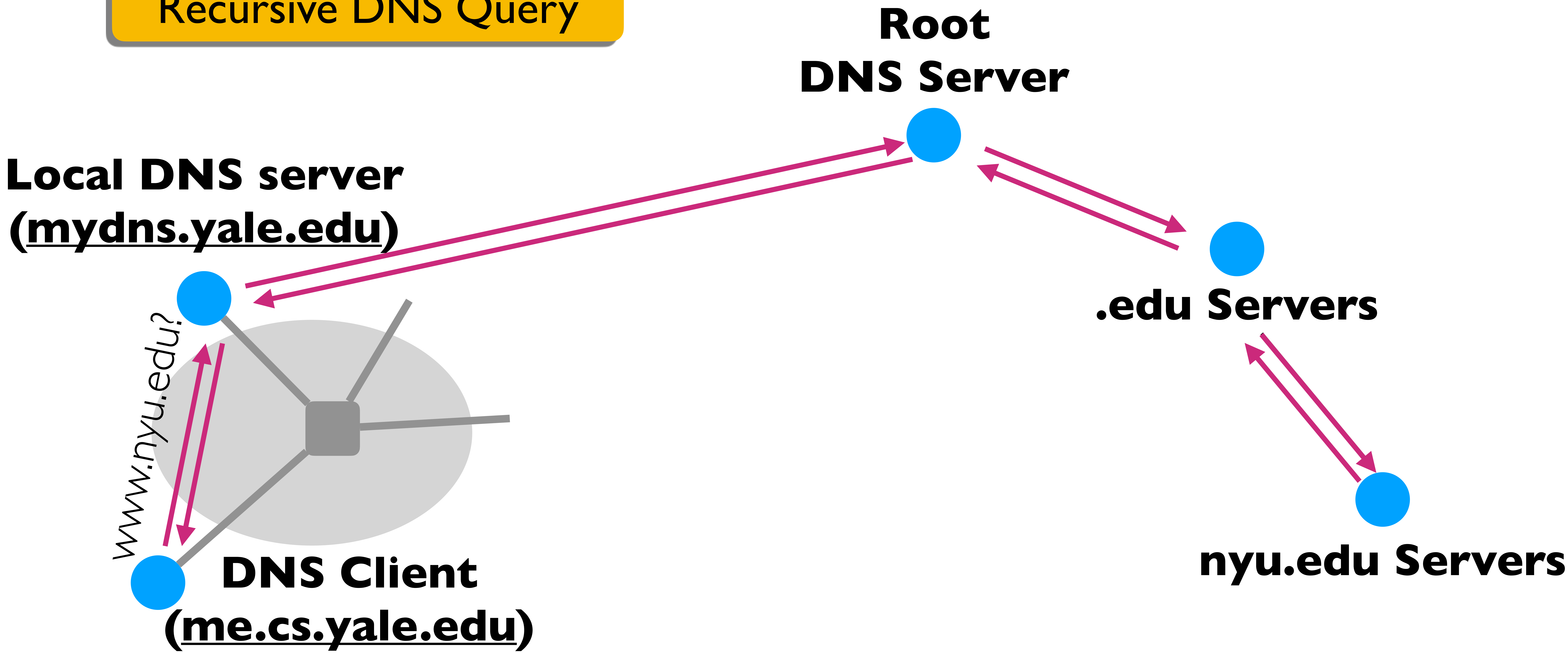
nyu.edu
Servers

DNS Client
(me.cs.yale.edu)



Root
DNS Server
Local DNS server
(mydns.yale.edu)
www.nyu.edu?
.edu Servers
nyu.edu
Servers
DNS Client
(me.cs.yale.edu)

Recursive DNS Query



Root DNS Server

Local DNS server (mydns.yale.edu)

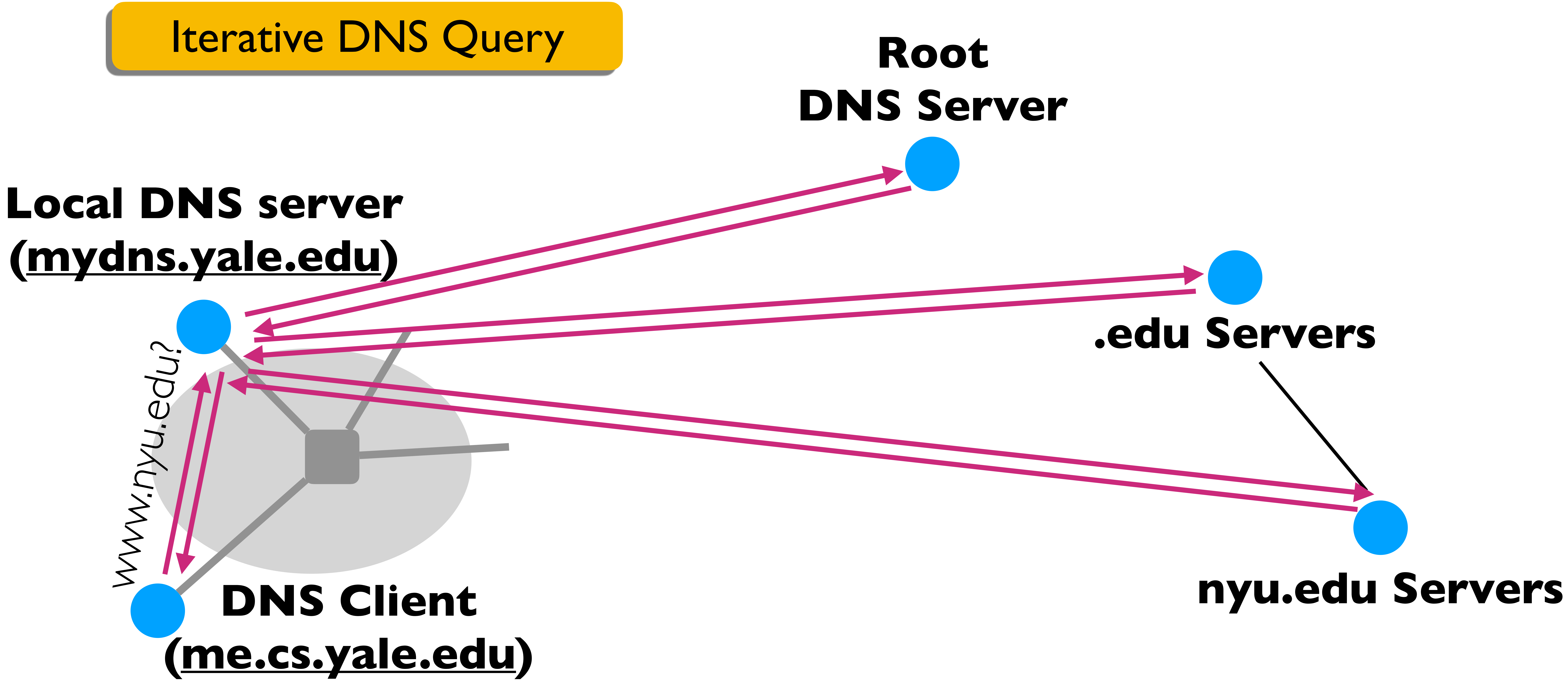
.edu Servers

www.nyu.edu?

DNS Client (me.cs.yale.edu)

nyu.edu Servers

Iterative DNS Query



Recap: DNS Protocol

- Query and Reply messages; both with the same message format
  - *See text for details*

- Client-Server Interaction on UDP Port 53
  - Spec. supports TCP too, but not always implemented

Questions?

Goals: How are we doing?

- Scalable
  - Many names
  - Many updates
  - Many users creating names
  - Many users looking up names
- Highly available
- Correct
  - No naming conflicts (uniqueness)
  - Consistency
- Lookups are fast

How?

Goals: How are we doing?

- Scalable
  - Many names
  - Many updates
  - Many users creating names
  - Many users looking up names
- Highly available

How?

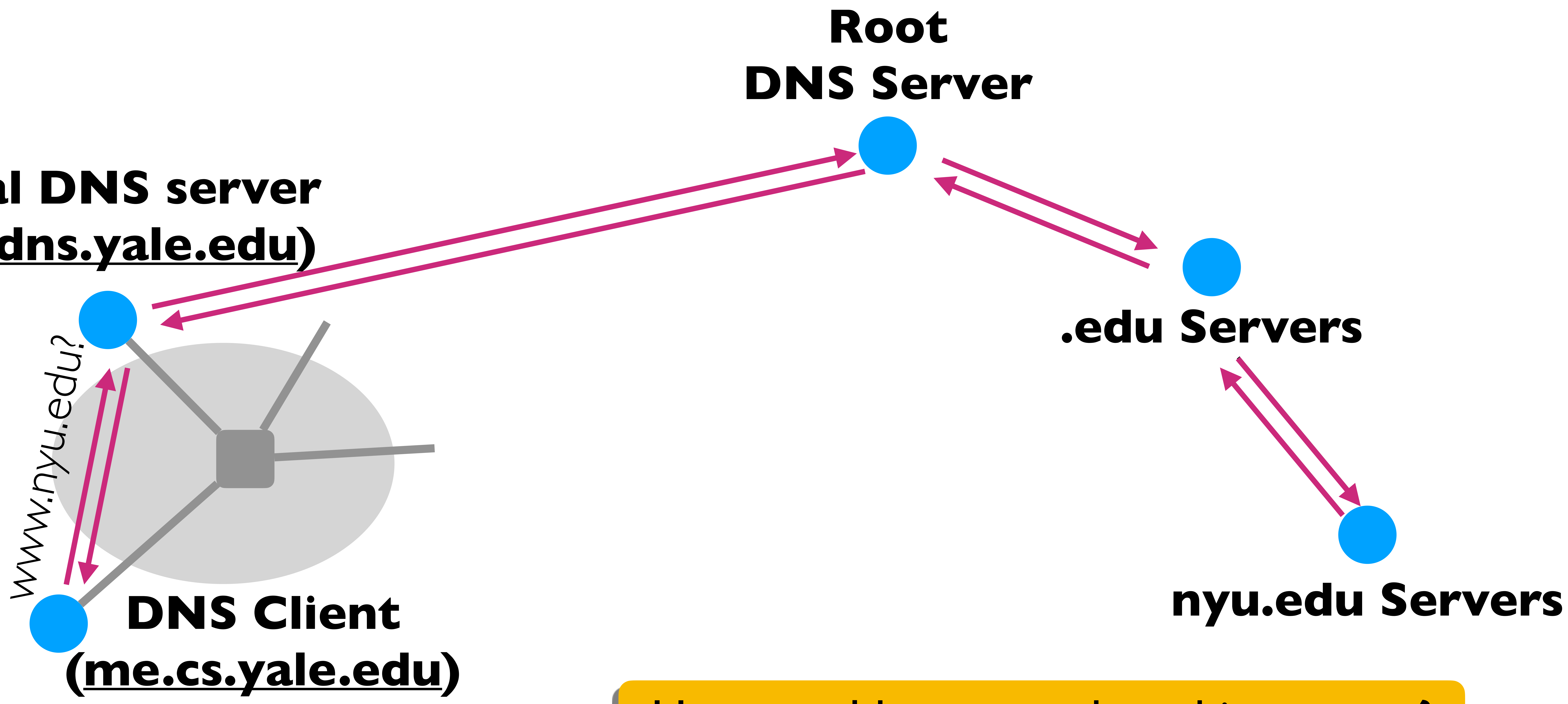
- Correct
  - No naming conflicts (uniqueness)
  - Consistency
- Lookups are fast

Per-domain Availability

- DNS servers are **replicated**
  - Primary and secondary name servers required
  - Name service available if at least one replica is up
  - Queries can be load-balanced between replicas
- Try alternate servers on timeout
  - *Exponential backoff* when retrying same server

Goals: How are we doing?

- **Scalable**
  - Many names
  - Many updates
  - Many users creating names
  - Many users looking up names
- **Highly available**
- **Correct**
  - No naming conflicts (uniqueness)
  - Consistency
- **Lookups are fast**



How would you speed up this process?

Caching

- Caching of DNS responses at all levels

- Reduces load at all levels
- Reduces delay experienced by DNS client

DNS Caching

- **How DNS caching works**
  - DNS servers cache responses to queries
  - Responses include a "time-to-live" (TTL) field
  - Server deletes cached entry after TTL expires

- **Why caching is effective**
  - The top-level servers very rarely change
  - Popular sites visited often → local DNS server often has the information cached

Negative Caching

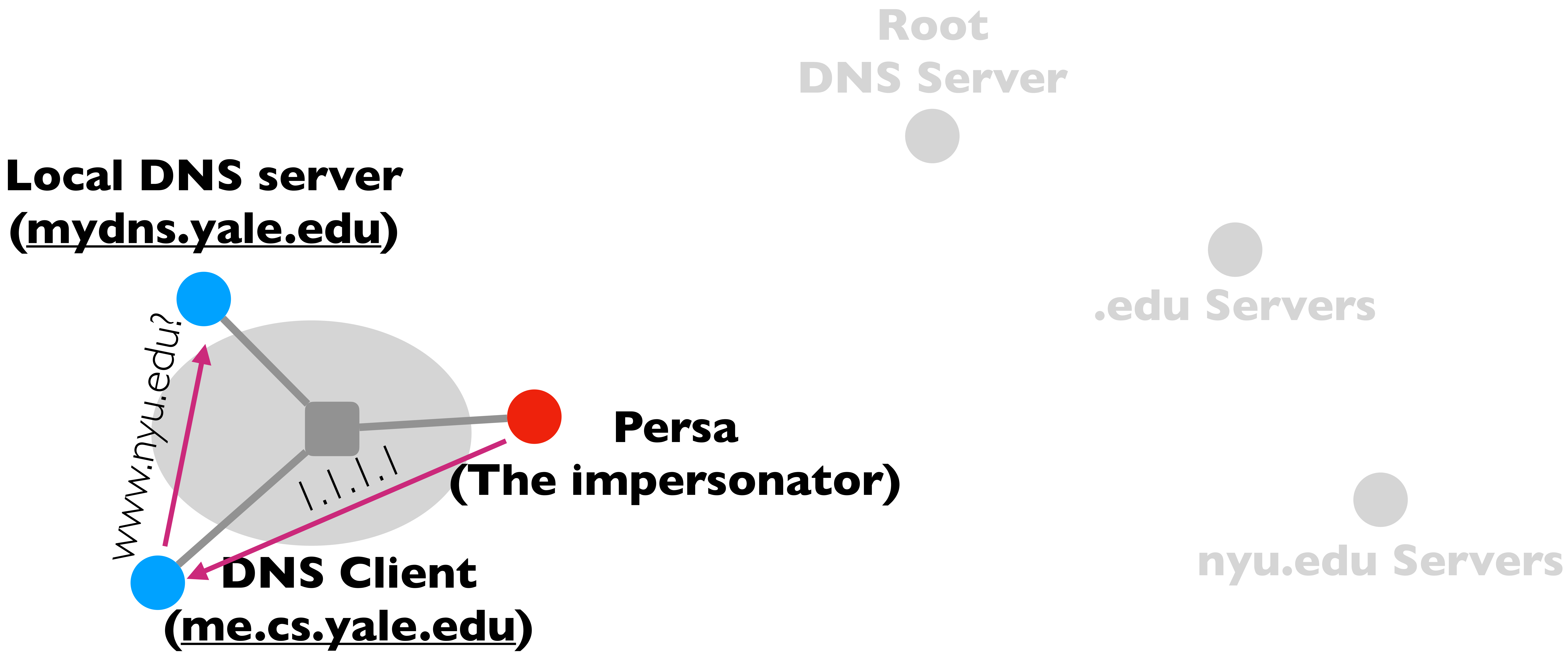
- **Remember things that don’t work**
  - Misspellings like *www.cnn.comm* and *www.cnnn.com*
  - These can take a long time to fail the first time
  - Good to remember that they don’t work
  - … so the failure takes less time the next time around

- **Negative caching is optional**

Questions?

Time to put your malicious hats on…

How can one attack DNS?



Root DNS Server

Local DNS server (mydns.yale.edu)

.edu Servers

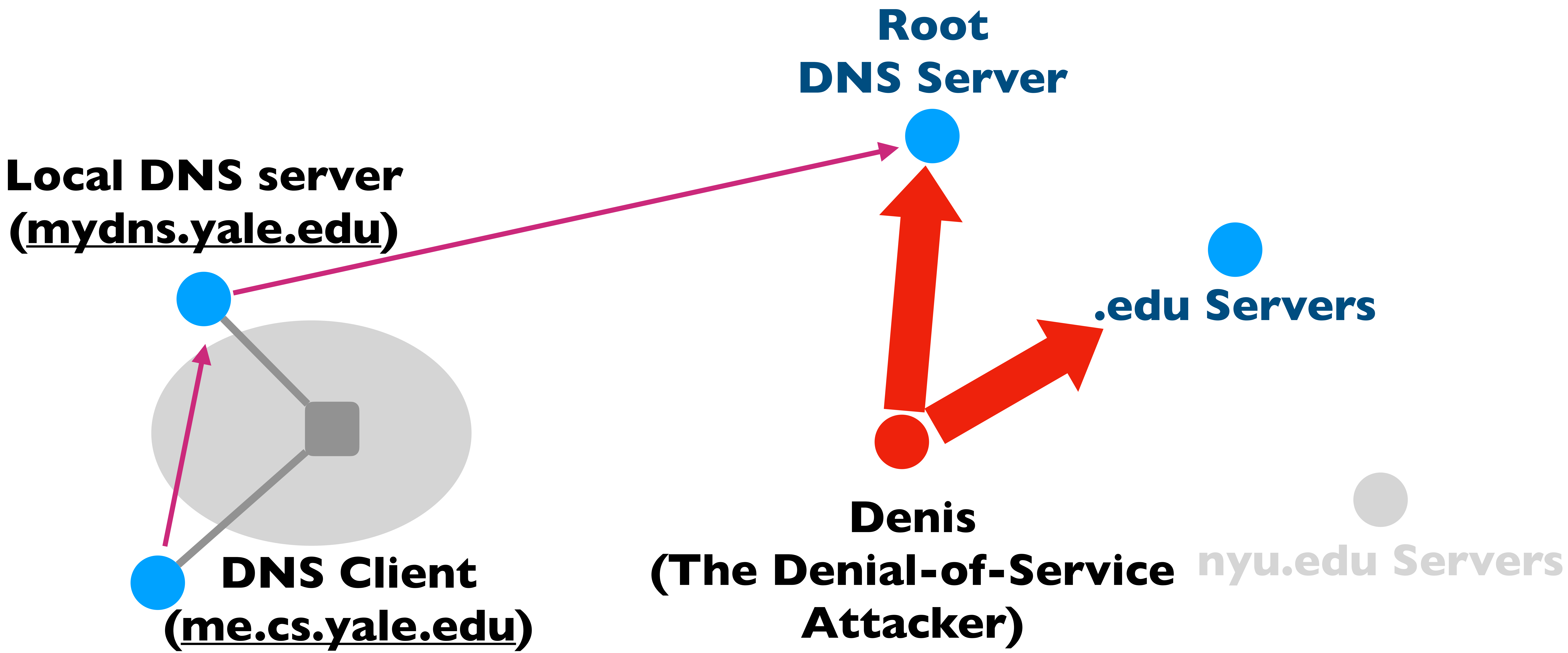
Persa (The impersonator)

DNS Client (me.cs.yale.edu)

nyu.edu Servers

How Can One Attack DNS?

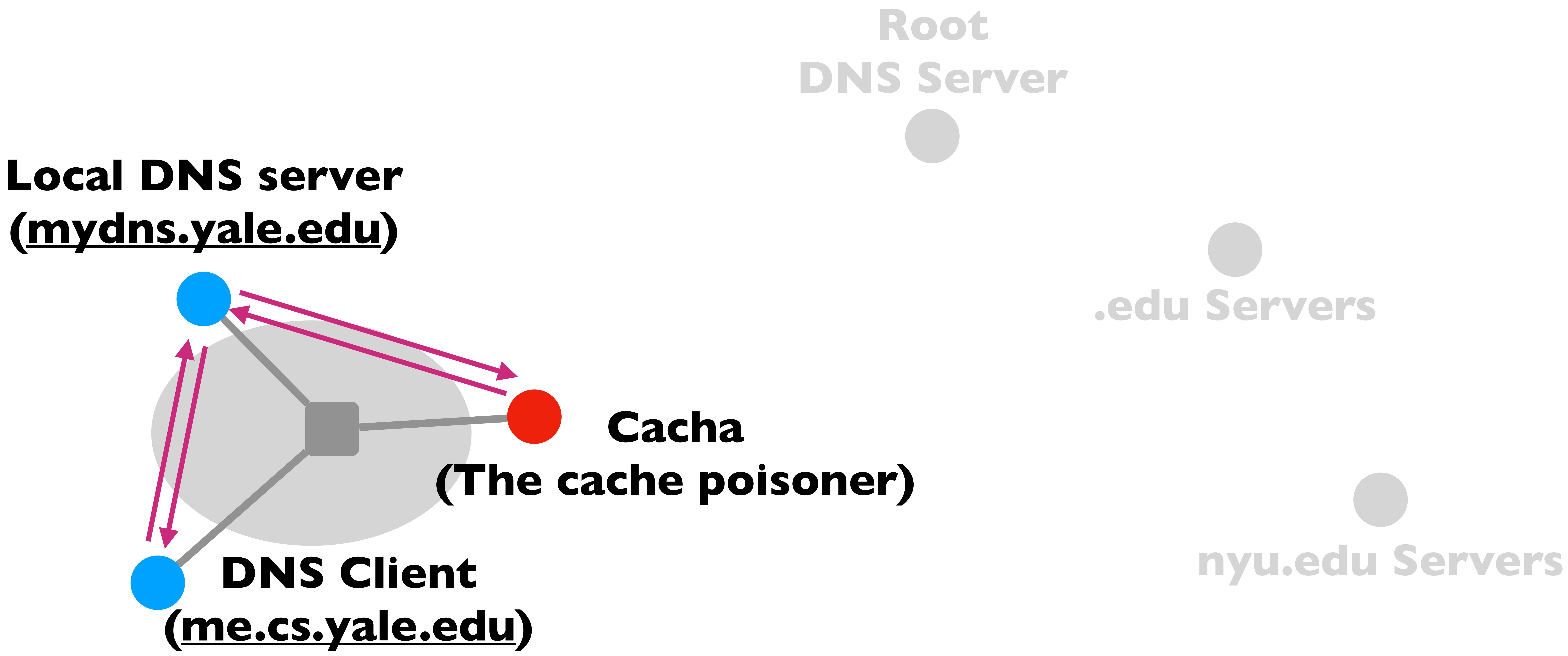
- **Impersonate the local DNS server**
  - Give the wrong IP address to the DNS client


Root
DNS Server
Local DNS server
(mydns.yale.edu)
.edu Servers
Denis
(The Denial-of-Service
Attacker)
nyu.edu Servers
DNS Client
(me.cs.yale.edu)

How Can One Attack DNS?

- **Impersonate the local DNS server**
  - Give the wrong IP address to the DNS client

- **Denial-of-service the root or TLD servers**
  - Make them unavailable to the rest of the world


Root
DNS Server
Local DNS server
(mydns.yale.edu)
.edu Servers
Cacha
(The cache poisoner)
nyu.edu Servers
DNS Client
(me.cs.yale.edu)

How Can One Attack DNS?

- **Impersonate the local DNS server**
  - Give the wrong IP address to the DNS client

- **Denial-of-service the root or TLD servers**
  - Make them unavailable to the rest of the world

- **Poison the cache of a DNS server**
  - Increase the delay experienced by DNS clients

Taking Stock: Important Properties of DNS

Administrative delegation and hierarchy results in:

- Easy unique naming
- “Fate sharing” for network failures
- Reasonable trust model
- Caching lends scalability, performance

Taking Stock: DNS Provides Indirection

- Addresses can **change** underneath
    - Move *www.cnn.com* to a new IP address
    - Humans/applications are unaffected
- Name could map to **multiple** IP addresses
    - Enables load-balancing
- Multiple names for the same addresses
    - E.g., many services (mail, www, ftp) on same machine
- Allowing "host" names to evolve into "service" names

Questions?