Comments about the quiz?

1

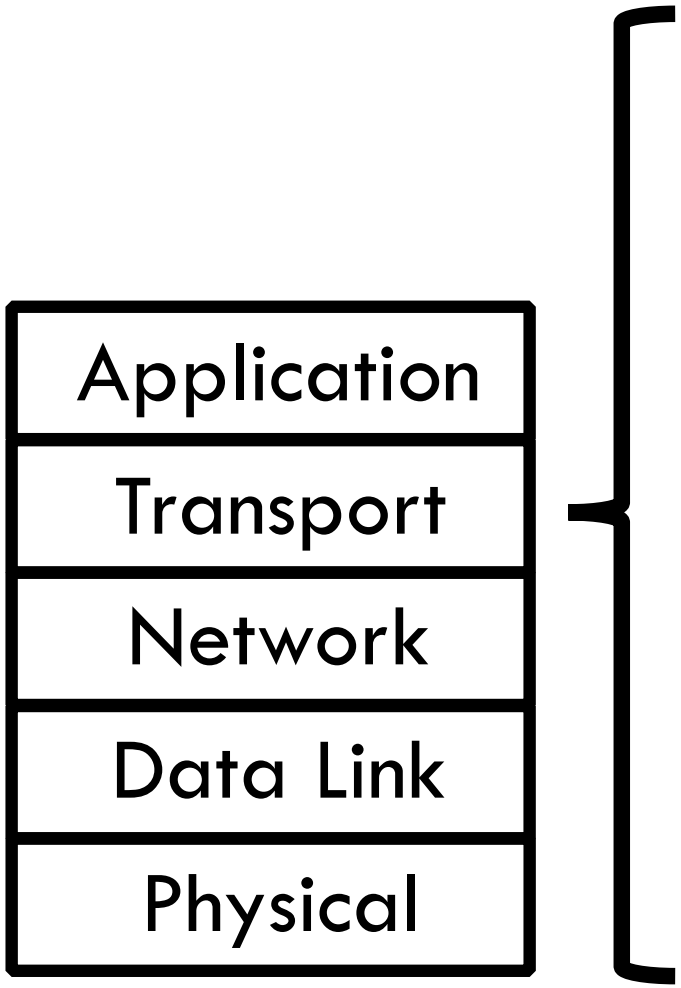
- No limit of challenges;
  - But you need to bring a document
  - Still need to come to office hours with the following in writing:
    - Specify the problem(s) you want regraded
    - For each problem, explain why the grade is in error

CSCI-351
DATA COMMUNICATION AND NETWORKS

Lecture 12: DNS and your Project2

The slide is built with the help of Prof. Alan Mislove, Christo Wilson, and David Choffnes's class

Why Skipping Transport Layer?



- No; we will cover at the next class
- Project 2 is announced: DNS

Project 2

4

Data Communication and Networks	Project 2: Simple DNS Client
CSCI-351 Fall 2018	October 2, 2018

This project is due at 11:59:59pm on October 18, 2018 and is worth 20% of your grade. You must complete it with a partner. You may only complete it alone or in a group of three if you have the instructor's explicit permission to do so for this project.

Note that there is a milestone deadline for this project, at 11:59:59pm on October 11, 2018. More details are in the Milestone section below.

1 Description

The Domain Name System (DNS) is a hierarchical system for converting domain names (e.g., `www.google.com`) to Internet Protocol (IP) addresses (e.g., 209.85.129.99). DNS is often referred to as a "phone book" for the Internet, translating human-friendly domain names into machine-friendly IP addresses. In this project, you will implement a DNS client program, which handles DNS requests by querying other machines. Note that the graduate version of this project has additional requirements, which serve as an opportunity for extra credit for students enrolled in the undergraduate version of this course.

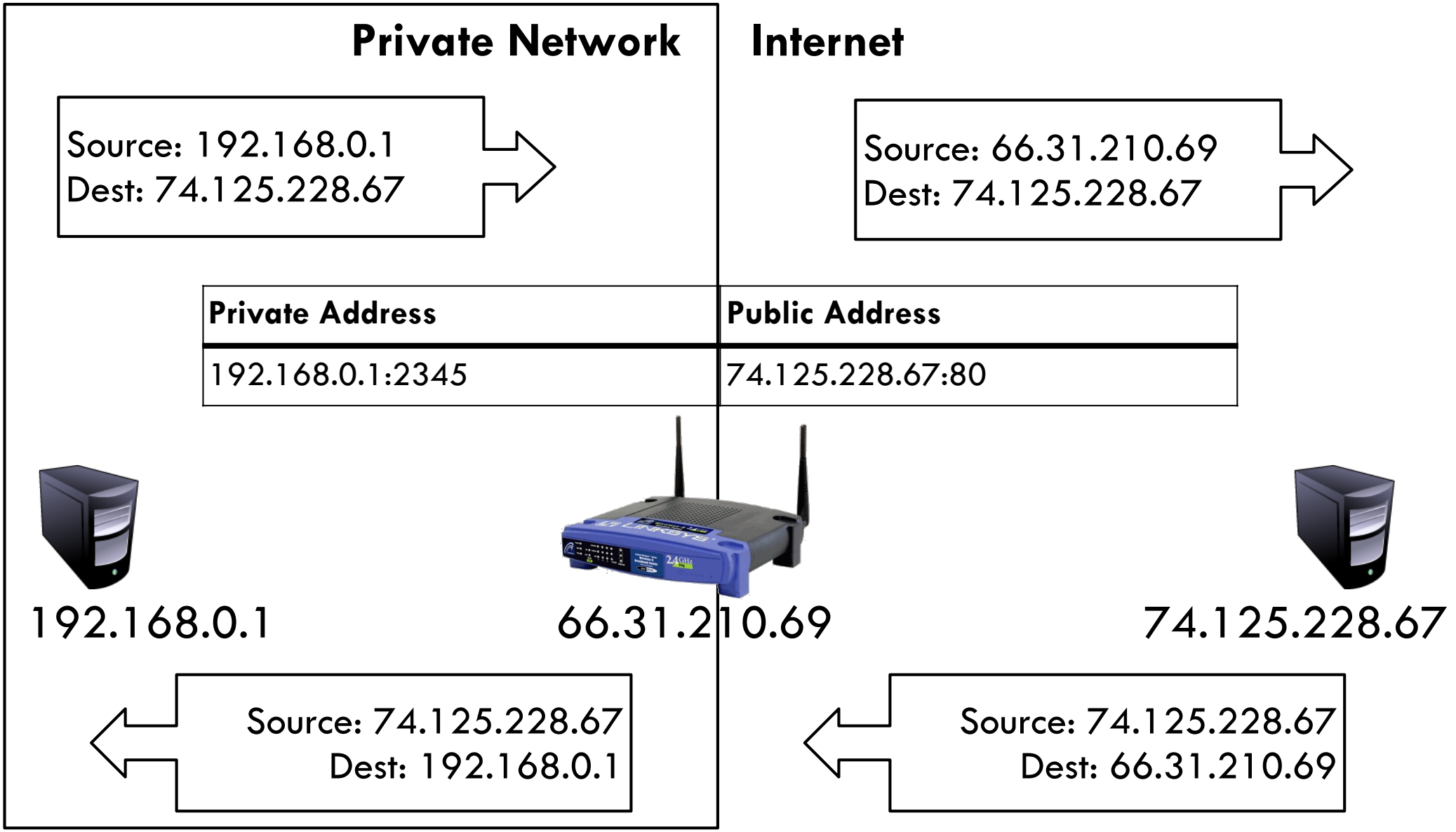
Project 2

5

The IPv4 Shortage

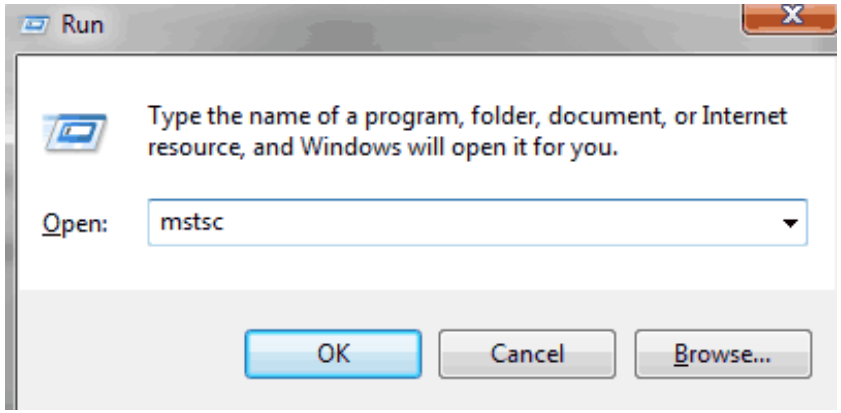
- Problem: consumer ISPs typically only give one IP address per-household
  - Additional IPs cost extra
  - More IPs may not be available
- NAT and DHCP
  - NAT + DHCP

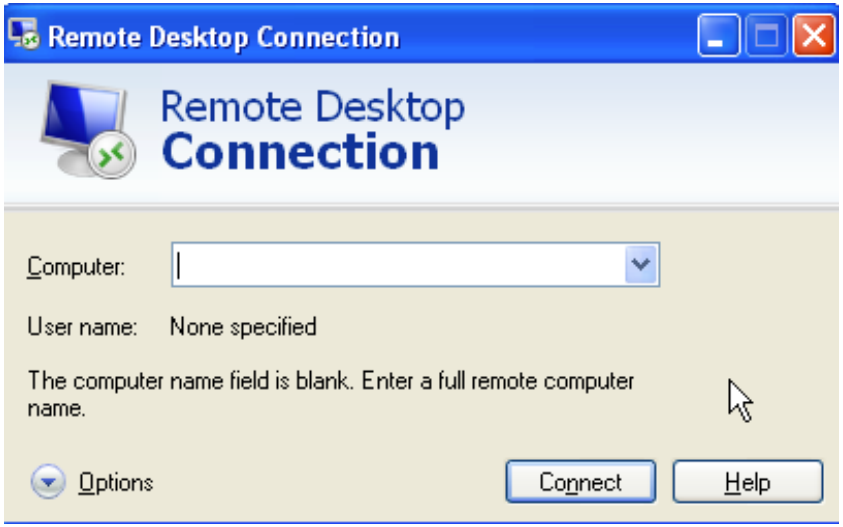
Basic NAT Operation



Port-forwarding

8

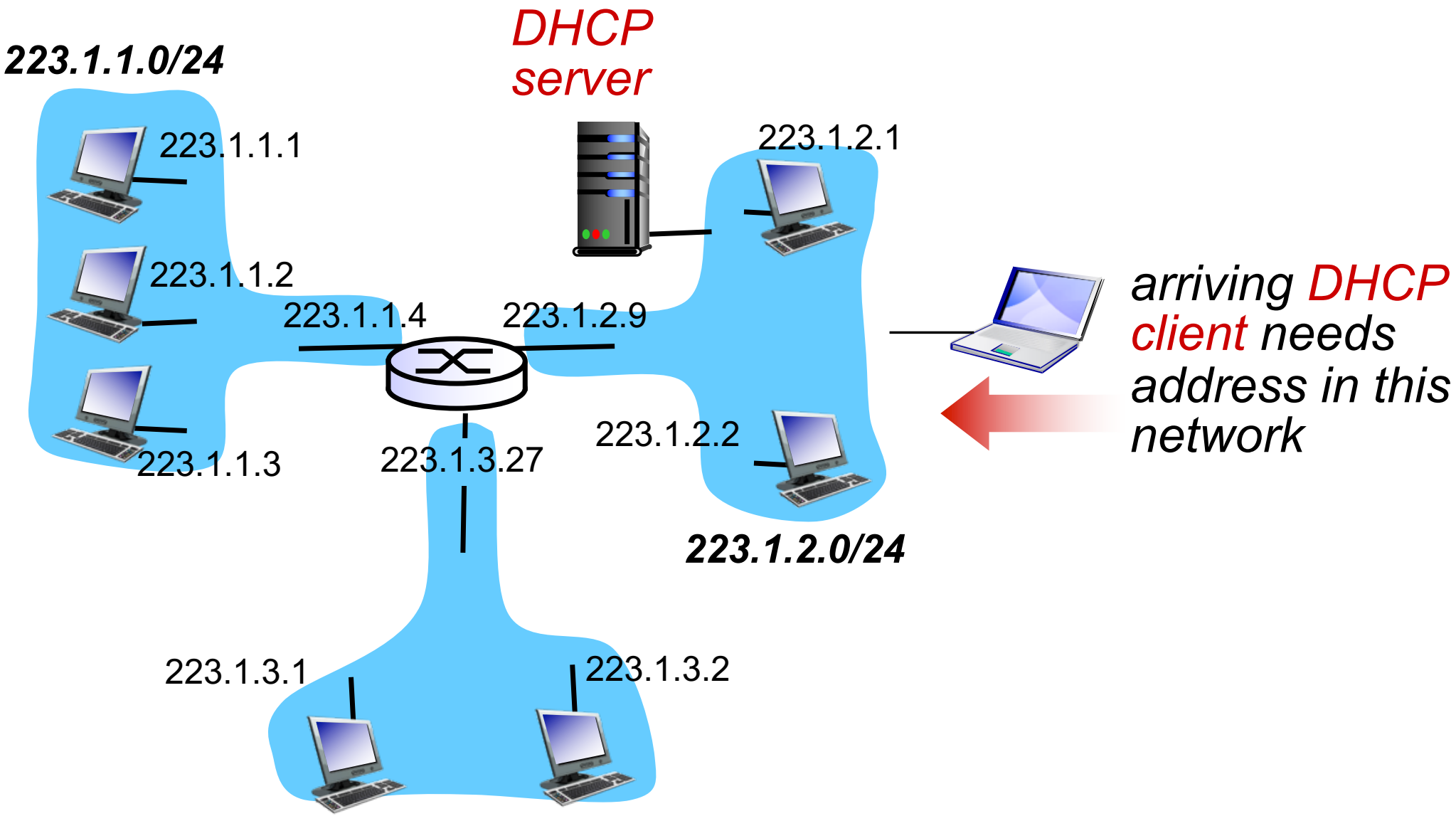




DHCP: Dynamic Host Configuration Protocol

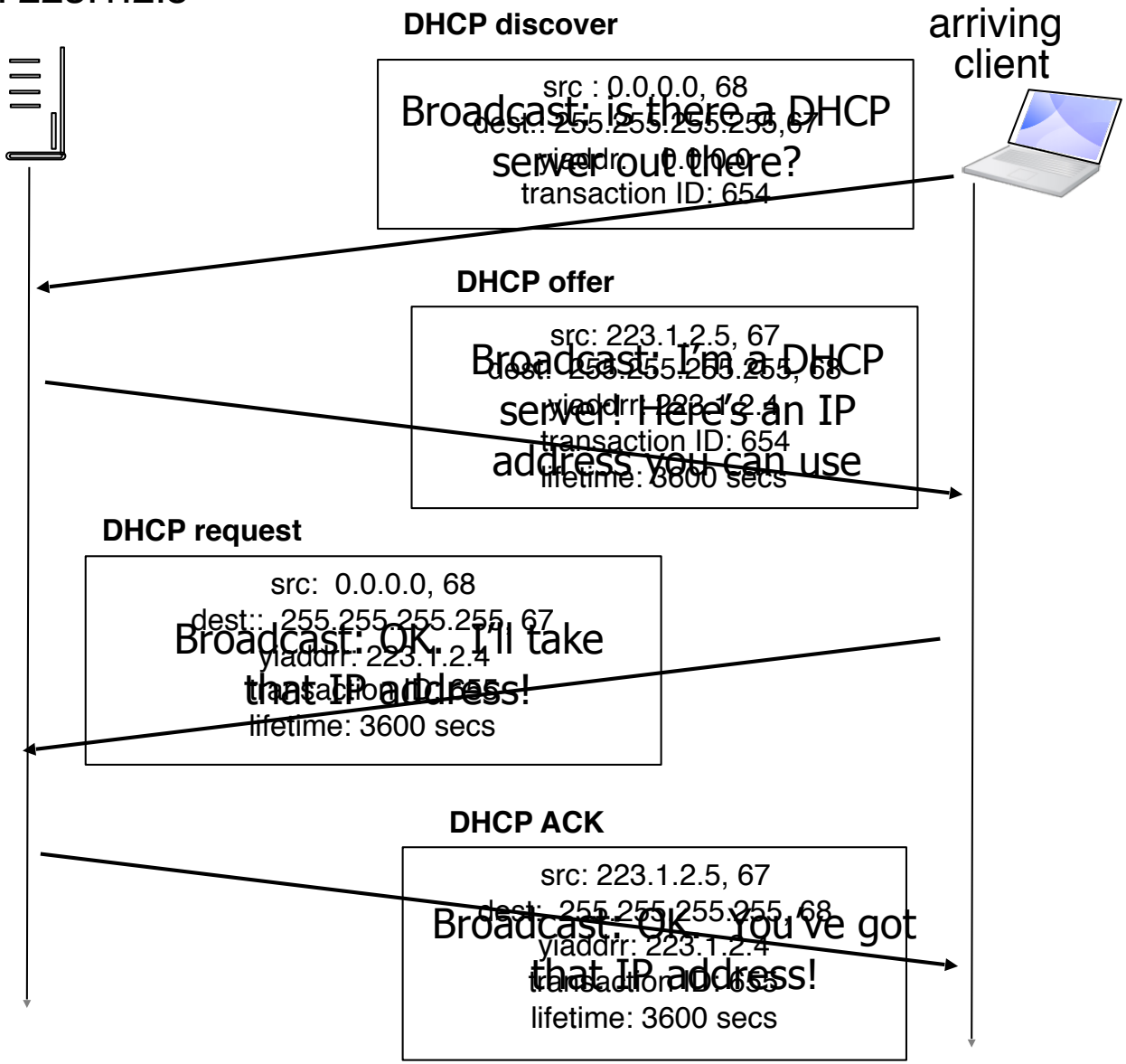
- Let's say that a ISP has X customers, How many IPs does it need to have?
  - X?
- Goal: allow host to *dynamically* obtain its IP address from network server when it joins network
  - can renew its lease on address in use
  - allows reuse of addresses (only hold address while connected/"on")
  - support for mobile users who want to join network (more shortly)

DHCP Client-Server



DHCP Client-Server

DHCP server: 223.1.2.5



DHCP: More than IP address

- DHCP can return more than just allocated IP address on subnet
  - address of first-hop router for client
  - name and IP address of DNS sever
  - network mask (indicating network versus host portion of address)

DHCP Header (Do not memorize)

Dynamic Host Configuration Protocol				
Bit Offset	0–15		16–31	
0	OpCode	Hardware Type	Hardware Length	Hops
32	Transaction ID			
64	Seconds Elapsed		Flags	
96	Client IP Address			
128	Your IP Address			
160	Server IP Address			
196	Gateway IP Address			
228+	Client Hardware Address (16 bytes)			
	Server Host Name (64 bytes)			
	Boot File (128 bytes)			
	Options			

CSCI-351
DATA COMMUNICATION AND NETWORKS

Lecture 12: DNS

The slide is built with the help of Prof. Alan Mislove, Christo Wilson, and David Choffnes's class

Layer 8 (The Carbon-based nodes)

- If you want to…
  - Call someone, you need to ask for their phone number
    - You can't just dial "P R O F C H U N G"
  - Mail someone, you need to get their address first
- What about the Internet?
  - If you need to reach Google, you need their IP
  - Does anyone know Google's IP?
- Problem:
  - People can't remember IP addresses
  - Need human readable names that map to IPs

Internet Names and Addresses

- Addresses, e.g. 129.10.117.100
  - Computer usable labels for machines
  - Conform to structure of the network
- Names, e.g. www.rit.edu
  - Human usable labels for machines
  - Conform to organizational structure
- How do you map from one to the other?
  - Domain Name System (DNS)

History

- Before DNS, all mappings were in *hosts.txt*
  - */etc/hosts* on Linux
  - C:*Windows**System32**drivers**etc**hosts* on Windows
- Centralized, manual system
  - Changes were submitted to SRI via email
  - Machines periodically FTP new copies of *hosts.txt*
  - Administrators could pick names at their discretion
  - Any name was allowed
    - tijay_server_at_rit_pwns_joo_lol_kthxbye

Towards DNS

- Eventually, the *hosts.txt* system fell apart
  - Not scalable, SRI couldn't handle the load
  - Hard to enforce uniqueness of names
    - e.g RIT
      - Rochester Institute of Technology?
      - Revolution in Training (US Navy)
  - Many machines had inaccurate copies of *hosts.txt*
- Thus, DNS was born

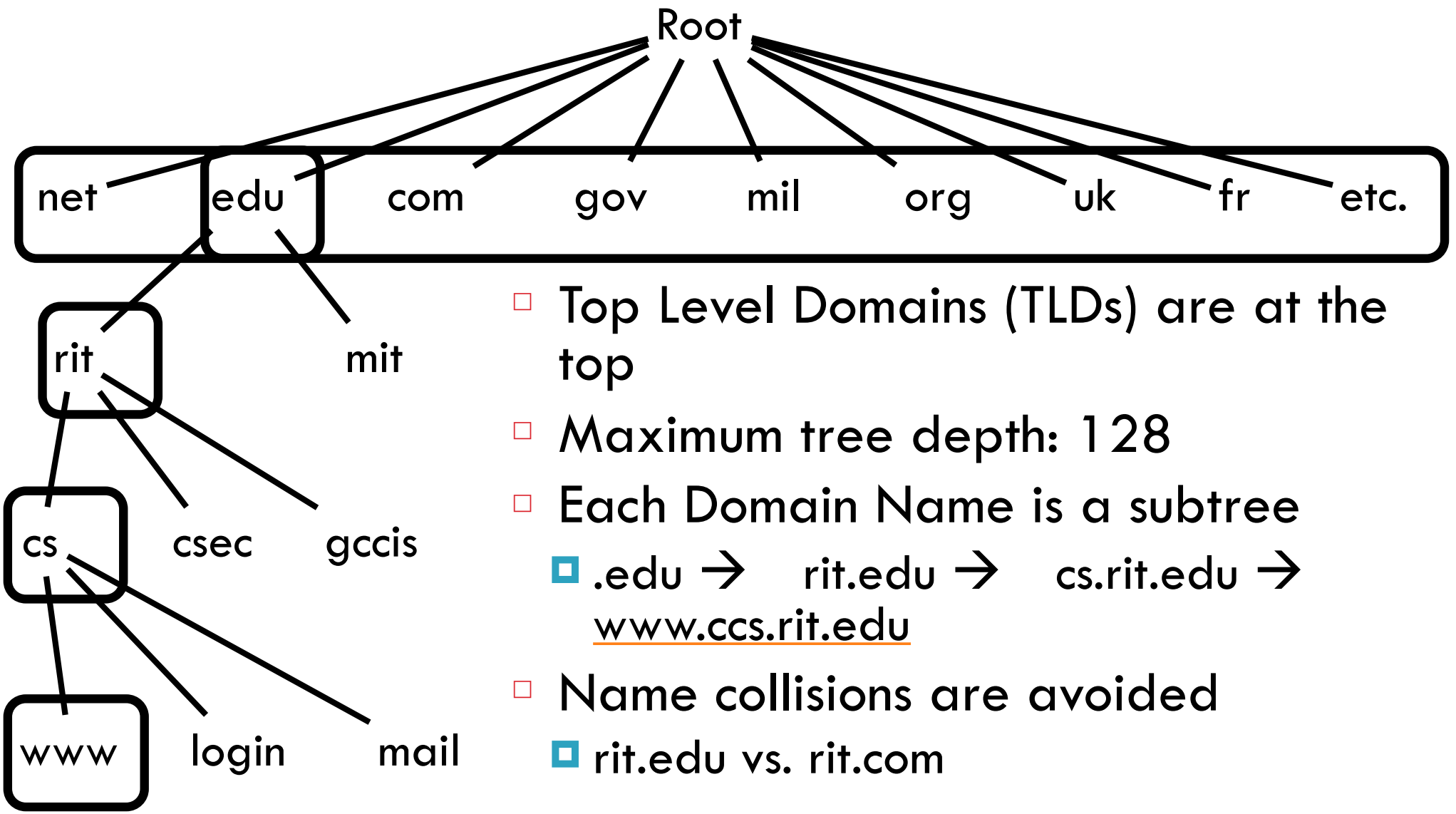
Outline

- DNS Basics
- DNS Security

DNS at a High-Level

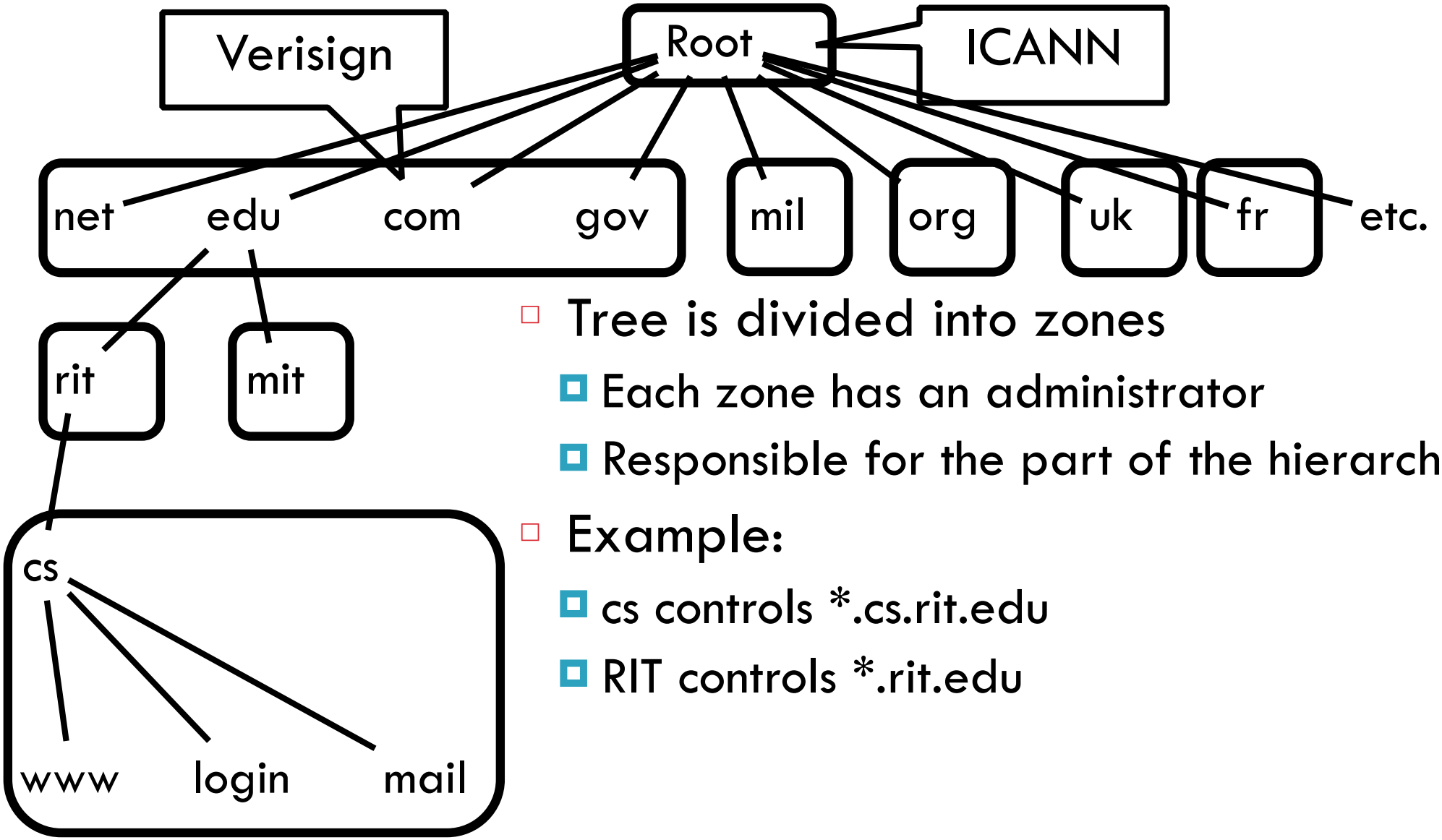
- Domain Name System
- Distributed database
  - No centralization
- Simple client/server architecture
  - UDP port 53, some implementations also use TCP
  - Why? (You will learn at the TCP-lecture)
- Hierarchical namespace
  - As opposed to original, flat namespace
  - e.g. .com → google.com → mail.google.com

Naming Hierarchy



- Top Level Domains (TLDs) are at the top
- Maximum tree depth: 128
- Each Domain Name is a subtree
  - .edu → rit.edu → cs.rit.edu → www.ccs.rit.edu
- Name collisions are avoided
  - rit.edu vs. rit.com

Hierarchical Administration



- Tree is divided into zones
  - Each zone has an administrator
  - Responsible for the part of the hierarchy
- Example:
  - cs controls *.cs.rit.edu
  - RIT controls *.rit.edu

Server Hierarchy

- Functions of each DNS server:
  - Authority over a portion of the hierarchy
    - No need to store all DNS names
  - Store all the records for hosts/domains in its zone
    - May be replicated for robustness
  - Know the addresses of the root servers
    - Resolve queries for unknown names
- Root servers know about all TLDs
  - The buck stops at the root servers

Root Name Servers

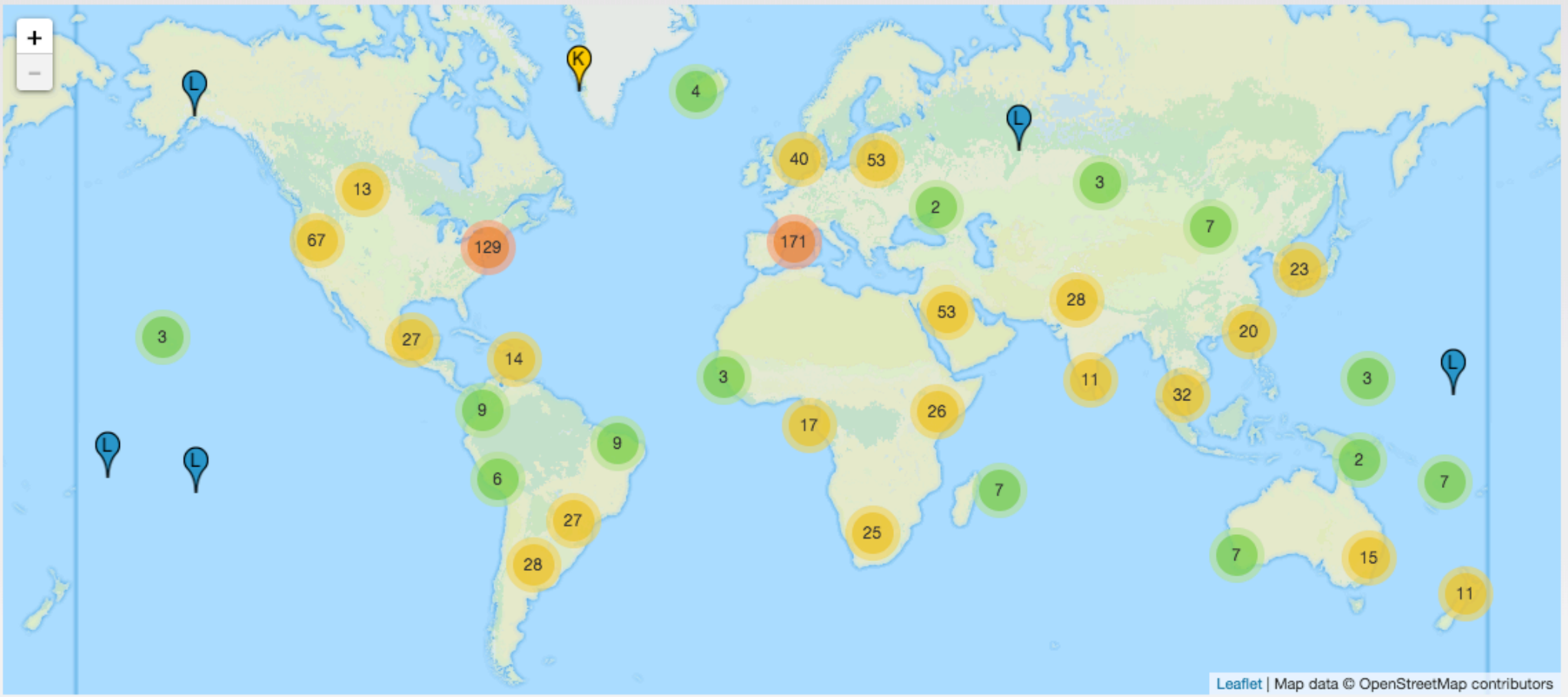
- Responsible for the Root Zone File
  - Lists the TLDs and who controls them
  - ~272KB in size

com.	172800	IN	NS	a.gtld-servers.net.
com.	172800	IN	NS	b.gtld-servers.net.
com.	172800	IN	NS	c.gtld-servers.net.

- Administered by ICANN
  - 13 root servers, labeled A→M
  - 6 are anycasted, i.e. they are globally replicated
- Contacted when names cannot be resolved
  - In practice, most systems cache this information

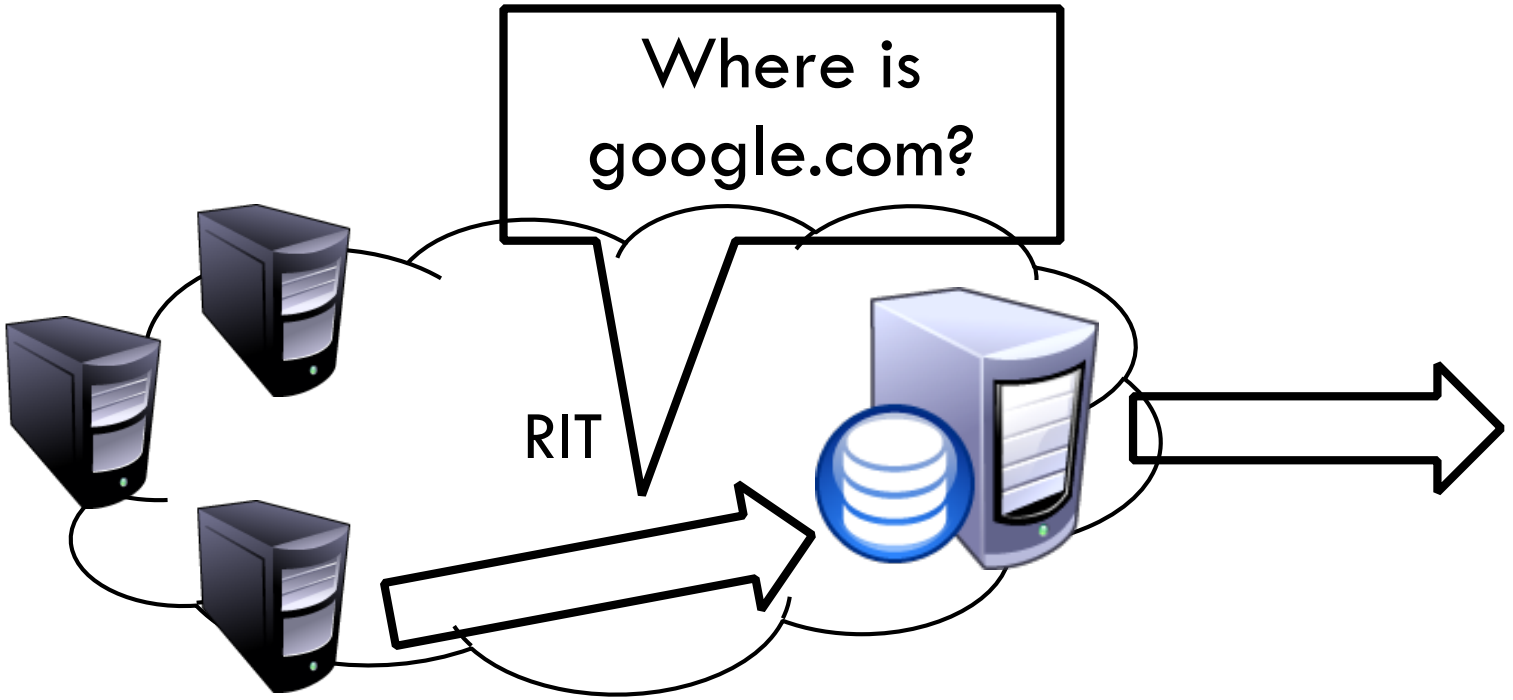
Map of the Roots (root-servers.org)

25



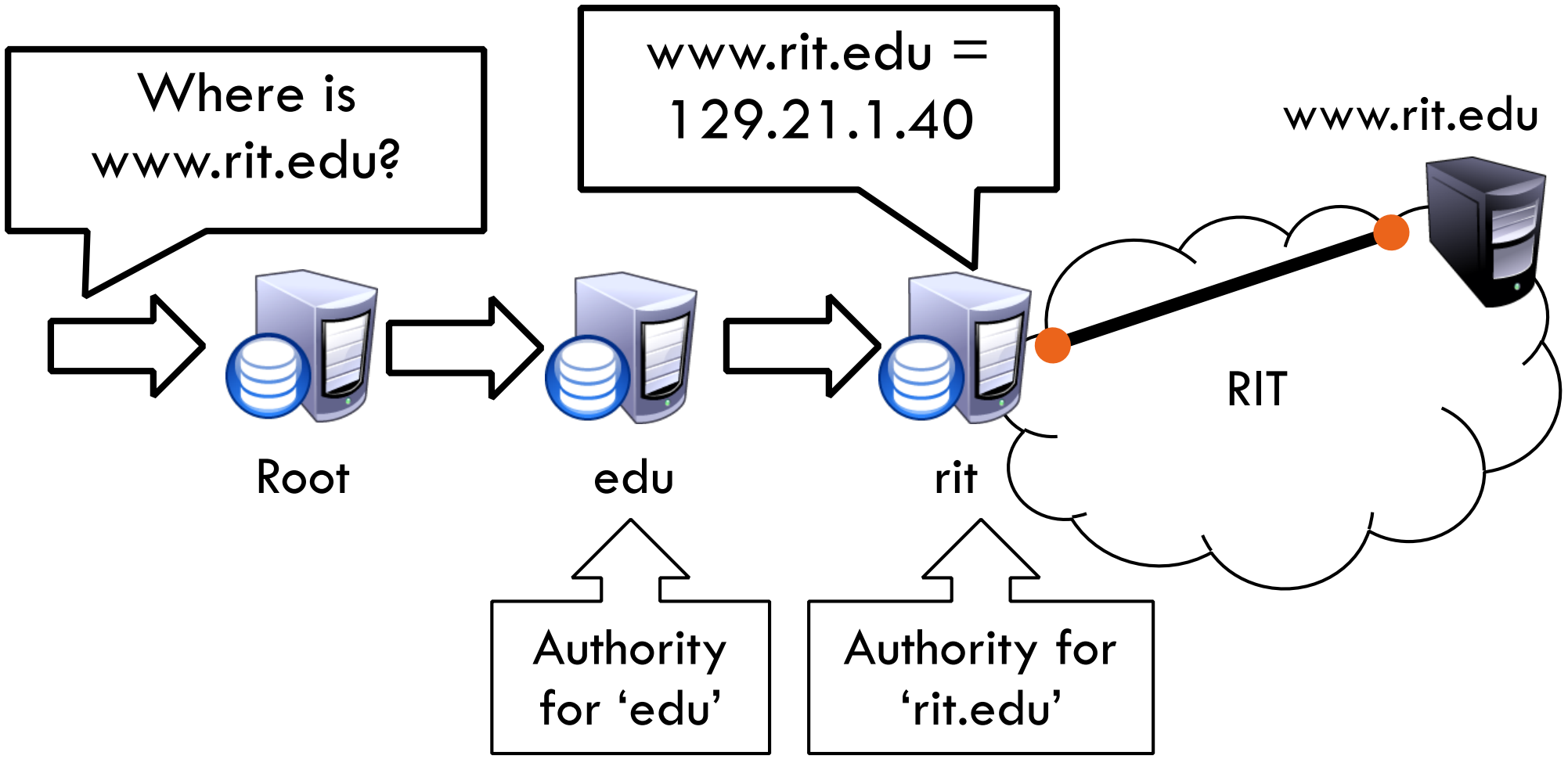
As of 2018-10-02, the root server system consists of 908 instances operated by the 12 independent root server operators.

Local Name Servers (Resolvers)



- Each ISP/company has a local, default name server
- Often configured via DHCP
- Hosts begin DNS queries by contacting the local name server
- Frequently cache query results

Authoritative Name Servers



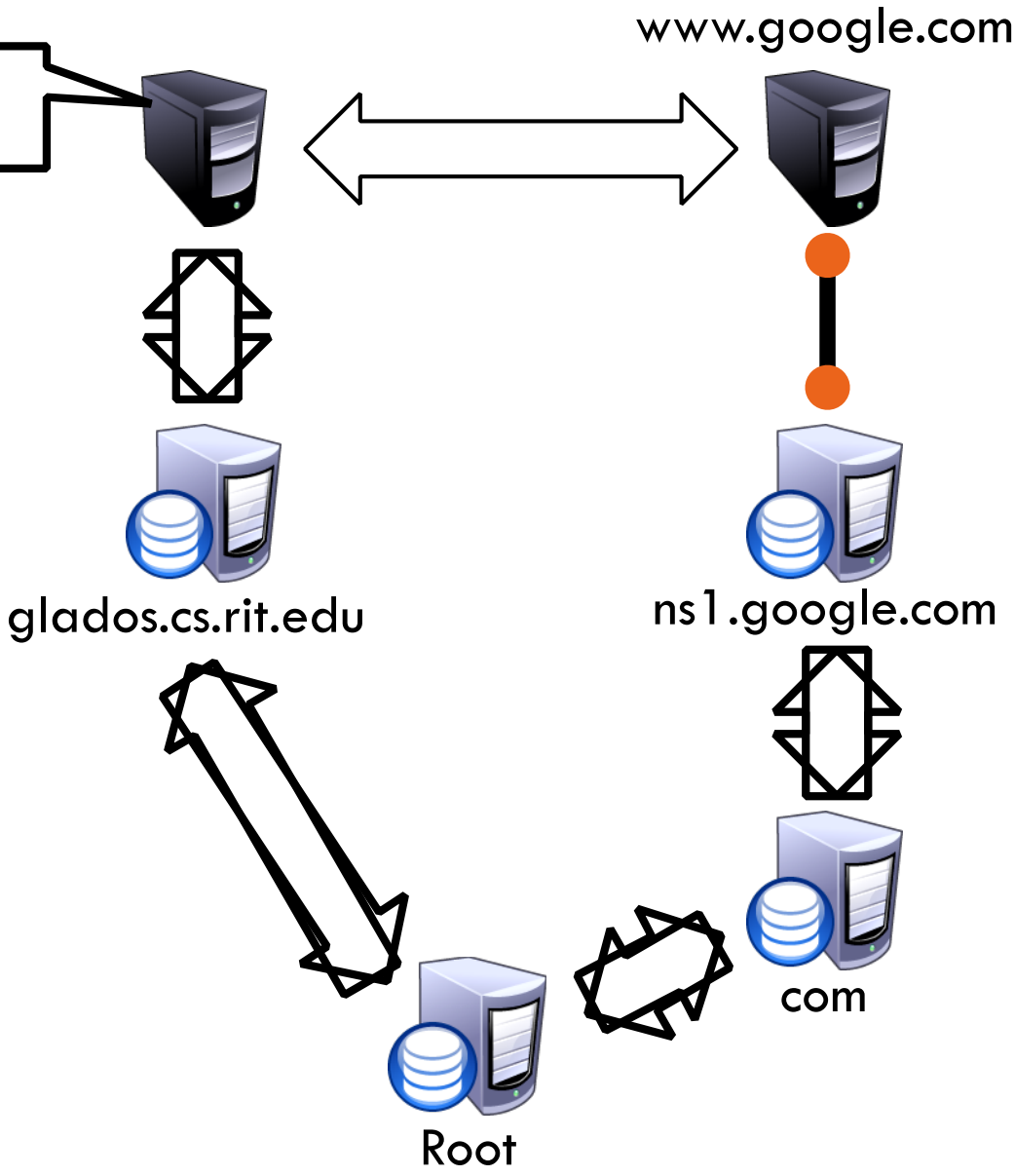
- Stores the name→IP mapping for a given host

Basic Domain Name Resolution

- Every host knows a local DNS server
  - Sends all queries to the local DNS server
- If the local DNS can answer the query, then you're done
  1. Local server has cached the record for that name
- Otherwise, go down the hierarchy and search for the authoritative name server
  - Every local DNS server knows the root servers
  - Use cache to skip steps if possible
    - e.g. skip the root and go directly to .edu if the root file is cached

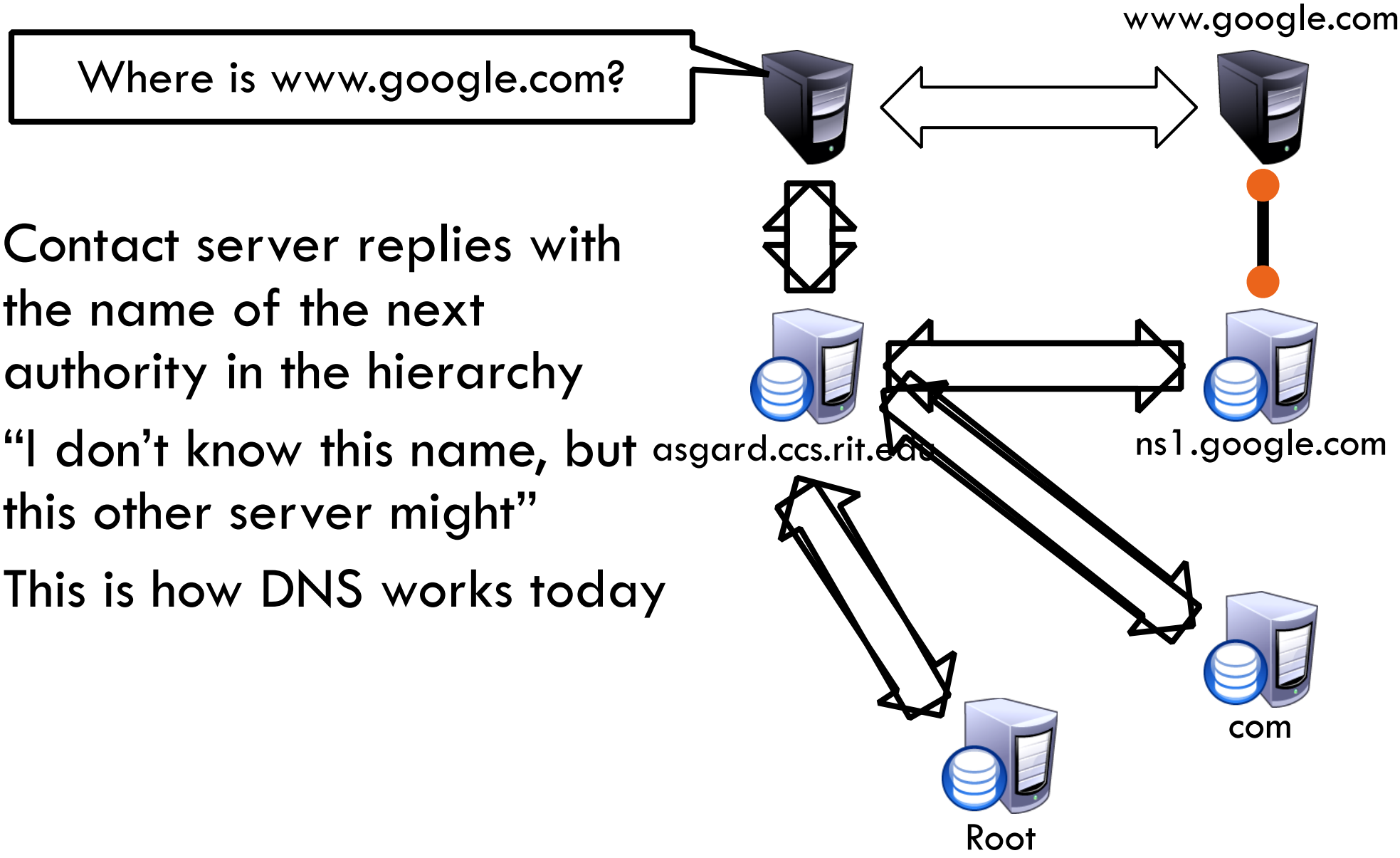
Recursive DNS Query

- Puts the burden of resolution on the contacted name server
- How does glados know who to forward responses too?
  - Random IDs embedded in DNS queries
- What have we said about keeping state in the network?



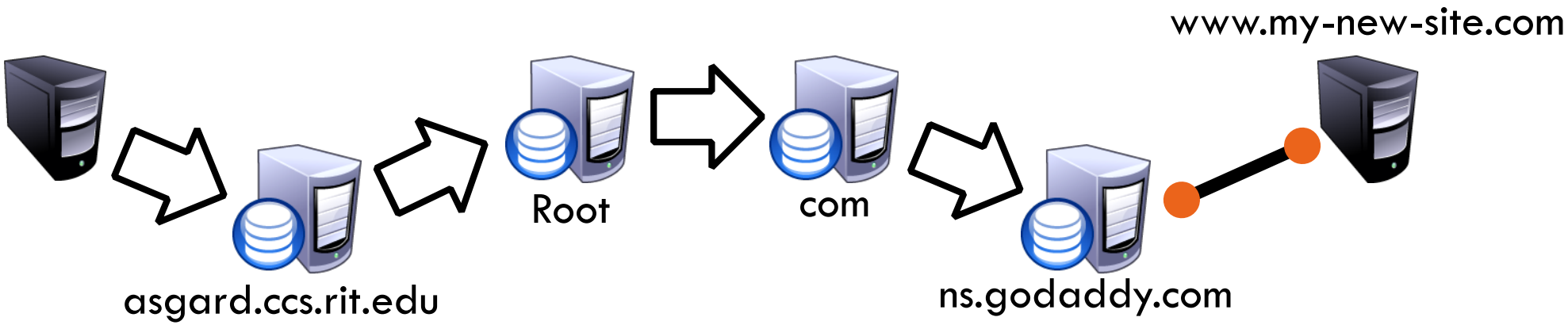
Iterated DNS query

- Contact server replies with the name of the next authority in the hierarchy
- "I don't know this name, but this other server might"
- This is how DNS works today



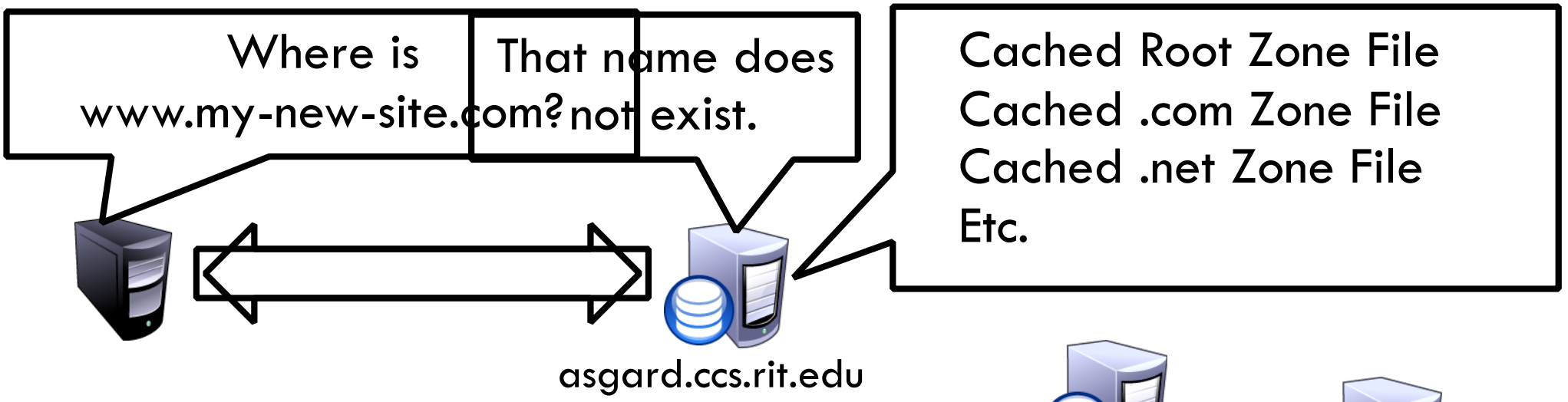
DNS Propagation

- How many of you have purchased a domain name?
  - Did you notice that it took ~72 hours for your name to become accessible?
  - This delay is called DNS Propagation

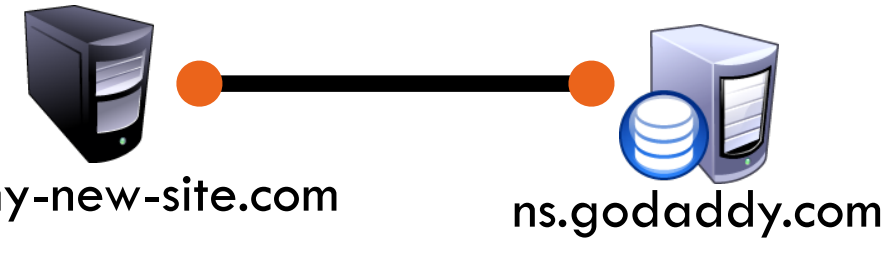


Caching vs. Freshness

- DNS Propagation delay is caused by caching



- Zone files may be cached for 1-72 hours



DNS Resource Records

- DNS queries have two fields: name and type
- Resource record is the response to a query
  - Four fields: (name, value, type, TTL)
  - There may be multiple records returned for one query
- What are do the name and value mean?
  - Depends on the type of query and response

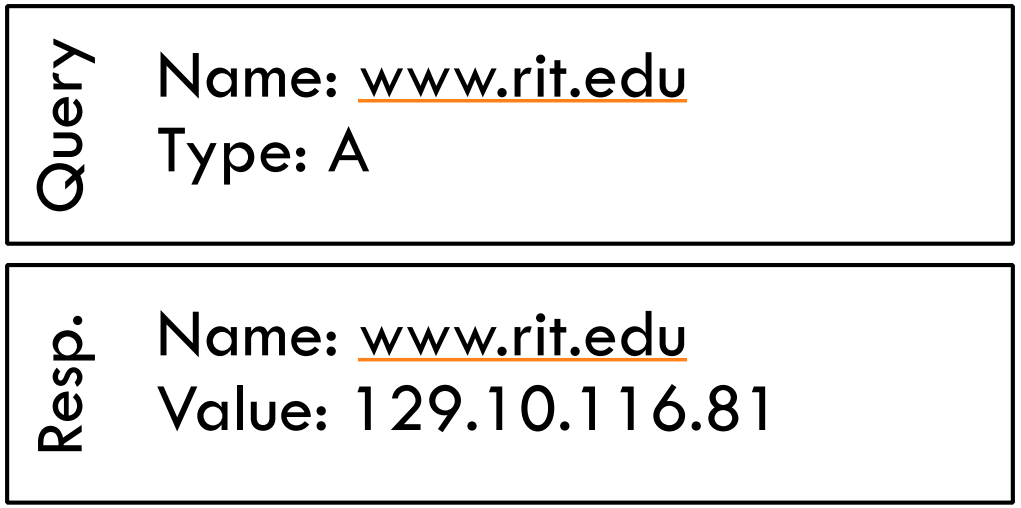
DNS header:

00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Identification																QR	Opcode				AA	TC	RD	RA	Z	AD	CD	Rcode				
Total Questions																Total Answer RRs															
Total Authority RRs																Total Additional RRs															
Questions [] :::																															
Answer RRs [] :::																															
Authority RRs [] :::																															
Additional RRs [] :::																															

DNS Types

- Type = NS
  - Name = partial domain
  - Value = name of DNS server for this domain
  - "Go send your query to this other server"

- Type = A / AAAA
  - Name = domain name
  - Value = IP address
  - A is IPv4, AAAA is IPv6



DNS Types, Continued

- Type = CNAME
  - Name = hostname
  - Value = canonical hostname
  - Useful for aliasing
  - CDNs use this (will be covered)

Query
Name: foo.mysite.com
Type: CNAME

Resp.
Name: foo.mysite.com
Value: bar.mysite.com

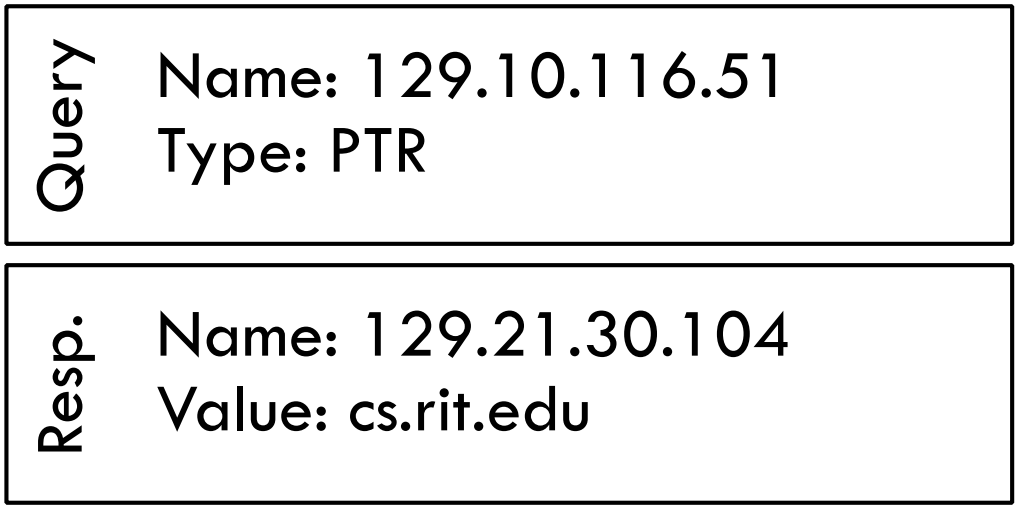
- Type = MX
  - Name = domain in email address
  - Value = canonical name of mail server

Query
Name: cs.rit.edu
Type: MX

Resp.
Name: cs.rit.edu
Value: pony-express.cs.rit.edu.

Reverse Lookups

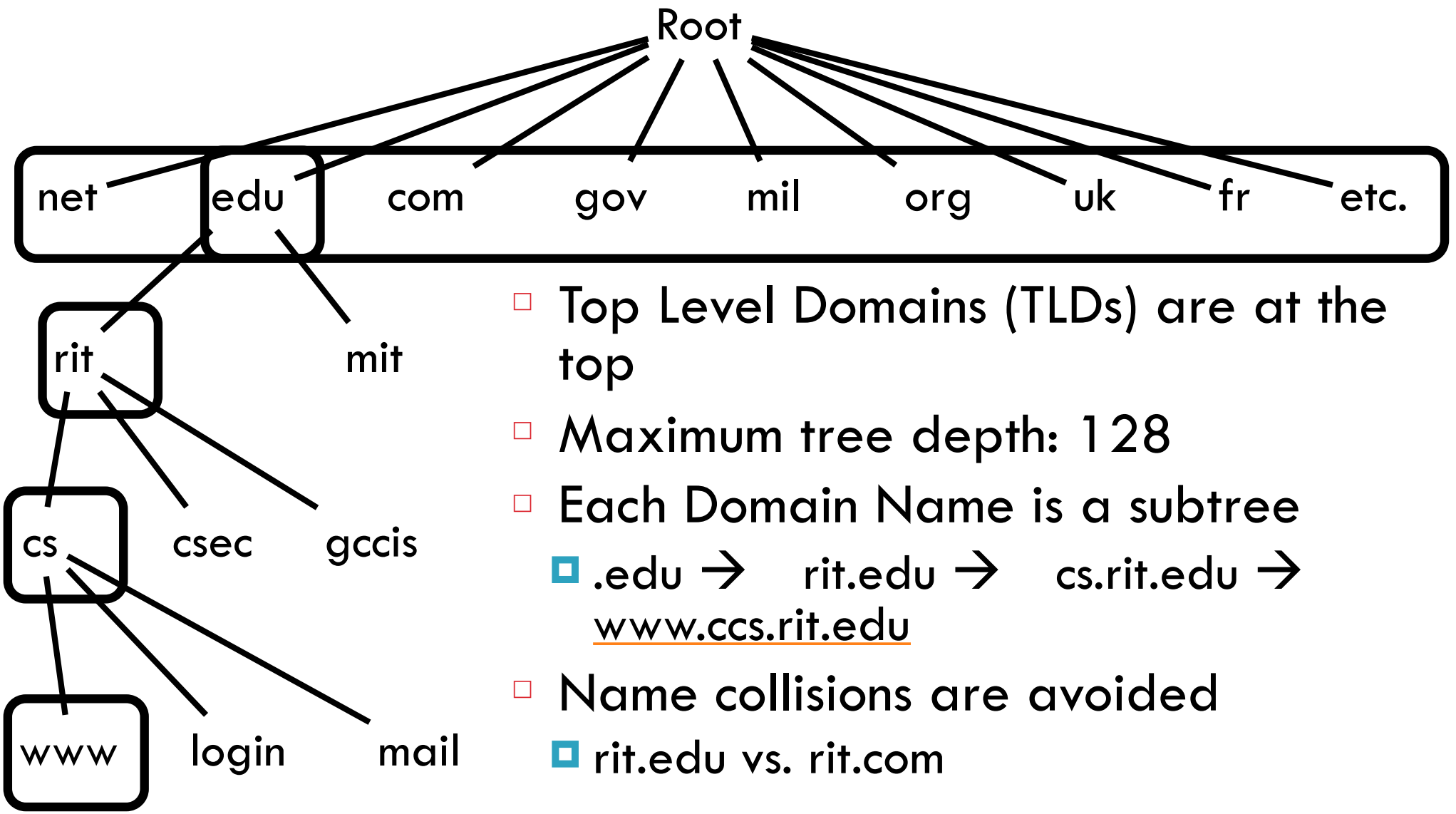
- What about the IP→name mapping?
- Separate server hierarchy stores reverse mappings
  - Rooted at in-addr.arpa and ip6.arpa
- Additional DNS record type: PTR
  - Name = IP address
  - Value = domain name
- Not guaranteed to exist for all IPs
- Why do we need this?



DNS Recap

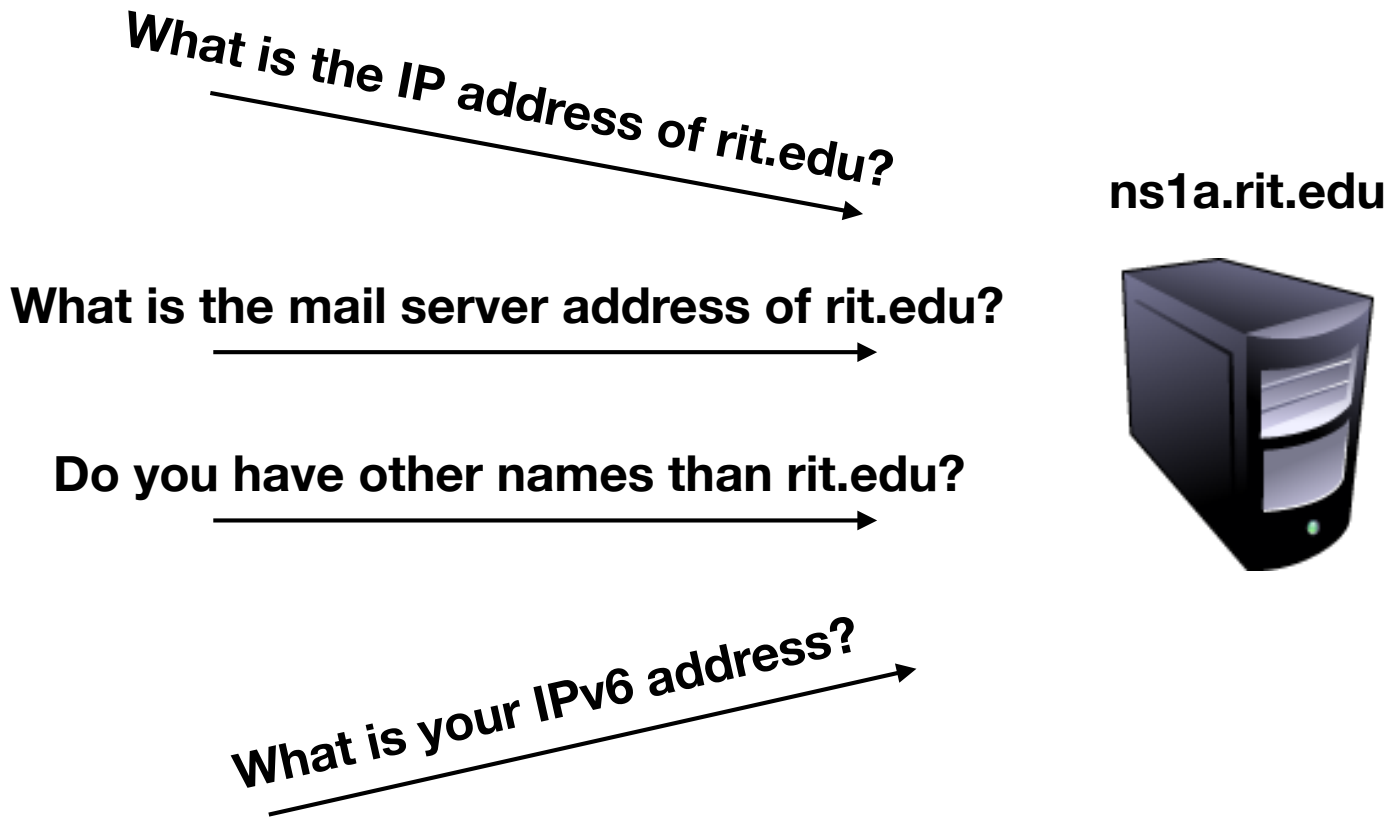
- Name hierarchy
- Two kinds of DNS servers:
  - Authoritative Nameserver
  - Resolver
- How to buy a domain name

Naming Hierarchy

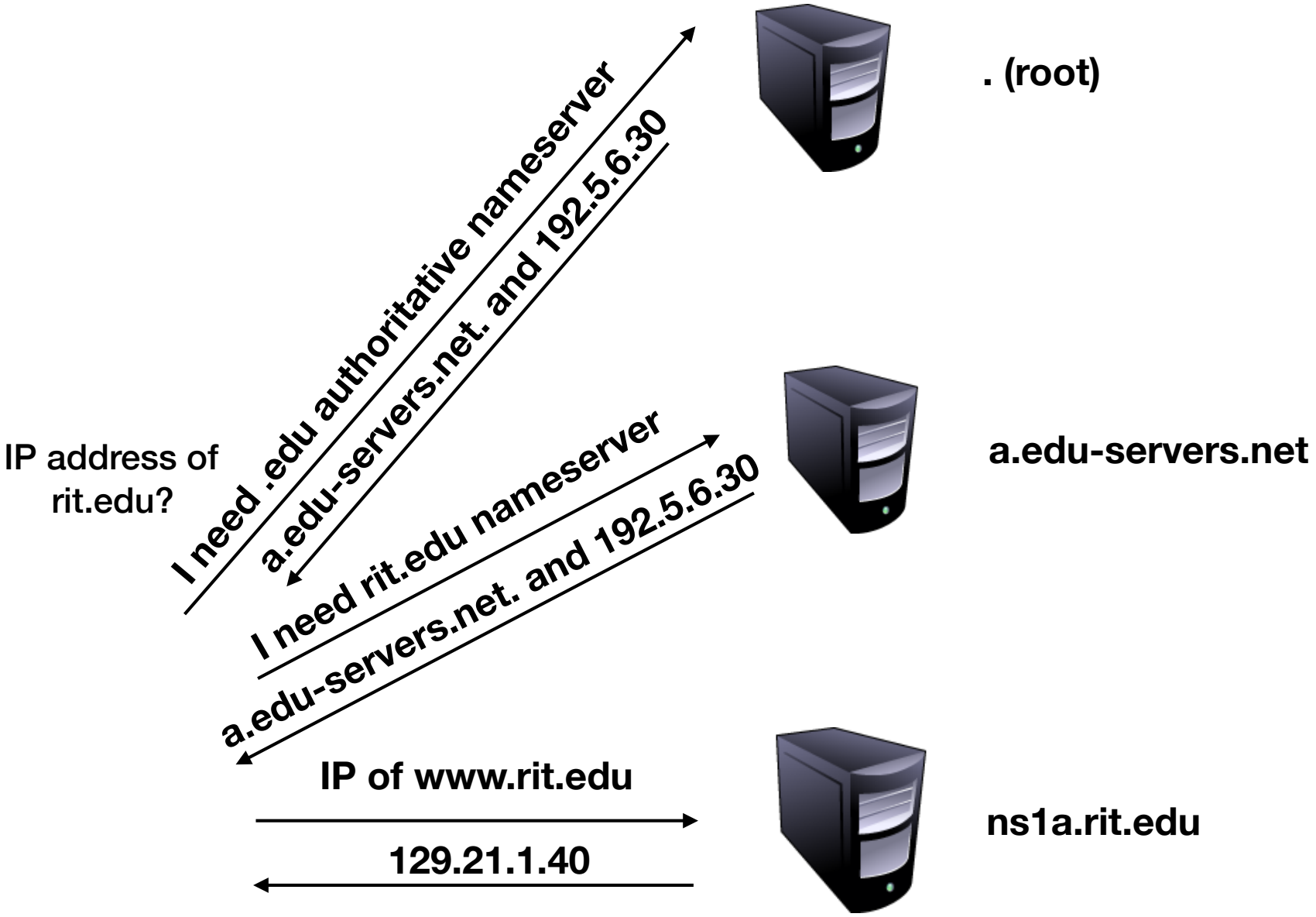


- Top Level Domains (TLDs) are at the top
- Maximum tree depth: 128
- Each Domain Name is a subtree
  - .edu → rit.edu → cs.rit.edu → www.ccs.rit.edu
- Name collisions are avoided
  - rit.edu vs. rit.com

Authoritative Nameserver



Authoritative Nameserver



Demo 1

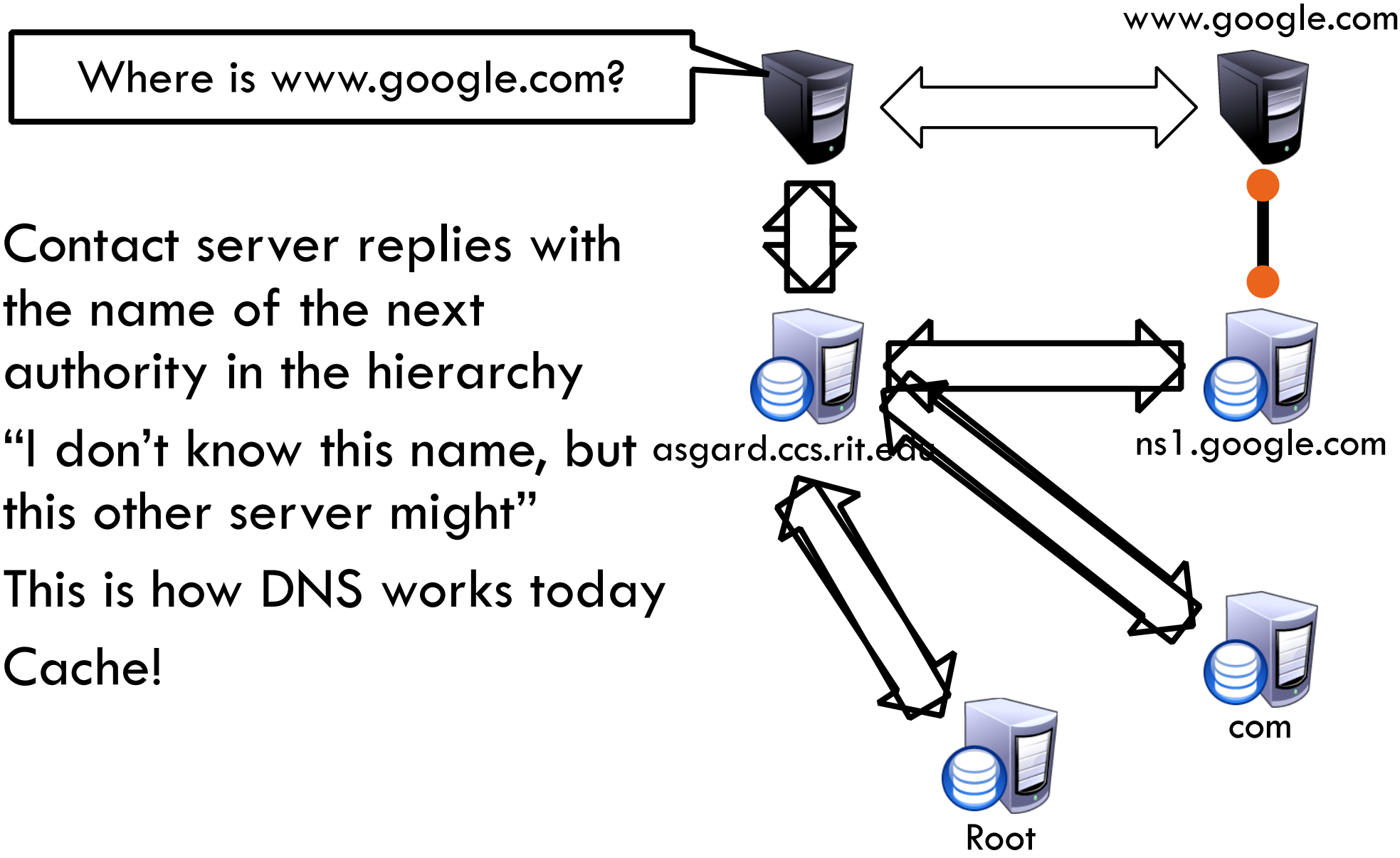
- Dig: (Domain Information Grouper)
  - Very useful tool to send a DNS request and parse the DNS response

List of Root Servers

HOSTNAME	IP ADDRESSES	MANAGER
a.root-servers.net	198.41.0.4, 2001:503:ba3e::2:30	VeriSign, Inc.
b.root-servers.net	199.9.14.201, 2001:500:200::b	University of Southern California (ISI)
c.root-servers.net	192.33.4.12, 2001:500:2::c	Cogent Communications
d.root-servers.net	199.7.91.13, 2001:500:2d::d	University of Maryland
e.root-servers.net	192.203.230.10, 2001:500:a8::e	NASA (Ames Research Center)
f.root-servers.net	192.5.5.241, 2001:500:2f::f	Internet Systems Consortium, Inc.
g.root-servers.net	192.112.36.4, 2001:500:12::d0d	US Department of Defense (NIC)
h.root-servers.net	198.97.190.53, 2001:500:1::53	US Army (Research Lab)
i.root-servers.net	192.36.148.17, 2001:7fe::53	Netnod
j.root-servers.net	192.58.128.30, 2001:503:c27::2:30	VeriSign, Inc.
k.root-servers.net	193.0.14.129, 2001:7fd::1	RIPE NCC
l.root-servers.net	199.7.83.42, 2001:500:9f::42	ICANN
m.root-servers.net	202.12.27.33, 2001:dc3::35	WIDE Project

Resolver

- Contact server replies with the name of the next authority in the hierarchy
- "I don't know this name, but this other server might"
- This is how DNS works today
- Cache!



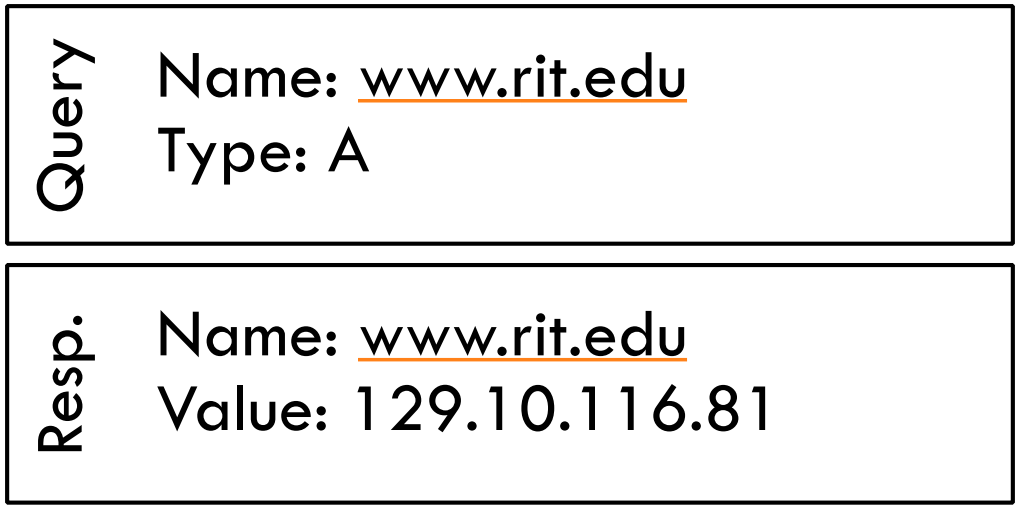
Demo 2

- Dig: (Domain Information Grouper)
  - Dig @1.1.1.1 rit.edu

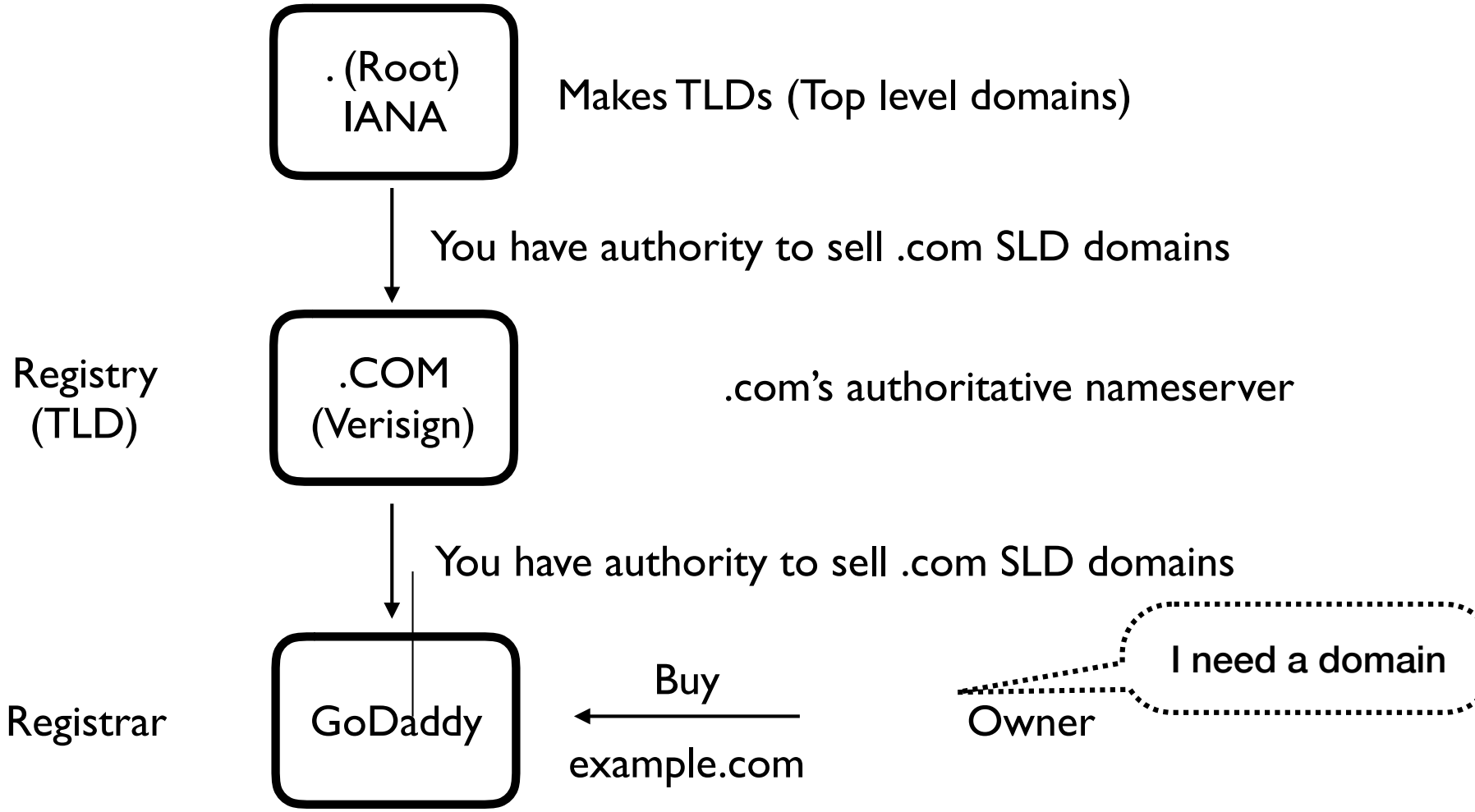
DNS Types

- Type = NS
  - Name = partial domain
  - Value = name of DNS server for this domain
  - "Go send your query to this other server"

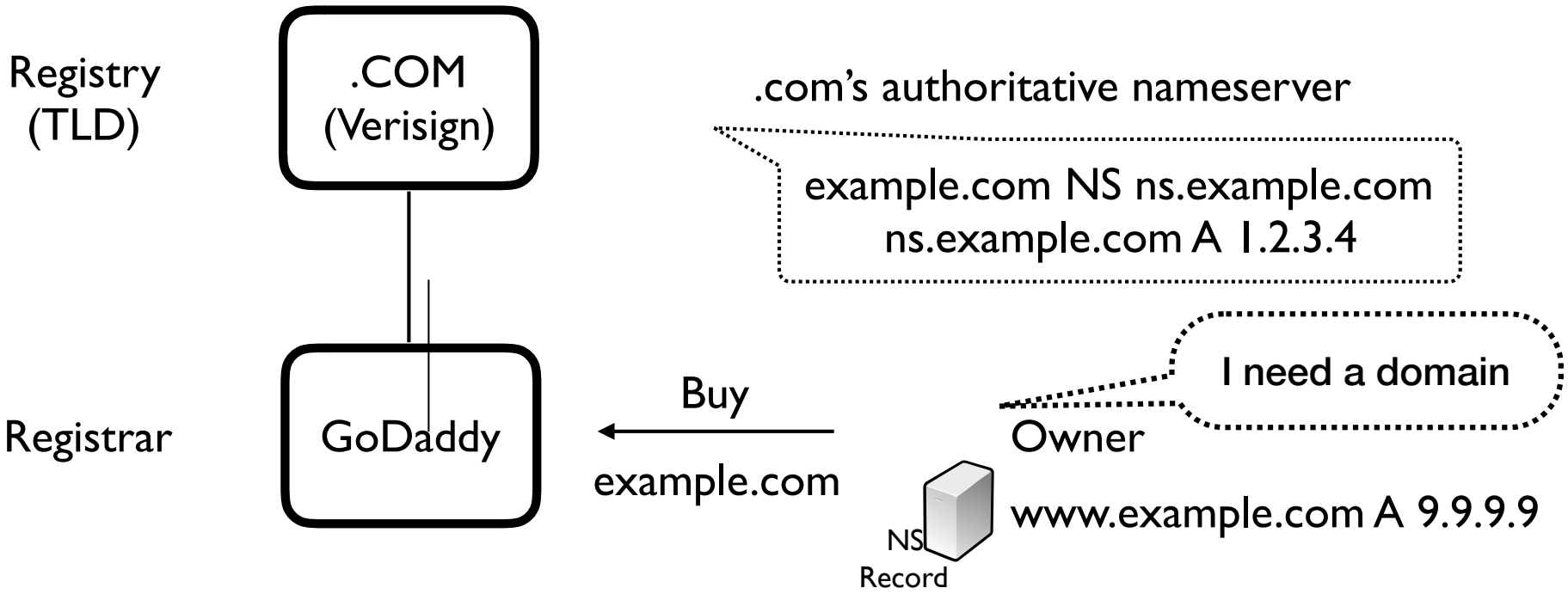
- Type = A / AAAA
  - Name = domain name
  - Value = IP address
  - A is IPv4, AAAA is IPv6



How to buy a domain name (1)

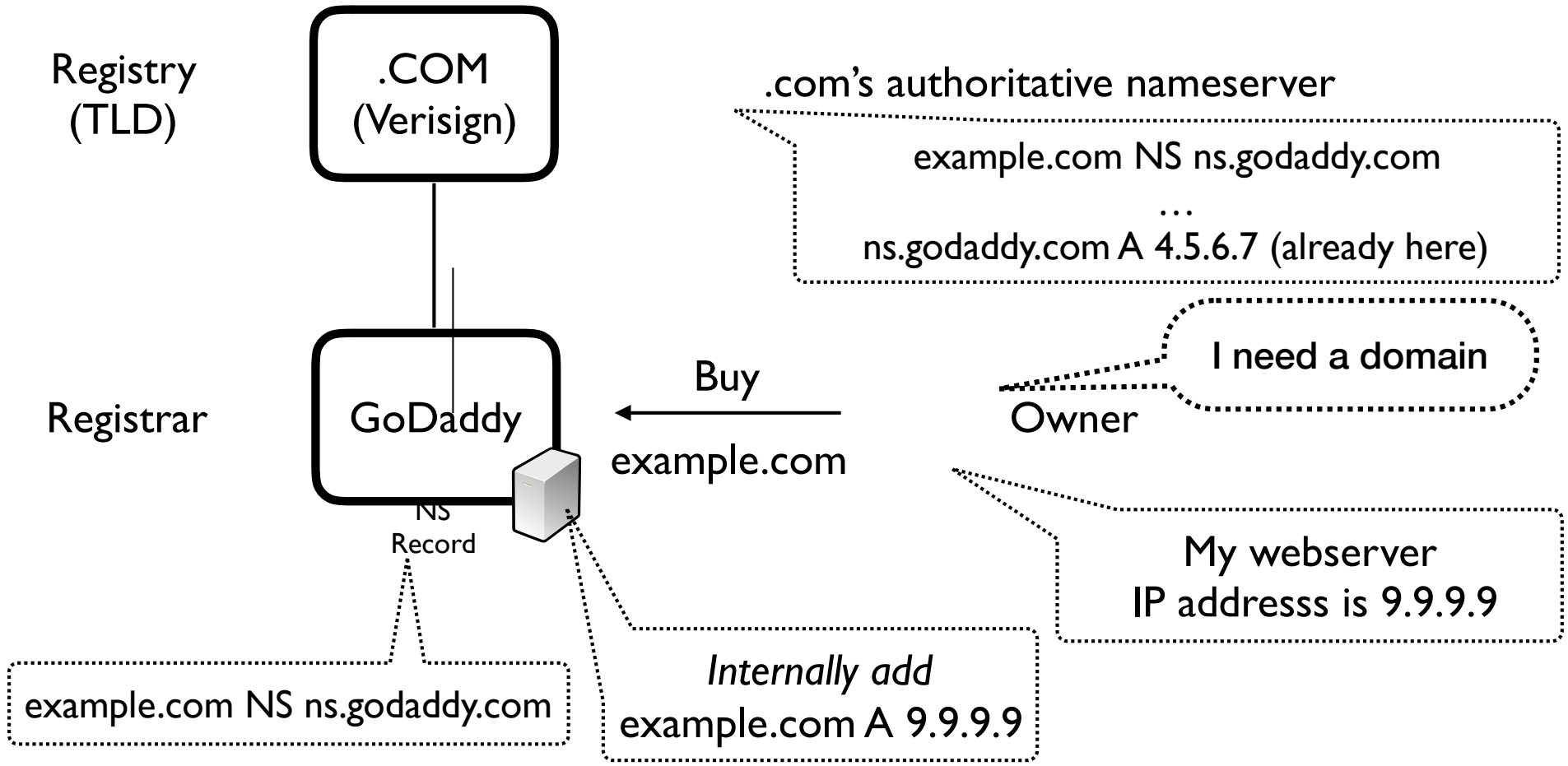


How to buy a domain name (2)
Using your own authoritative nameserver



My nameserver name is
ns.example.com and
this is the IP addresss: 1.2.3.4.

How to buy a domain name (3)
Using the registrar's default nameserver



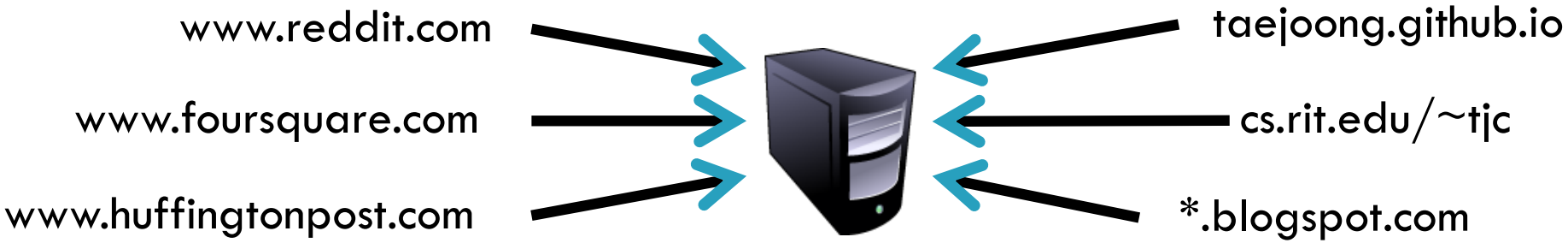
DNS as Indirection Service

- DNS gives us very powerful capabilities
  - Not only easier for humans to reference machines!

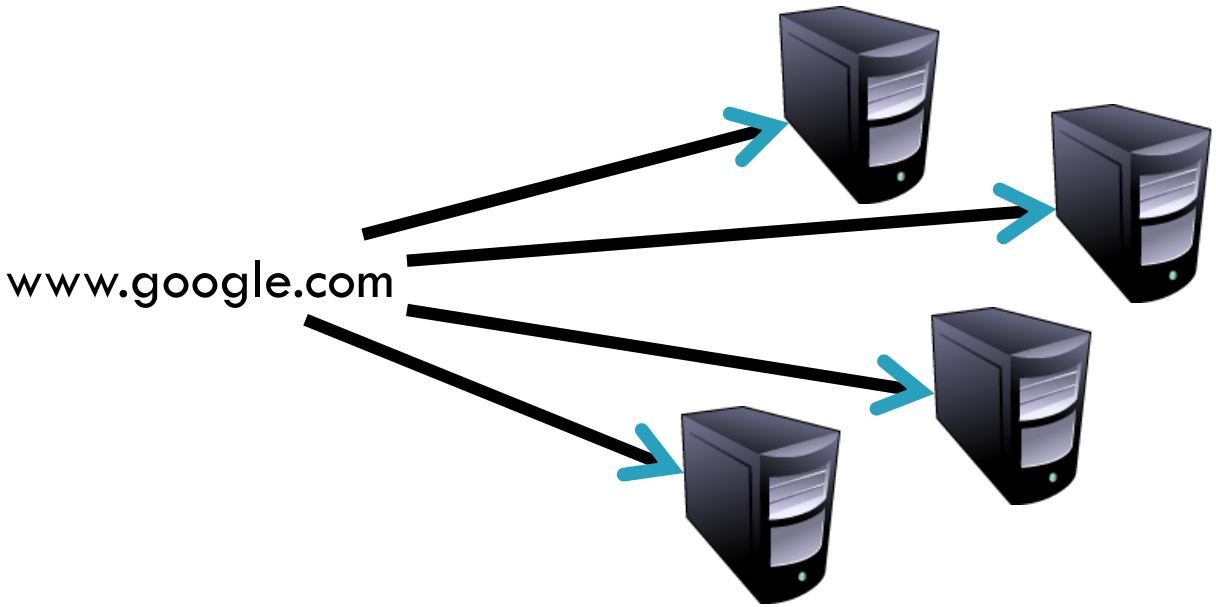
- Changing the IPs of machines becomes trivial
  - e.g. you want to move your web server to a new host
  - Just change the DNS record!

Aliasing and Load Balancing

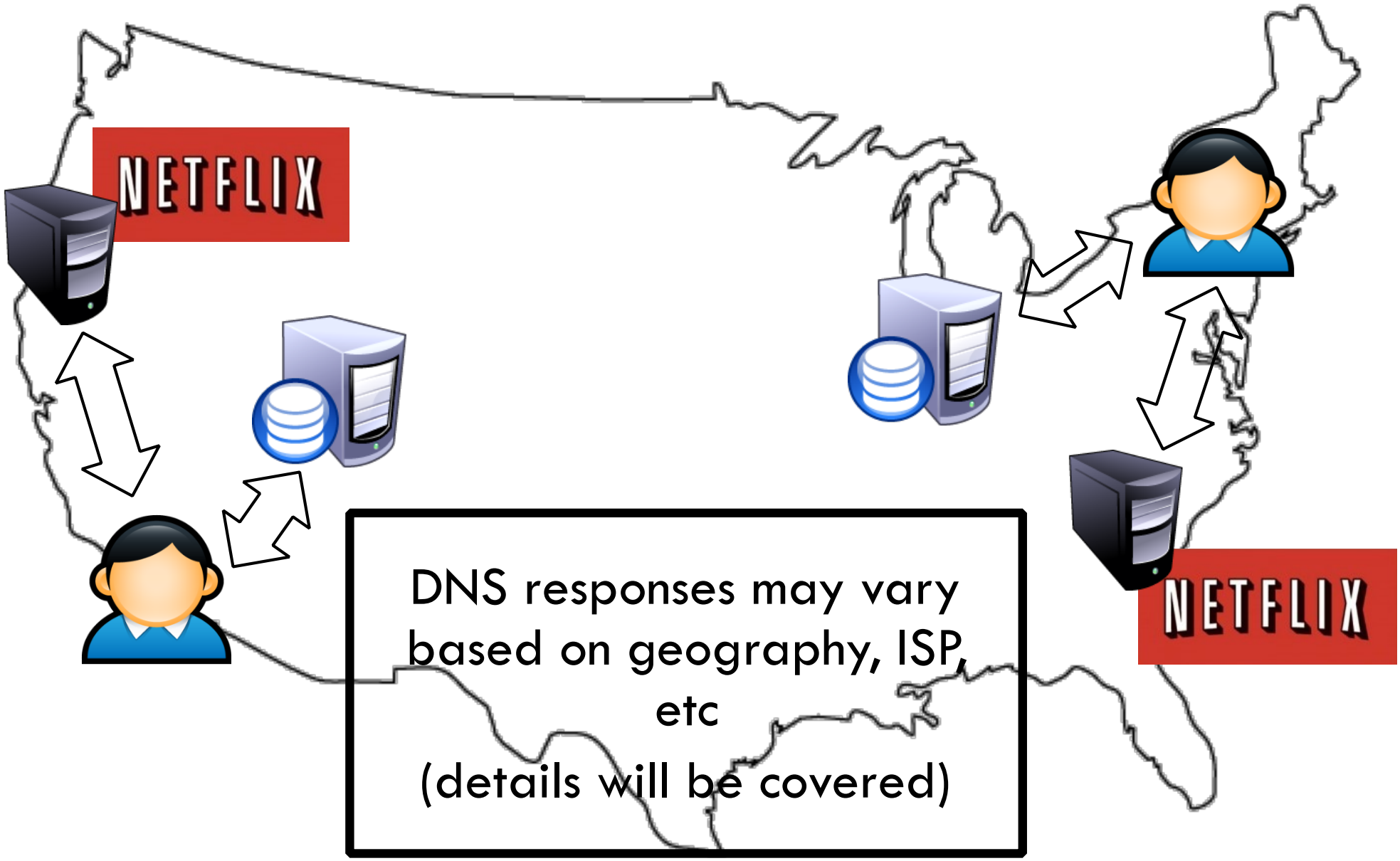
- One machine can have many aliases (virtual hosting)



- One domain can map to multiple machines



Content Delivery Networks



Outline

- DNS Basics
- DNS Security

The Importance of DNS

- Without DNS…
  - How could you get to any websites?
- You are your mailserver
  - When you sign up for websites, you use your email address
  - What if someone hijacks the DNS for your mail server?
- DNS is the root of trust for the web
  - When a user types www.bankofamerica.com, they expect to be taken to their bank's website
  - What if the DNS record is compromised?

Denial Of Service

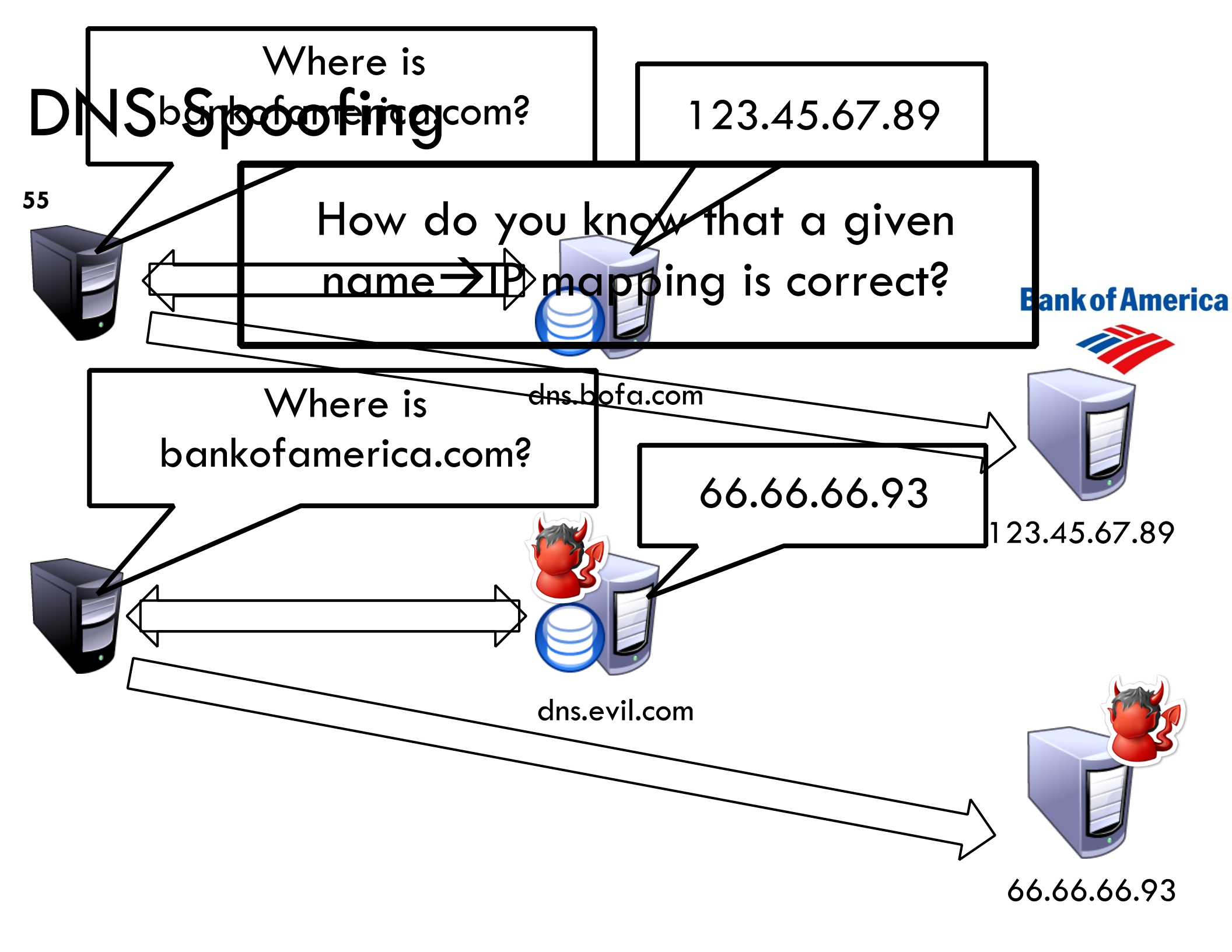
- Flood DNS servers with requests until they fail
- October 2002: massive DDoS against the root name servers
  - What was the effect?
  - … users didn't even notice
  - Root zone file is cached almost everywhere
- More targeted attacks can be effective
  - Local DNS server → cannot access DNS
  - Authoritative server → cannot access domain

DNS Hijacking

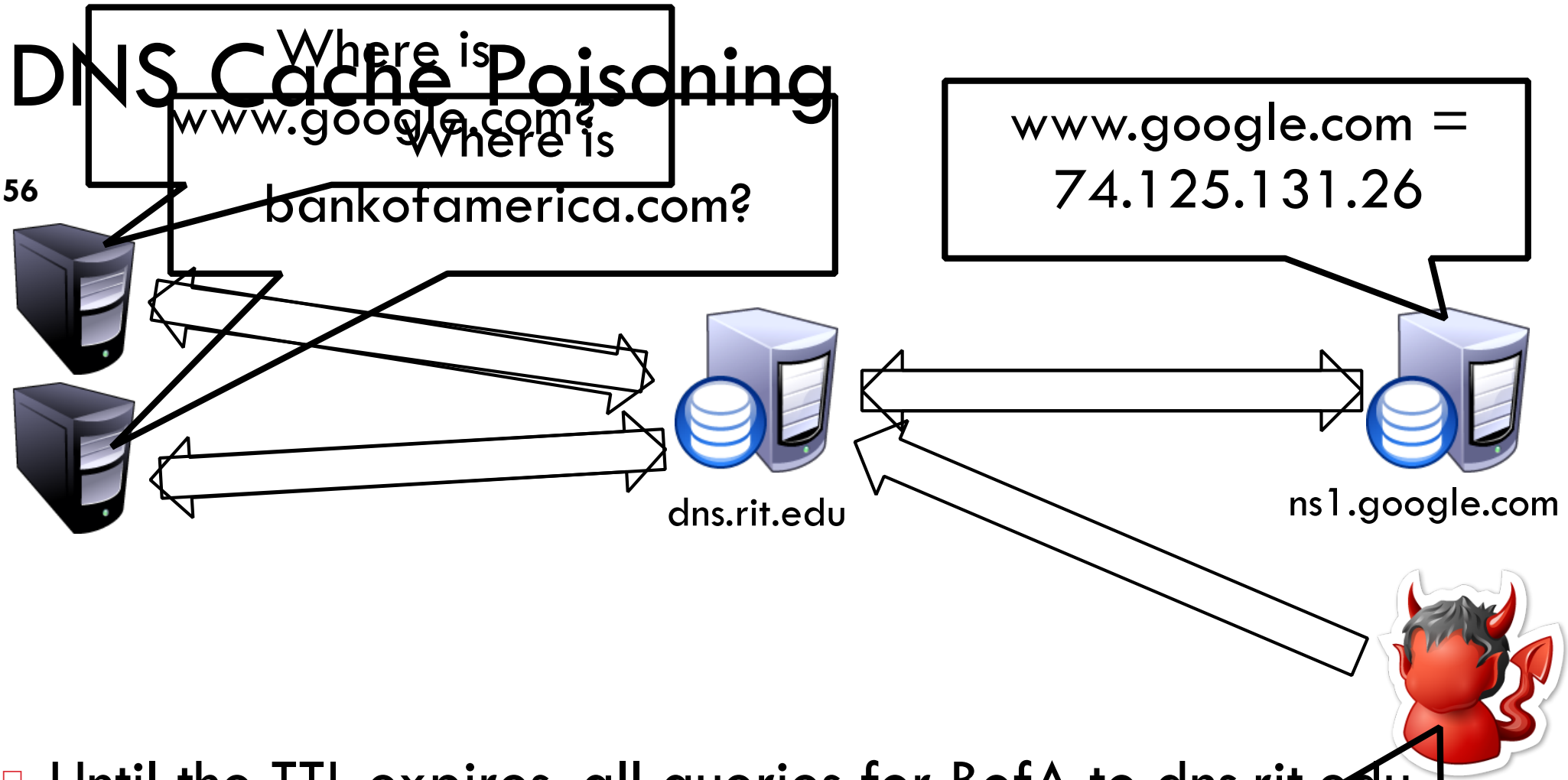
- Infect their OS or browser with a virus/trojan
  - e.g. Many trojans change entries in /etc/hosts
  - *.bankofamerica.com → evilbank.com
- Man-in-the-middle

- Response Spoofing
  - Eavesdrop on requests
  - Outrace the servers response

DNS Spoofing

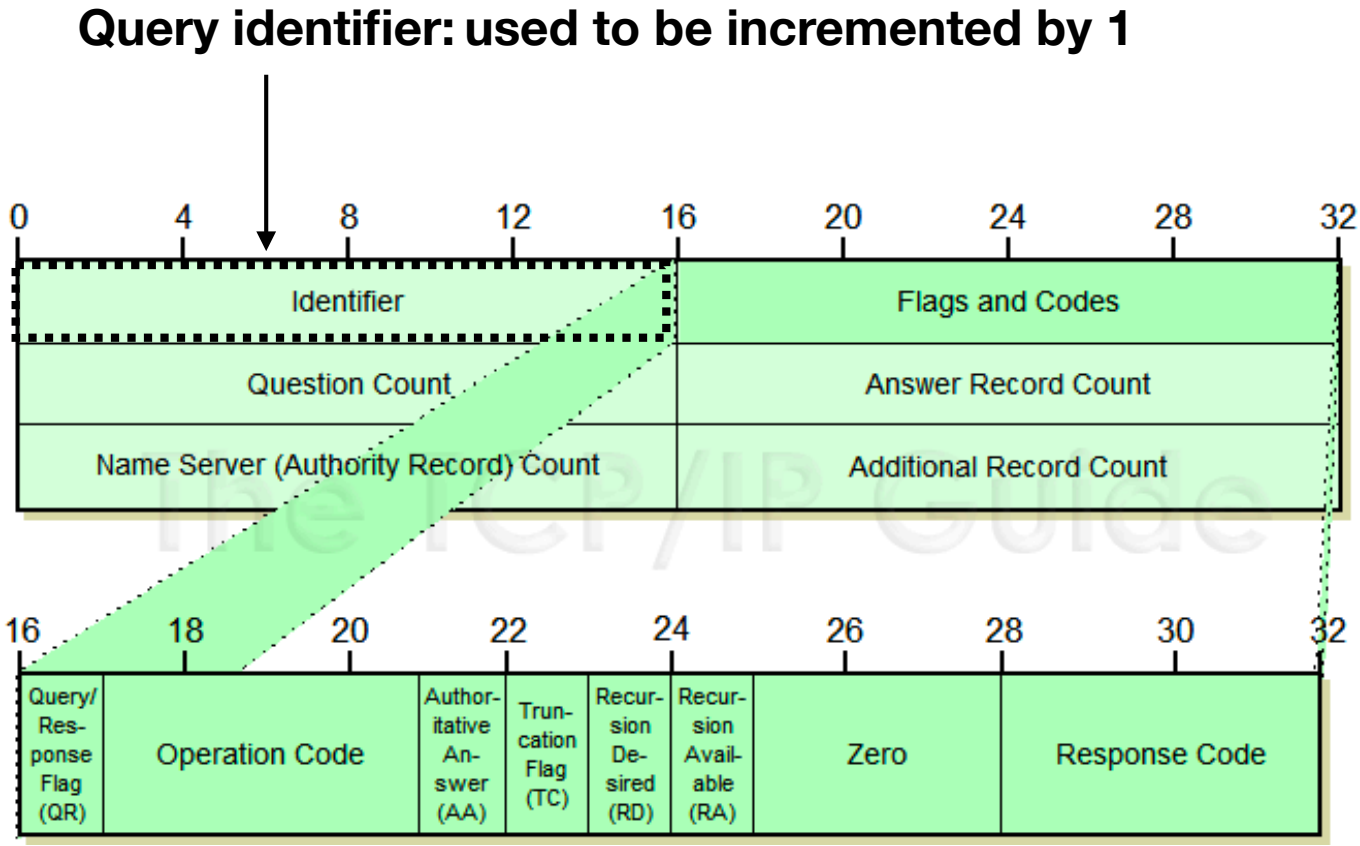


DNS Cache Poisoning



- Until the TTL expires, all queries for BofA to dns.rit.edu will return poisoned result
- Much worse than spoofing/man-in-the-middle
  - Whole ISPs can be impacted!

DNS Header


Query identifier: used to be incremented by 1
0
4
8
12
16
20
24
28
32
Identifier
Flags and Codes
Question Count
Answer Record Count
Name Server (Authority Record) Count
Additional Record Count
16
18
20
22
24
26
28
30
32
Query/ Response Flag (QR)
Operation Code
Authoritative Answer (AA)
Truncation Flag (TC)
Recursion Desired (RD)
Recursion Available (RA)
Zero
Response Code

Attacking DNS (only few examples)

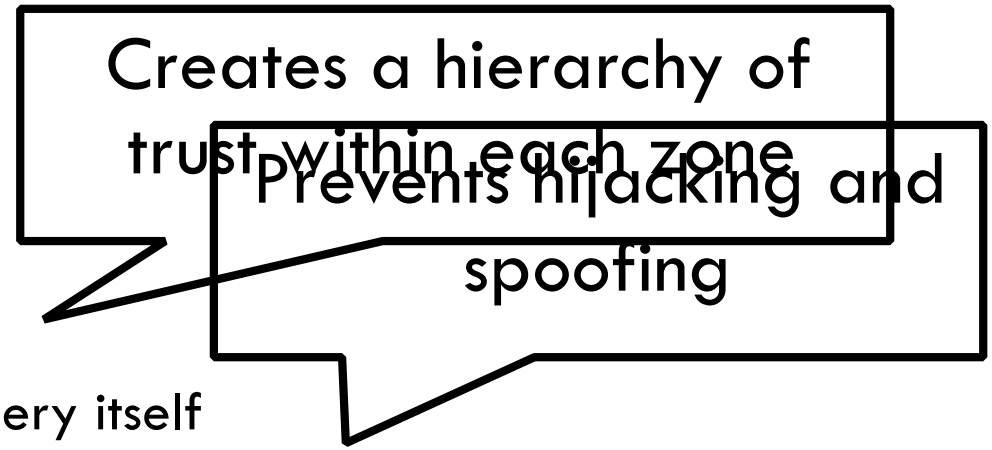
Kaminsky Attack
(QID bruteforcing)

Random QID and
Random Port

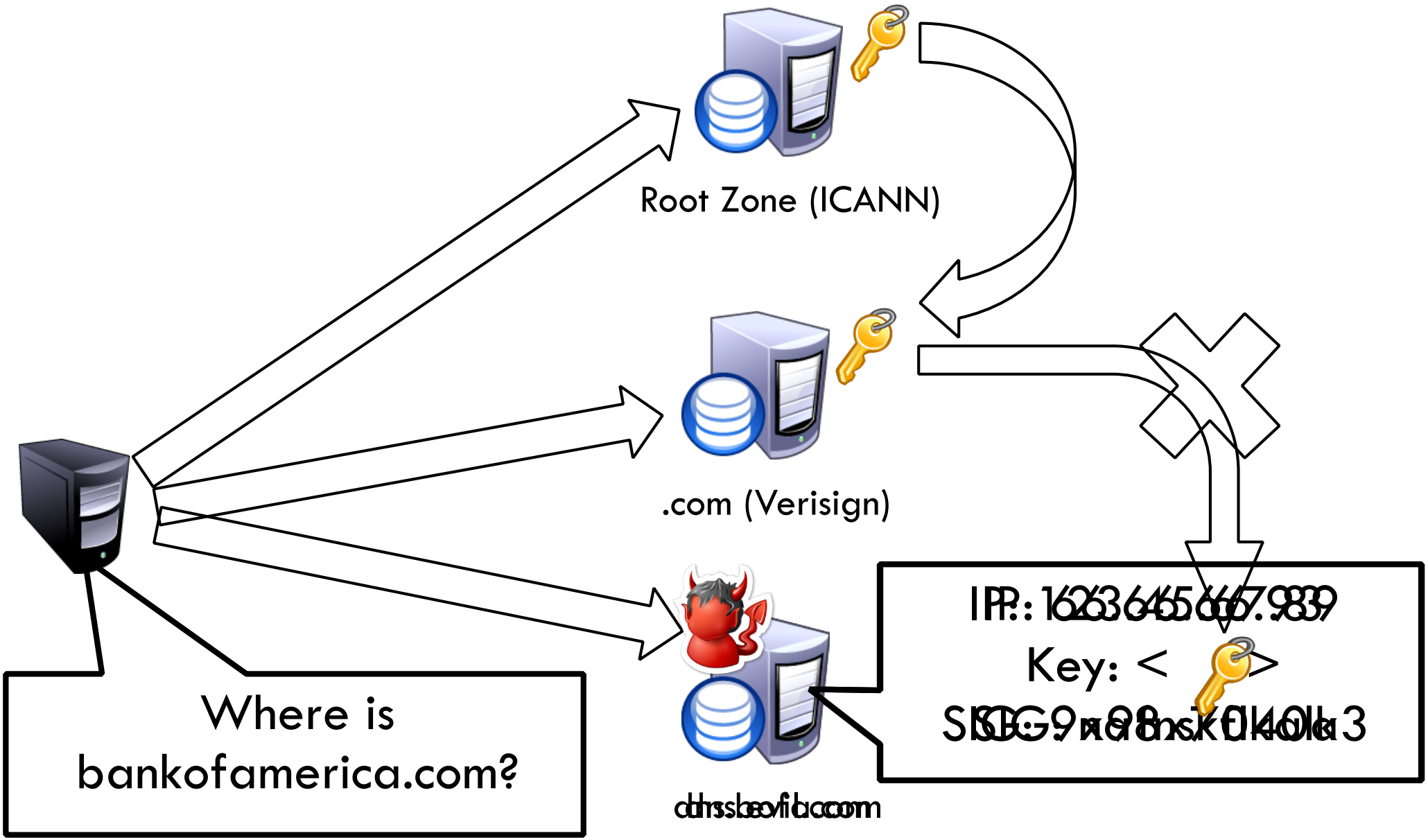
Man-in-the-Middle

Solution: DNSSEC

- Cryptographically sign critical resource records
  - Resolver can verify the cryptographic signature
- Two new resource types
  - Type = DNSKEY
    - Name = Zone domain name
    - Value = Public key for the zone
  - Type = RRSIG
    - Name = (type, name) tuple, i.e. the query itself
    - Value = Cryptographic signature of the query results
- Deployment
  - On the roots since July 2010
  - Verisign enabled it on .com and .net in January 2011
  - Comcast is the first major ISP to support it (January 2012)

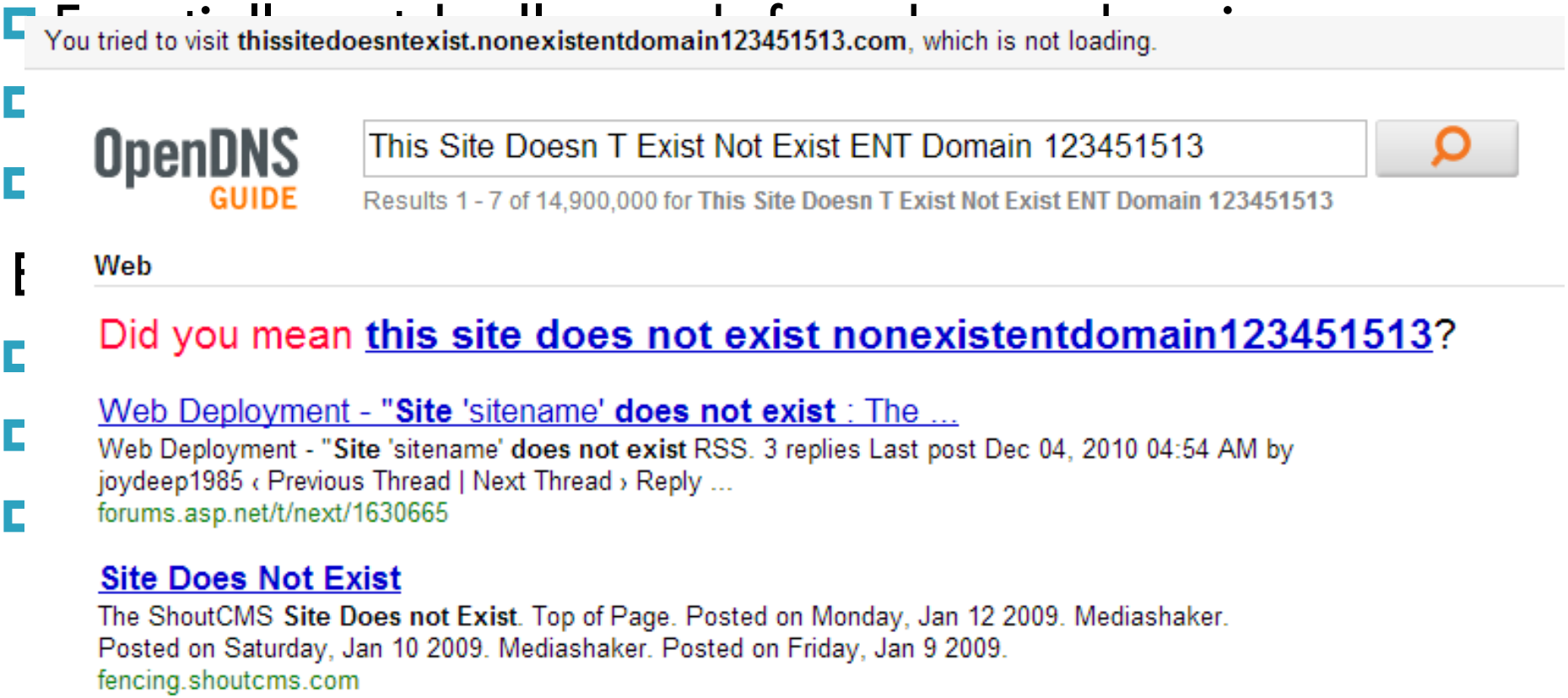


DNSSEC Hierarchy of Trust



Site Finder

- September 2003: Verisign created DNS wildcards for *.com and *.net
- [illegible]



Much More to DNS

- Caching: when, where, how much, etc.
- Other uses for DNS (i.e. DNS hacks)
  - Content Delivery Networks (CDNs)
  - Different types of DNS load balancing
  - Dynamic DNS (e.g. for mobile hosts)
- DNS and botnets
- Politics and growth of the DNS system
  - Governance
  - New TLDs (.xxx, .biz), eliminating TLDs altogether
  - Copyright, arbitration, squatting, typo-squatting