CS 3700
Networks and Distributed Systems

Lecture 11: DNS + NAT

Revised 3/10/14

Outline

- DNS
- NAT
- Other middleboxes

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 M I S L O V E”
  - Mail someone, you need to get their address first

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 M I S L O V E"
  - 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?

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 M I S L O V E"
  - 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

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.northeastern.edu
  - Human usable labels for machines
  - Conform to organizational structure

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.northeastern.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

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
    - alans_server_at_neu_pwns_joo_lol_kthxbye

Towards DNS

- Eventually, the *hosts.txt* system fell apart

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 MIT
      - Massachusetts Institute of Technology?
      - Melbourne Institute of Technology?
  - Many machines had inaccurate copies of *hosts.txt*

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 MIT
      - Massachusetts Institute of Technology?
      - Melbourne Institute of Technology?
  - Many machines had inaccurate copies of *hosts.txt*
- Thus, DNS was born

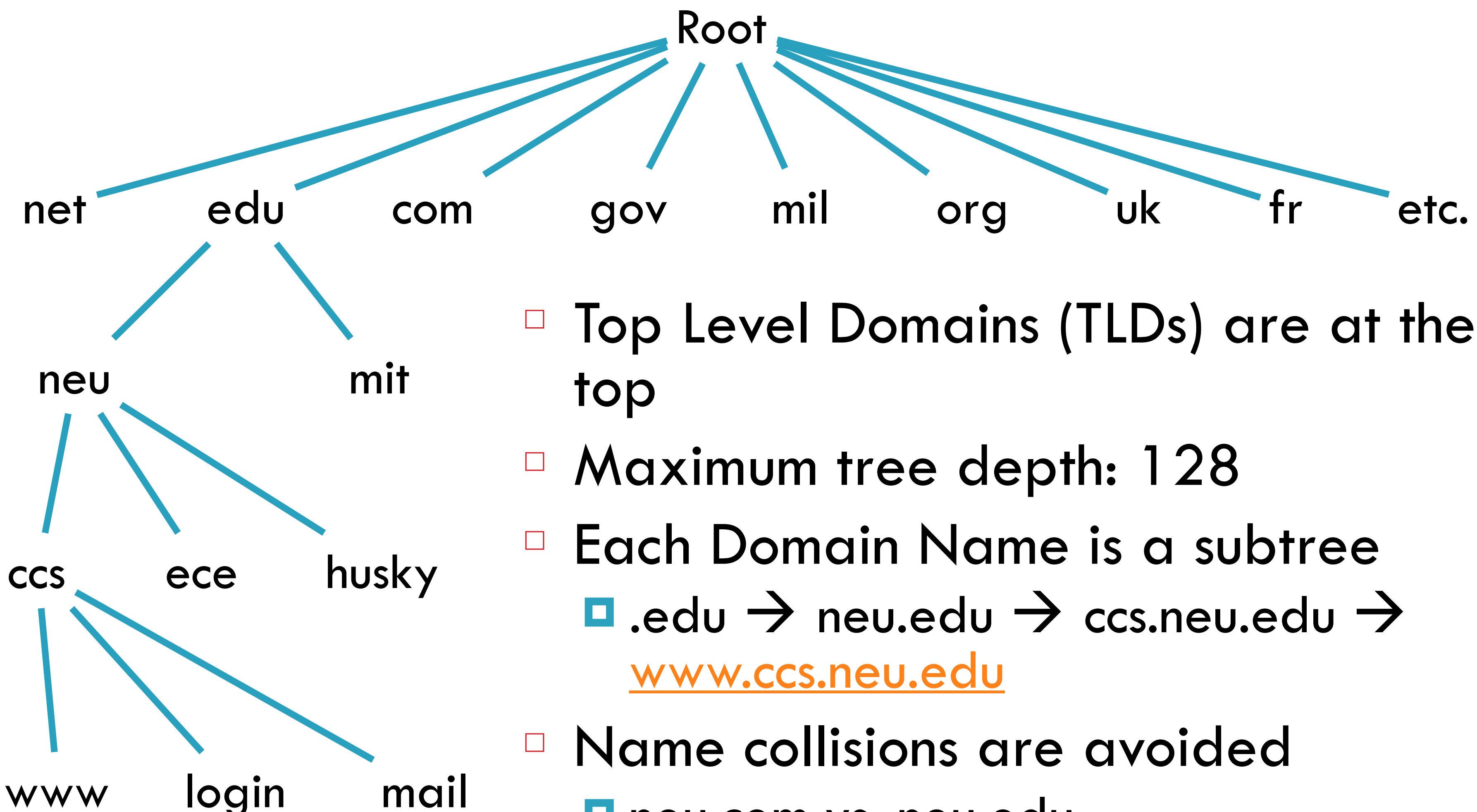
Outline

- 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?
- Hierarchical namespace
  - As opposed to original, flat namespace
  - e.g. .com → google.com → mail.google.com

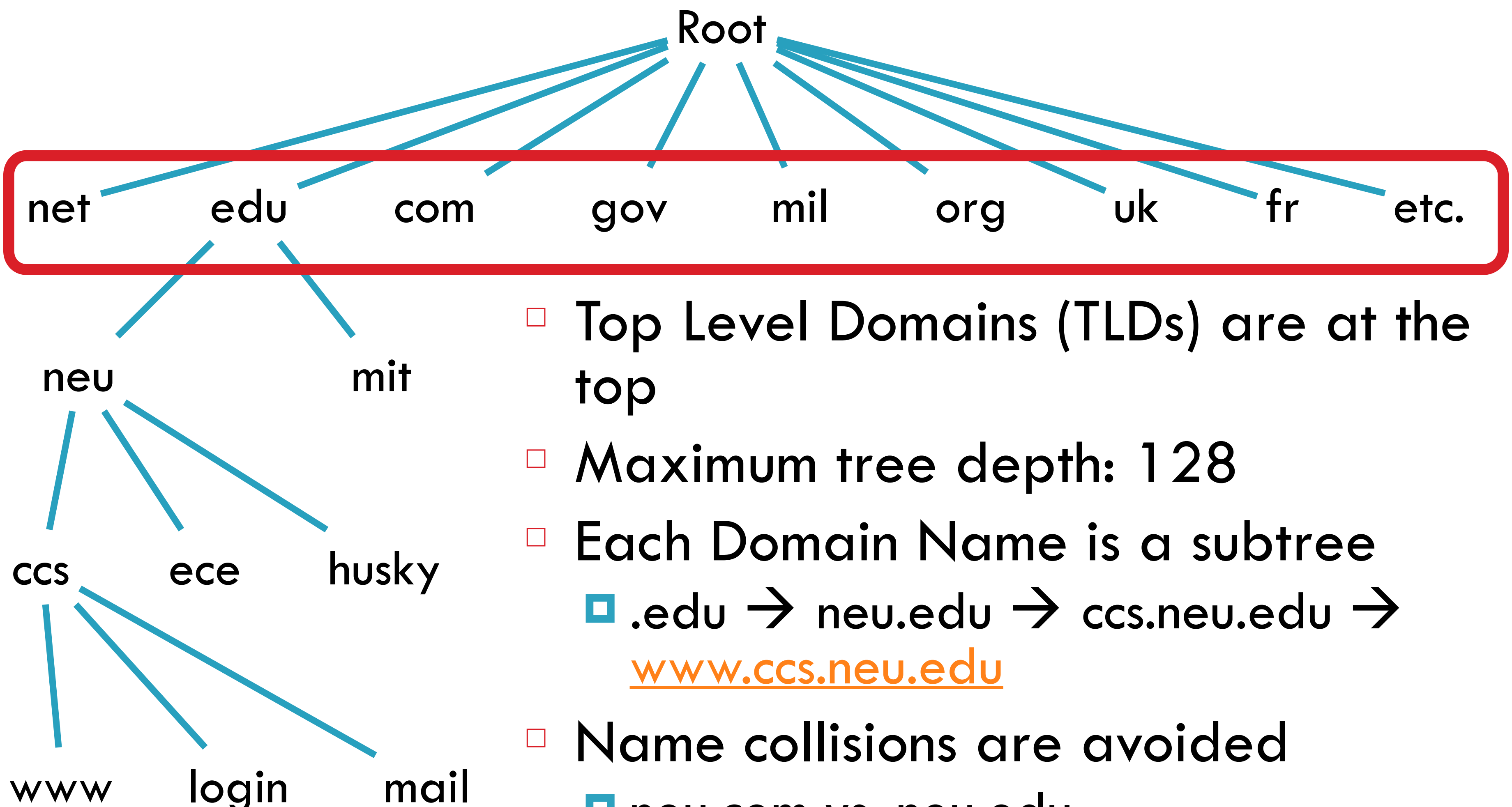
Naming Hierarchy



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

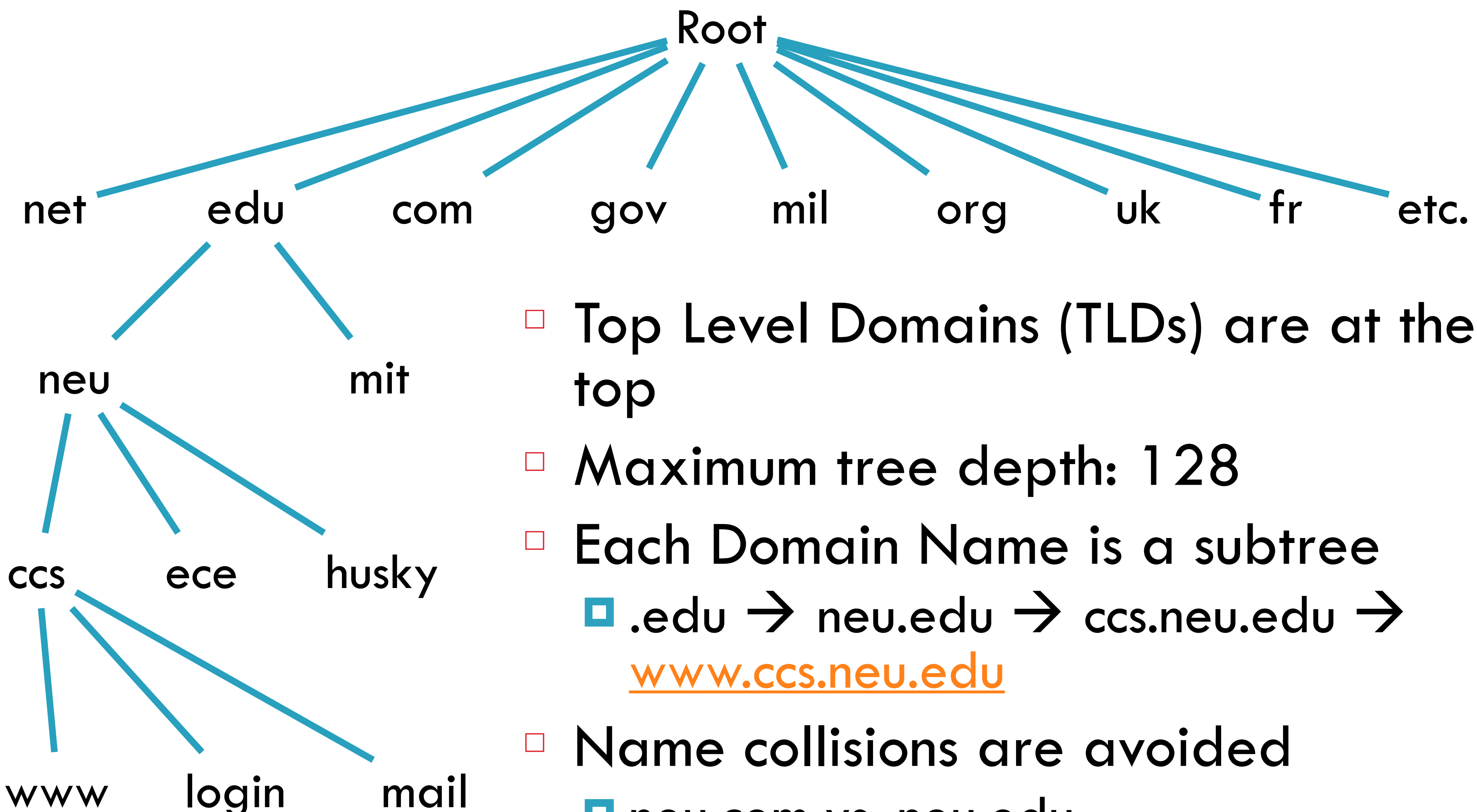
Naming Hierarchy

9



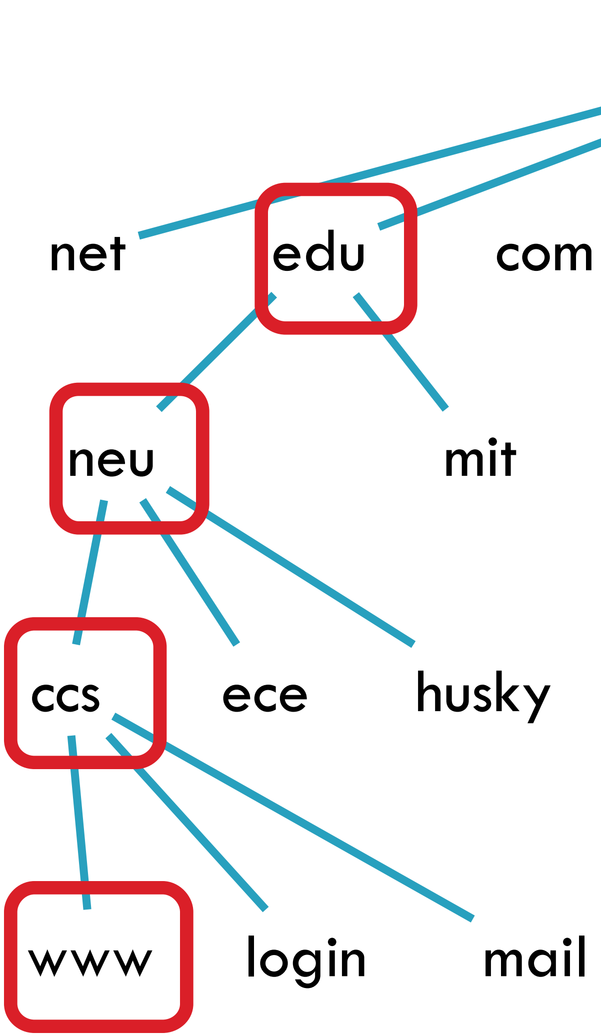
- Top Level Domains (TLDs) are at the top
- Maximum tree depth: 128
- Each Domain Name is a subtree
  - .edu → neu.edu → ccs.neu.edu → www.ccs.neu.edu
- Name collisions are avoided
  - neu.com vs. neu.edu

Naming Hierarchy



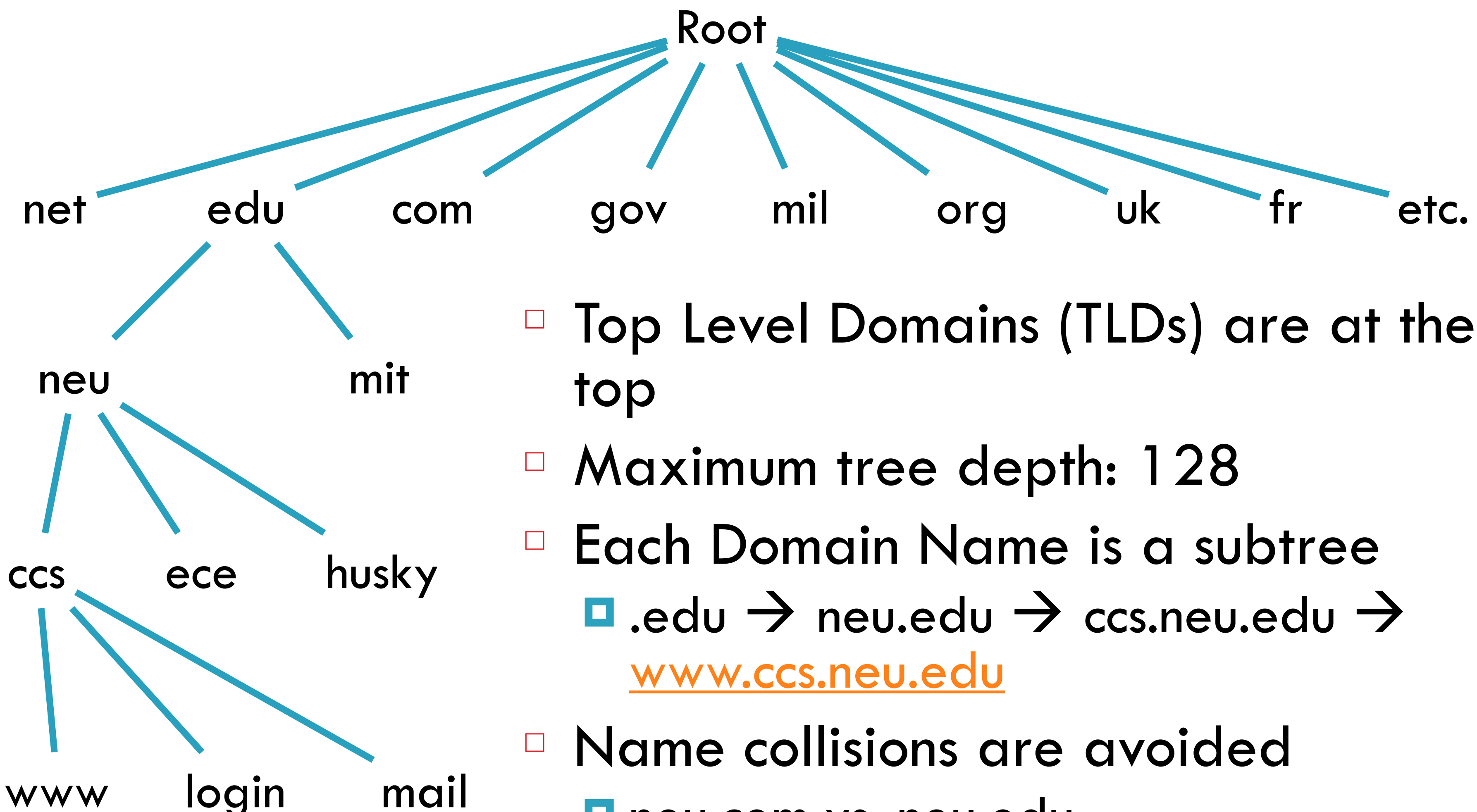
- Top Level Domains (TLDs) are at the top
- Maximum tree depth: 128
- Each Domain Name is a subtree
  - .edu → neu.edu → ccs.neu.edu → www.ccs.neu.edu
- Name collisions are avoided
  - neu.com vs. neu.edu

Naming Hierarchy



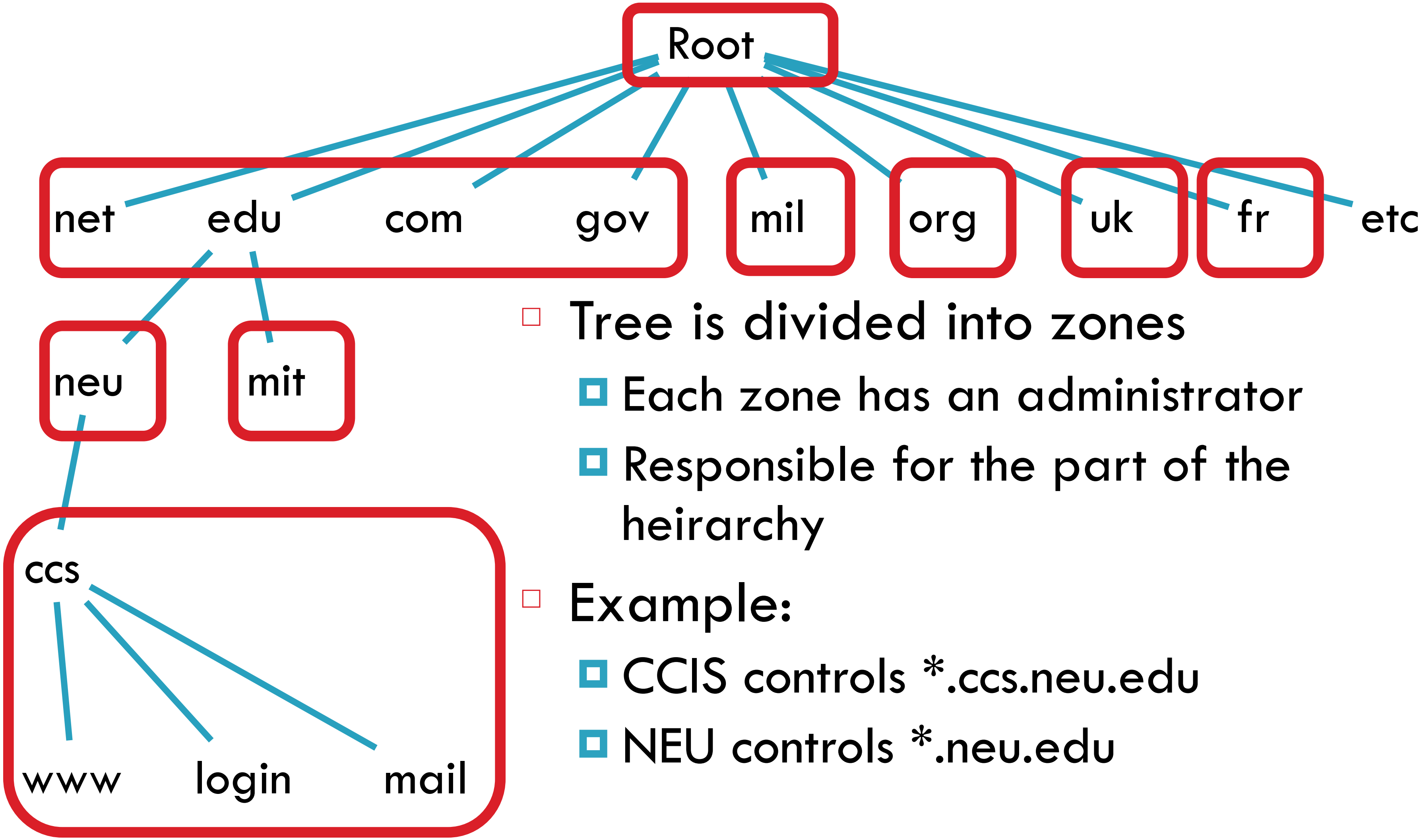
- Top Level Domains (TLDs) are at the top
- Maximum tree depth: 128
- Each Domain Name is a subtree
  - .edu → neu.edu → ccs.neu.edu → www.ccs.neu.edu
- Name collisions are avoided
  - neu.com vs. neu.edu

Naming Hierarchy



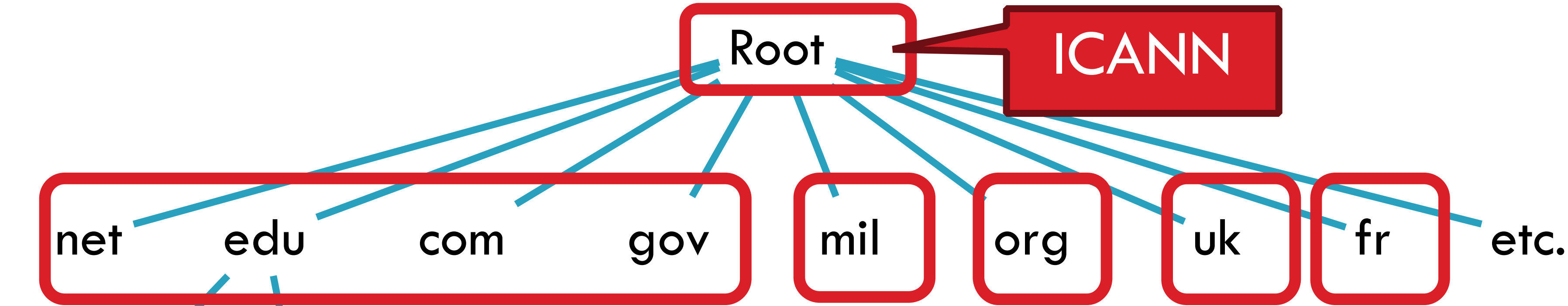
- Top Level Domains (TLDs) are at the top
- Maximum tree depth: 128
- Each Domain Name is a subtree
  - .edu → neu.edu → ccs.neu.edu → www.ccs.neu.edu
- Name collisions are avoided
  - neu.com vs. neu.edu

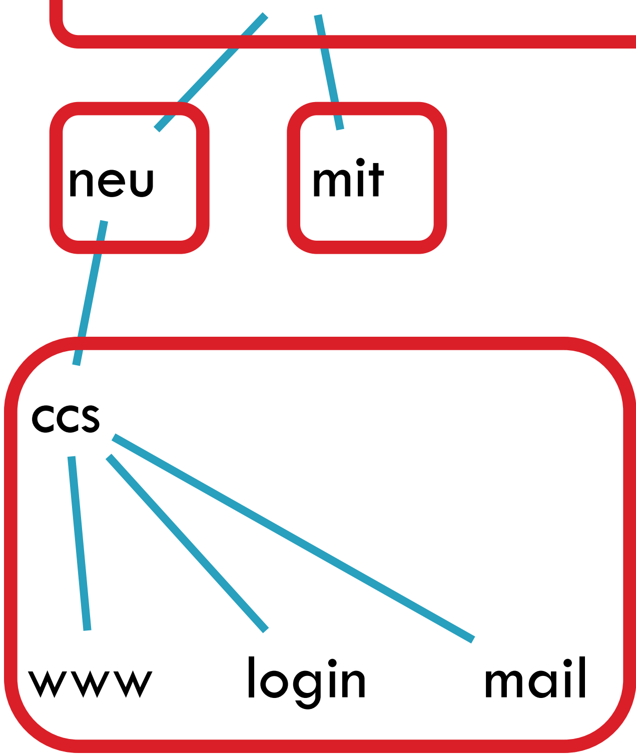
Hierarchical Administration



- Tree is divided into zones
  - Each zone has an administrator
  - Responsible for the part of the heirarchy
- Example:
  - CCIS controls *.ccs.neu.edu
  - NEU controls *.neu.edu

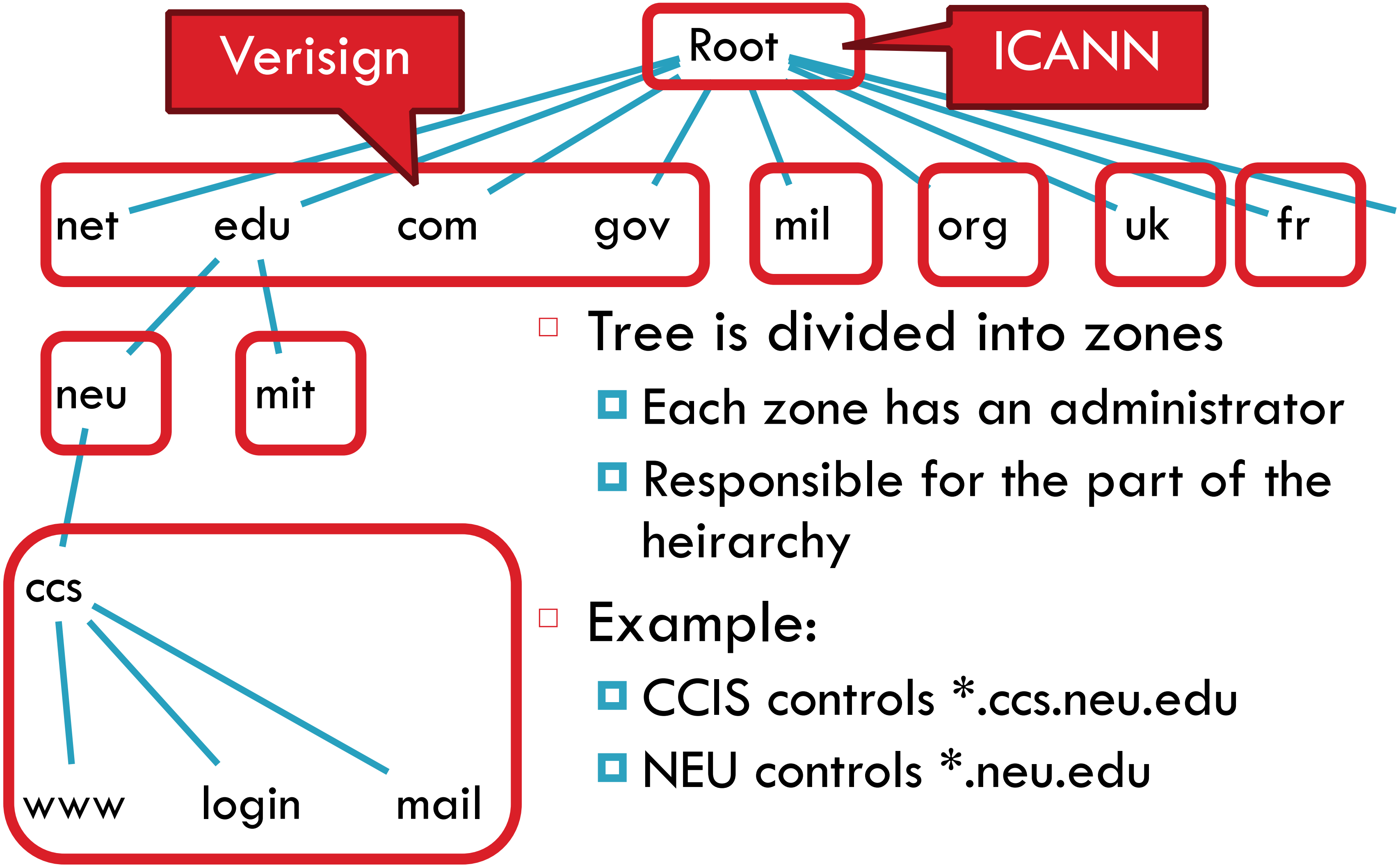
Hierarchical Administration





- Tree is divided into zones
  - Each zone has an administrator
  - Responsible for the part of the heirarchy
- Example:
  - CCIS controls *.ccs.neu.edu
  - NEU controls *.neu.edu

Hierarchical Administration



- Tree is divided into zones
  - Each zone has an administrator
  - Responsible for the part of the heirarchy
- Example:
  - CCIS controls *.ccs.neu.edu
  - NEU controls *.neu.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

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

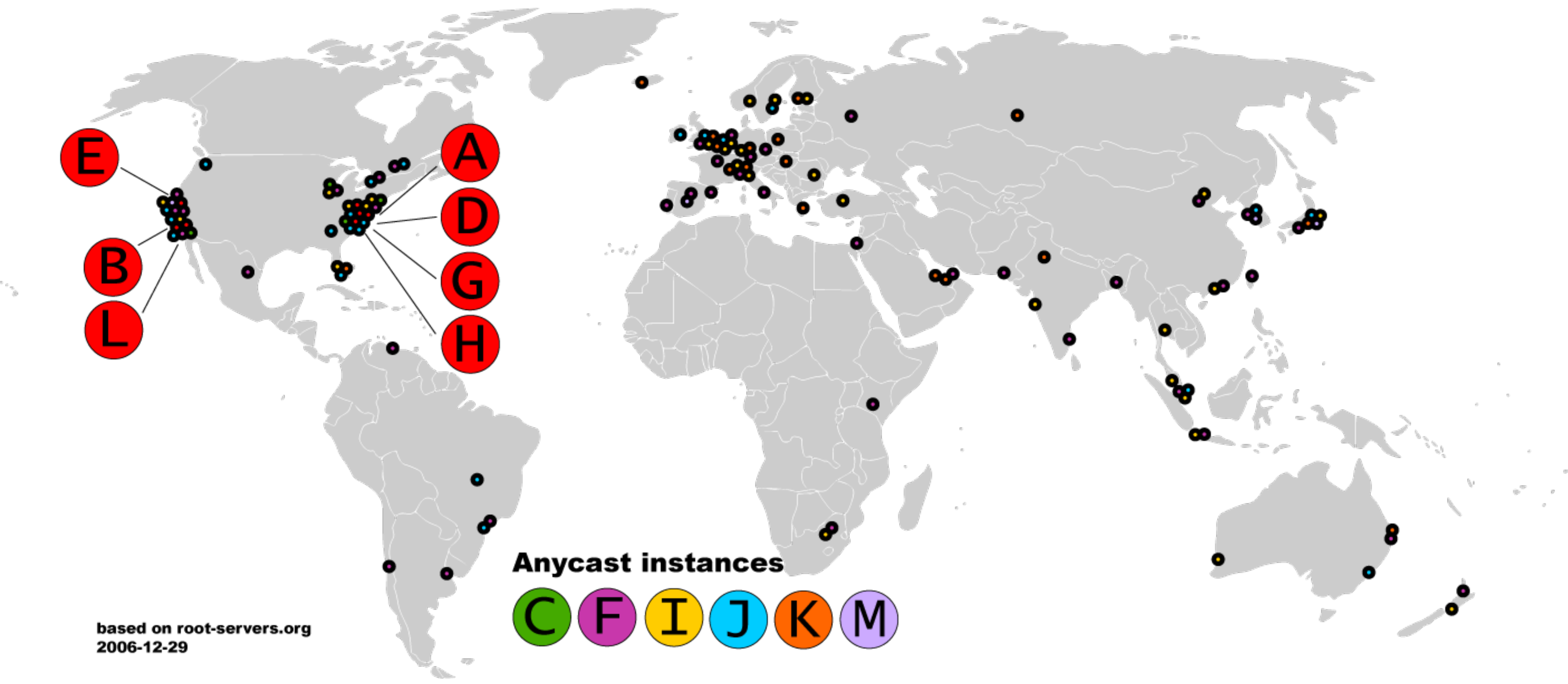
Root Name Servers

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

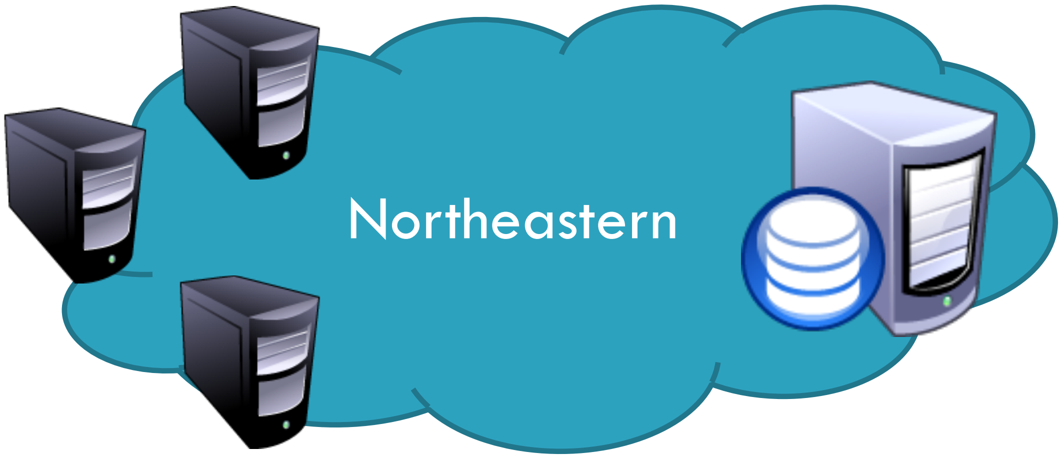
com.	172800	IN	NS	a.gtld-servers.net.
com.	172800	IN	NS	b.gtld-servers.net.
com.	172800	IN	NS	c.gtld-servers.net.

- 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

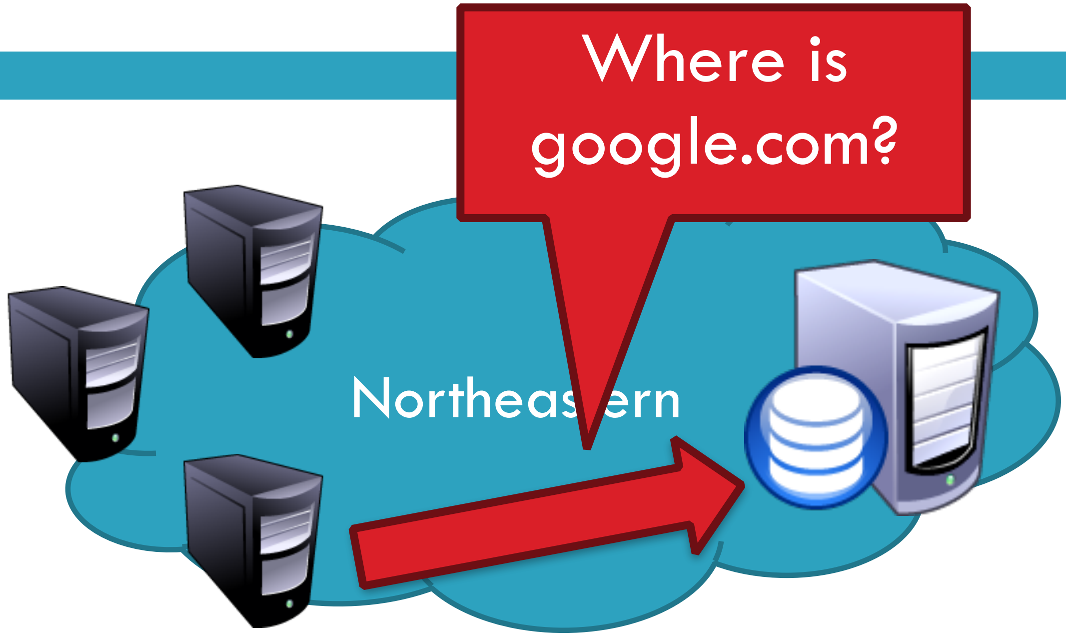


Local Name Servers



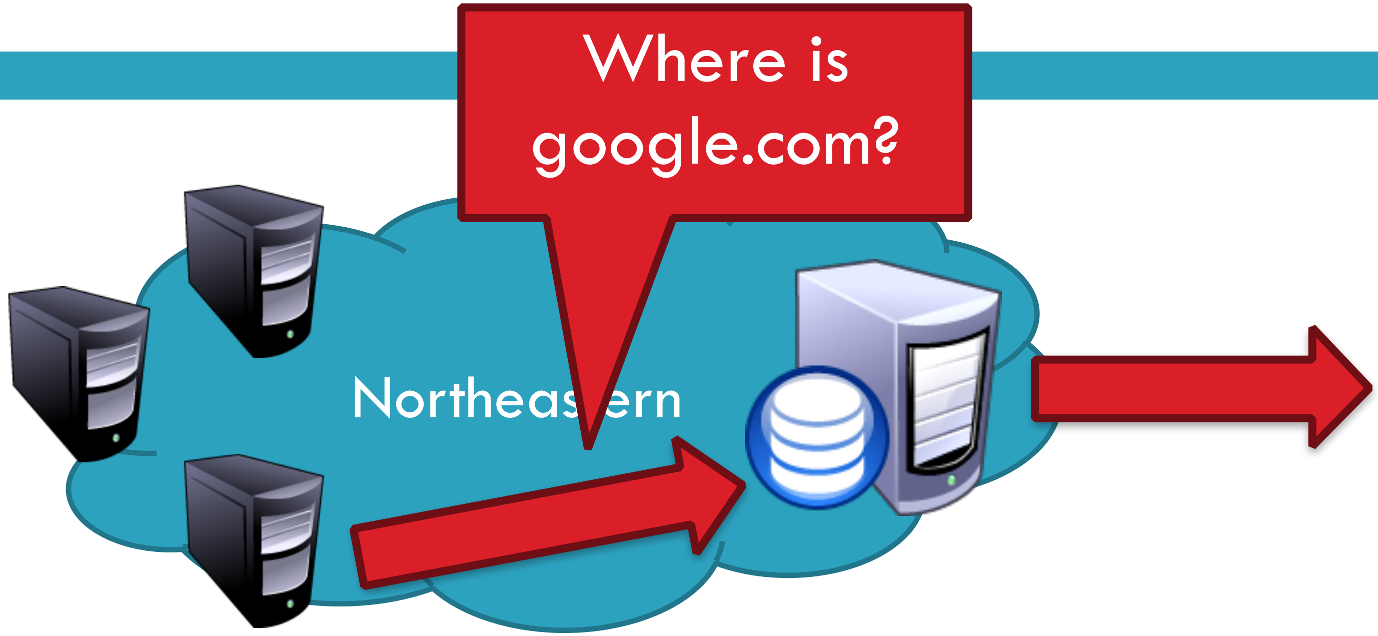
- Each ISP/company has a local, default name server
- Often configured via DHCP
- Hosts begin DNS queries by contacting the local name server
- Frequently cache query results

Local Name Servers



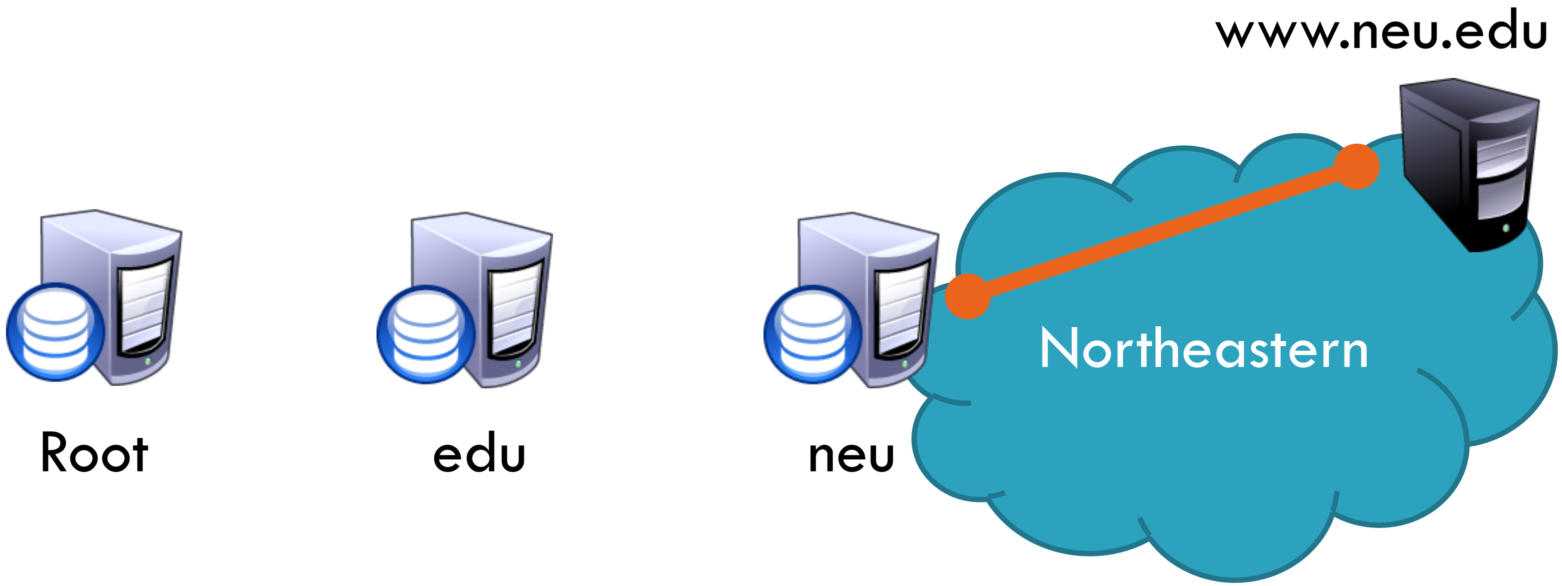
- Each ISP/company has a local, default name server
- Often configured via DHCP
- Hosts begin DNS queries by contacting the local name server
- Frequently cache query results

Local Name Servers



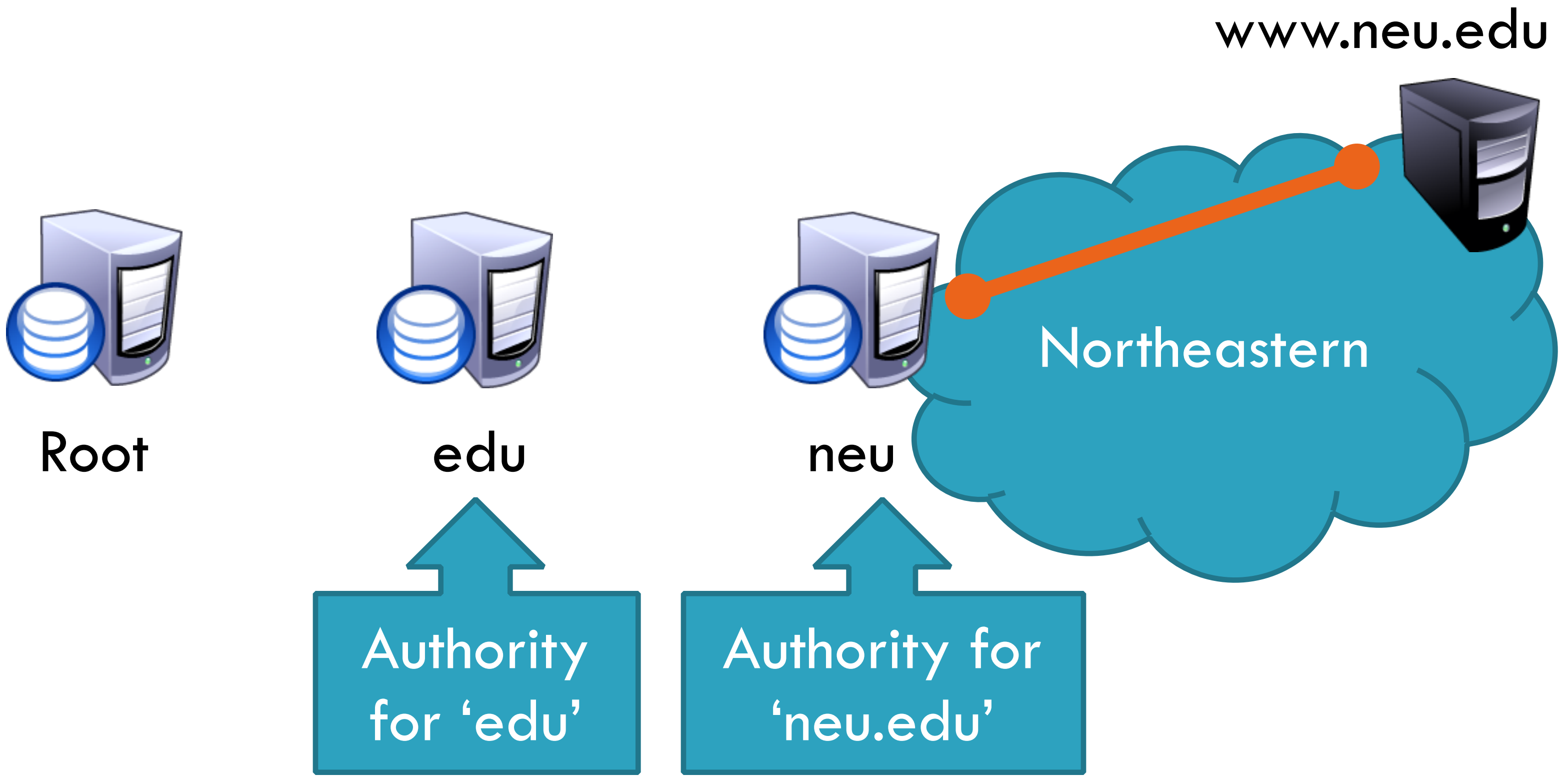
- Each ISP/company has a local, default name server
- Often configured via DHCP
- Hosts begin DNS queries by contacting the local name server
- Frequently cache query results

Authoritative Name Servers



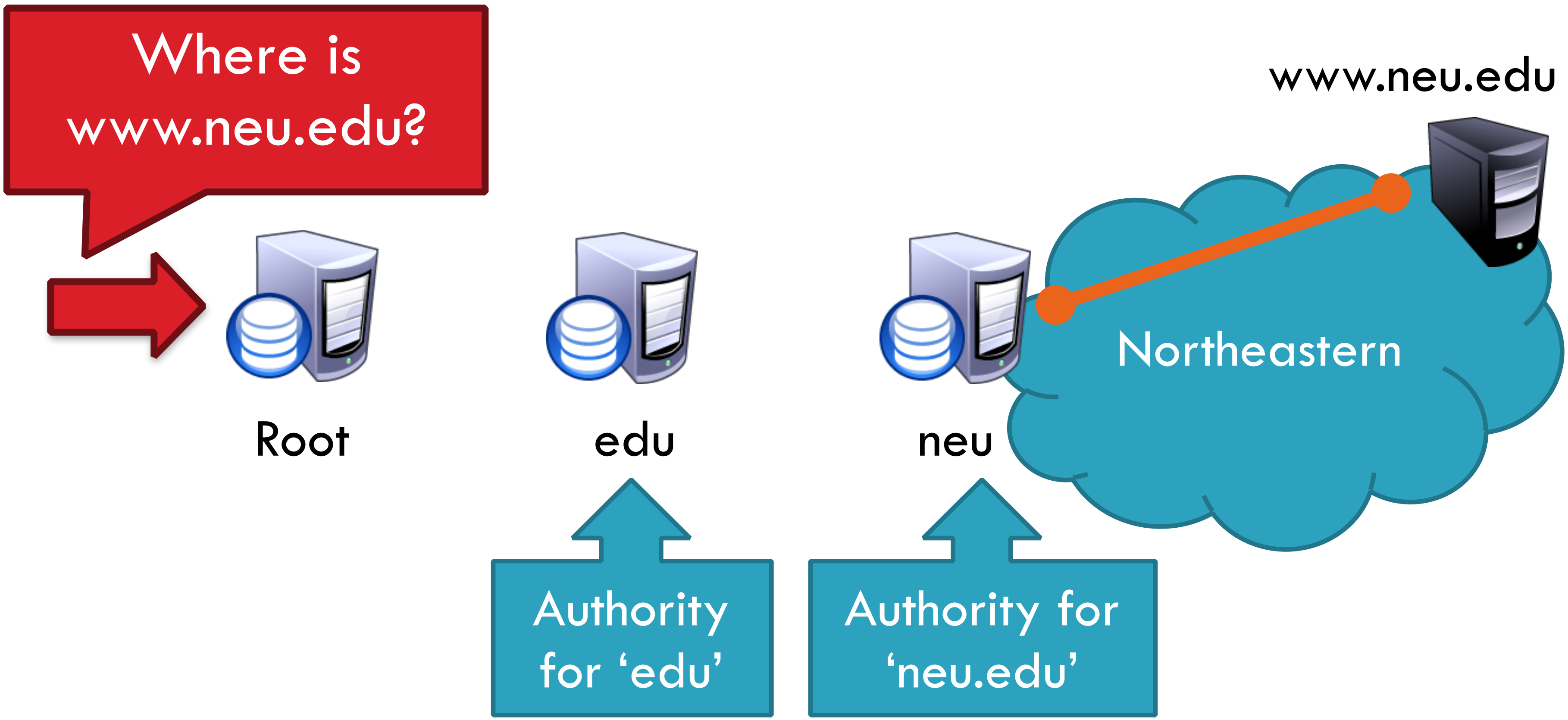
- Stores the name→IP mapping for a given host

Authoritative Name Servers



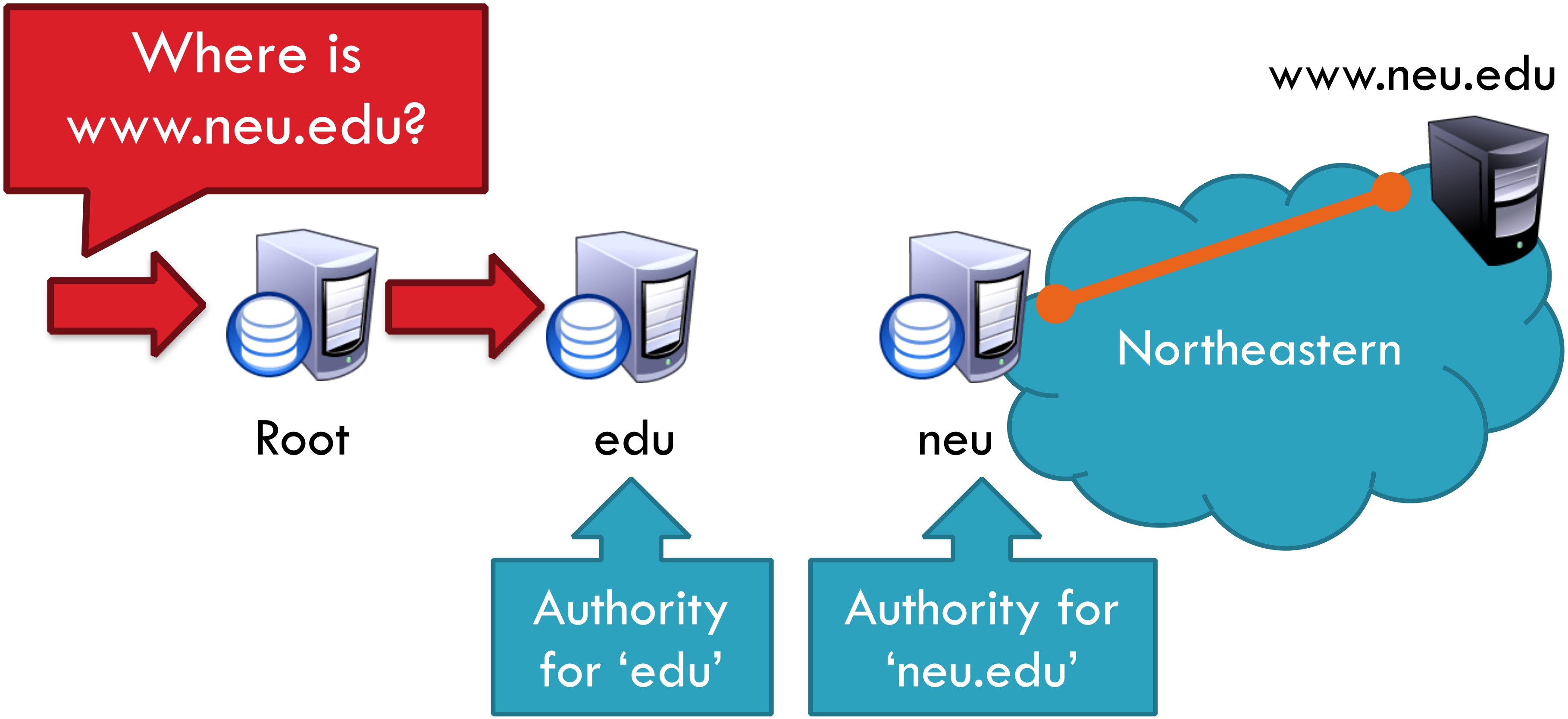
- Stores the name→IP mapping for a given host

Authoritative Name Servers



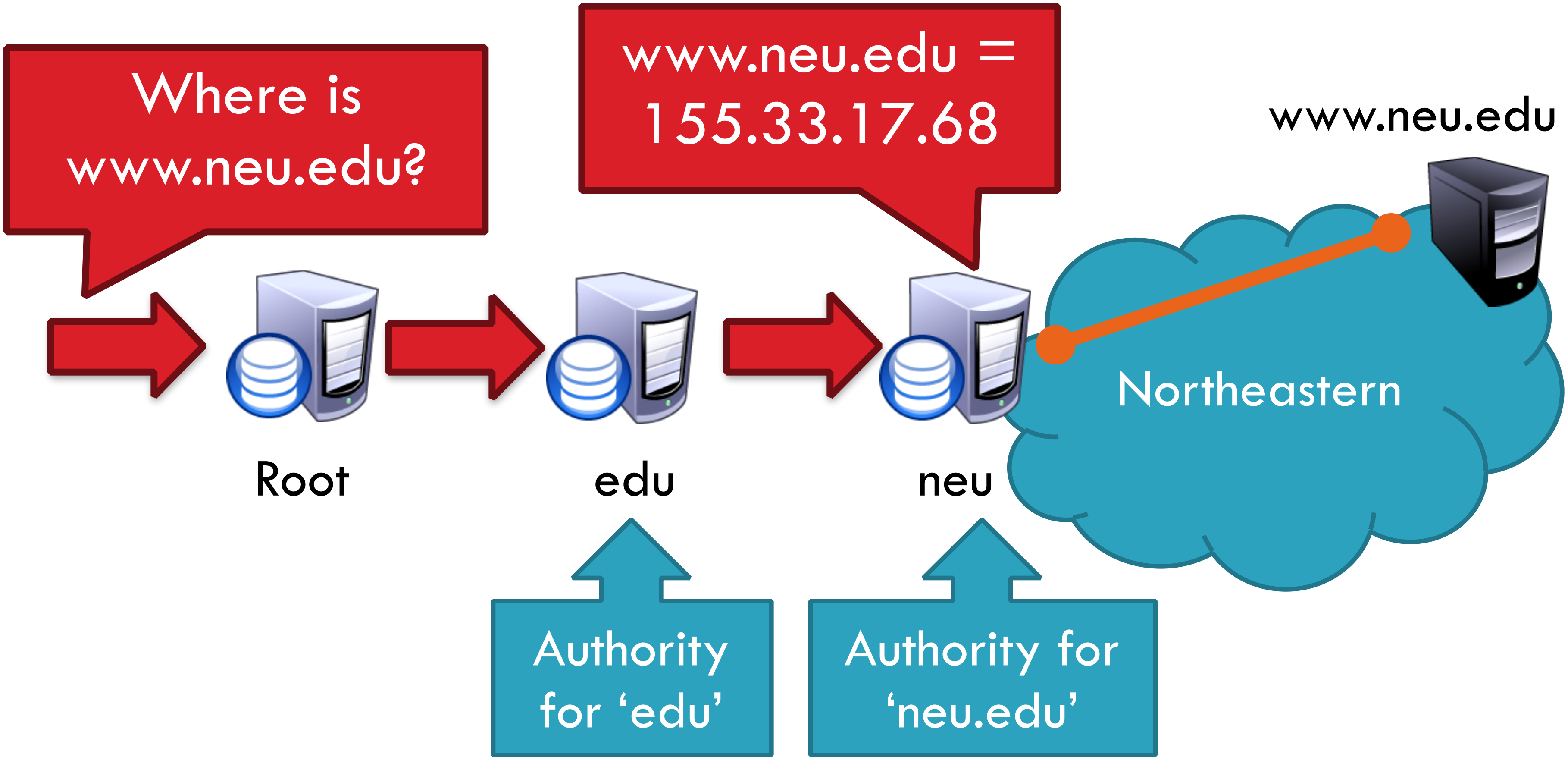
- Stores the name→IP mapping for a given host

Authoritative Name Servers



- Stores the name→IP mapping for a given host

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

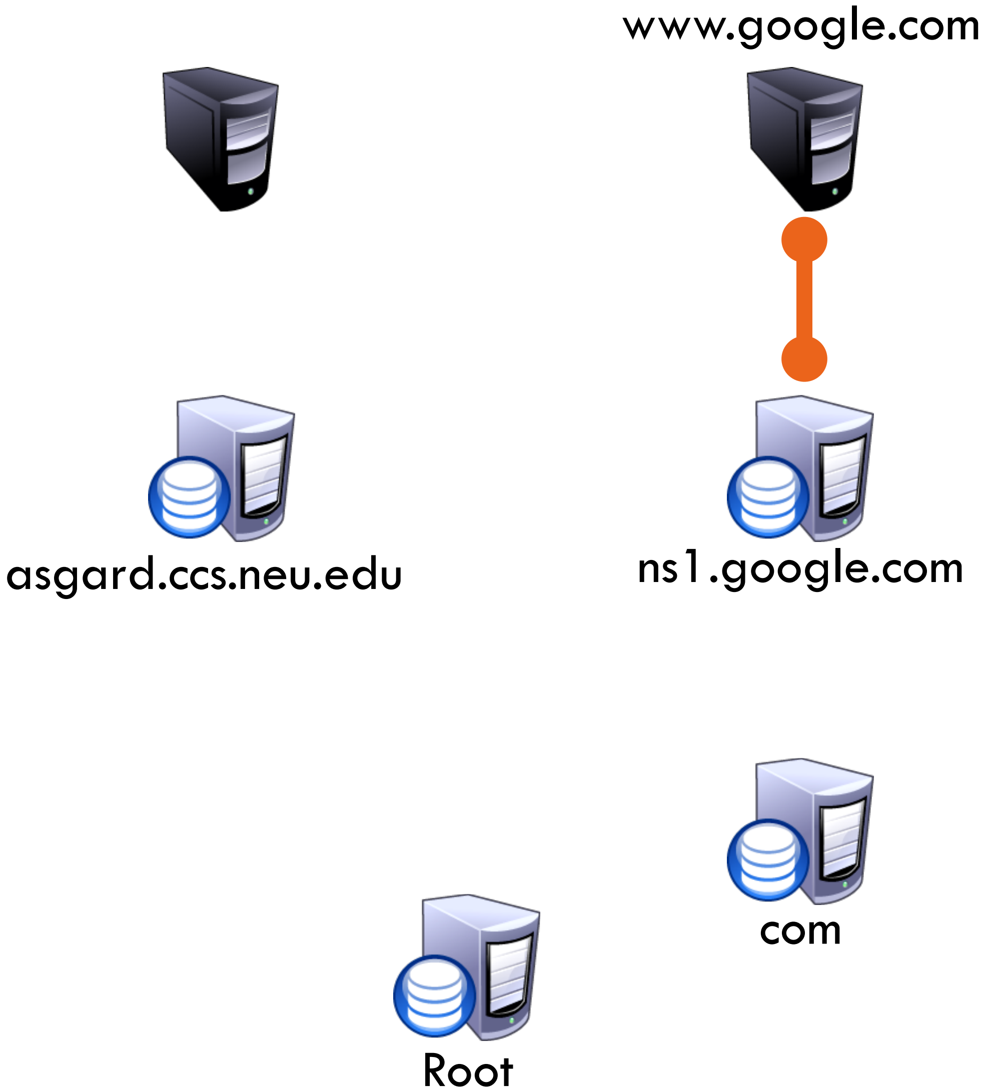
Basic Domain Name Resolution

- 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 is also the authoritative server for that name
  2. Local server has cached the record for that name

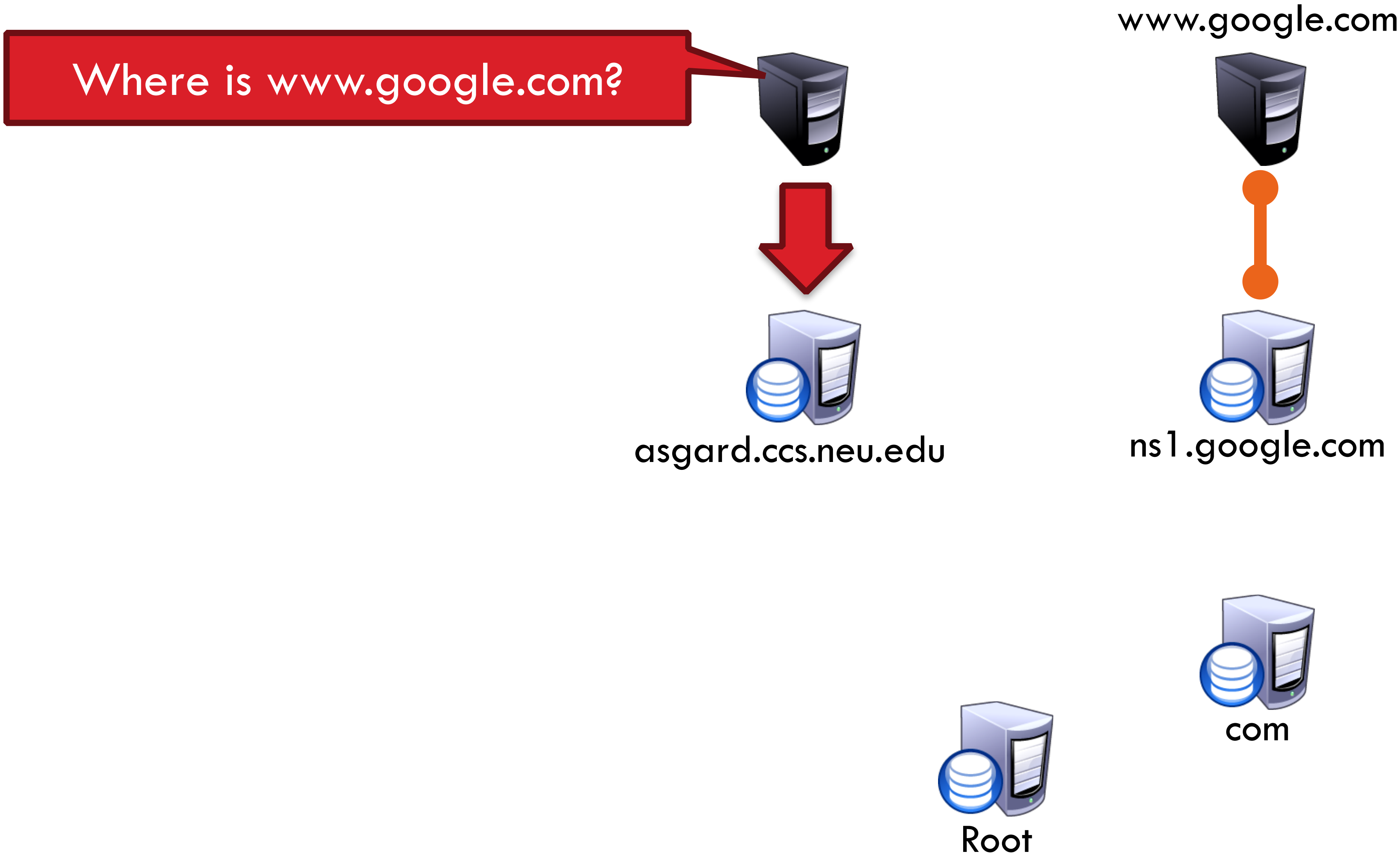
Basic Domain Name Resolution

- 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 is also the authoritative server for that name
  2. 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

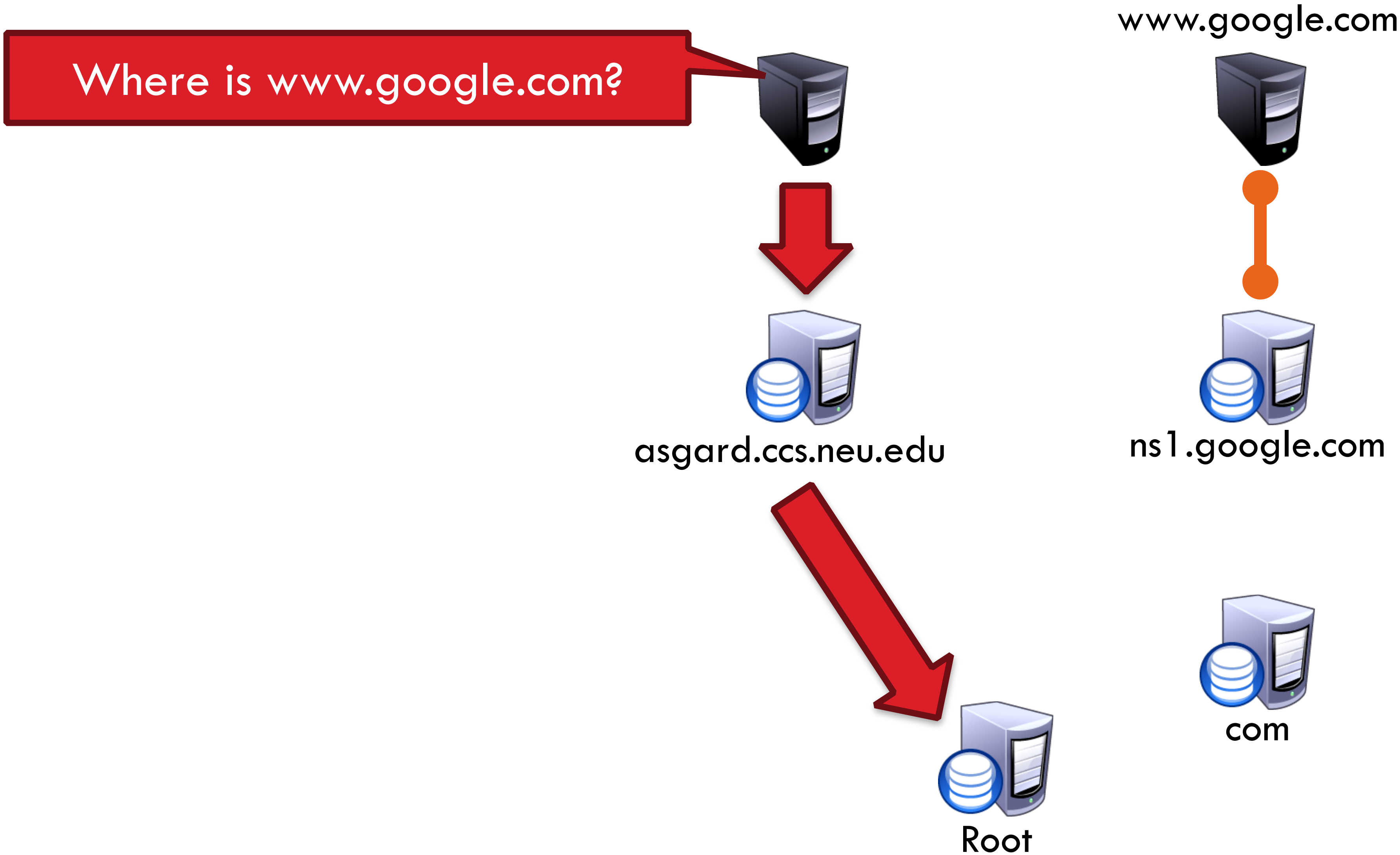
Recursive DNS Query



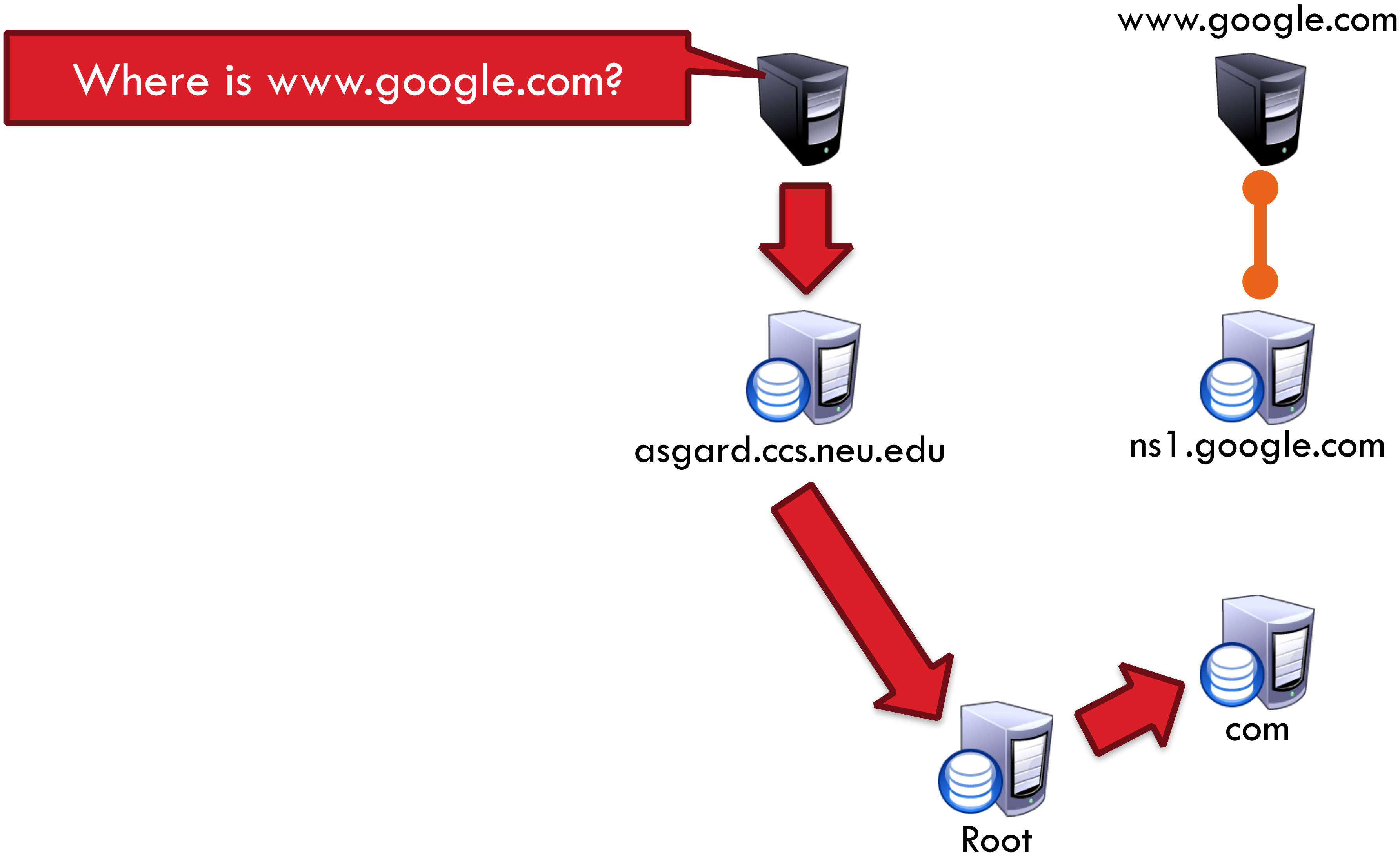
Recursive DNS Query



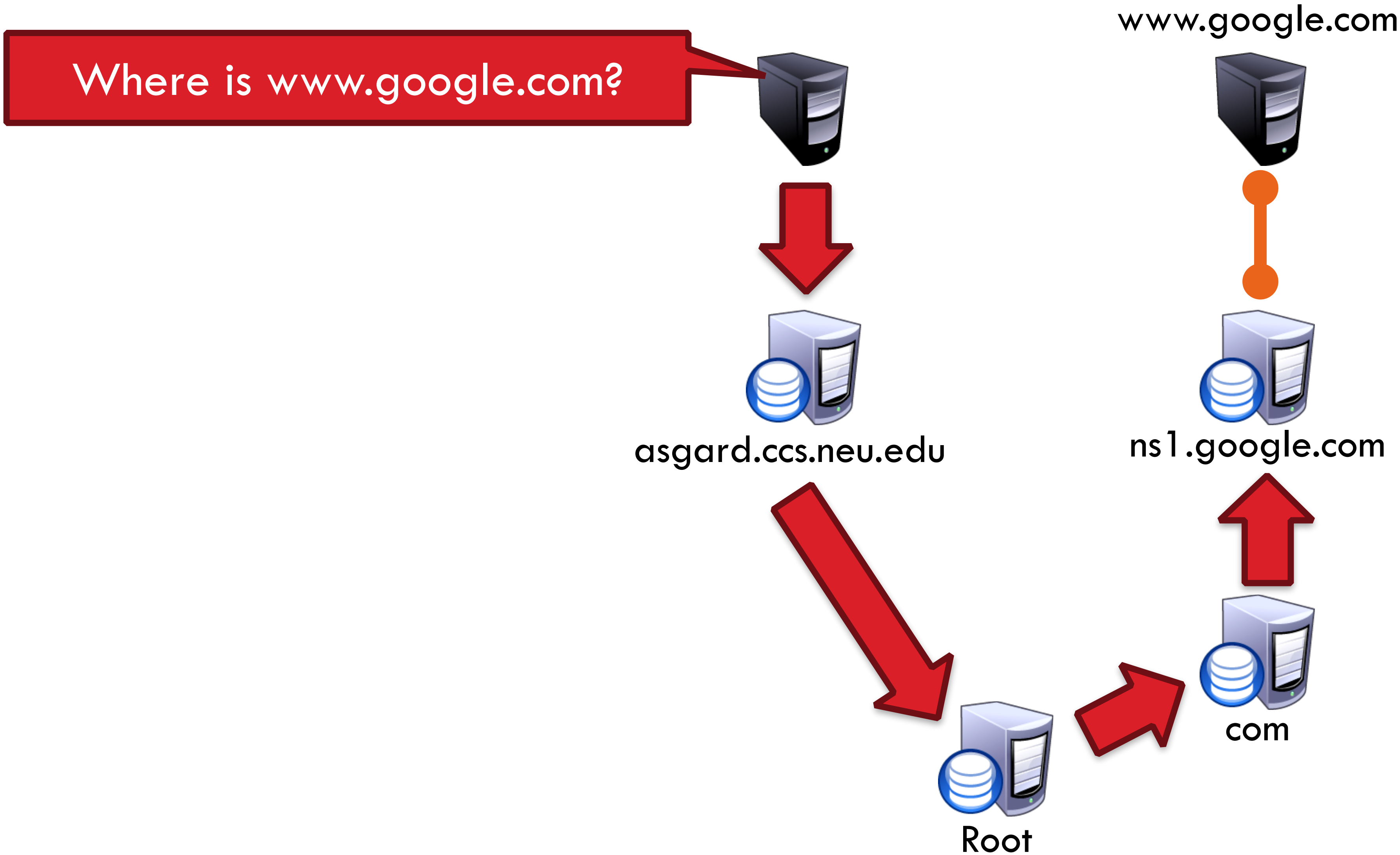
Recursive DNS Query



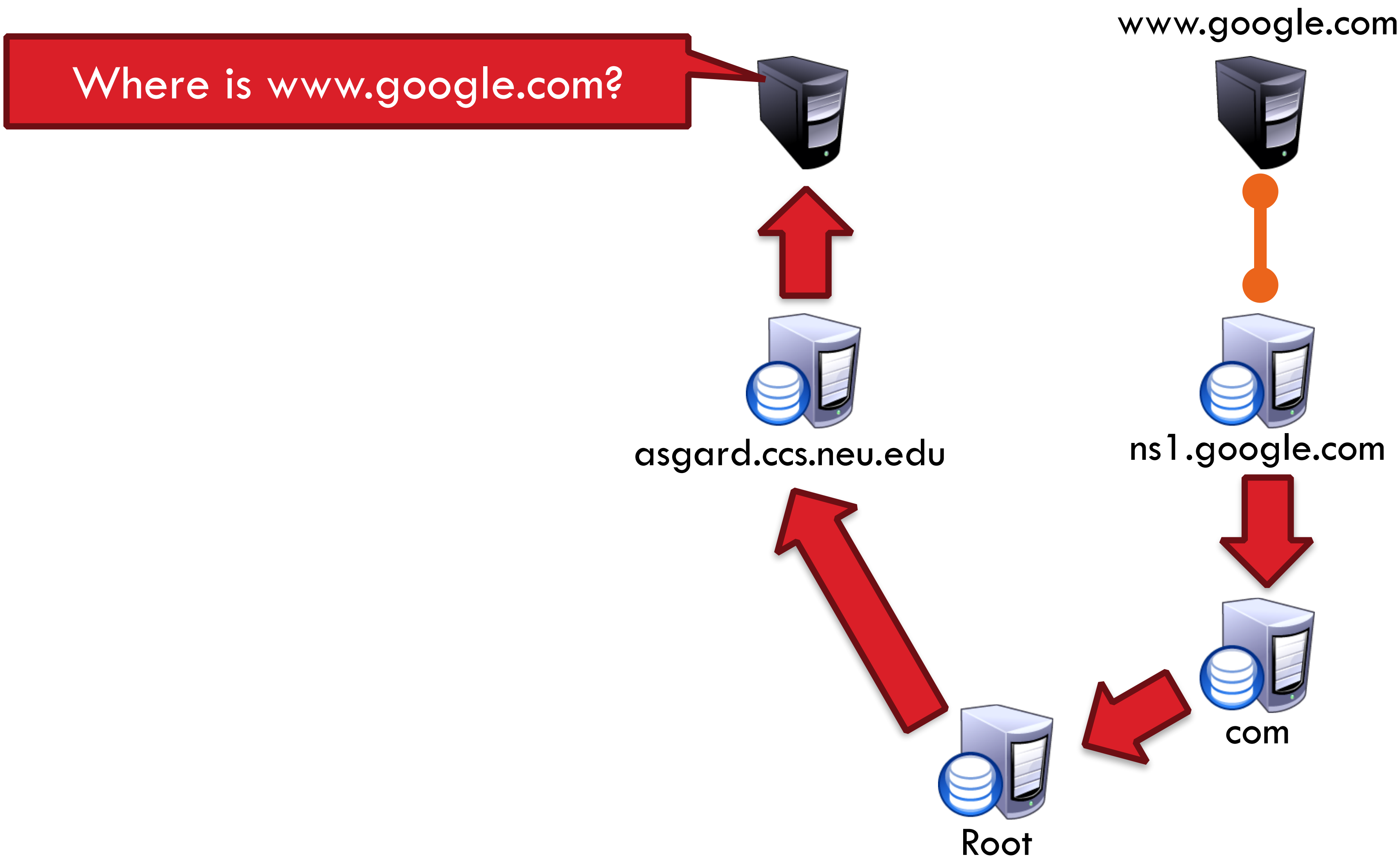
Recursive DNS Query



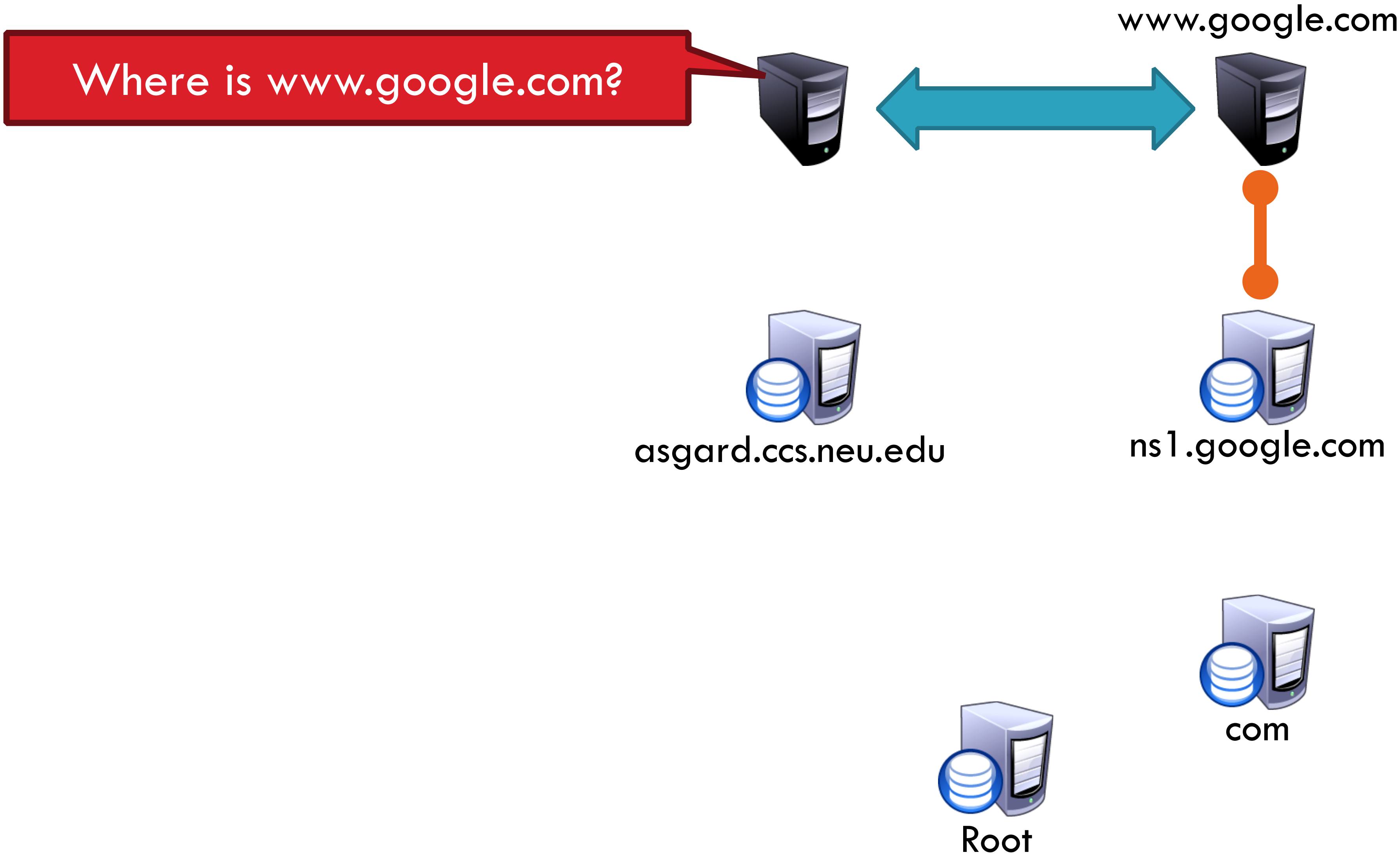
Recursive DNS Query



Recursive DNS Query

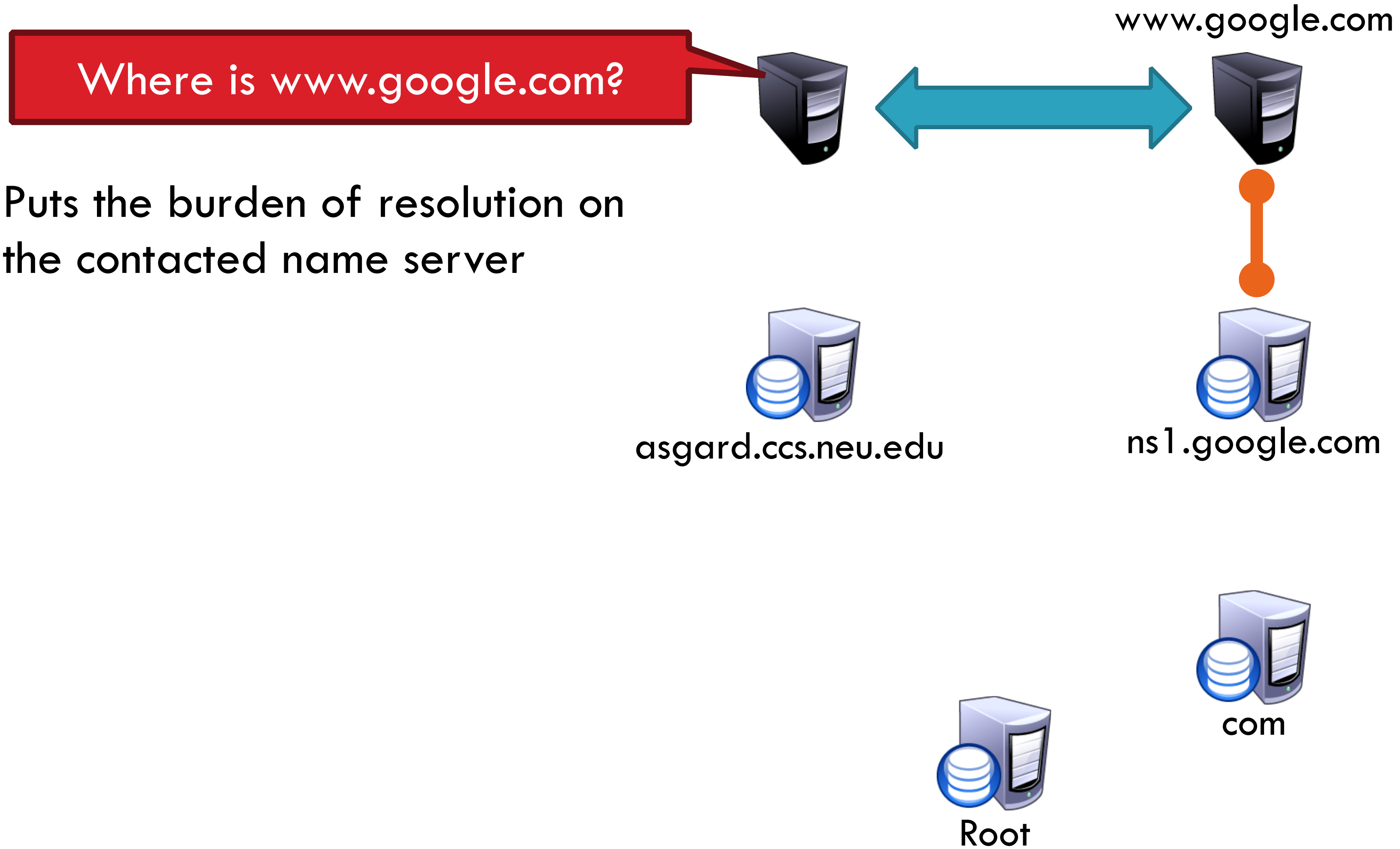


Recursive DNS Query



Recursive DNS Query

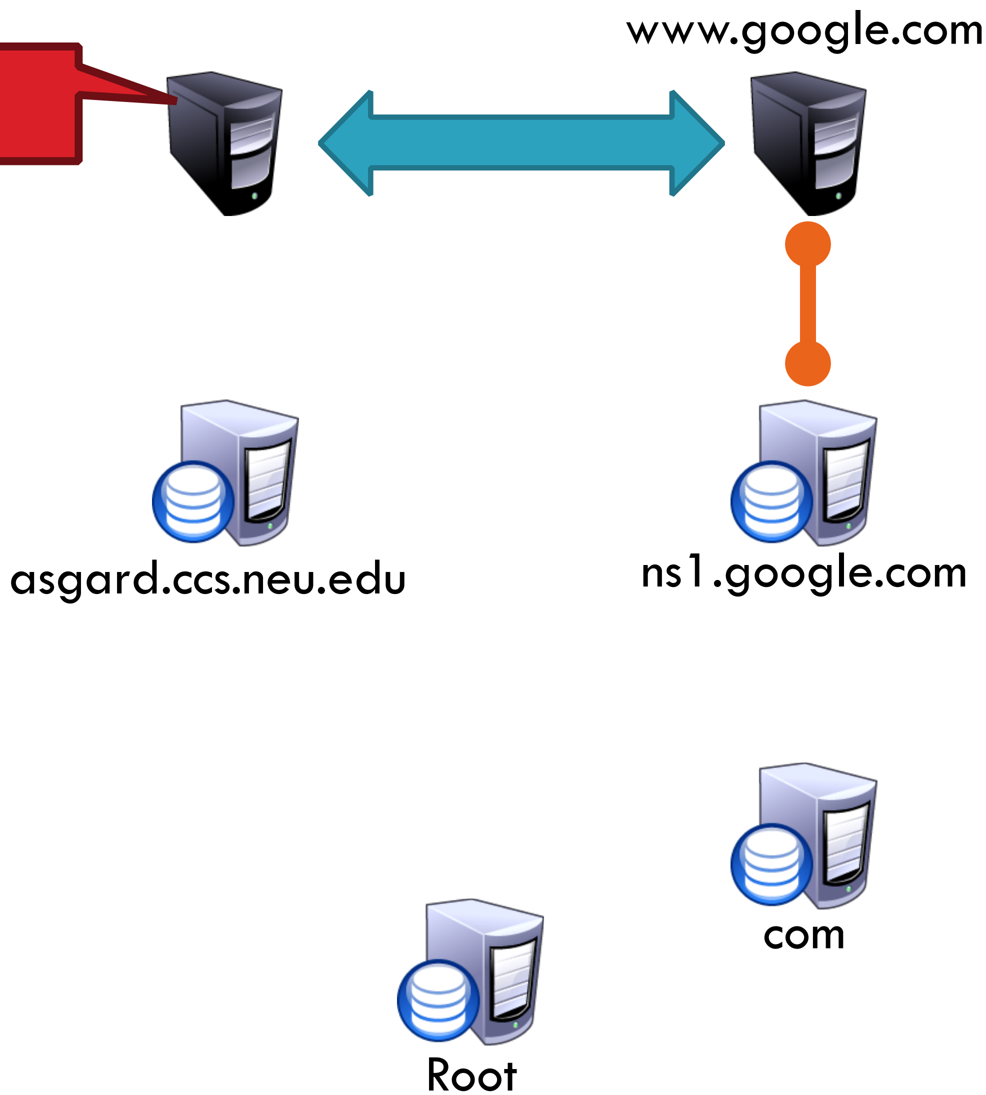
- Puts the burden of resolution on the contacted name server



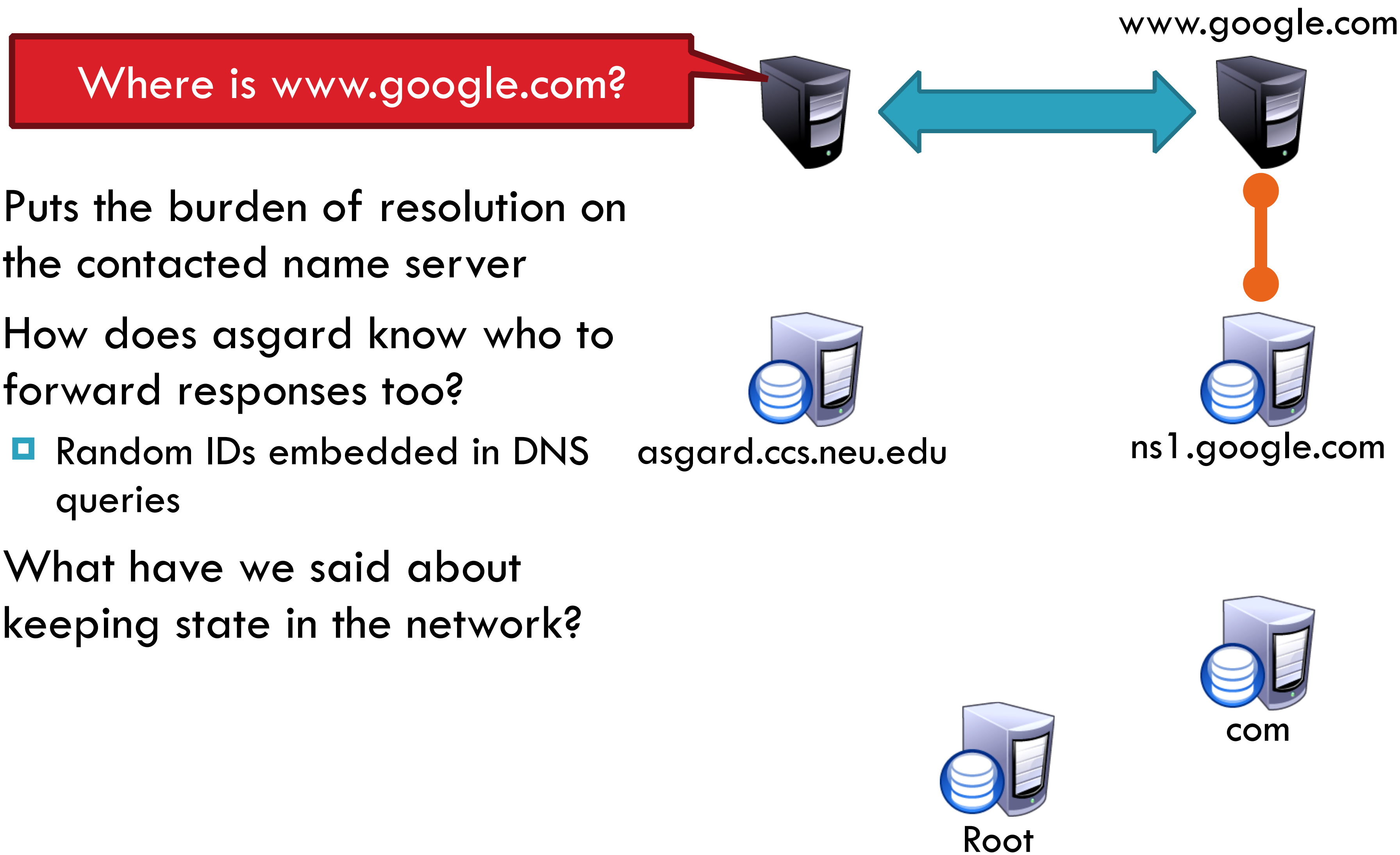
Recursive DNS Query

Where is www.google.com?

- Puts the burden of resolution on the contacted name server
- How does asgard know who to forward responses too?
  - Random IDs embedded in DNS queries

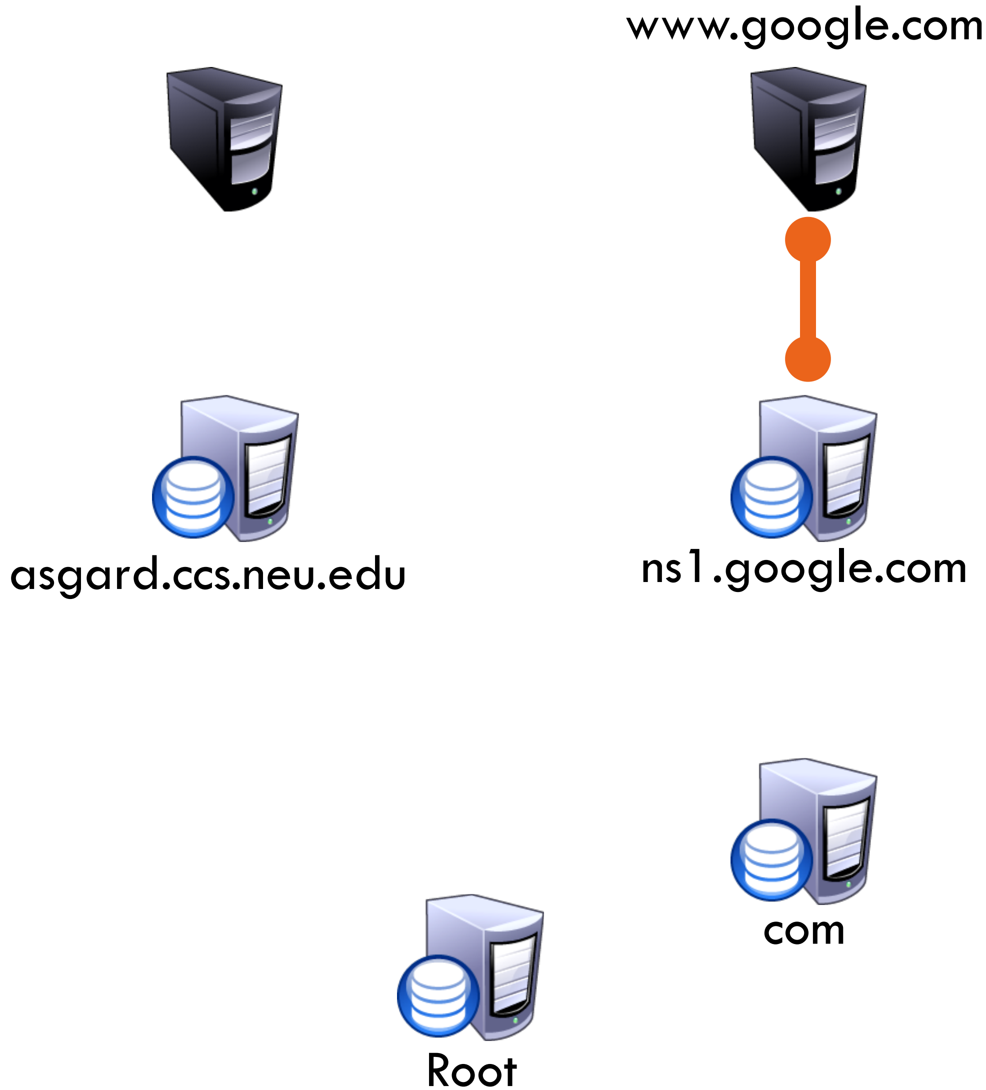


Recursive DNS Query

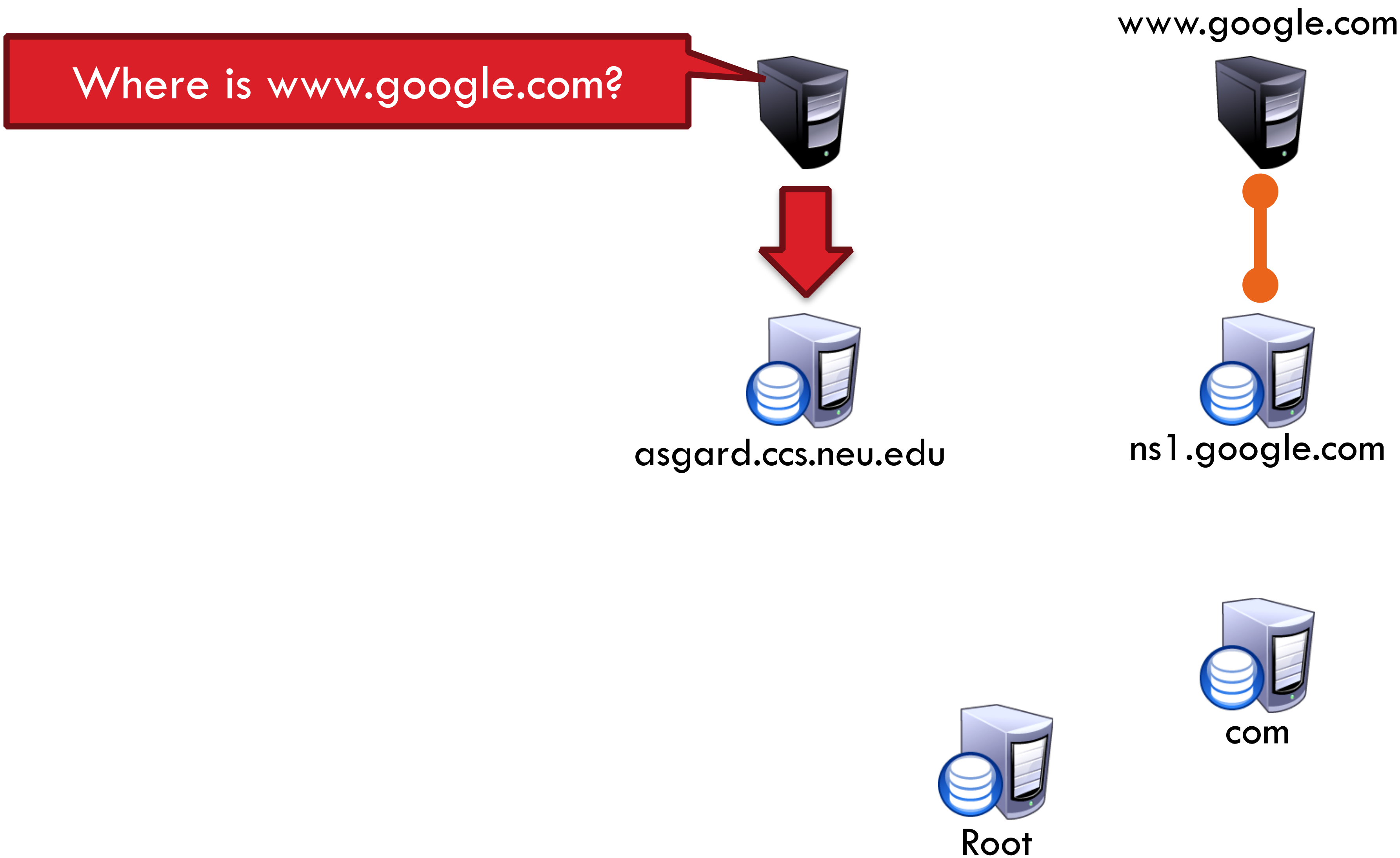


- Puts the burden of resolution on the contacted name server
- How does asgard know who to forward responses too?
  - Random IDs embedded in DNS queries
- What have we said about keeping state in the network?

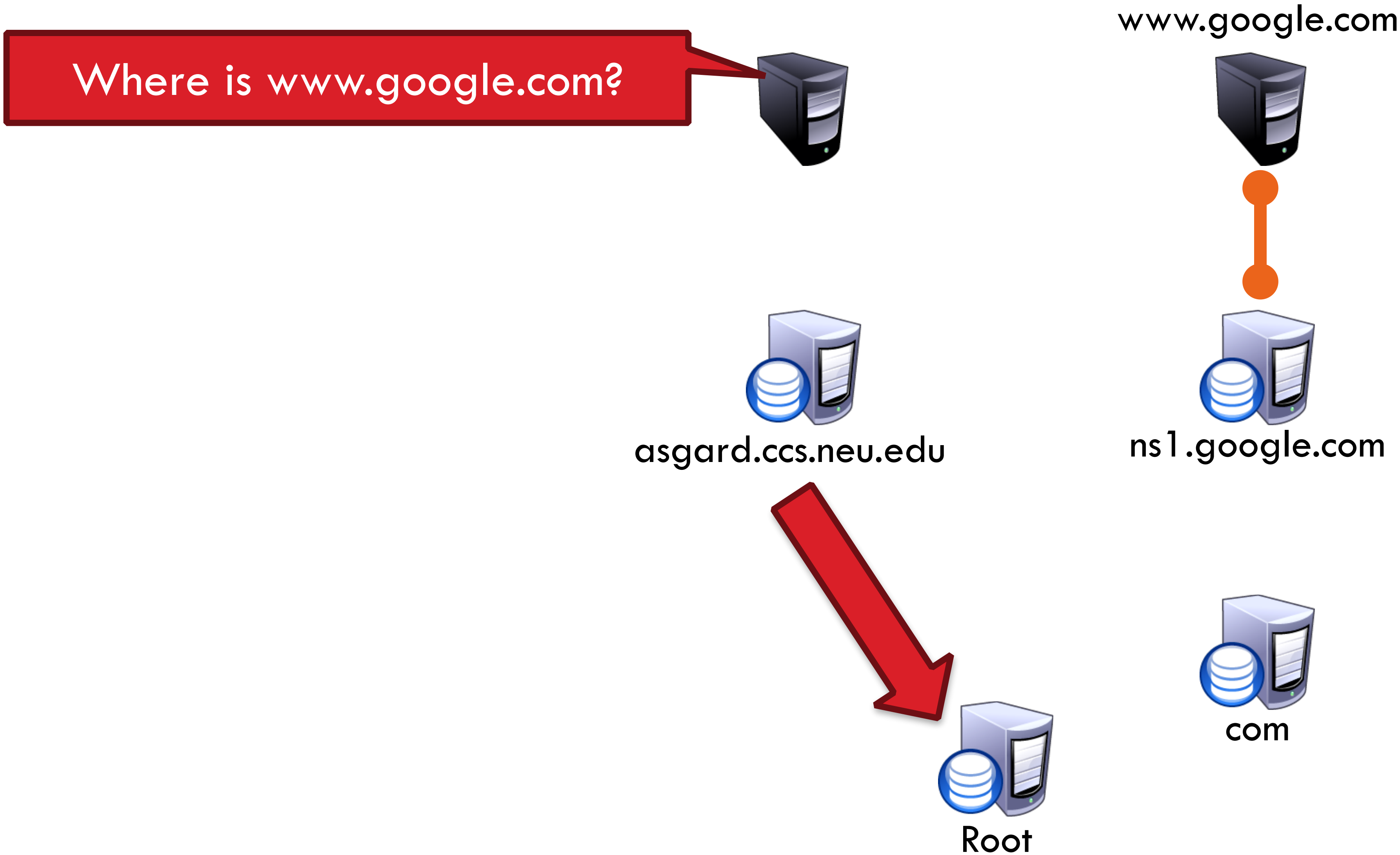
Iterated DNS query



Iterated DNS query

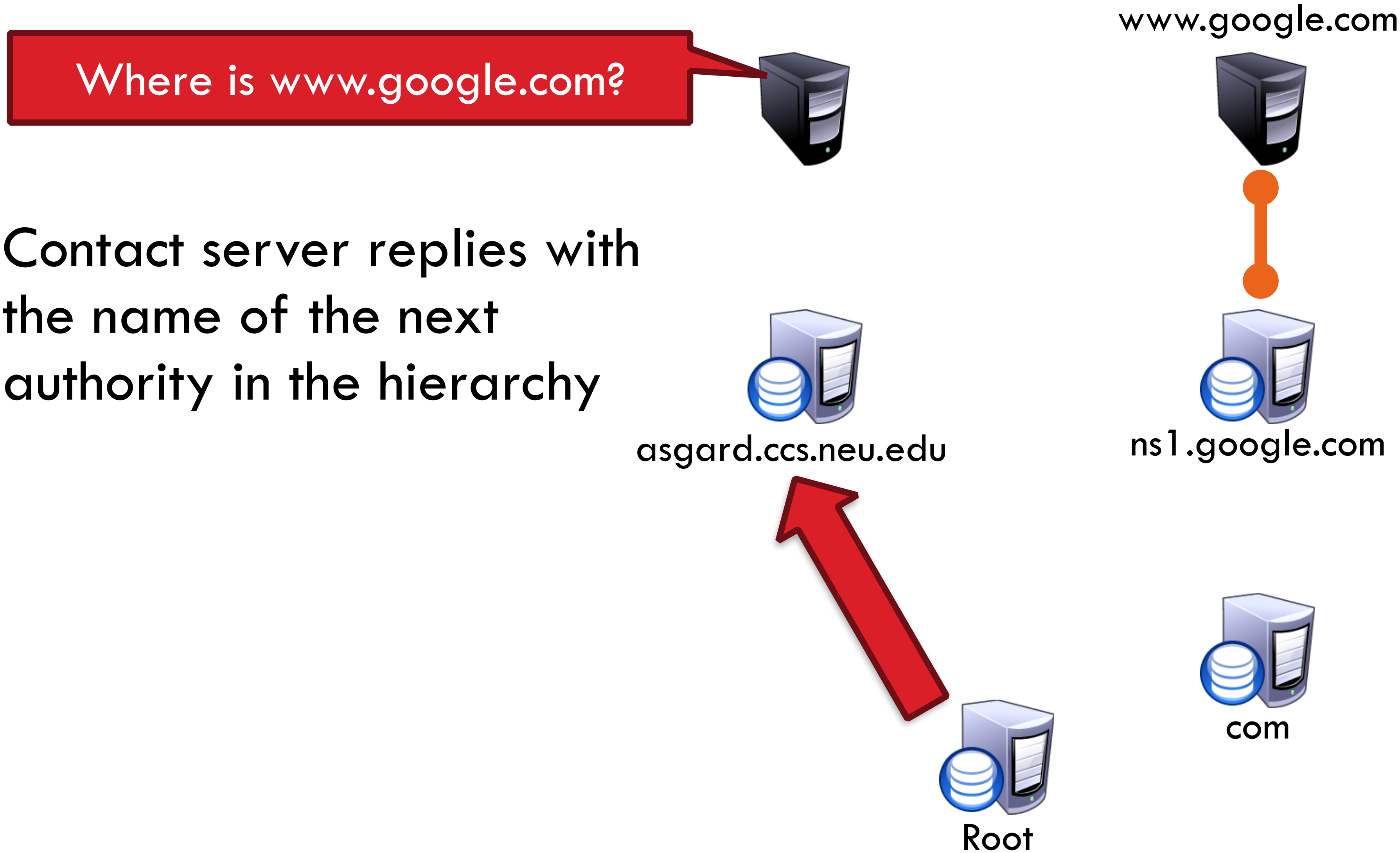


Iterated DNS query



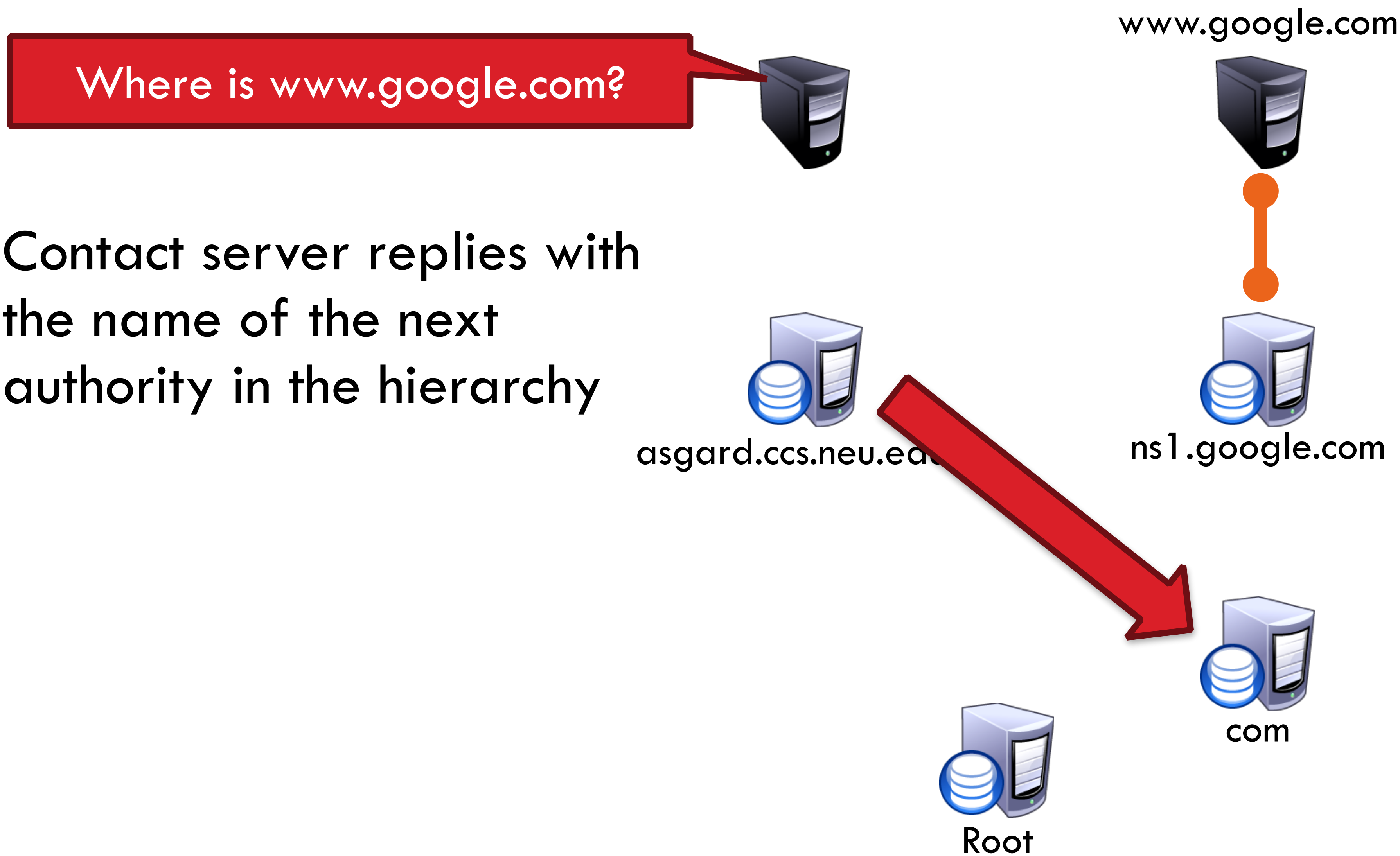
Iterated DNS query

- Contact server replies with the name of the next authority in the hierarchy



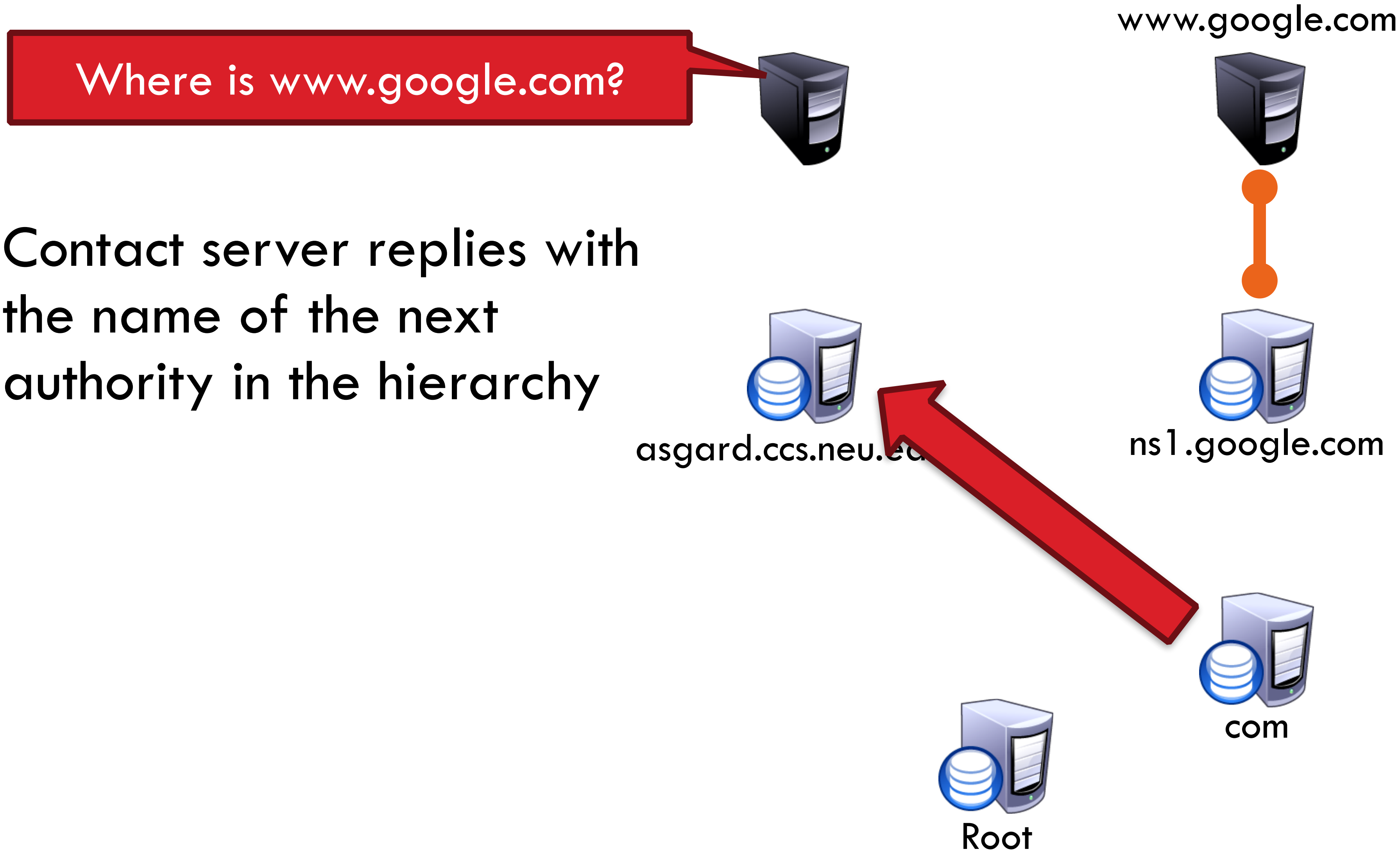
Iterated DNS query

- Contact server replies with the name of the next authority in the hierarchy



Iterated DNS query

- Contact server replies with the name of the next authority in the hierarchy

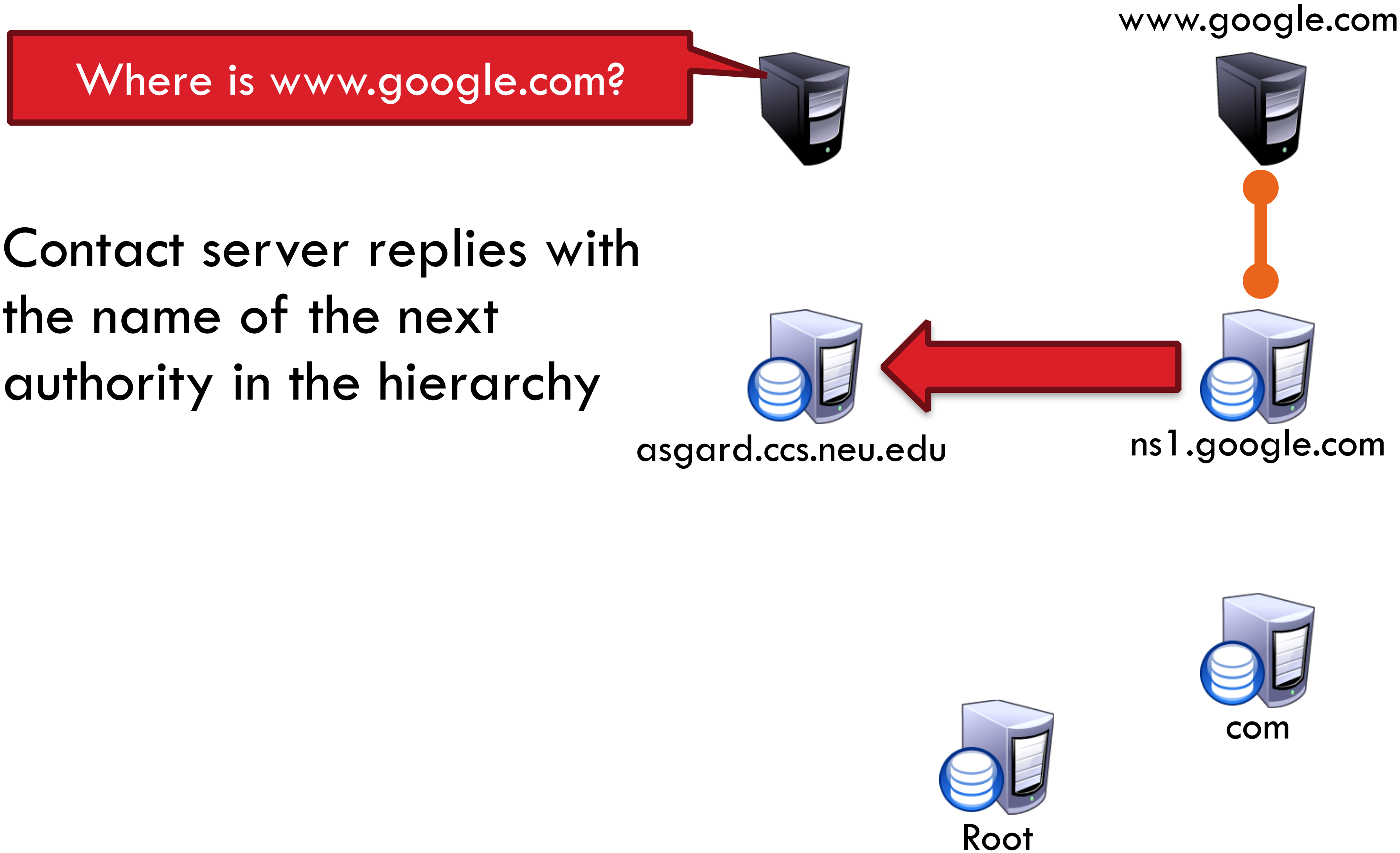


Iterated DNS query

- Contact server replies with the name of the next authority in the hierarchy

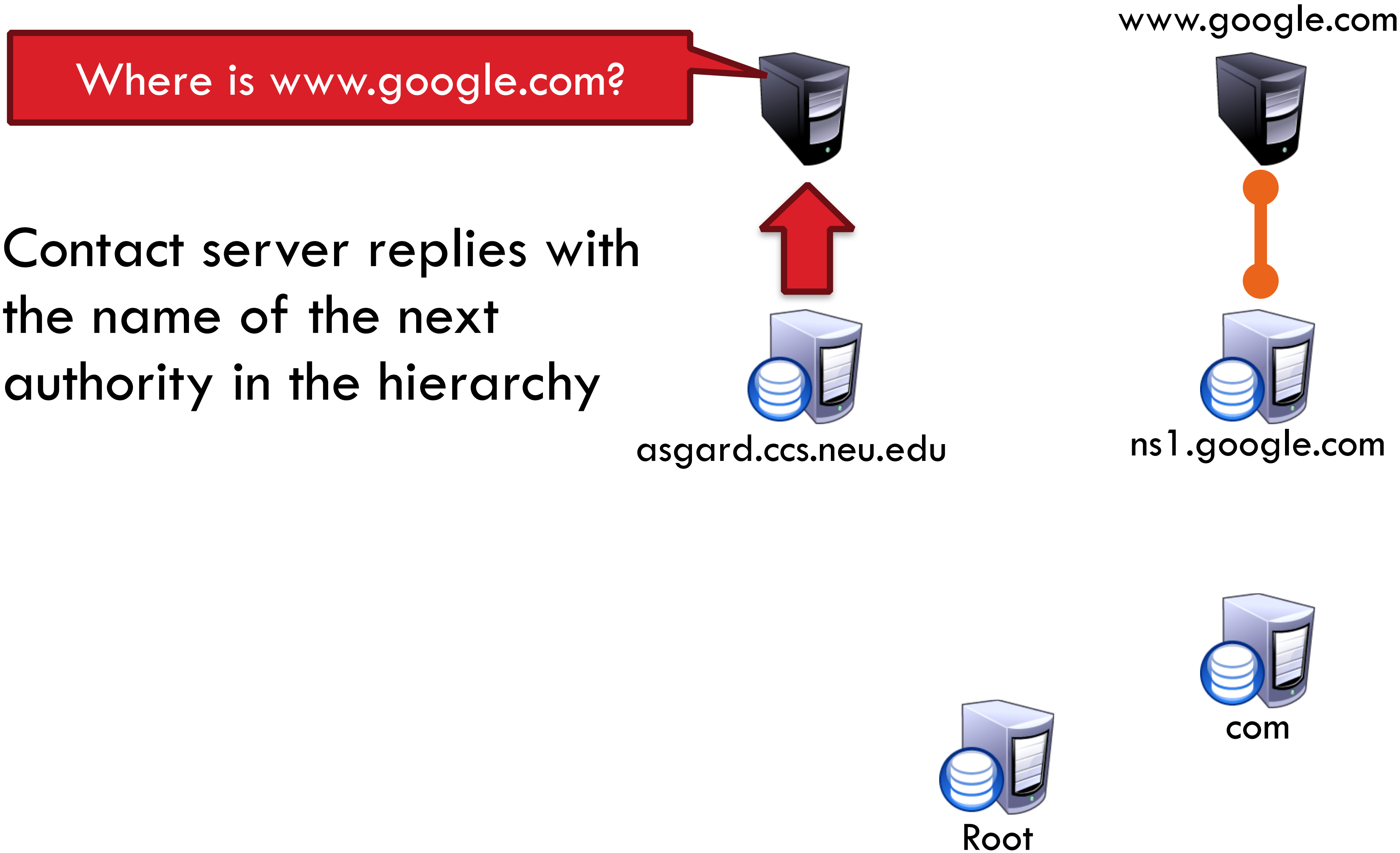
Iterated DNS query

- Contact server replies with the name of the next authority in the hierarchy

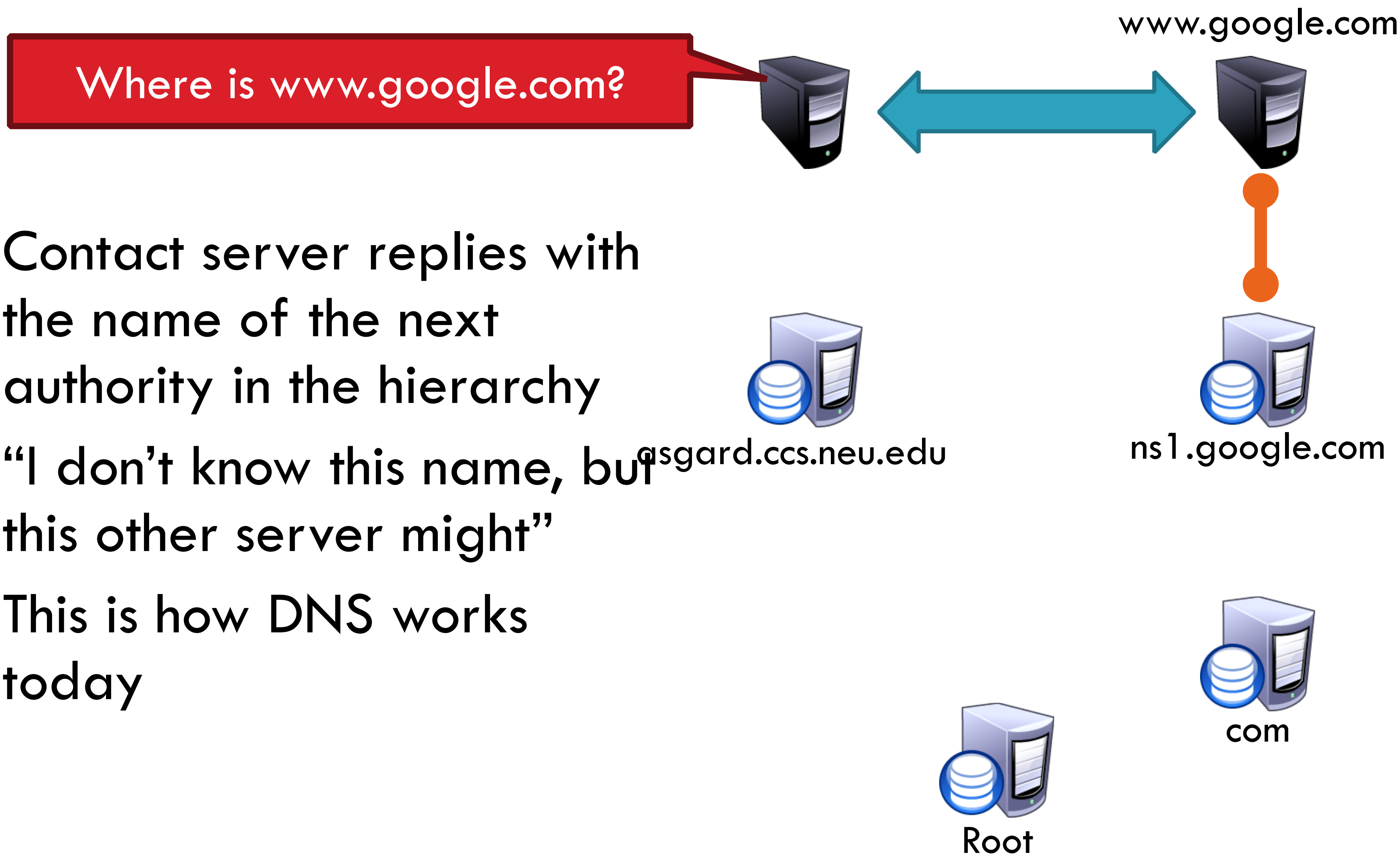


Iterated DNS query

- Contact server replies with the name of the next authority in the hierarchy



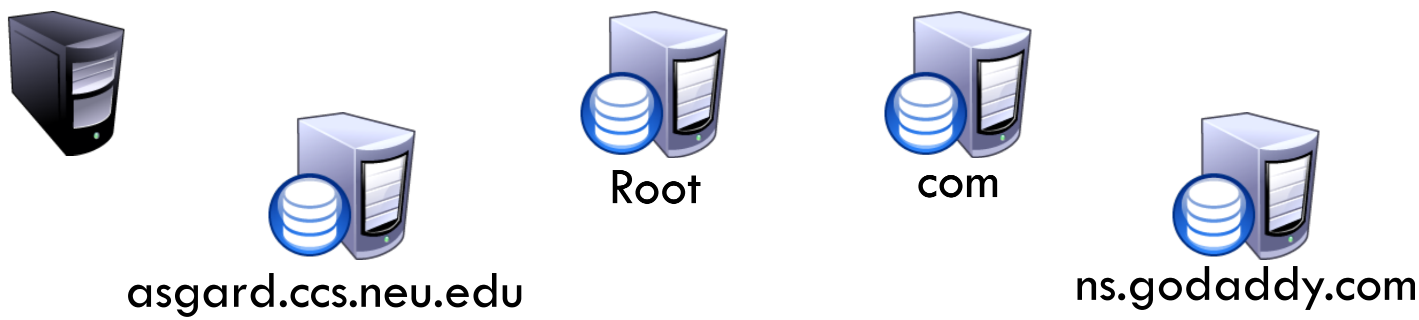
Iterated DNS query



- 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

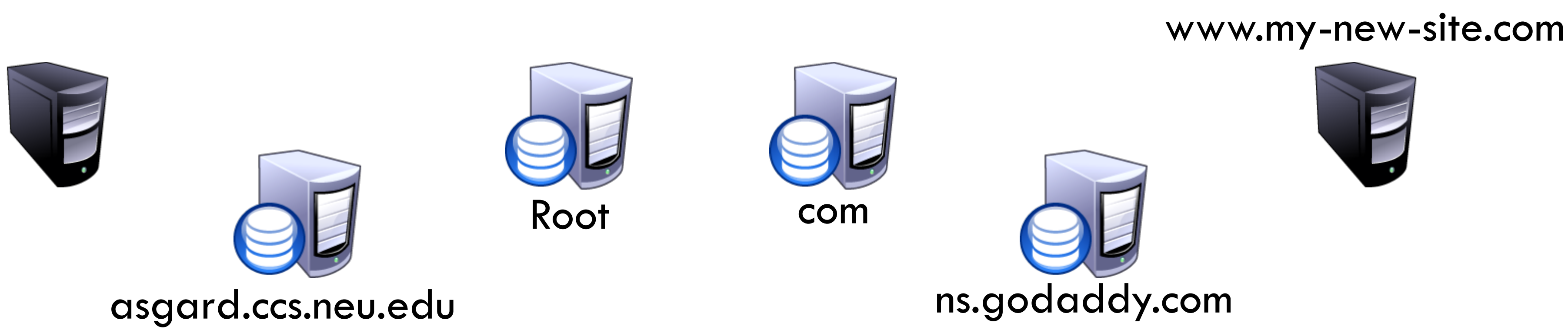
DNS Propagation

- 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



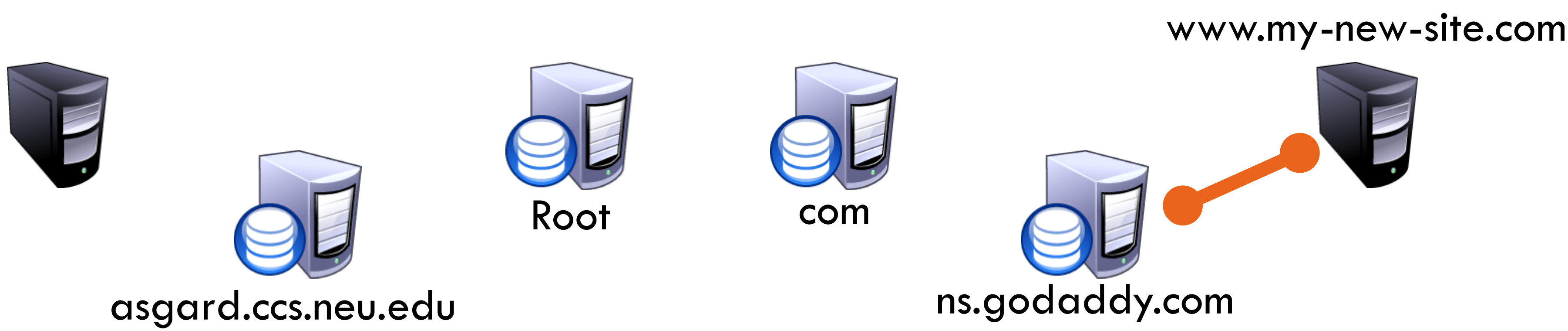
DNS Propagation

- 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



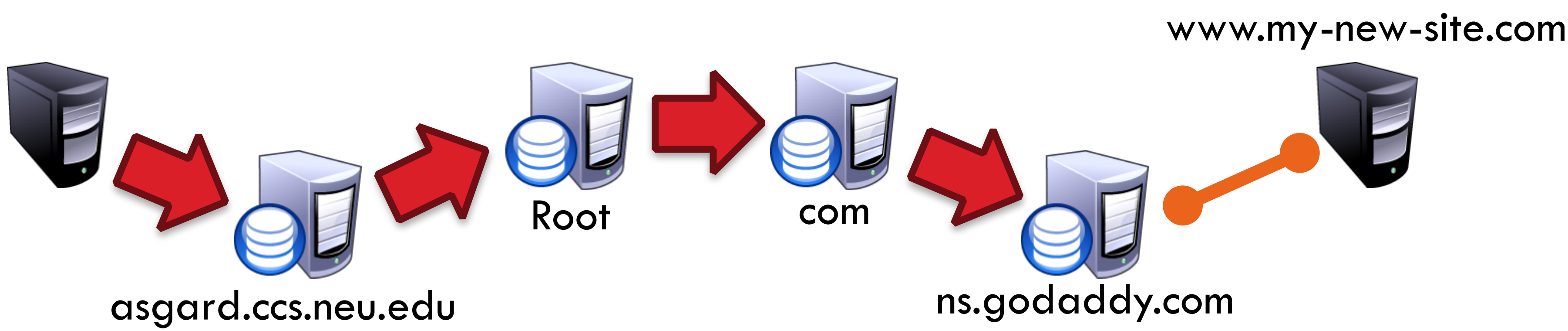
DNS Propagation

- 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



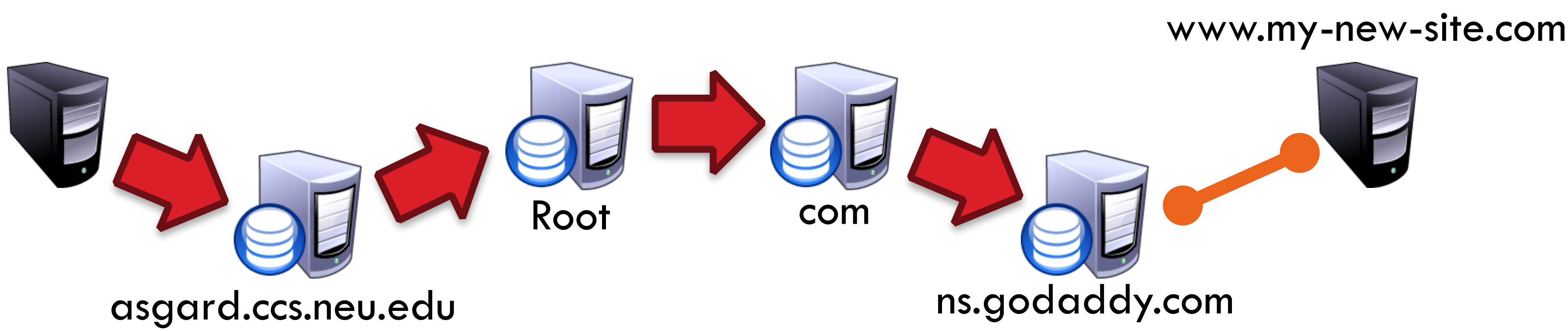
DNS Propagation

- 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



DNS Propagation

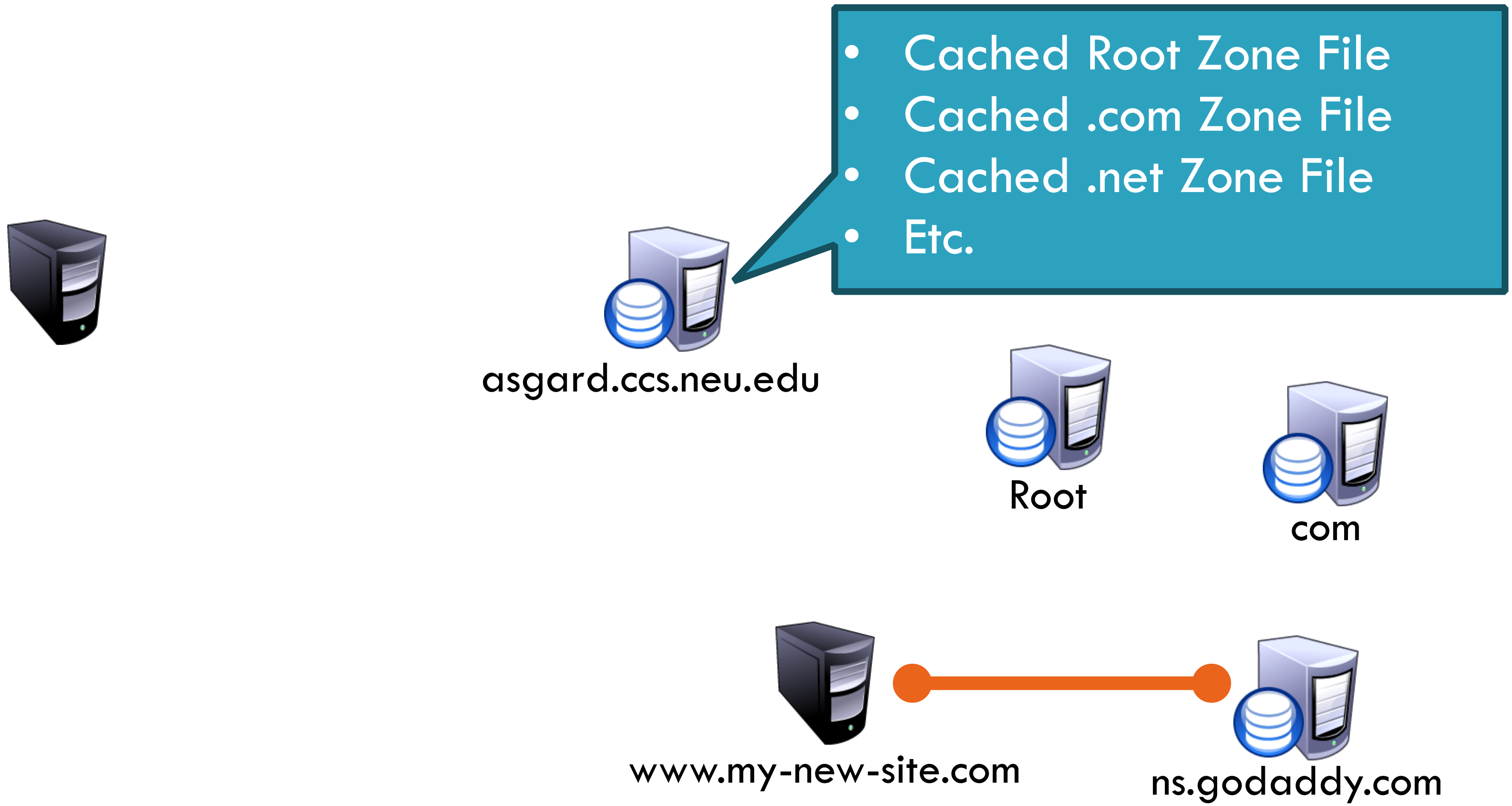
- How many of you have purchased a domain name?
  - Did you notice that it took ~72 hours for your name to become accessible?
  - This delay is called DNS Propagation



- Why would this process fail for a new DNS name?

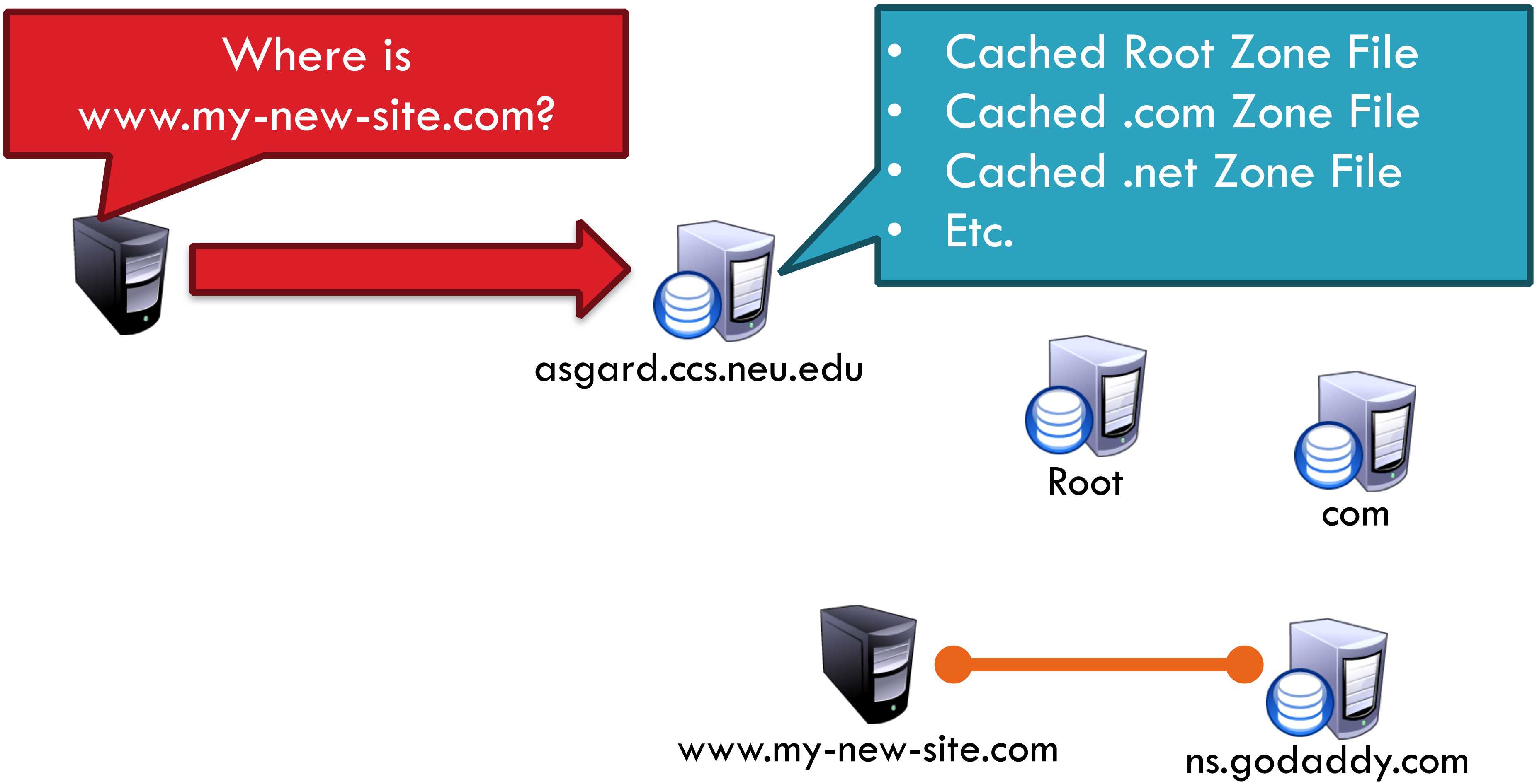
Caching vs. Freshness

- DNS Propagation delay is caused by caching



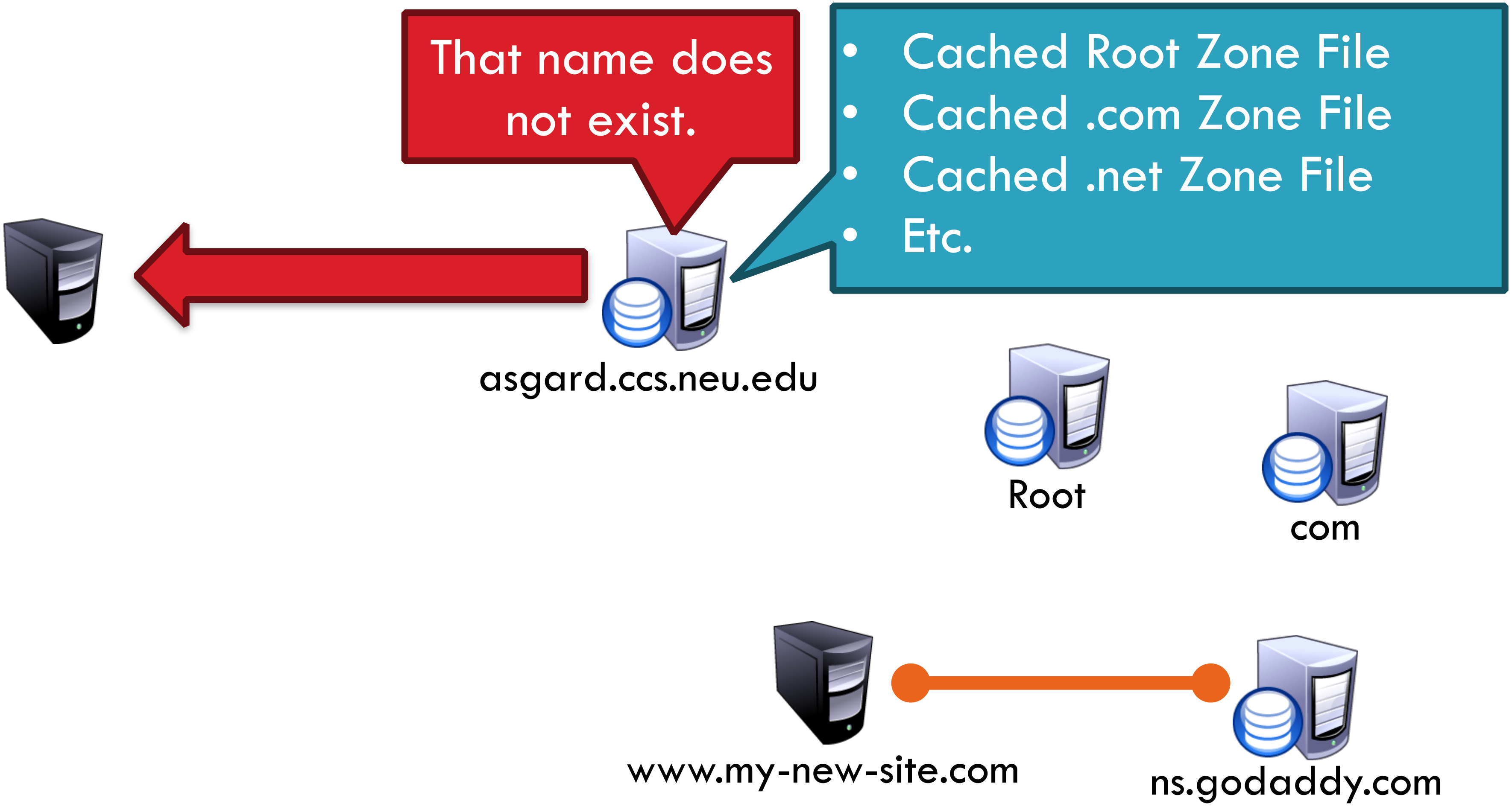
Caching vs. Freshness

- DNS Propagation delay is caused by caching



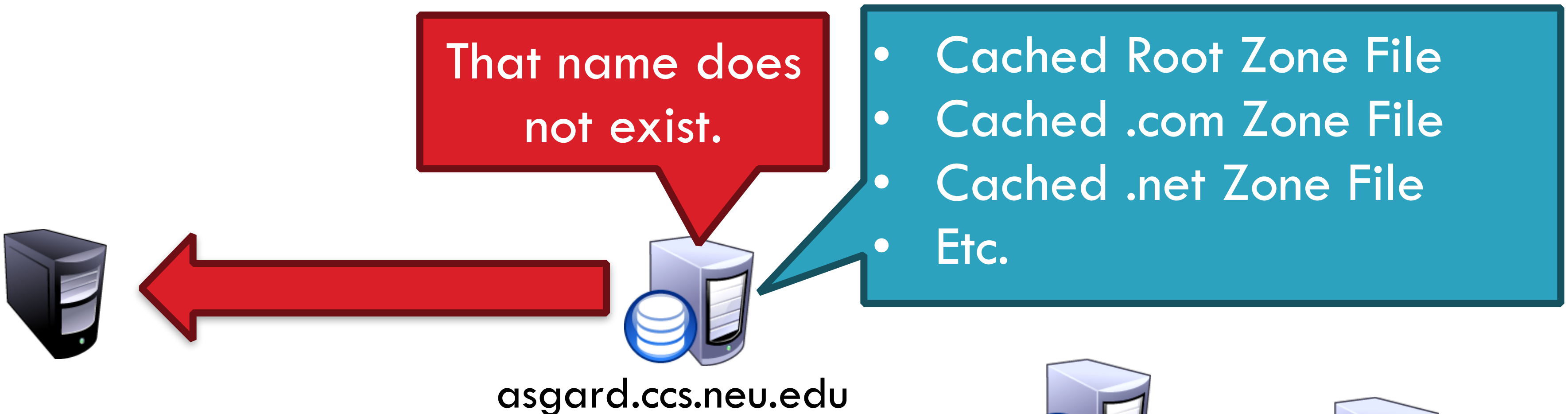
Caching vs. Freshness

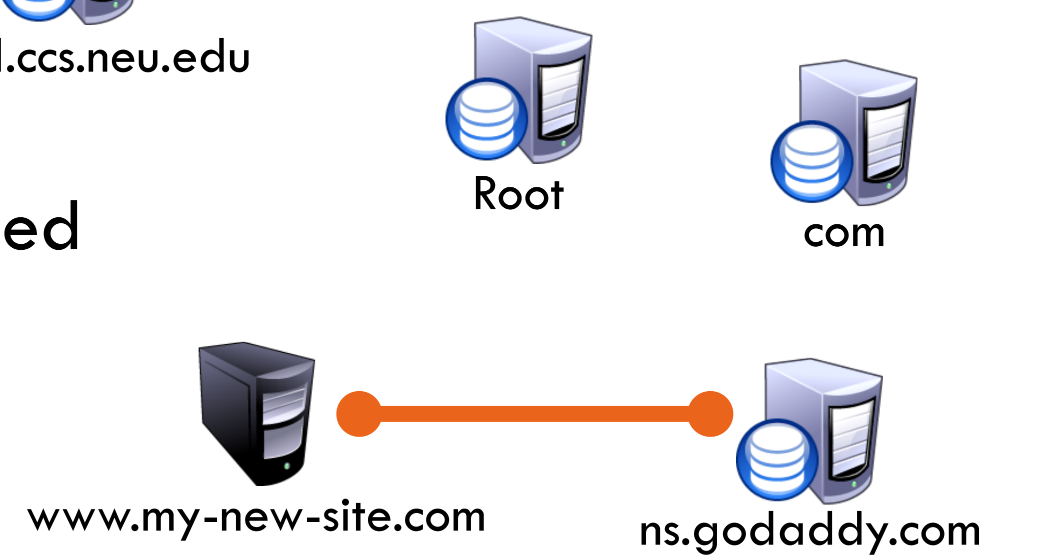
- DNS Propagation delay is caused by caching



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

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 Types

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

DNS Types

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

Query

Name: www.ccs.neu.edu
Type: A

DNS Types

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

Query

Name: www.ccs.neu.edu
Type: A

Resp.

Name: www.ccs.neu.edu
Value: 129.10.116.81

DNS Types

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

Query	Name: www.ccs.neu.edu Type: A
Resp.	Name: www.ccs.neu.edu Value: 129.10.116.81

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

DNS Types

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

Query
Name: www.ccs.neu.edu
Type: A

Resp.
Name: www.ccs.neu.edu
Value: 129.10.116.81

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

Query
Name: ccs.neu.edu
Type: NS

DNS Types

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

Query
Name: www.ccs.neu.edu
Type: A

Resp.
Name: www.ccs.neu.edu
Value: 129.10.116.81

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

Query
Name: ccs.neu.edu
Type: NS

Resp.
Name: ccs.neu.edu
Value: 129.10.116.51

DNS Types, Continued

- Type = CNAME
  - Name = hostname
  - Value = canonical hostname
  - Useful for aliasing
  - CDNs use this

DNS Types, Continued

- Type = CNAME
  - Name = hostname
  - Value = canonical hostname
  - Useful for aliasing
  - CDNs use this

Query

Name: foo.mysite.com
Type: CNAME

DNS Types, Continued

- Type = CNAME
  - Name = hostname
  - Value = canonical hostname
  - Useful for aliasing
  - CDNs use this

Query

Name: foo.mysite.com
Type: CNAME

Resp.

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

DNS Types, Continued

- Type = CNAME
  - Name = hostname
  - Value = canonical hostname
  - Useful for aliasing
  - CDNs use this

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

DNS Types, Continued

- Type = CNAME
  - Name = hostname
  - Value = canonical hostname
  - Useful for aliasing
  - CDNs use this

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: ccs.neu.edu
Type: MX

DNS Types, Continued

- Type = CNAME
  - Name = hostname
  - Value = canonical hostname
  - Useful for aliasing
  - CDNs use this

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: ccs.neu.edu
Type: MX

Resp.
Name: ccs.neu.edu
Value: amber.ccs.neu.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

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

Query
Name: 129.10.116.51
Type: PTR

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

Query	Name: 129.10.116.51 Type: PTR
Resp.	Name: 129.10.116.51 Value: ccs.neu.edu

DNS as Indirection Service

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

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

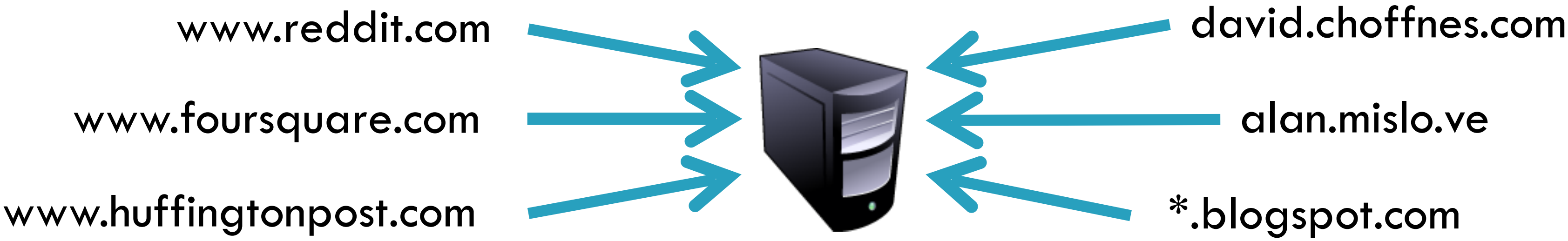
- One machine can have many aliases

Aliasing and Load Balancing

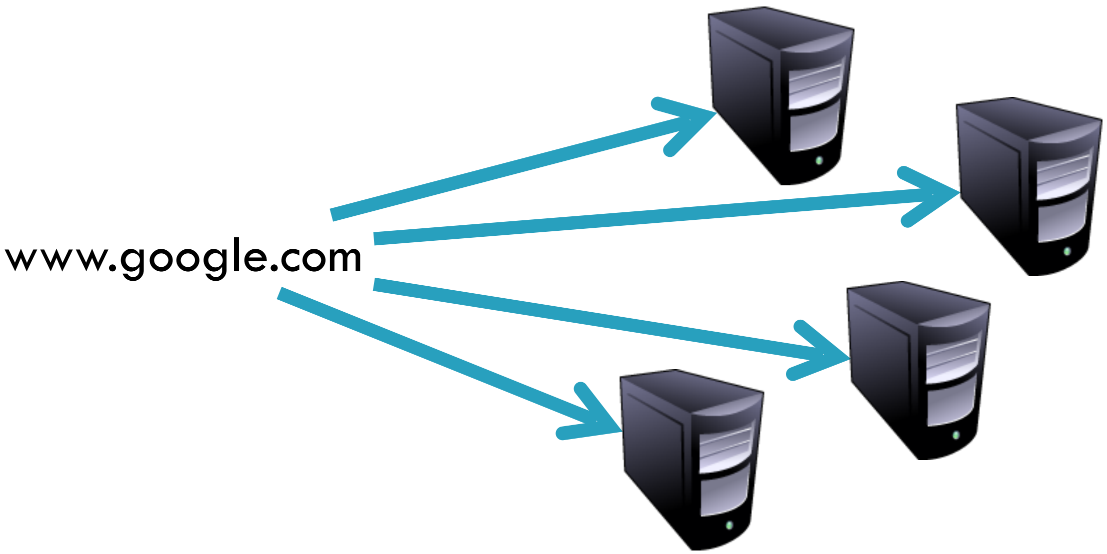
- One machine can have many aliases

Aliasing and Load Balancing

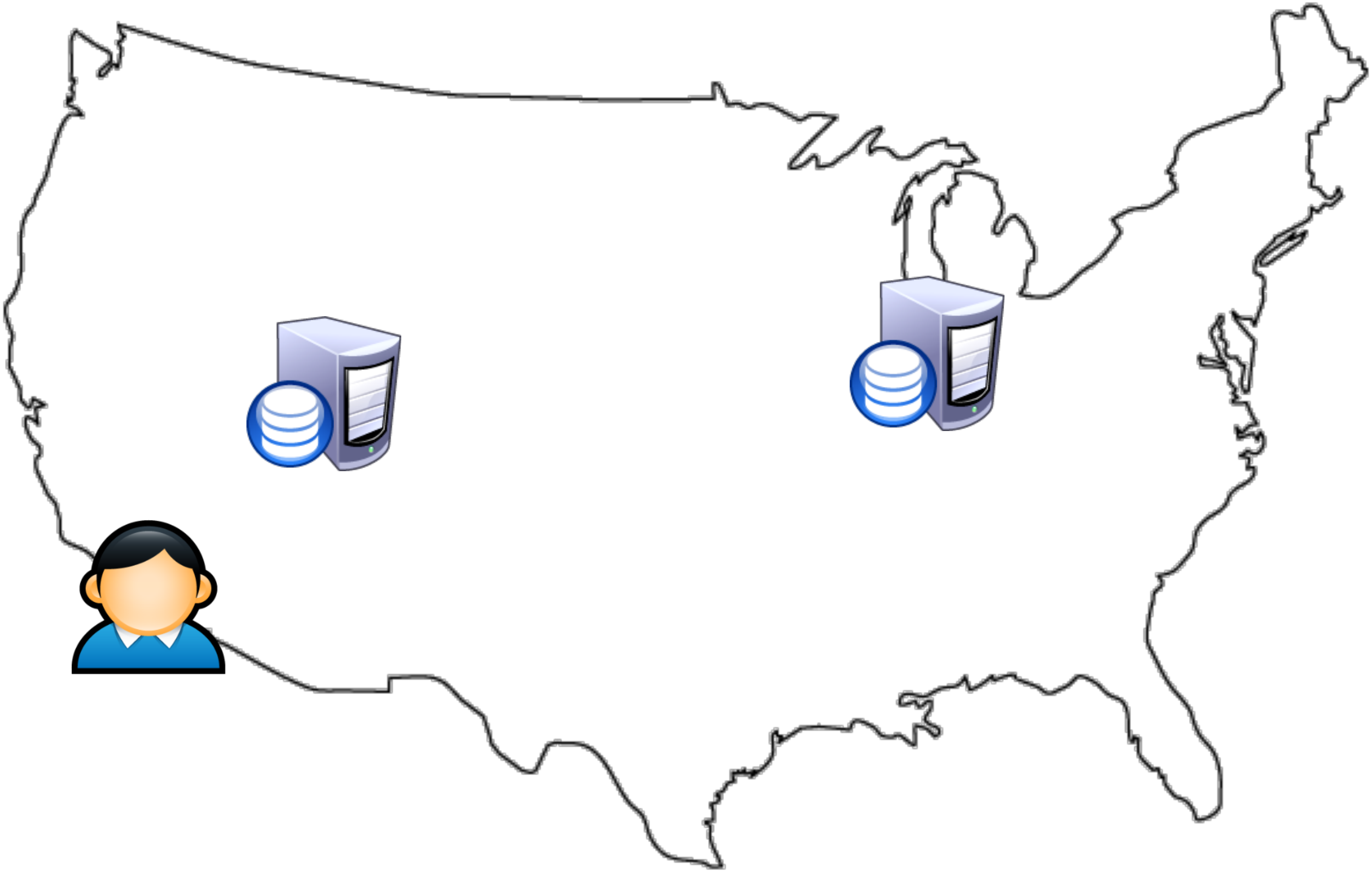
- One machine can have many aliases



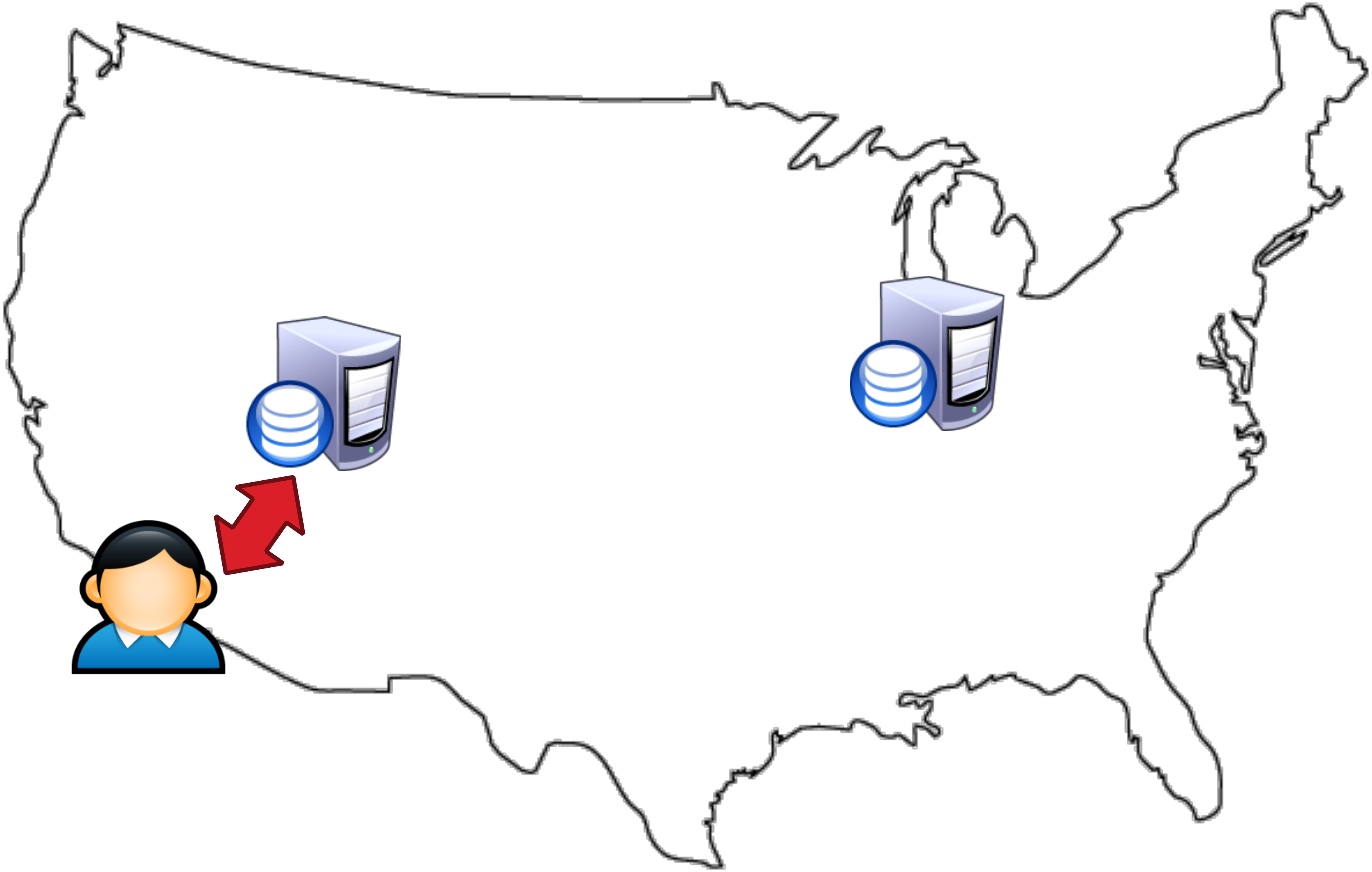
- One domain can map to multiple machines



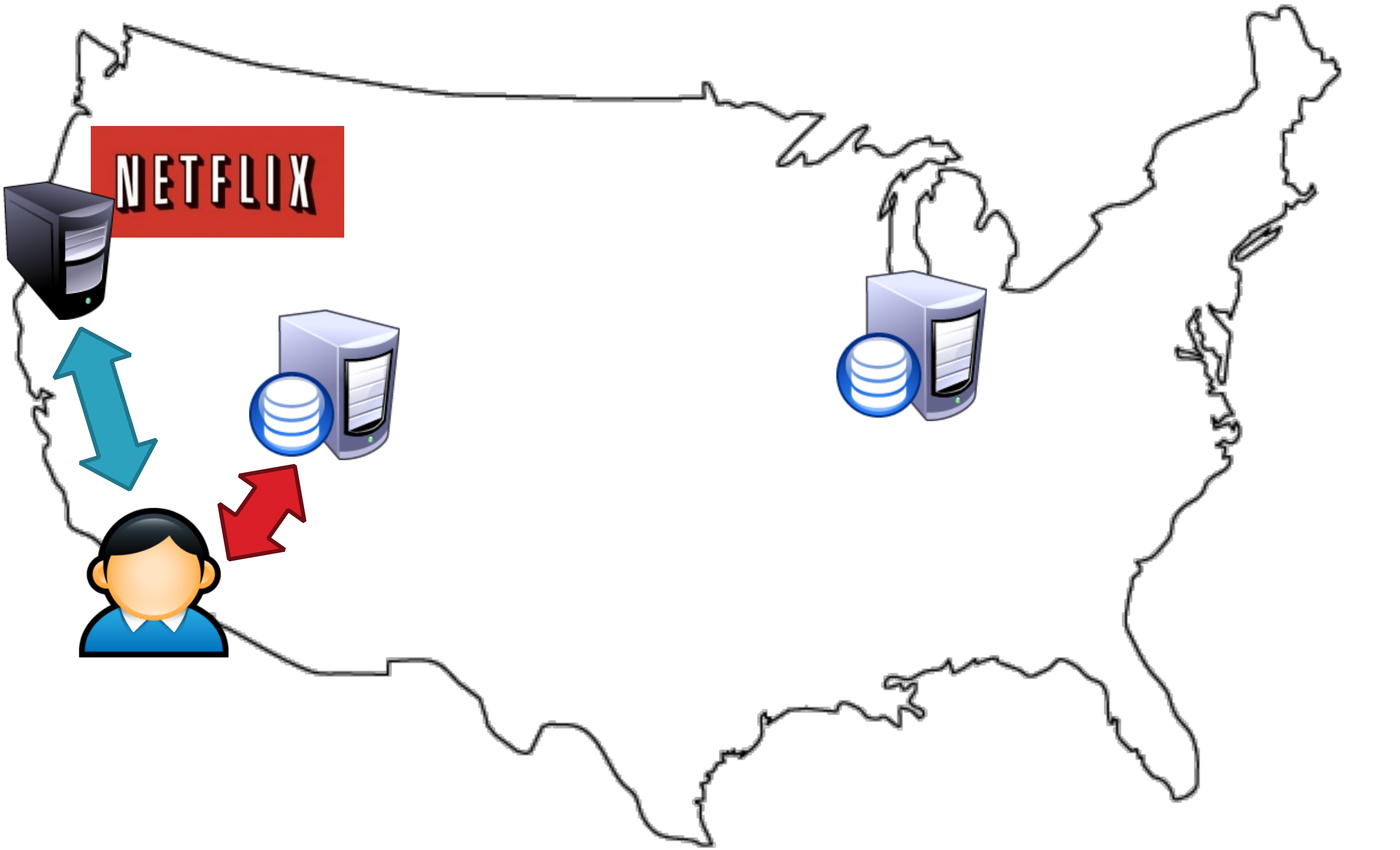
Content Delivery Networks



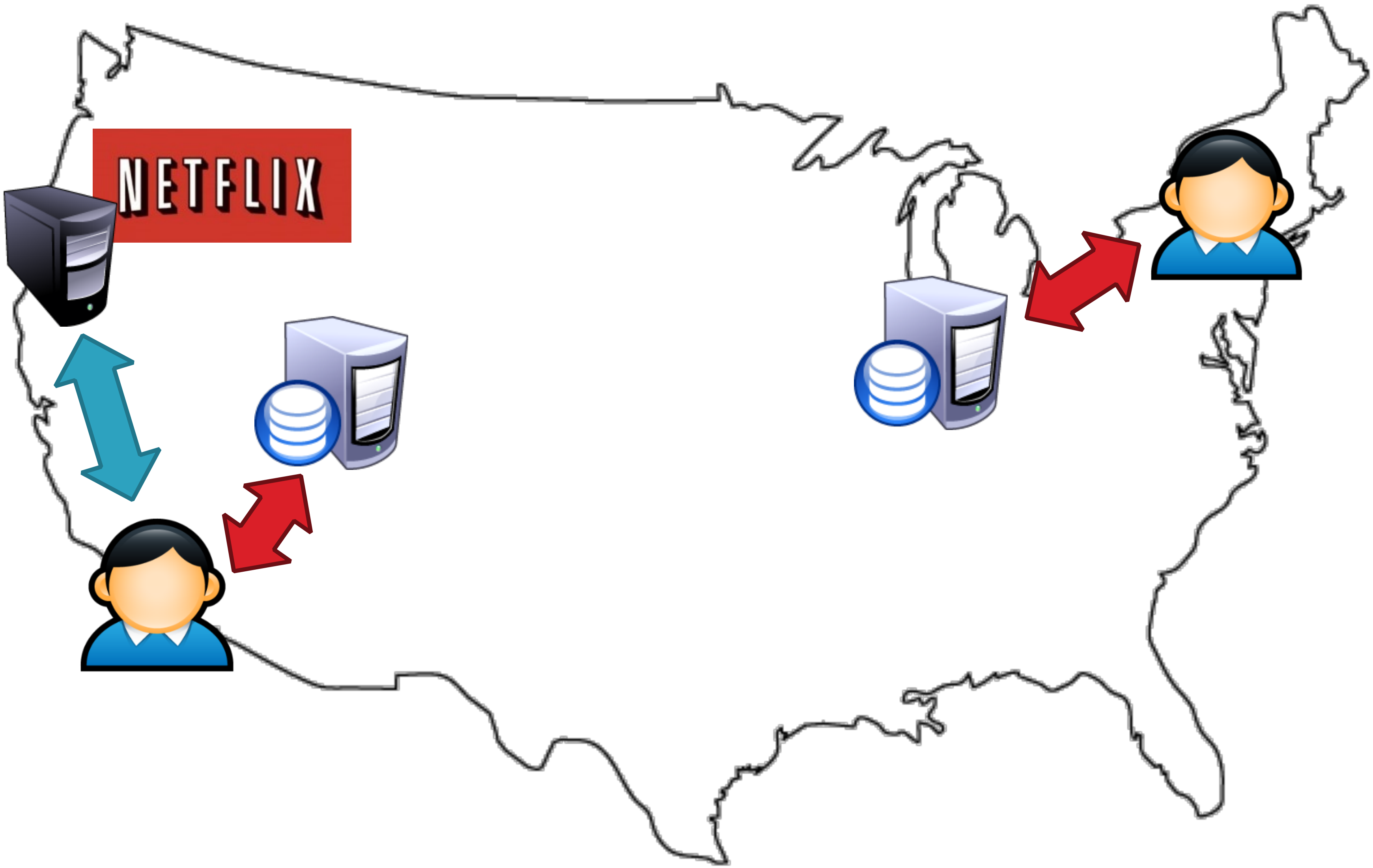
Content Delivery Networks



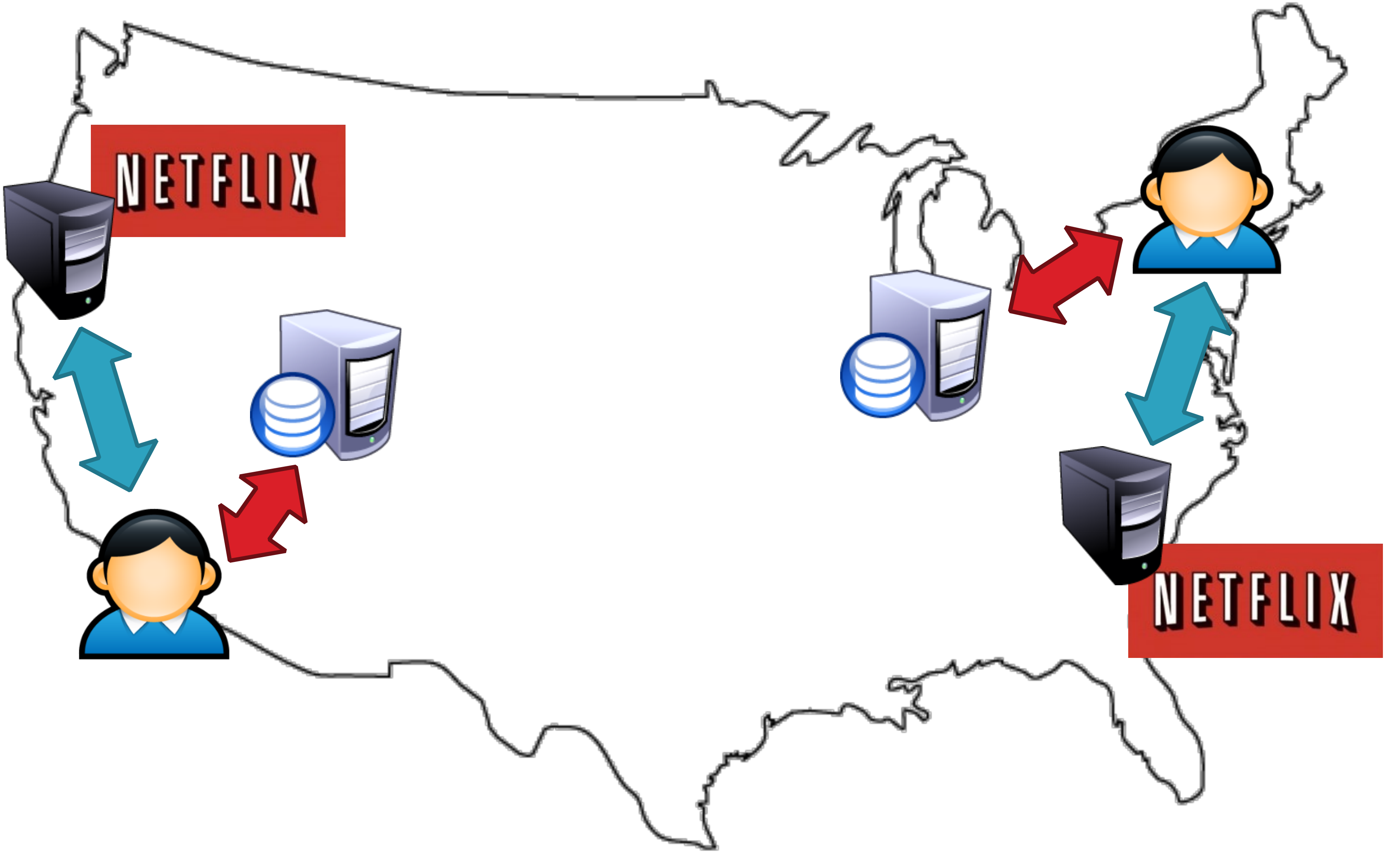
Content Delivery Networks



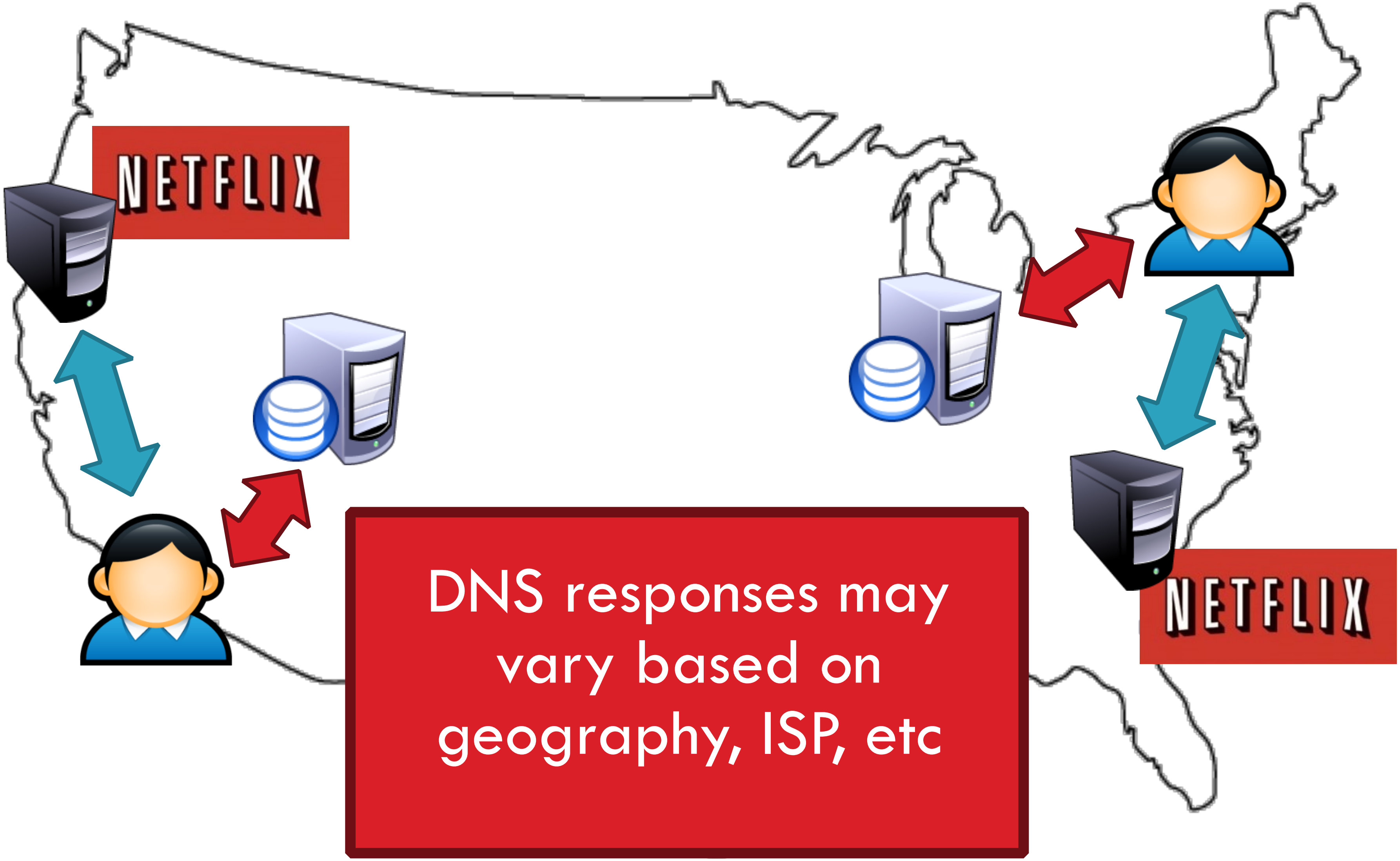
Content Delivery Networks



Content Delivery Networks



Content Delivery Networks



Outline

- DNS Basics
- DNS Security

The Importance of DNS

- Without DNS…
  - How could you get to any websites?

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?

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?

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

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 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

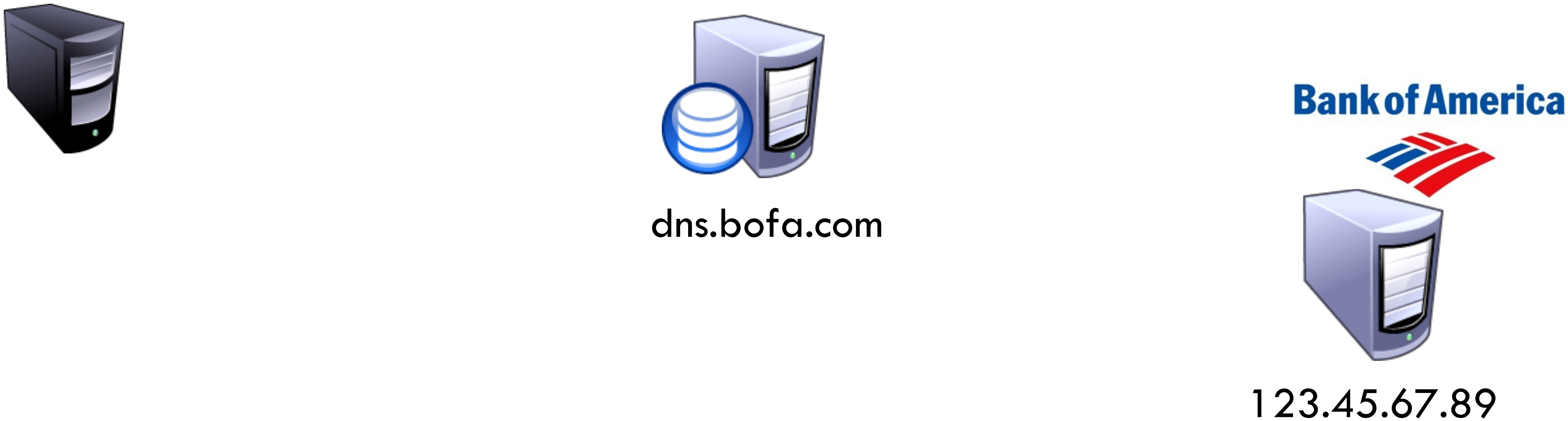
- Response Spoofing
  - Eavesdrop on requests
  - Outrace the servers response

DNS Hijacking

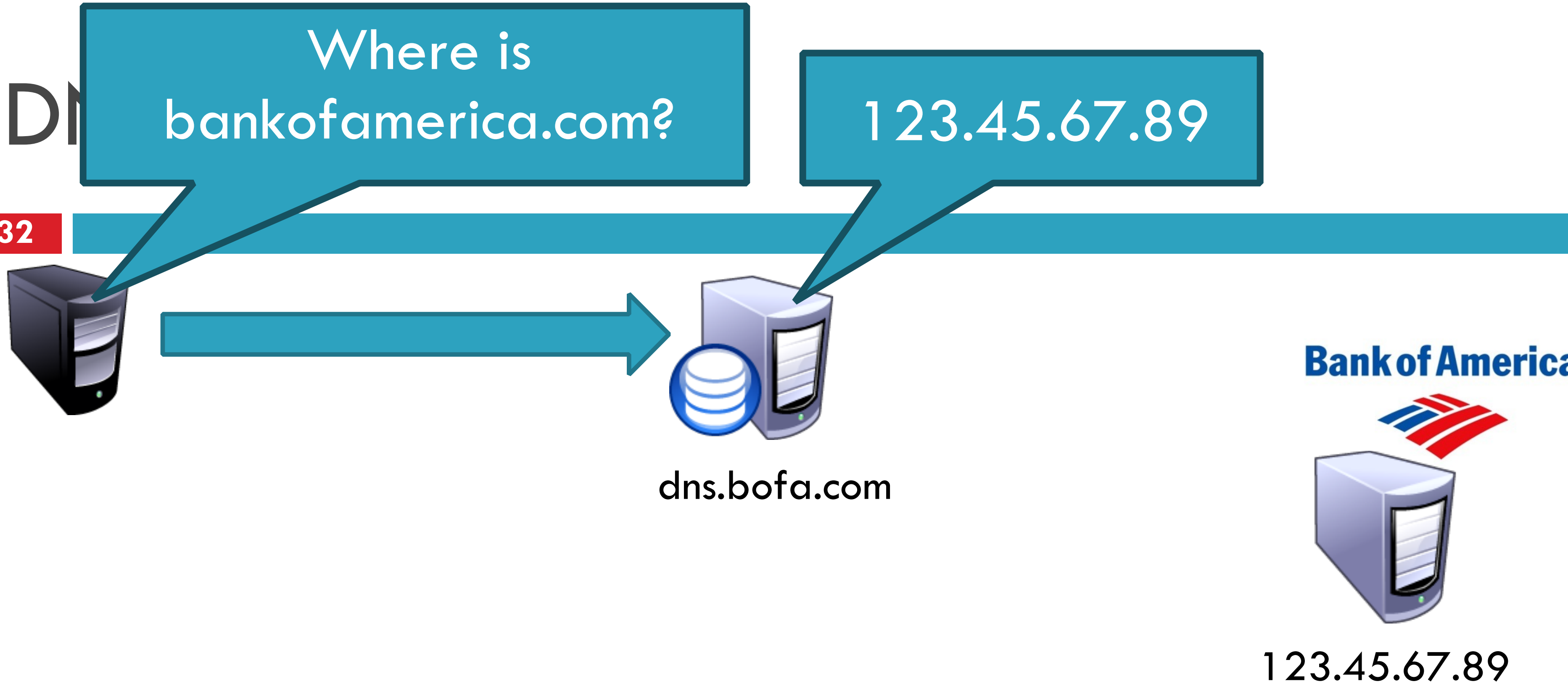
- Infect their OS or browser with a virus/trojan
  - e.g. Many trojans change entries in /etc/hosts
  - *.bankofamerica.com → evilbank.com
- Man-in-the-middle

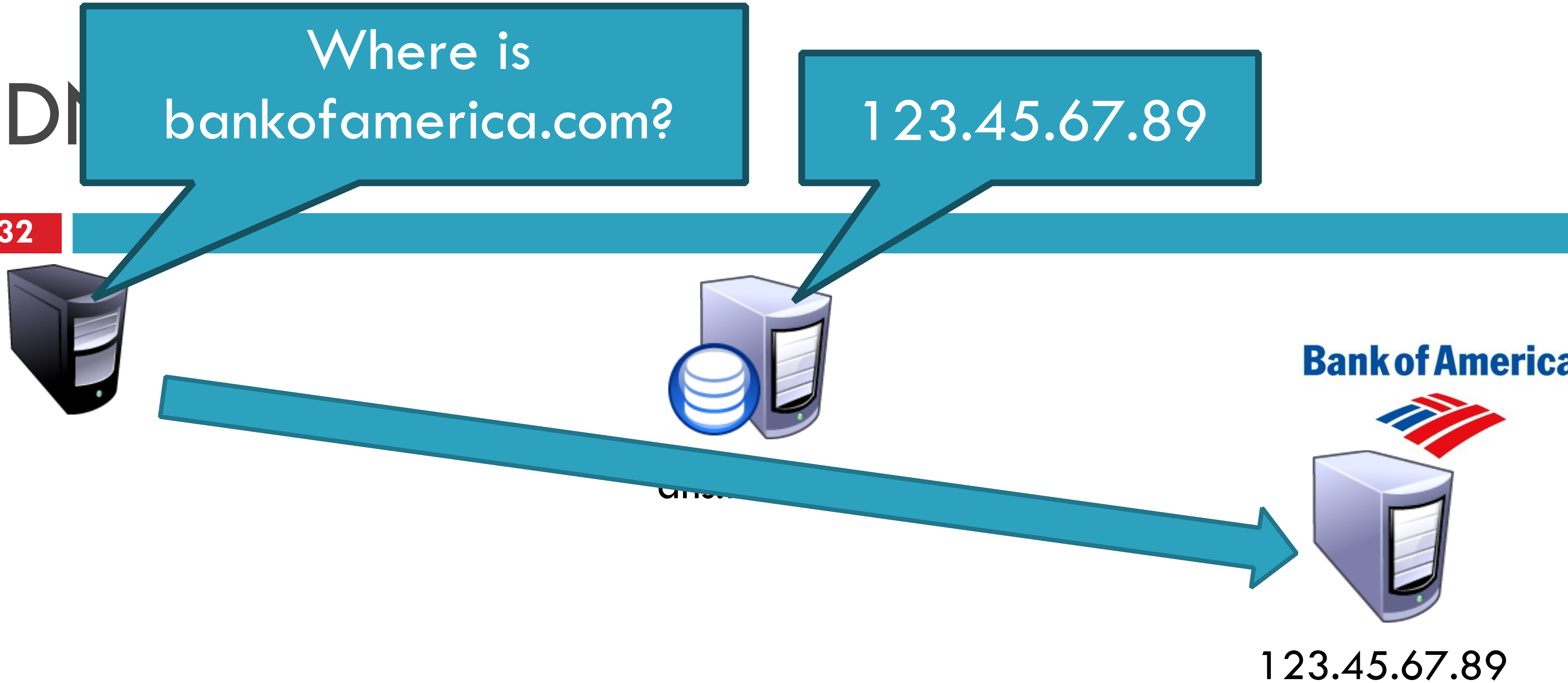
- Response Spoofing
  - Eavesdrop on requests
  - Outrace the servers response

DNS Spoofing

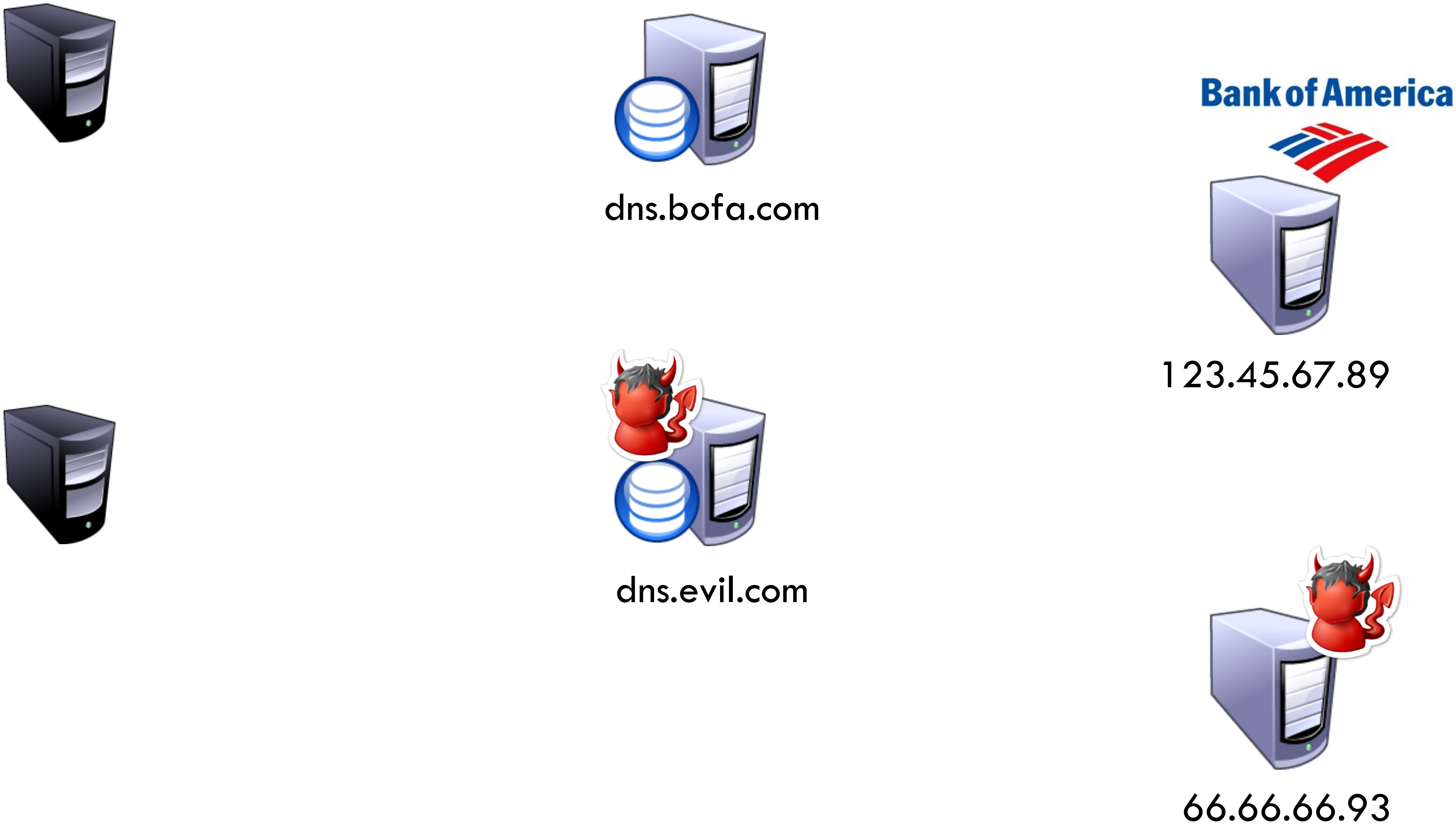


DN

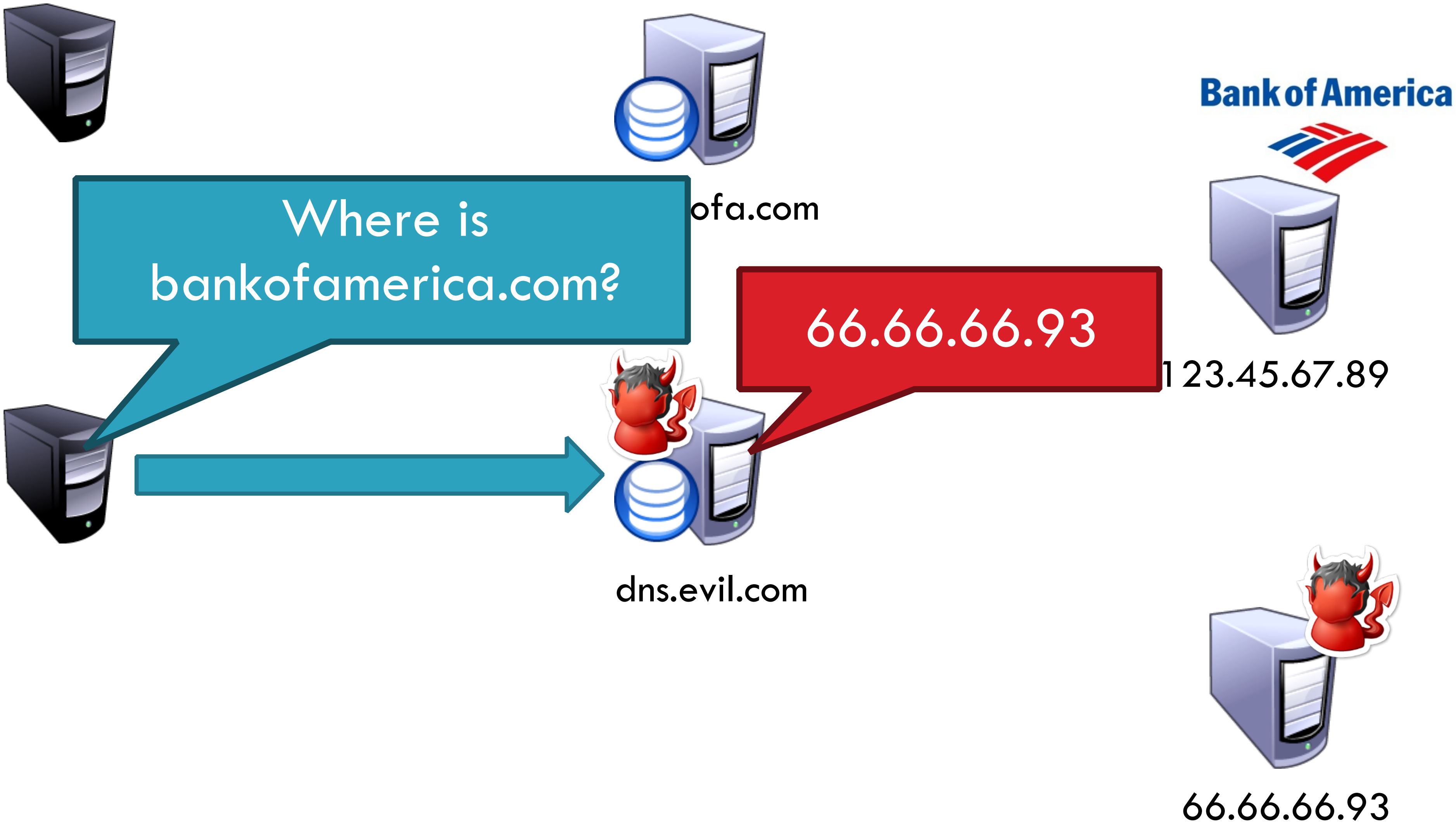

Where is bankofamerica.com?
123.45.67.89
dns.bofa.com
Bank of America
123.45.67.89


Where is bankofamerica.com?
123.45.67.89
Bank of America
123.45.67.89

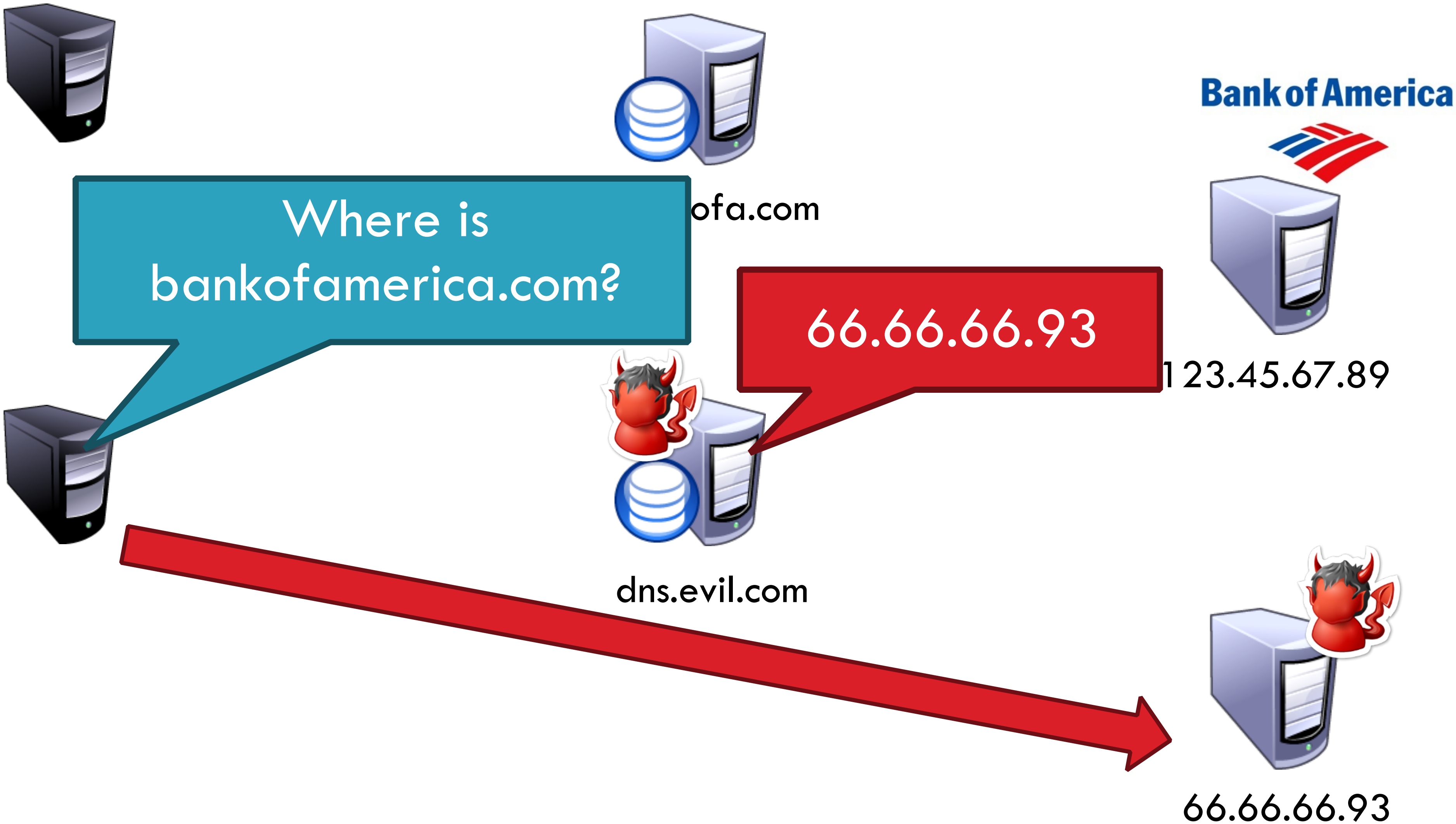
DNS Spoofing



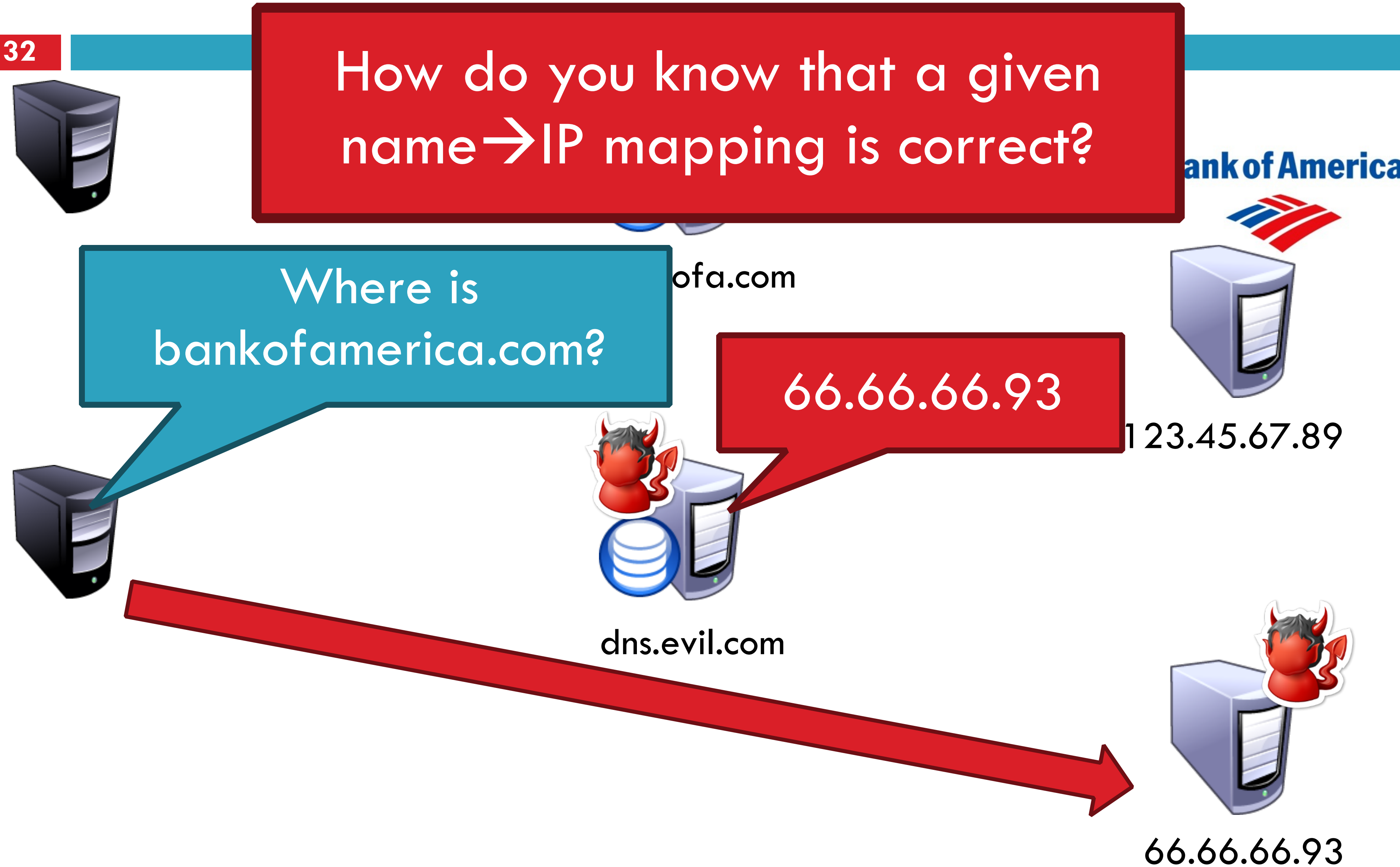
DNS Spoofing



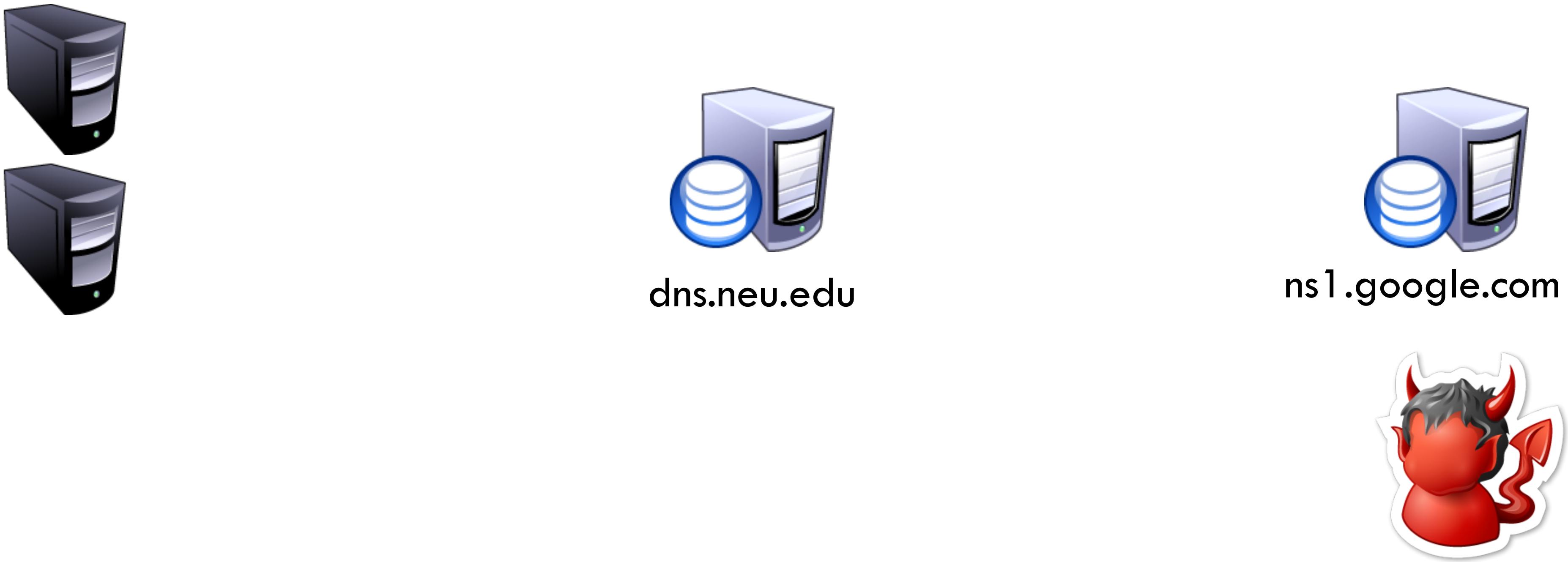
DNS Spoofing



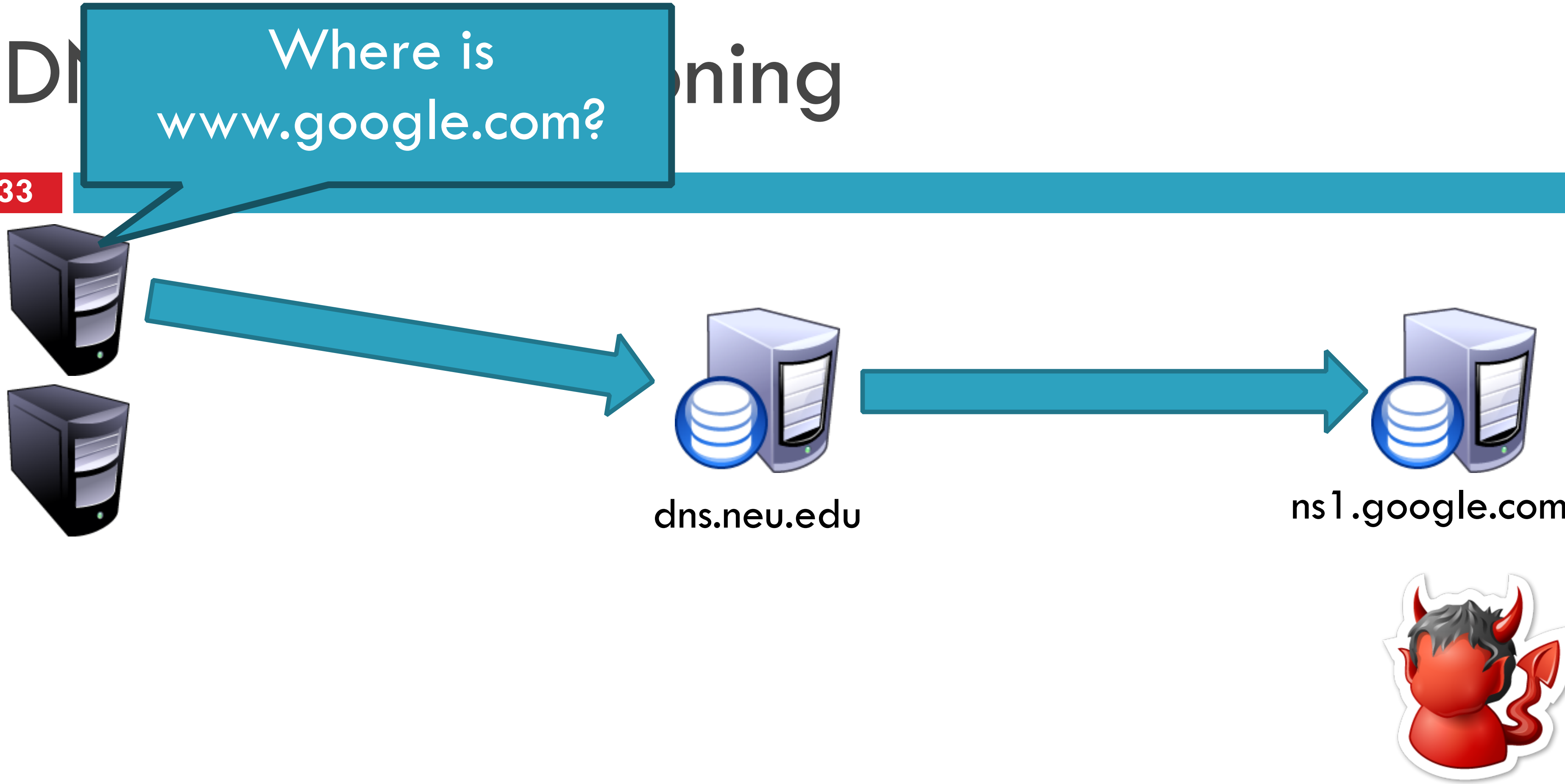
DNS Spoofing



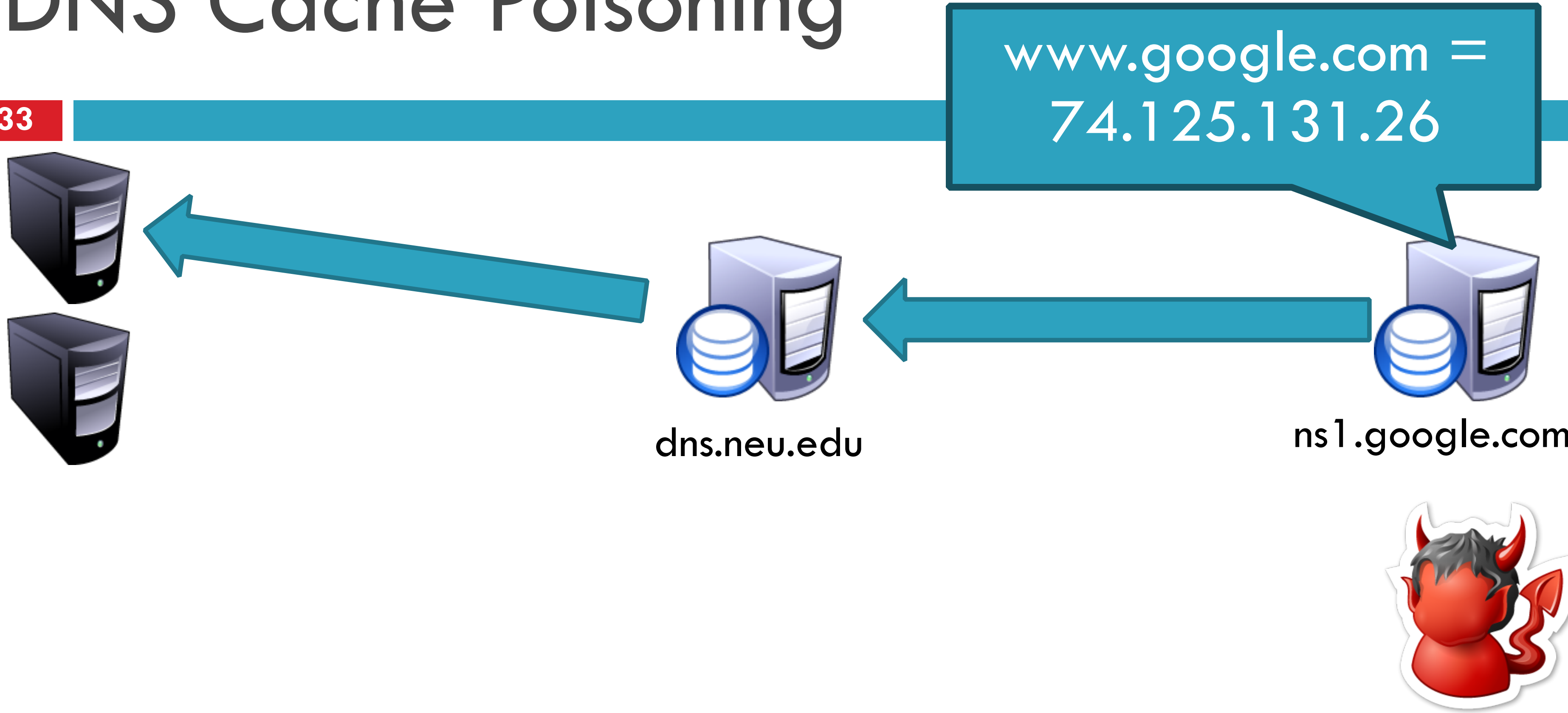
DNS Cache Poisoning



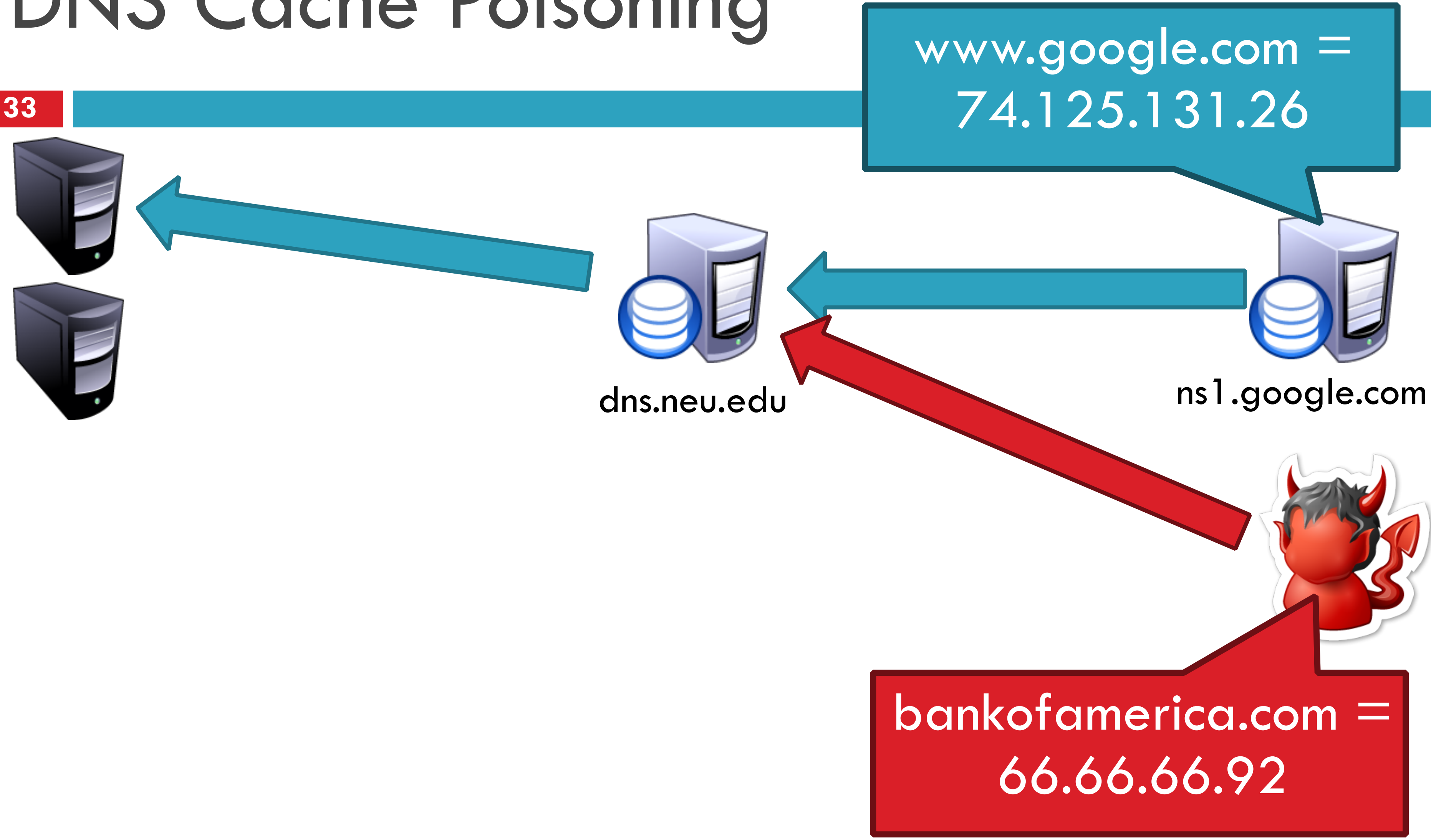
DN ning



DNS Cache Poisoning

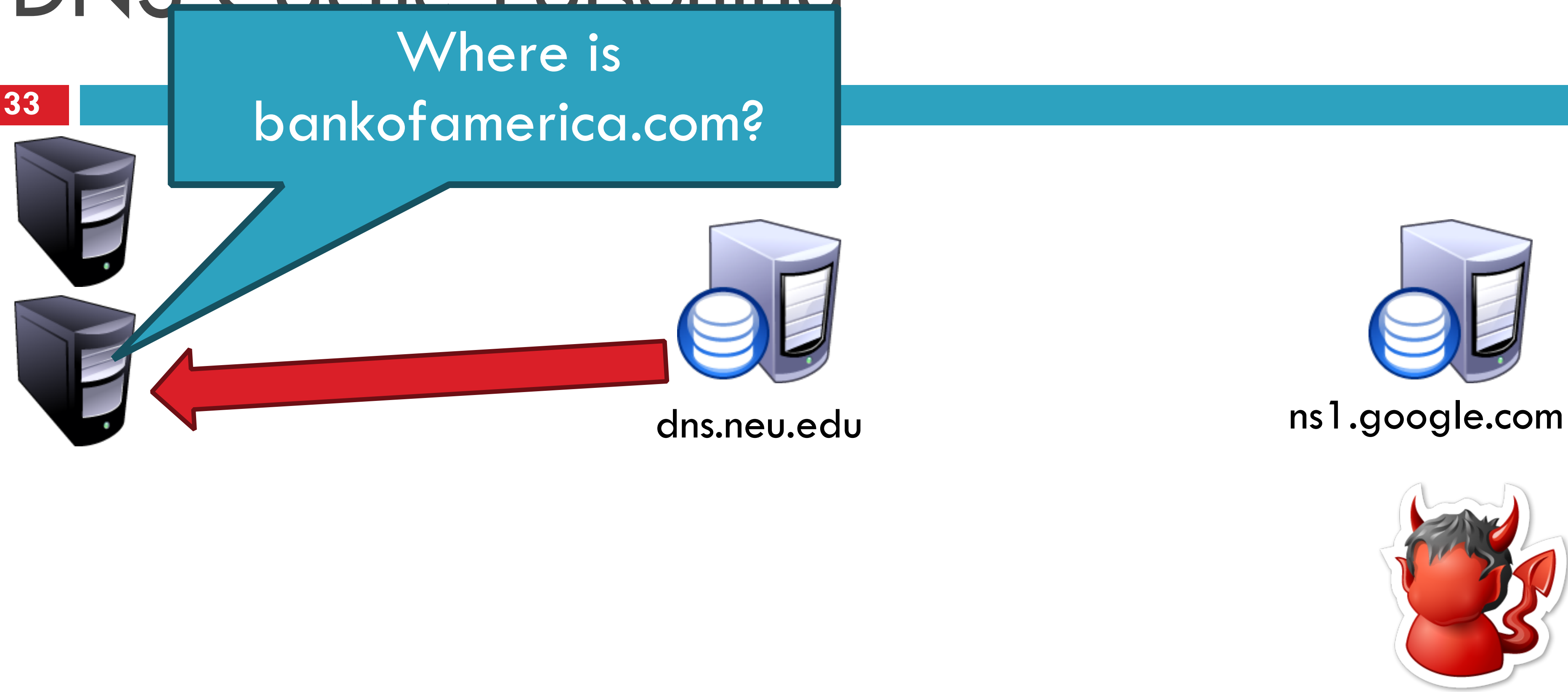


DNS Cache Poisoning

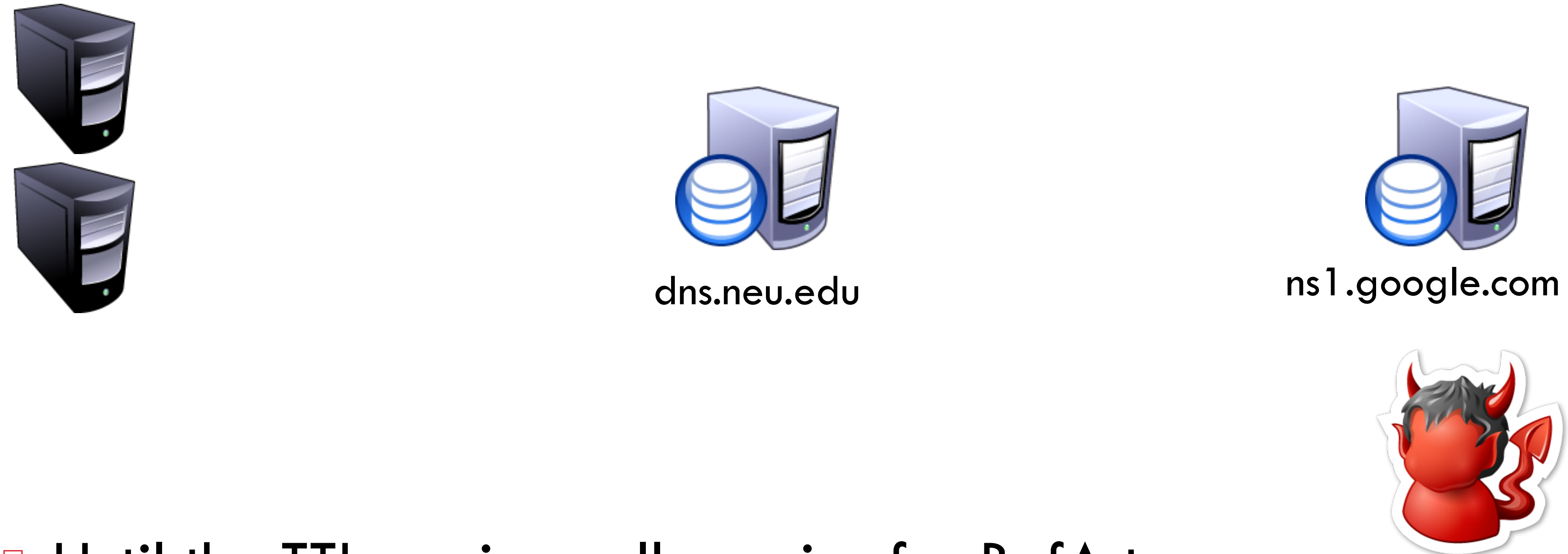


DNS Cache Poisoning

DNS Cache Poisoning



DNS Cache Poisoning



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

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

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

Creates a hierarchy of trust within each zone

Solution: DNSSEC

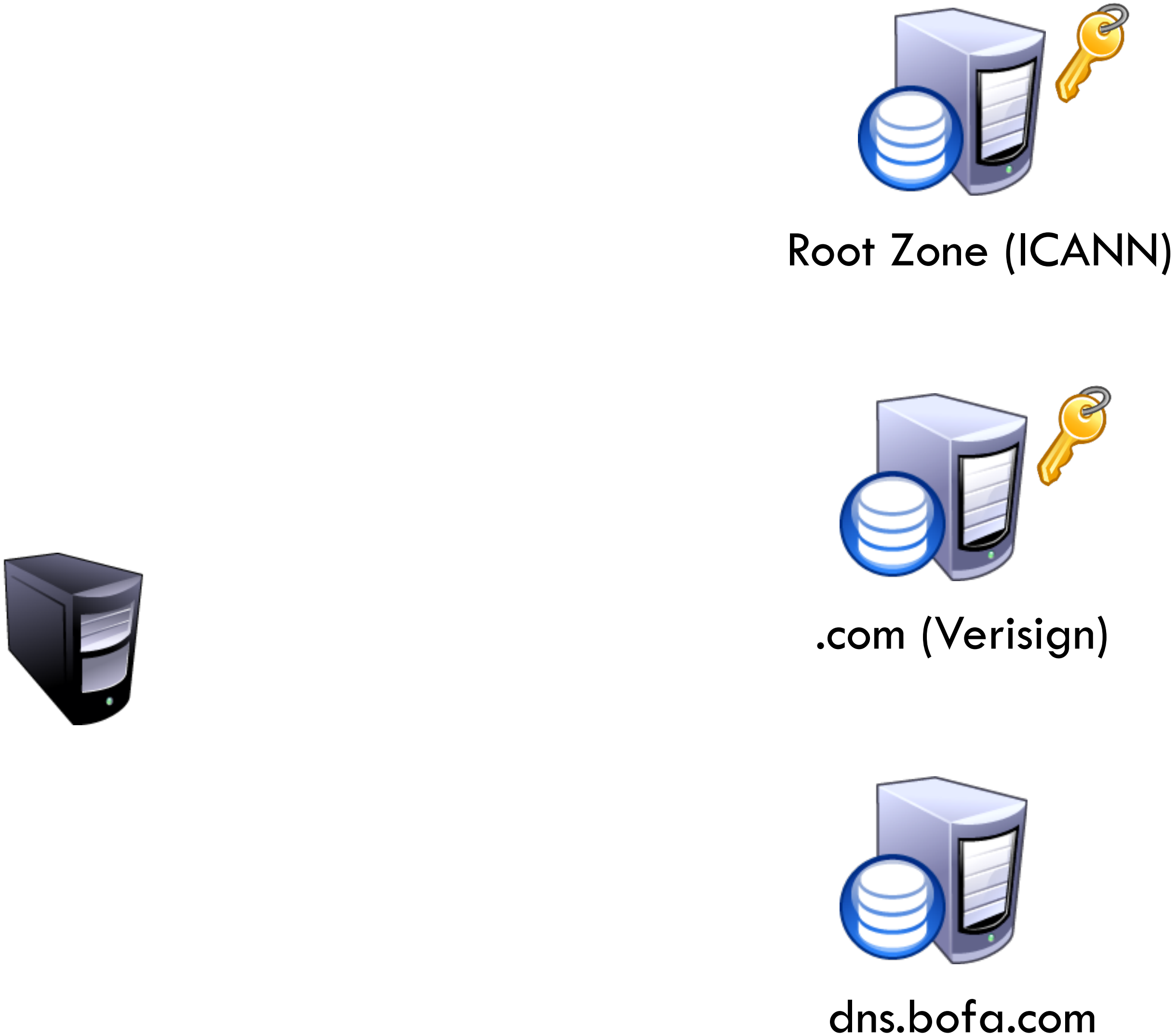
- Cryptographically sign critical resource records
  - Resolver can verify the cryptographic signature
- Two new resource types
  - Type = DNSKEY
    - Name = Zone domain name
    - Value = Public key for the zone
  - Type = RRSIG
    - Name = (type, name) tuple, i.e. the query itself
    - Value = Cryptographic signature of the query results

Prevents hijacking and spoofing

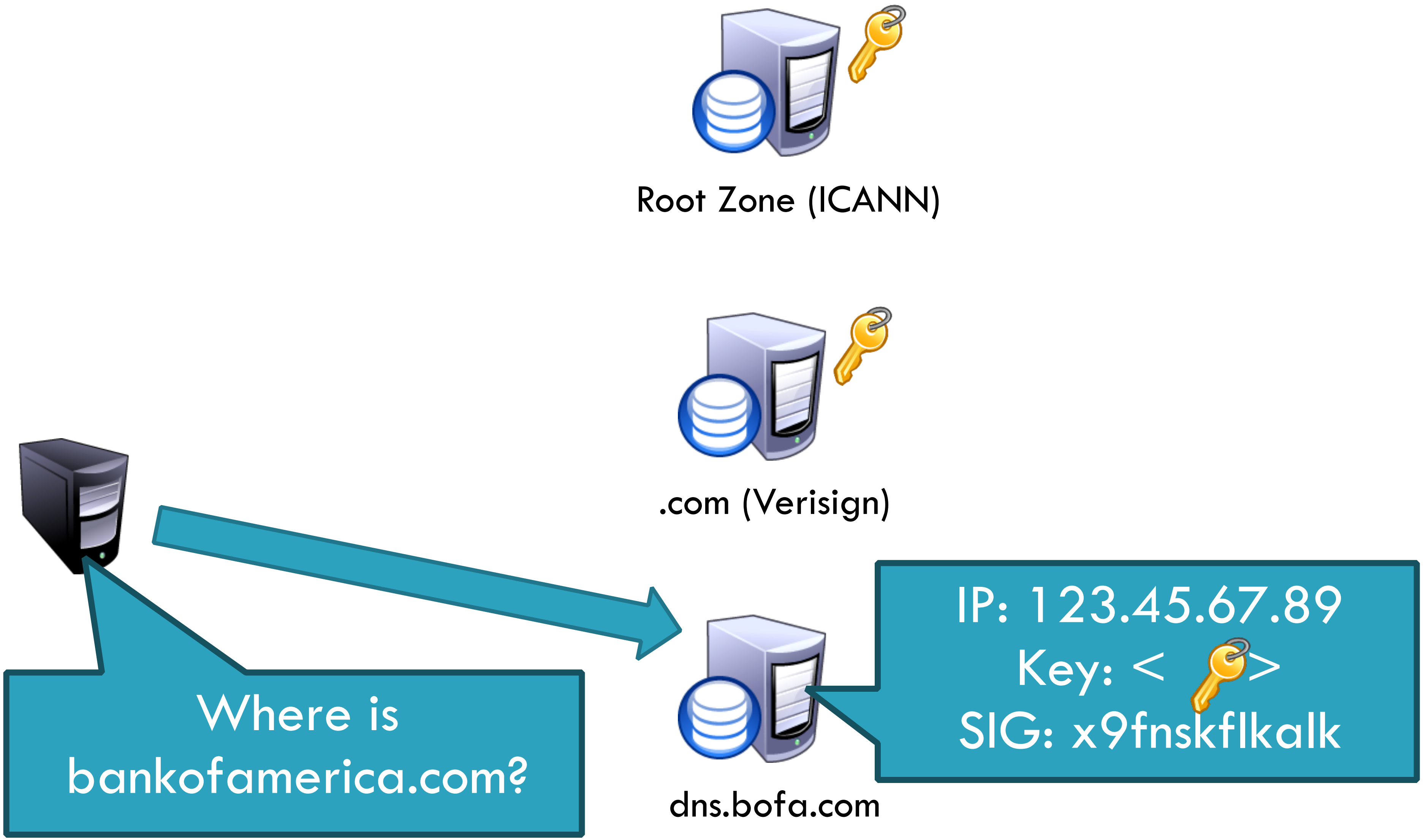
Solution: DNSSEC

- 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)

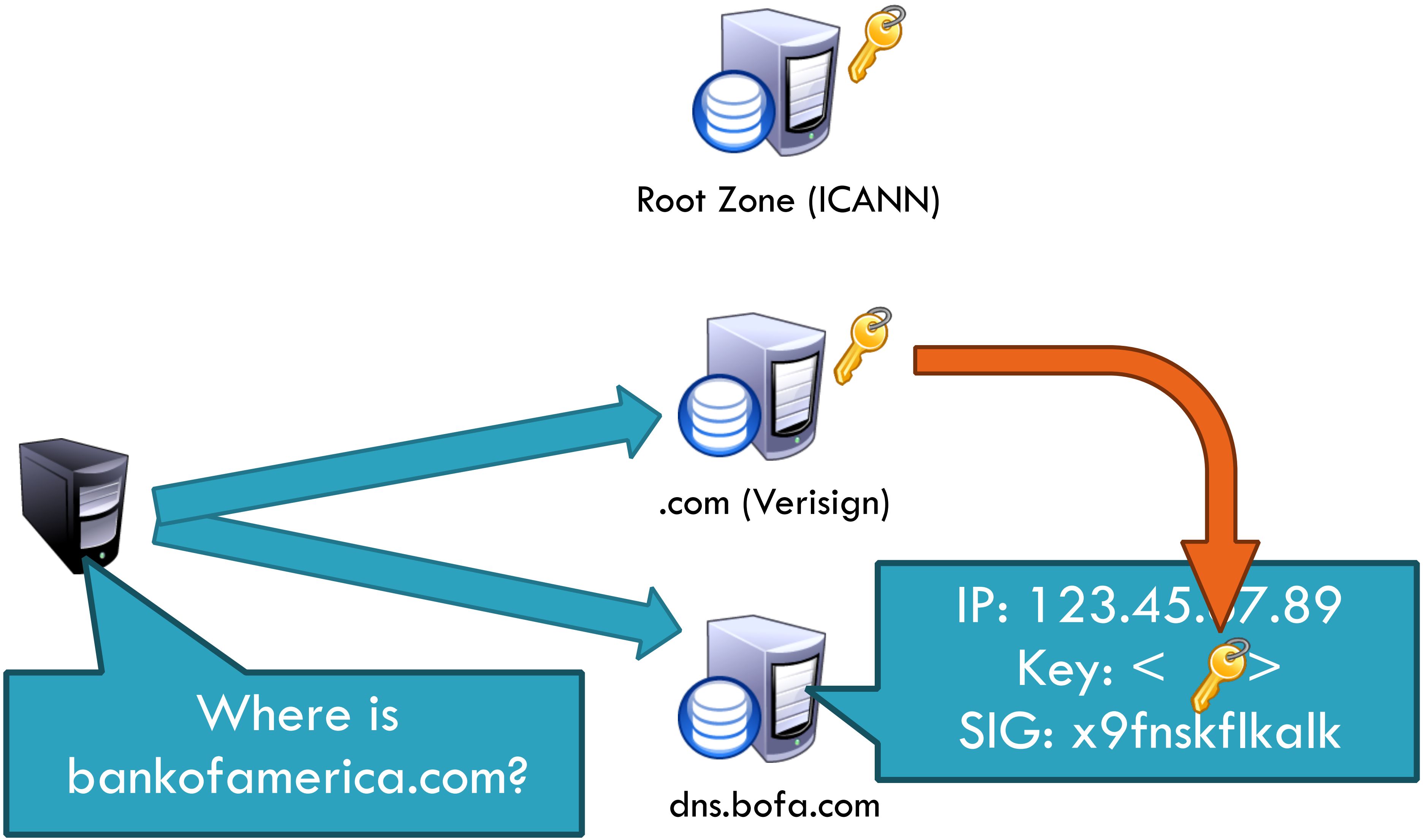
DNSSEC Hierarchy of Trust



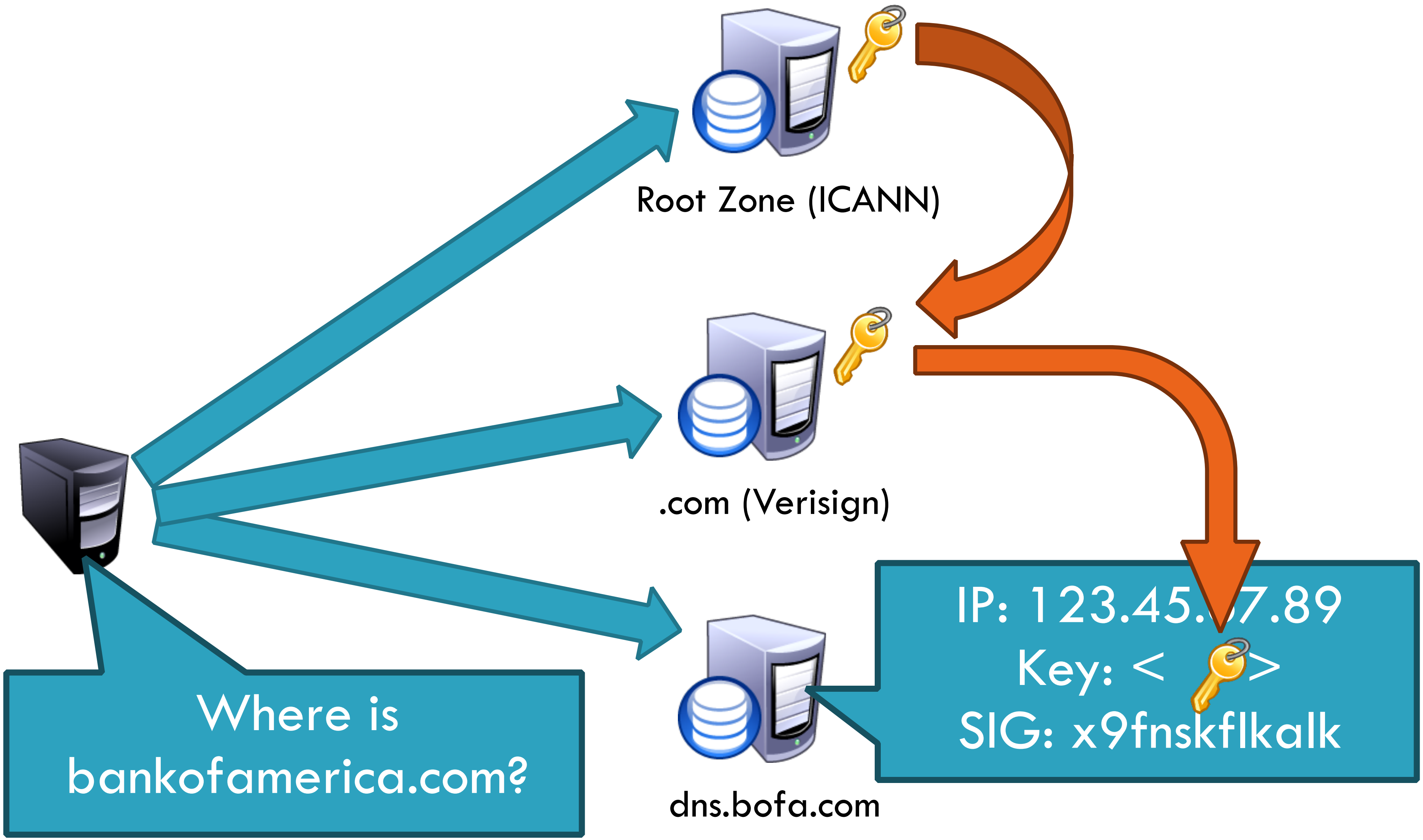
DNSSEC Hierarchy of Trust



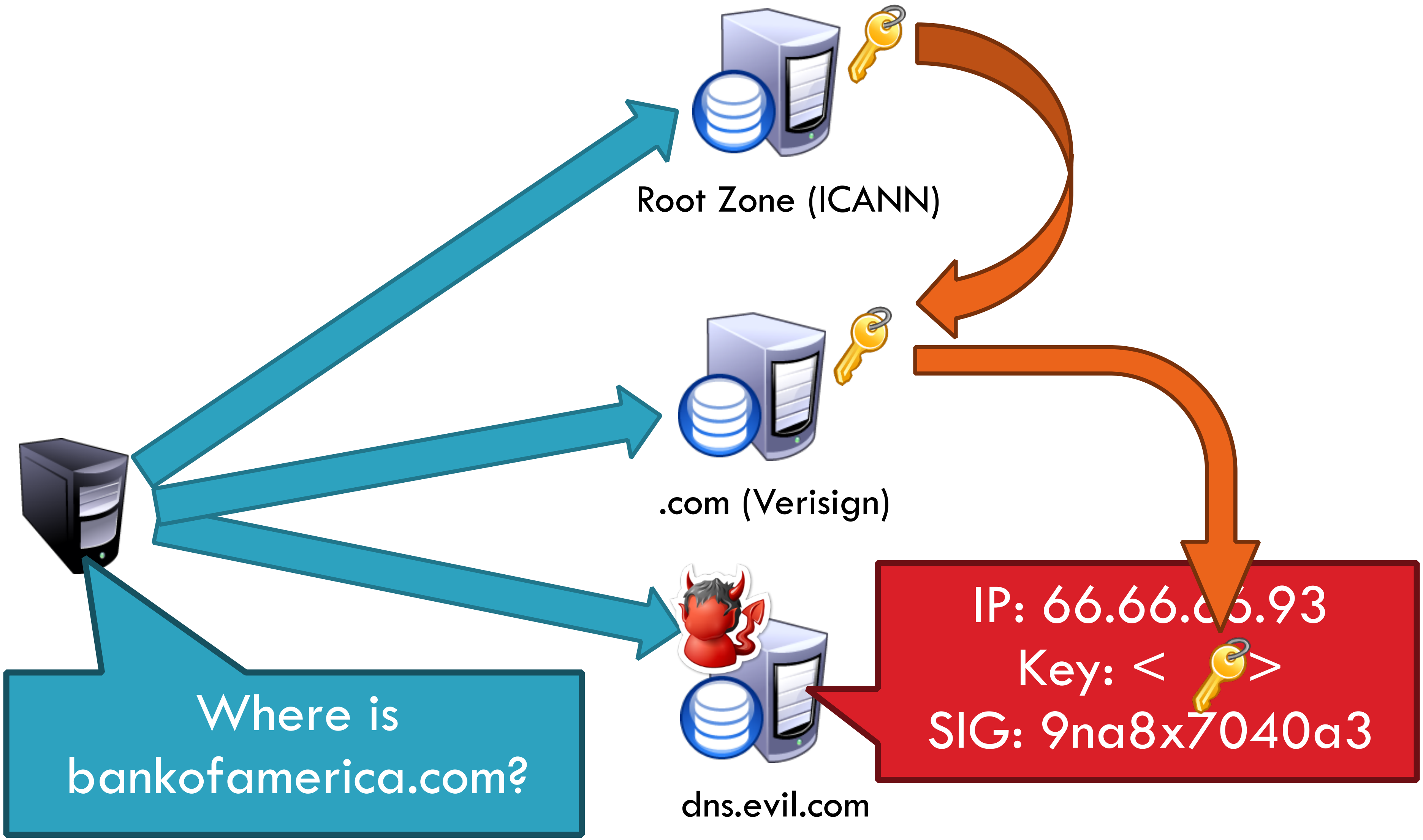
DNSSEC Hierarchy of Trust



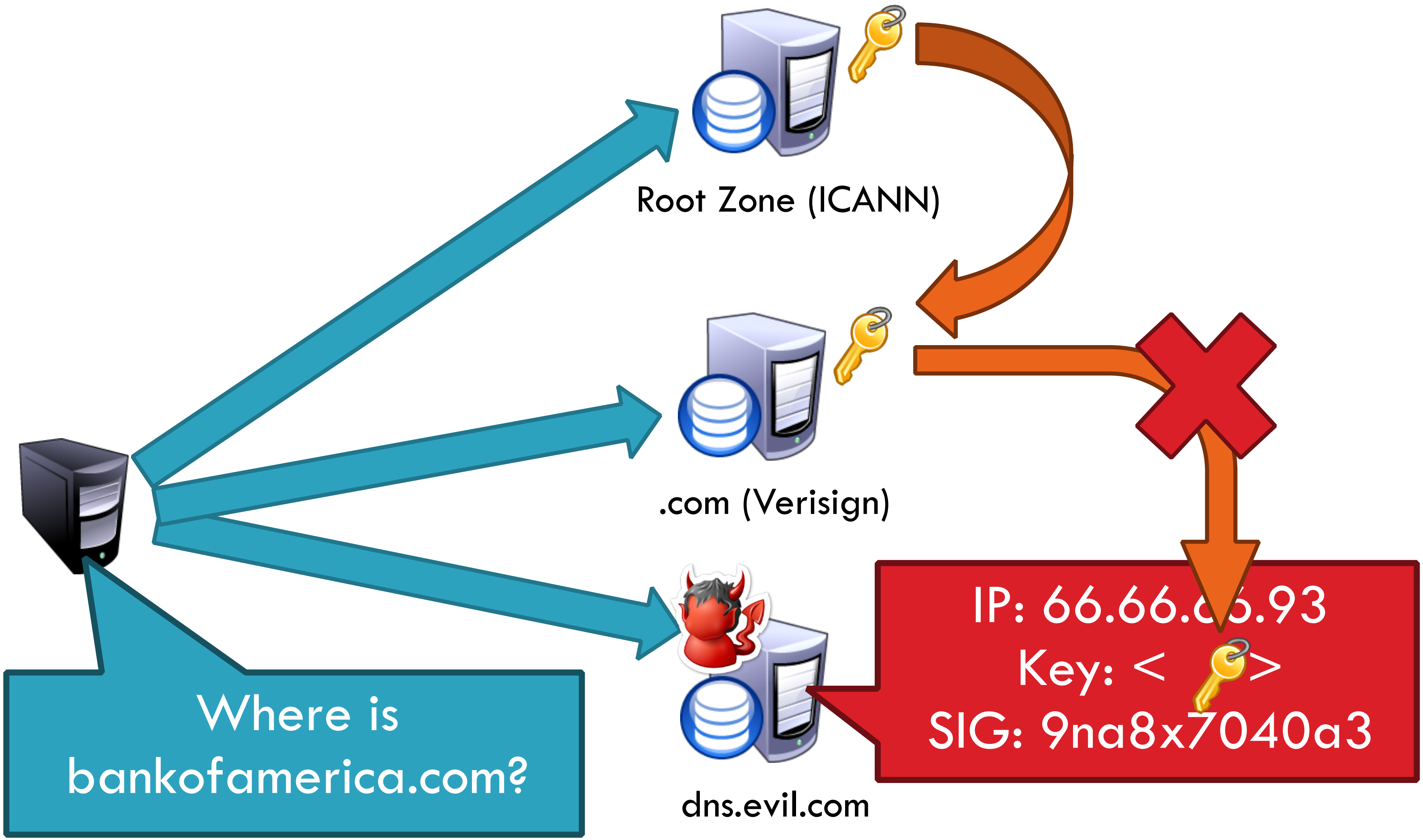
DNSSEC Hierarchy of Trust



DNSSEC Hierarchy of Trust



DNSSEC Hierarchy of Trust

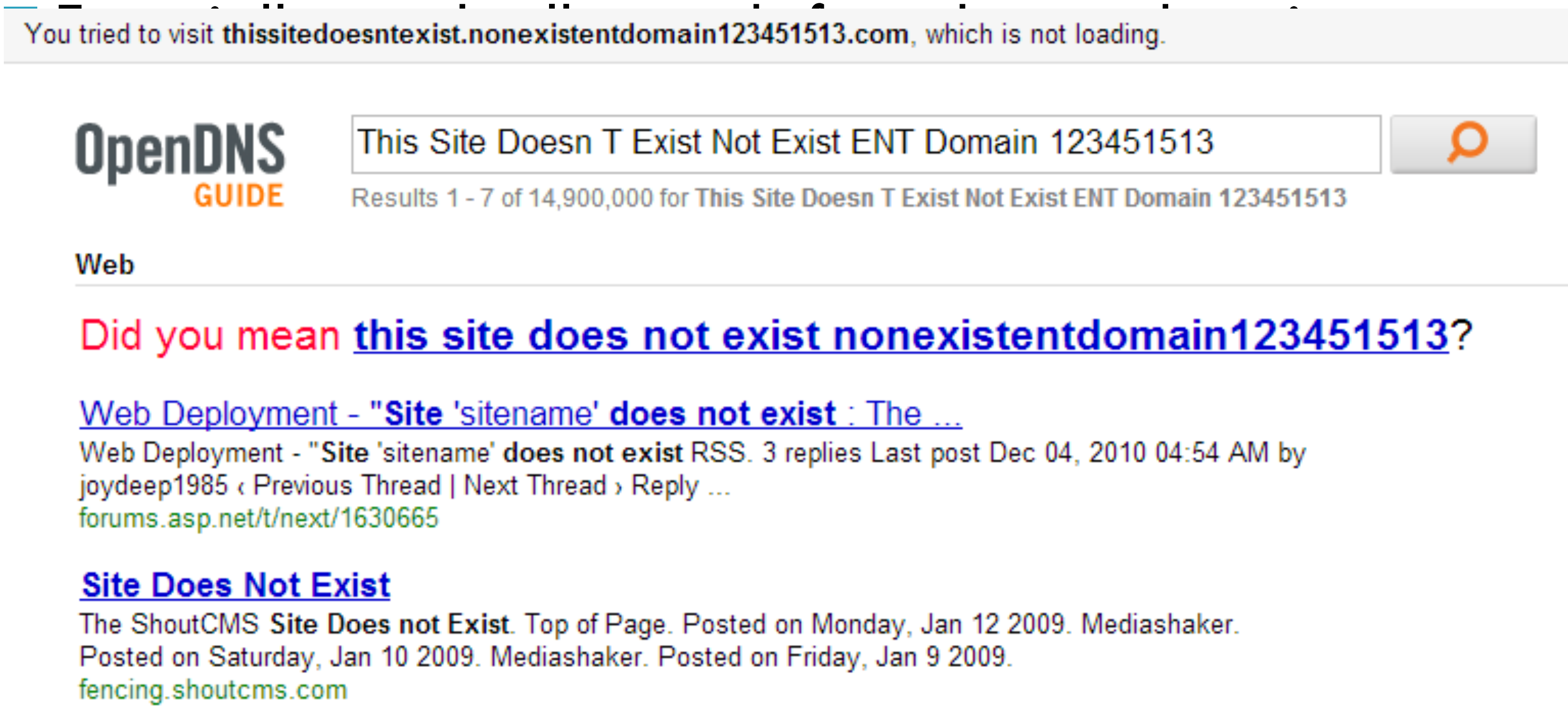


Site Finder

- September 2003: Verisign created DNS wildcards for *.com and *.net
  - Essentially, catch-all records for unknown domains
  - Pointed to a search website run by Verisign
  - Search website was full of advertisements

Site Finder

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



Site Finder

- September 2003: Verisign created DNS wildcards for *.com and *.net
  - Essentially, catch-all records for unknown domains
  - Pointed to a search website run by Verisign
  - Search website was full of advertisements

Site Finder

- September 2003: Verisign created DNS wildcards for *.com and *.net
  - Essentially, catch-all records for unknown domains
  - Pointed to a search website run by Verisign
  - Search website was full of advertisements
- Extremely controversial move
  - Is this DNS hijacking?
  - Definitely abuse of trust by Verisign
  - Site Finder was quickly shut down, lawsuits ensued

Much More to DNS

- Caching: when, where, how much, etc.

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)

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

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

Outline

- DNS
- NAT
- Other middleboxes

The IPv4 Shortage

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

The IPv4 Shortage

- Problem: consumer ISPs typically only give one IP address per-household
  - Additional IPs cost extra
  - More IPs may not be available
- Today's households have more networked devices than ever
  - Laptops and desktops
  - TV, bluray players, game consoles
  - Tablets, smartphones, eReaders

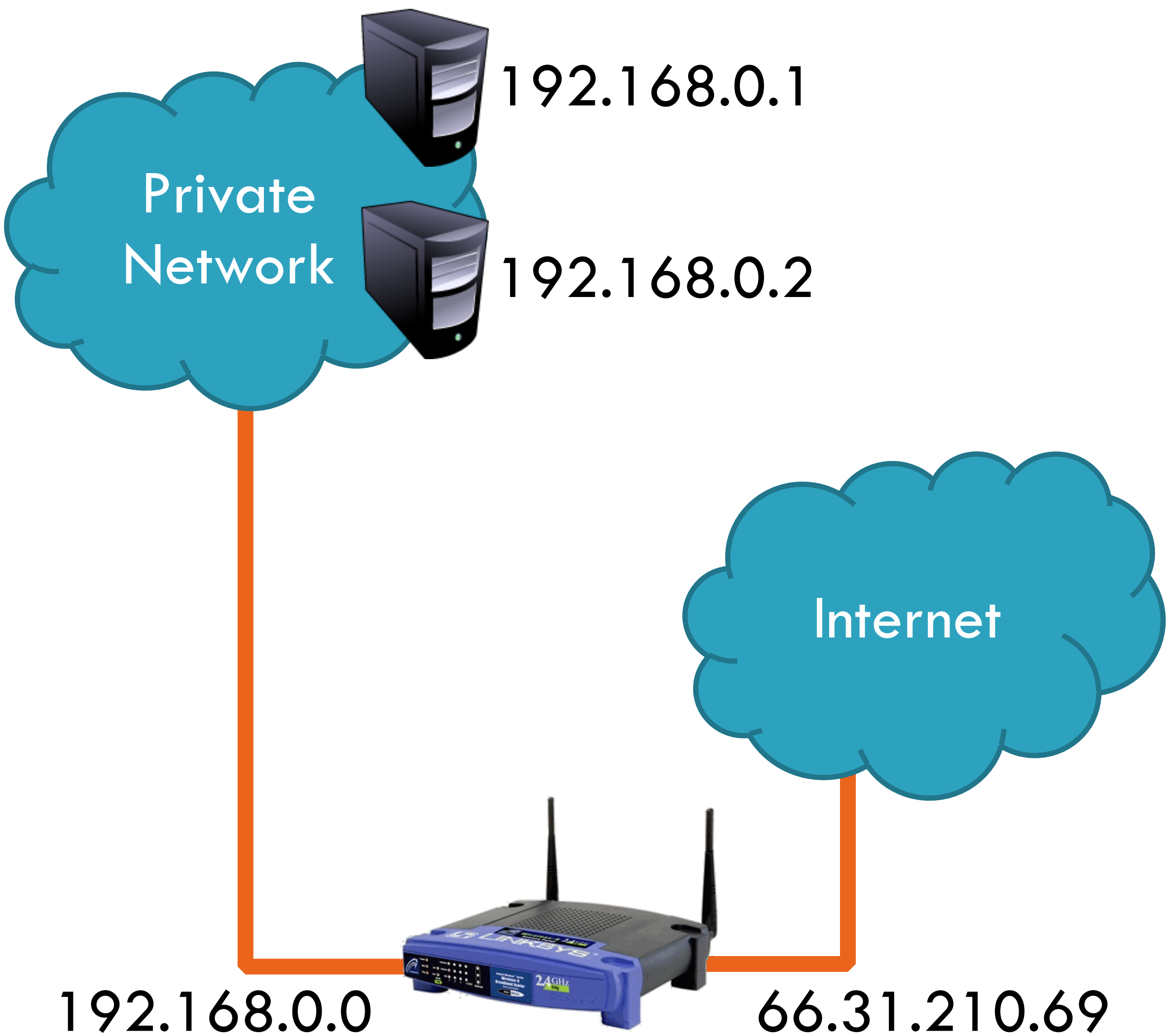
The IPv4 Shortage

- Problem: consumer ISPs typically only give one IP address per-household
  - Additional IPs cost extra
  - More IPs may not be available
- Today's households have more networked devices than ever
  - Laptops and desktops
  - TV, bluray players, game consoles
  - Tablets, smartphones, eReaders
- How to get all these devices online?

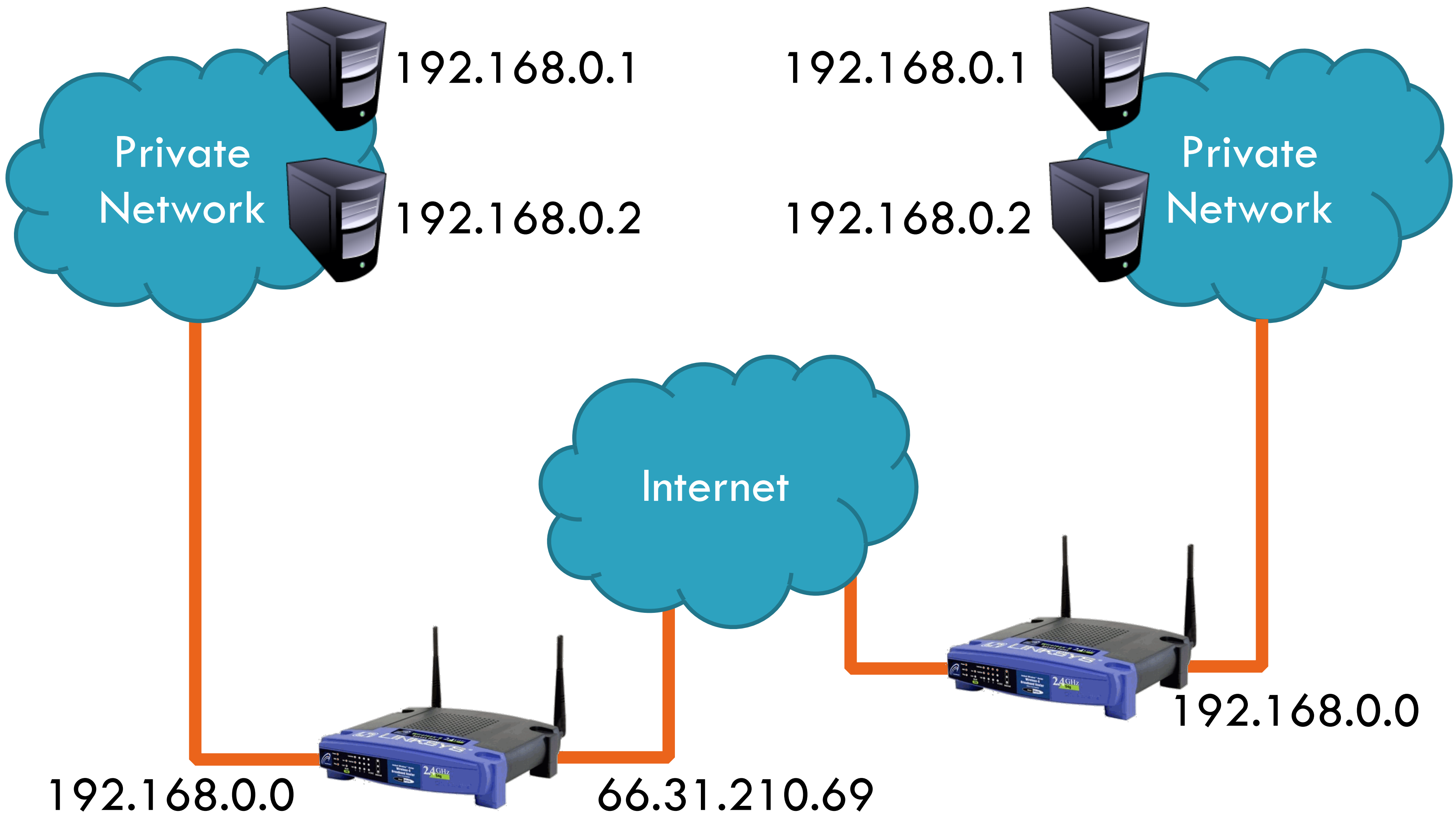
Private IP Networks

- Idea: create a range of private IPs that are separate from the rest of the network
  - Use the private IPs for internal routing
  - Use a special router to bridge the LAN and the WAN
- Properties of private IPs
  - Not globally unique
  - Usually taken from non-routable IP ranges (why?)
- Typical private IP ranges
  - 10.0.0.0 – 10.255.255.255
  - 172.16.0.0 – 172.31.255.255
  - 192.168.0.0 – 192.168.255.255

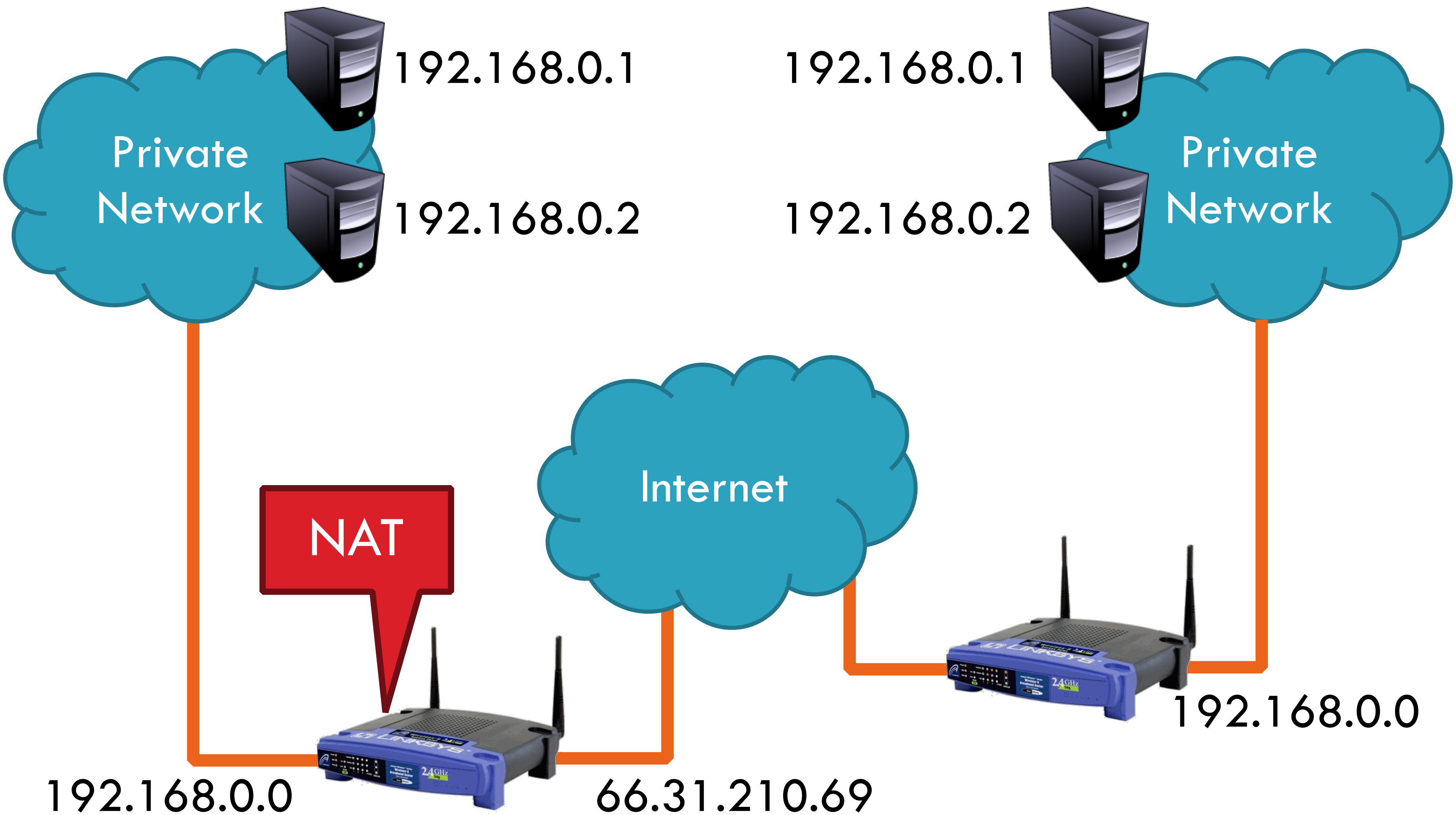
Private Networks



Private Networks



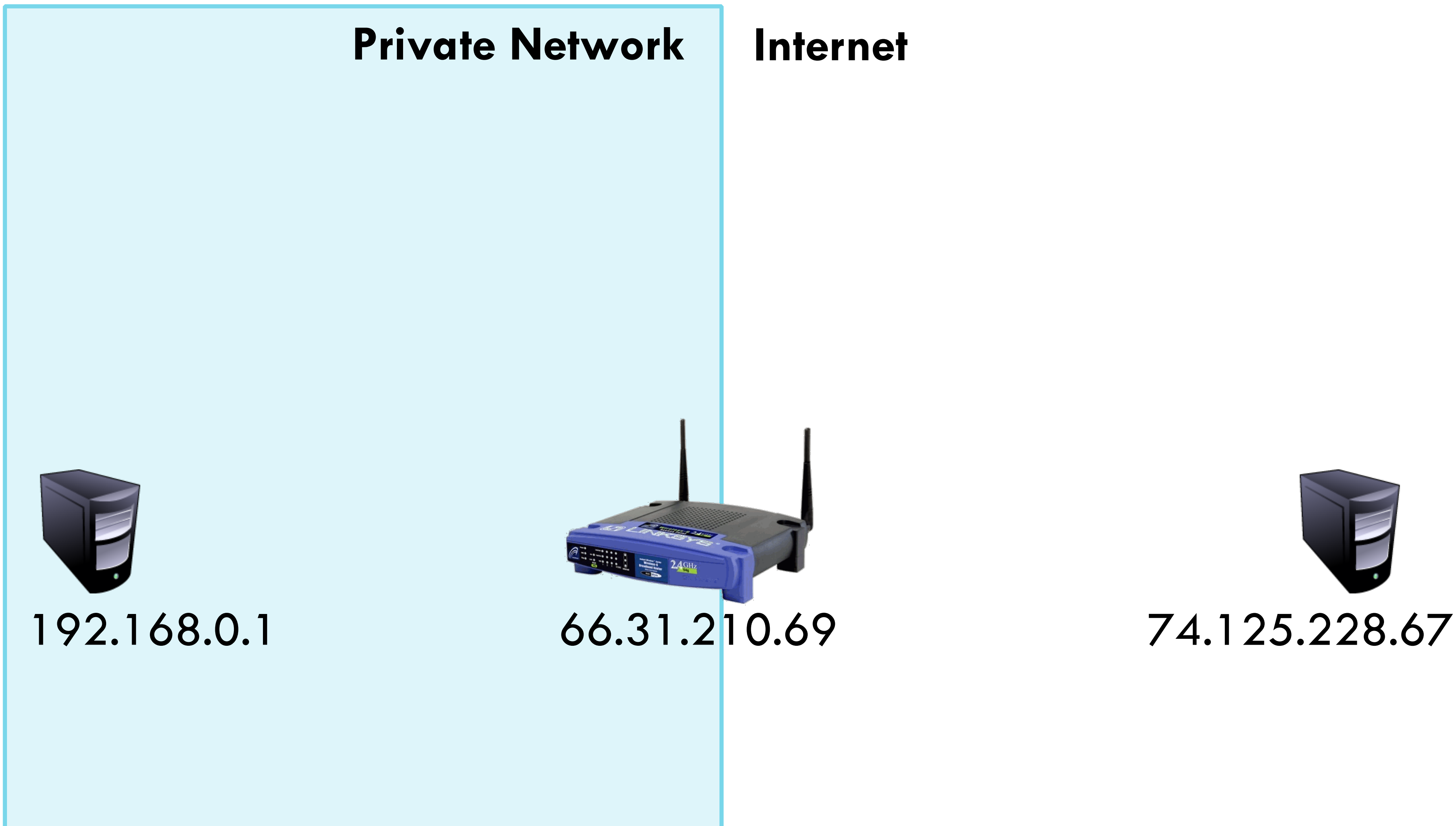
Private Networks



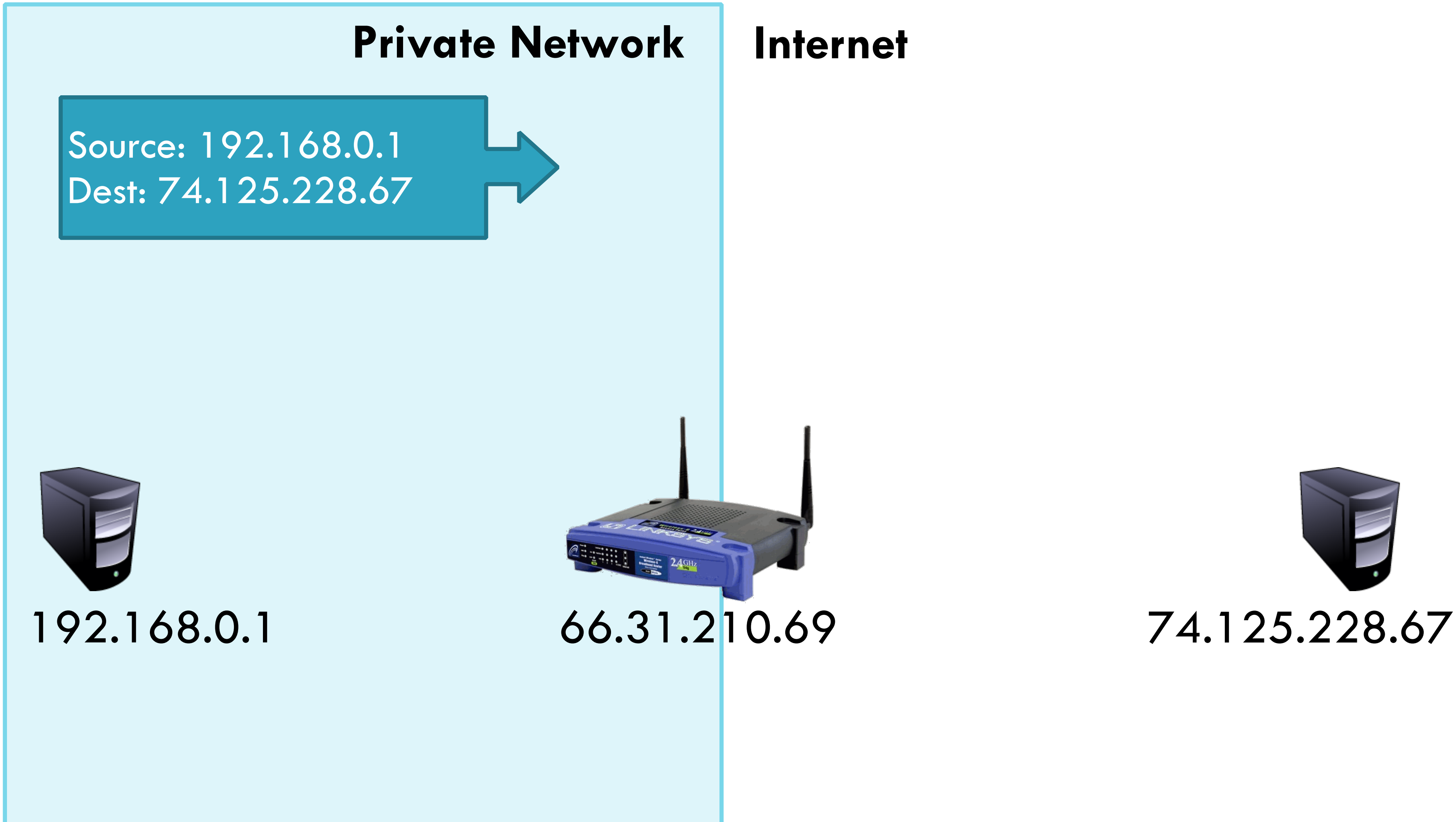
Network Address Translation (NAT)

- NAT allows hosts on a private network to communicate with the Internet
  - Warning: connectivity is not seamless
- Special router at the boundary of a private network
  - Replaces internal IPs with external IP
    - This is "Network Address Translation"
  - May also replace TCP/UDP port numbers
- Maintains a table of active flows
  - Outgoing packets initialize a table entry
  - Incoming packets are rewritten based on the table

Basic NAT Operation



Basic NAT Operation



Basic NAT Operation

Private Network **Internet**

Source: 192.168.0.1
Dest: 74.125.228.67

Private Address	**Public Address**
192.168.0.1:2345	***74.125.228.67:80***

192.168.0.1

66.31.210.69

74.125.228.67

Basic NAT Operation

Private Network

Internet

Source: 192.168.0.1
Dest: 74.125.228.67

Source: 66.31.210.69
Dest: 74.125.228.67

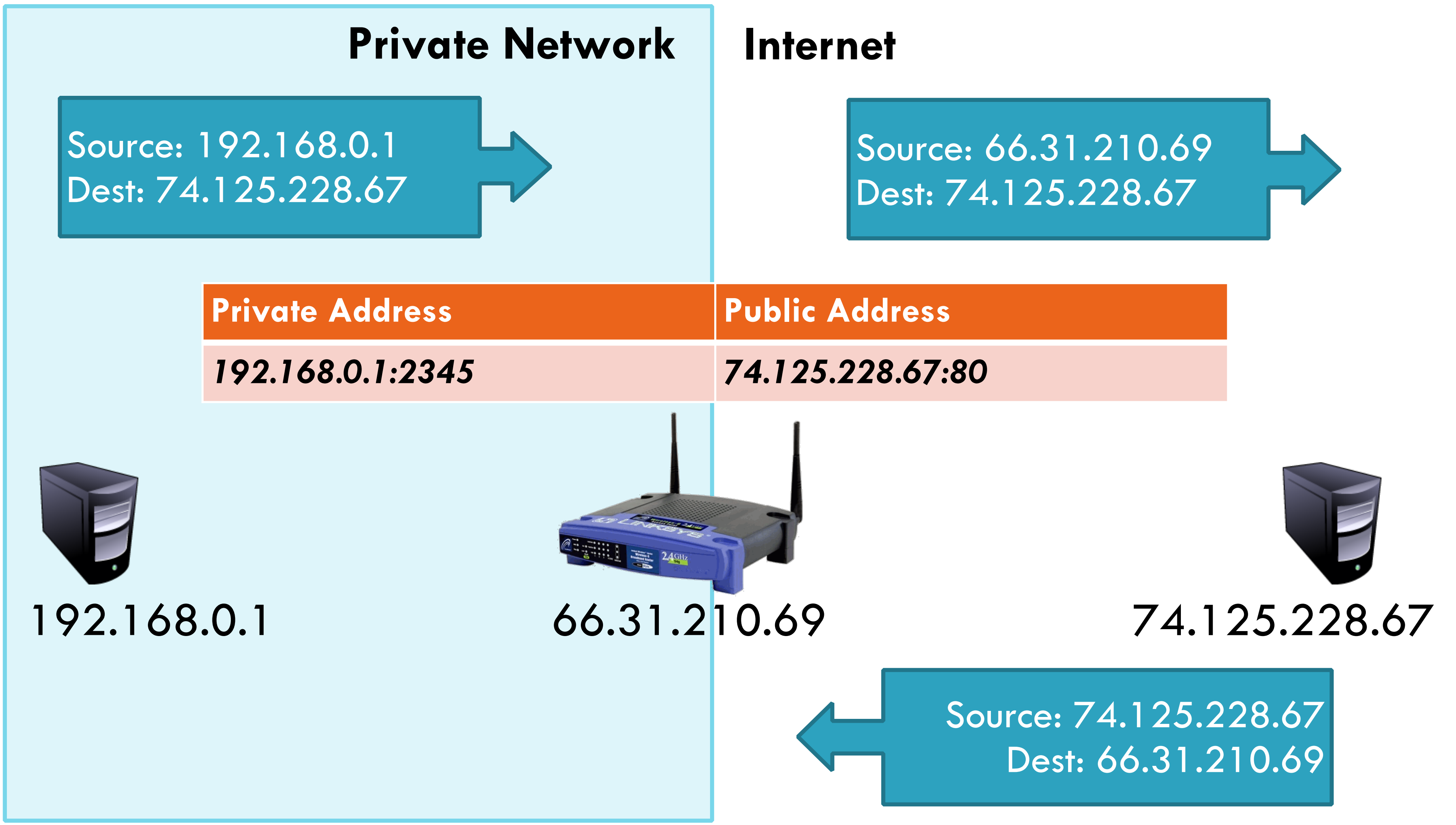
Private Address	Public Address
192.168.0.1:2345	***74.125.228.67:80***

192.168.0.1

66.31.210.69

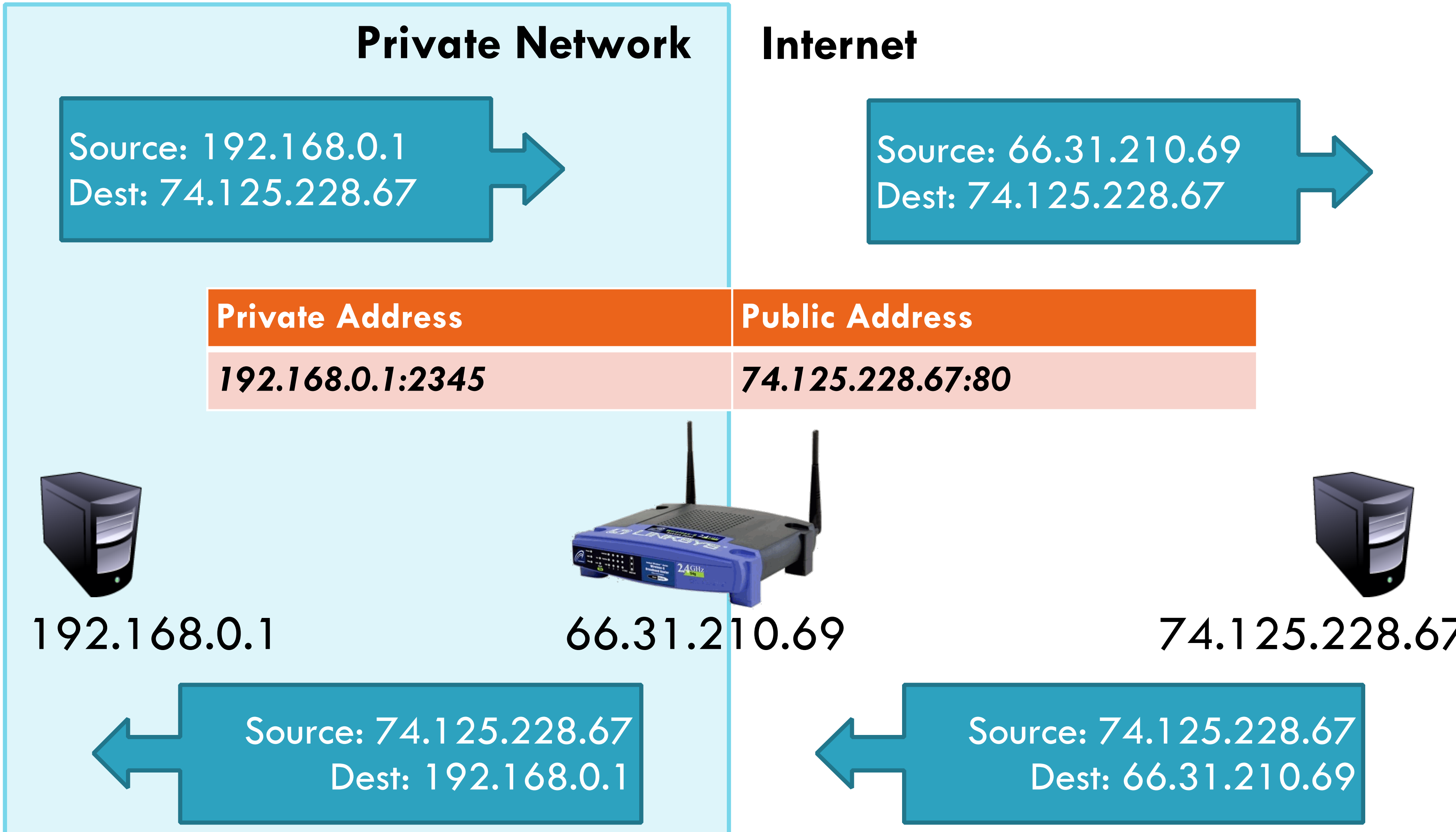
74.125.228.67

Basic NAT Operation



Private Address	Public Address
192.168.0.1:2345	***74.125.228.67:80***

Basic NAT Operation



Private Address	Public Address
192.168.0.1:2345	*74.125.228.67:80*

Advantages of NATs

- Allow multiple hosts to share a single public IP

Advantages of NATs

- Allow multiple hosts to share a single public IP
- Allow migration between ISPs
  - Even if the public IP address changes, you don't need to reconfigure the machines on the LAN

Advantages of NATs

- Allow multiple hosts to share a single public IP
- Allow migration between ISPs
  - Even if the public IP address changes, you don't need to reconfigure the machines on the LAN
- Load balancing
  - Forward traffic from a single public IP to multiple private hosts

Natural Firewall

Private Network **Internet**

Private Address	Public Address

192.168.0.1

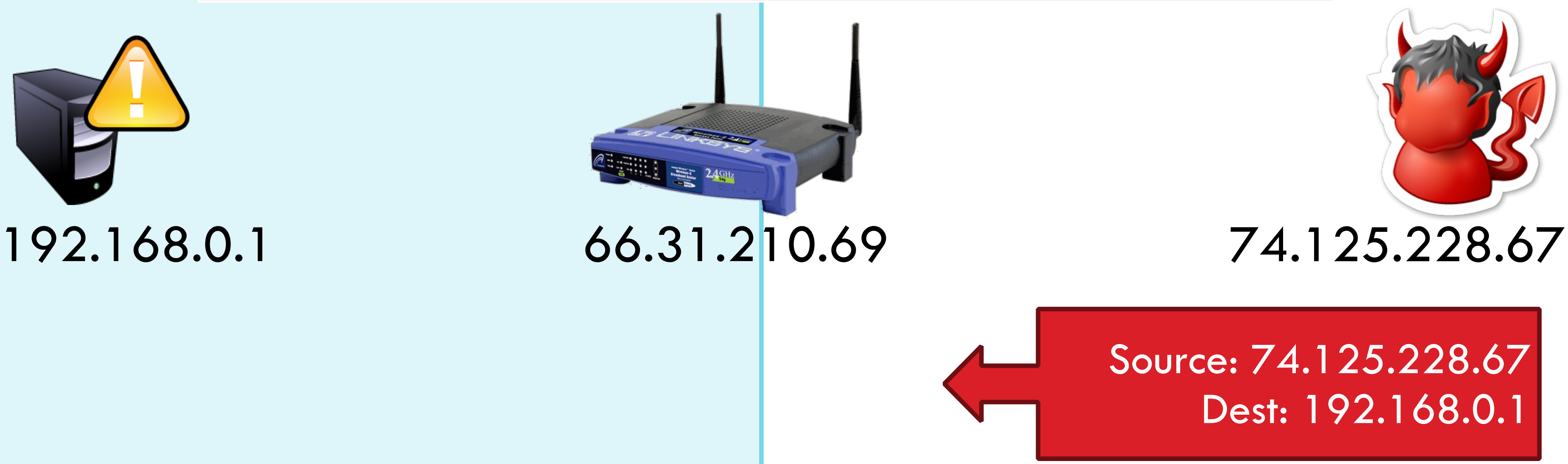
66.31.210.69

74.125.228.67

Natural Firewall

Private Network **Internet**

Private Address	Public Address



Natural Firewall

Private Network **Internet**

Private Address	Public Address

192.168.0.1

66.31.210.69

74.125.228.67

Natural Firewall

Private Network **Internet**

Private Address	Public Address

192.168.0.1

66.31.210.69

74.125.228.67

Source: 74.125.228.67
Dest: 66.31.210.69

Natural Firewall

Private Network | **Internet**

Private Address	Public Address

192.168.0.1

66.31.210.69

74.125.228.67

Source: 74.125.228.67
Dest: 66.31.210.69

Natural Firewall

Private Network **Internet**

Private Address	Public Address

192.168.0.1

66.31.210.69

74.125.228.67

Concerns About NAT

- Performance/scalability issues
  - Per flow state!
  - Modifying IP and Port numbers means NAT must recompute IP and TCP checksums

Concerns About NAT

- Performance/scalability issues
  - Per flow state!
  - Modifying IP and Port numbers means NAT must recompute IP and TCP checksums
- Breaks the layered network abstraction

Concerns About NAT

- Performance/scalability issues
  - Per flow state!
  - Modifying IP and Port numbers means NAT must recompute IP and TCP checksums
- Breaks the layered network abstraction
- Breaks end-to-end Internet connectivity
  - 192.168.*.* addresses are private
  - Cannot be routed to on the Internet
  - Problem is worse when both hosts are behind NATs

Concerns About NAT

- Performance/scalability issues
  - Per flow state!
  - Modifying IP and Port numbers means NAT must recompute IP and TCP checksums
- Breaks the layered network abstraction
- Breaks end-to-end Internet connectivity
  - 192.168.*.* addresses are private
  - Cannot be routed to on the Internet
  - Problem is worse when both hosts are behind NATs
- What about IPs embedded in data payloads?

Port Forwarding

Private Network **Internet**

Private Address	Public Address
192.168.0.1:7000	*.*.*.*:*

192.168.0.1

66.31.210.69

74.125.228.67

Port Forwarding

Private Network **Internet**

Private Address	Public Address
192.168.0.1:7000	*.*.*.*:*

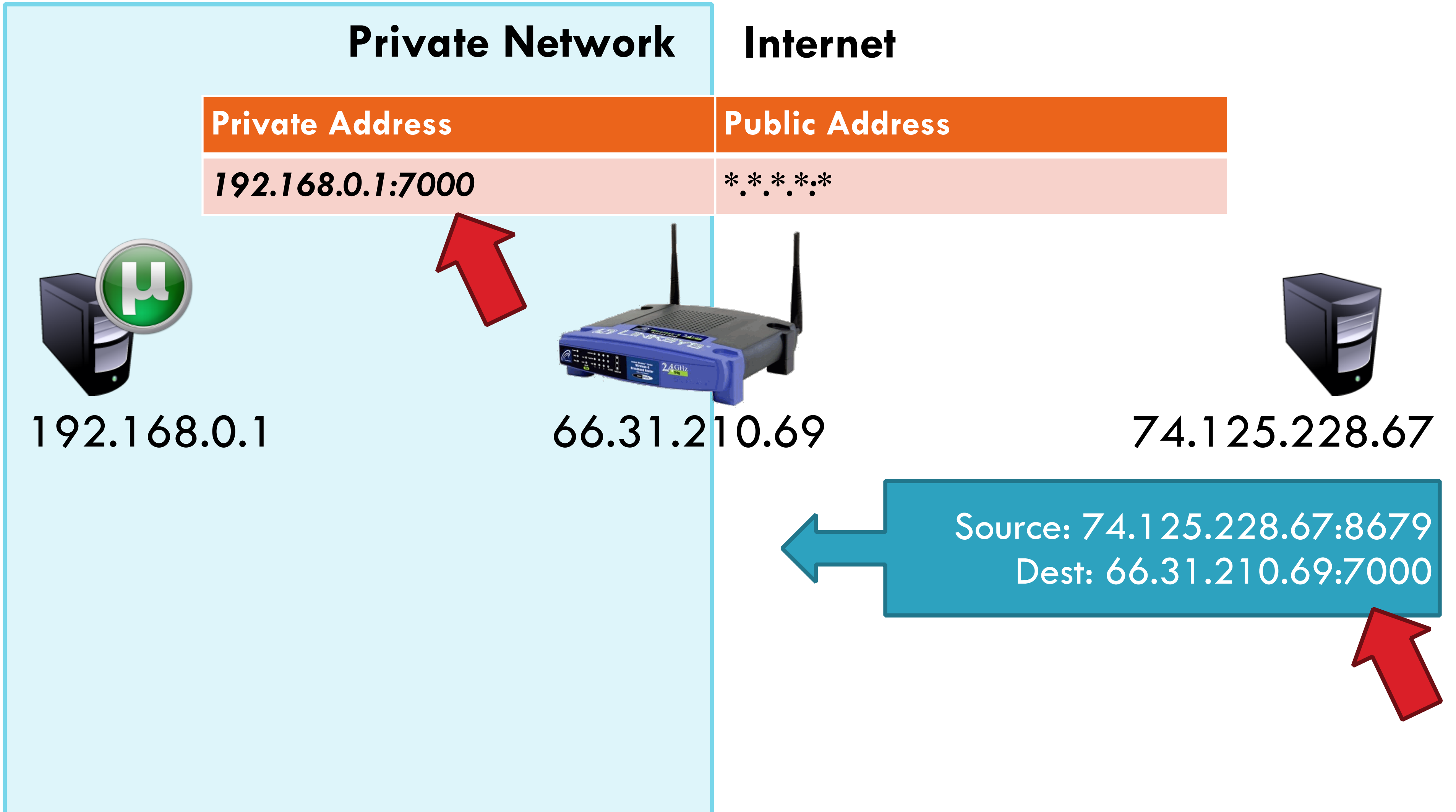
192.168.0.1

66.31.210.69

74.125.228.67

Source: 74.125.228.67:8679
Dest: 66.31.210.69:7000

Port Forwarding

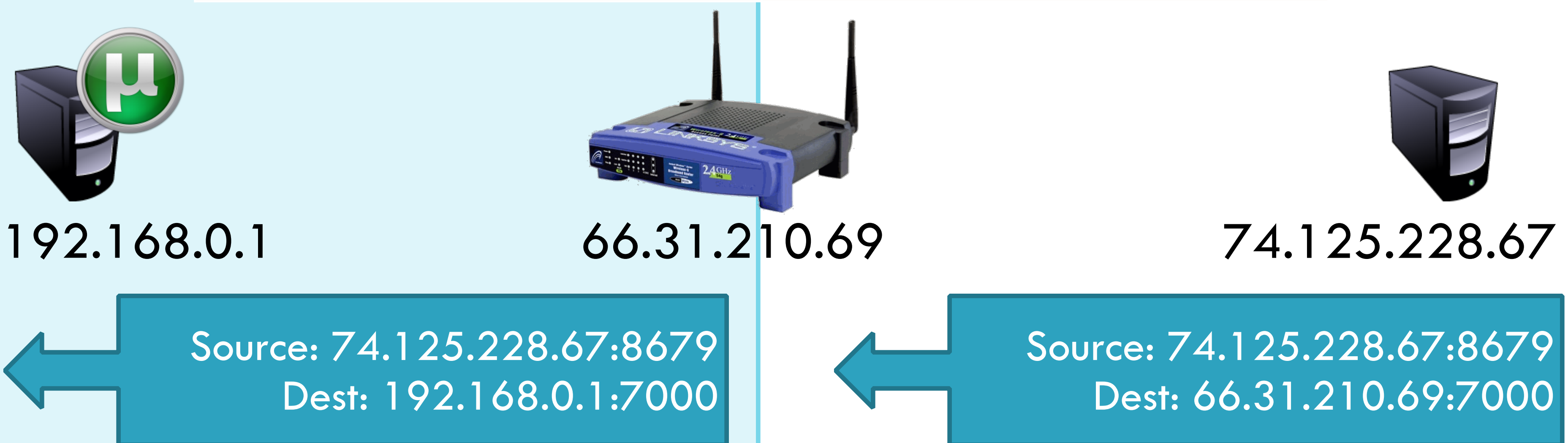


Private Address	Public Address
192.168.0.1:7000	*.*.*.*:*

Port Forwarding

Private Network **Internet**

Private Address	Public Address
192.168.0.1:7000	*.*.*.*:*



Hole Punching

- Problem: How to enable connectivity through NATs?

Hole Punching

- Problem: How to enable connectivity through NATs?

Hole Punching

- Problem: How to enable connectivity through NATs?

Hole Punching

- Problem: How to enable connectivity through NATs?

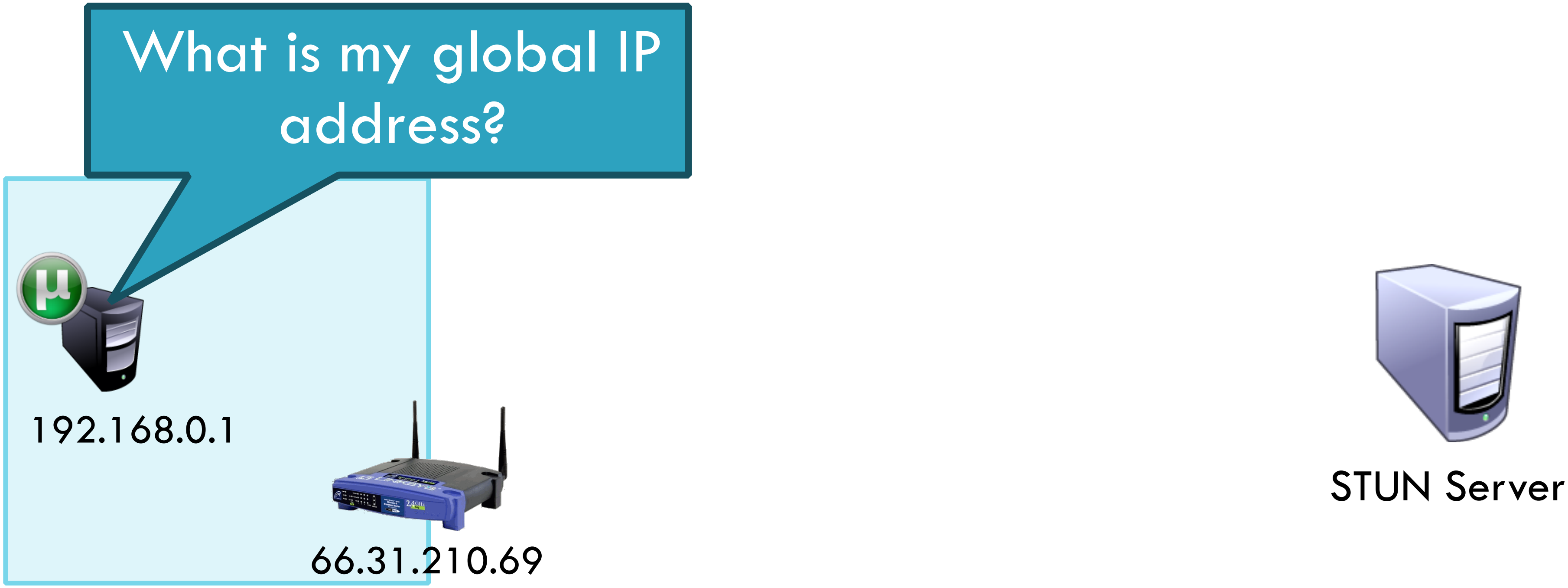
- Two application-level protocols for hole punching
  - STUN
  - TURN

STUN

- **S**ession **T**raversal **U**tilities for **N**AT
  - Use a third-party to echo your global IP address
  - Also used to probe for symmetric NATs/firewalls
    - i.e. are external ports open or closed?

STUN

- **S**ession **T**raversal **U**tilities for **N**AT
  - Use a third-party to echo your global IP address
  - Also used to probe for symmetric NATs/firewalls
    - i.e. are external ports open or closed?



STUN

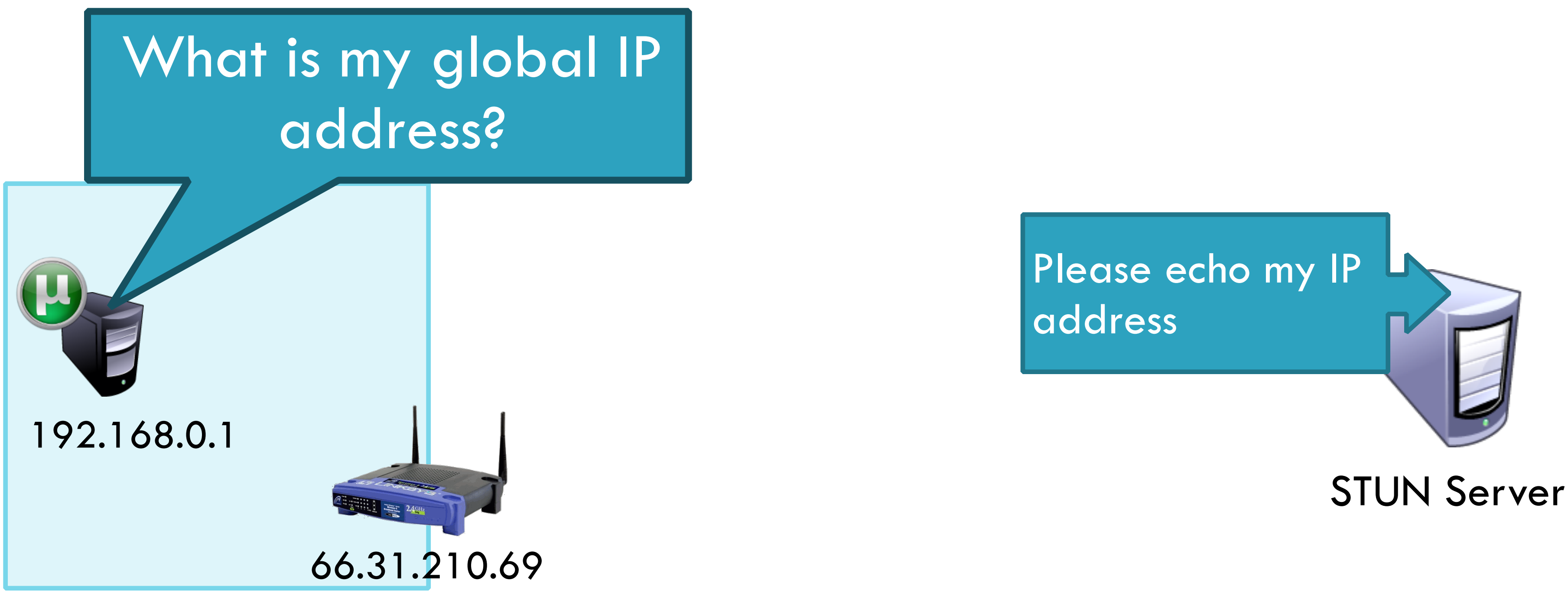
- **S**ession **T**raversal **U**tilities for **N**AT
  - Use a third-party to echo your global IP address
  - Also used to probe for symmetric NATs/firewalls
    - i.e. are external ports open or closed?

STUN

- **S**ession **T**raversal **U**tilities for **N**AT
  - Use a third-party to echo your global IP address
  - Also used to probe for symmetric NATs/firewalls
    - i.e. are external ports open or closed?

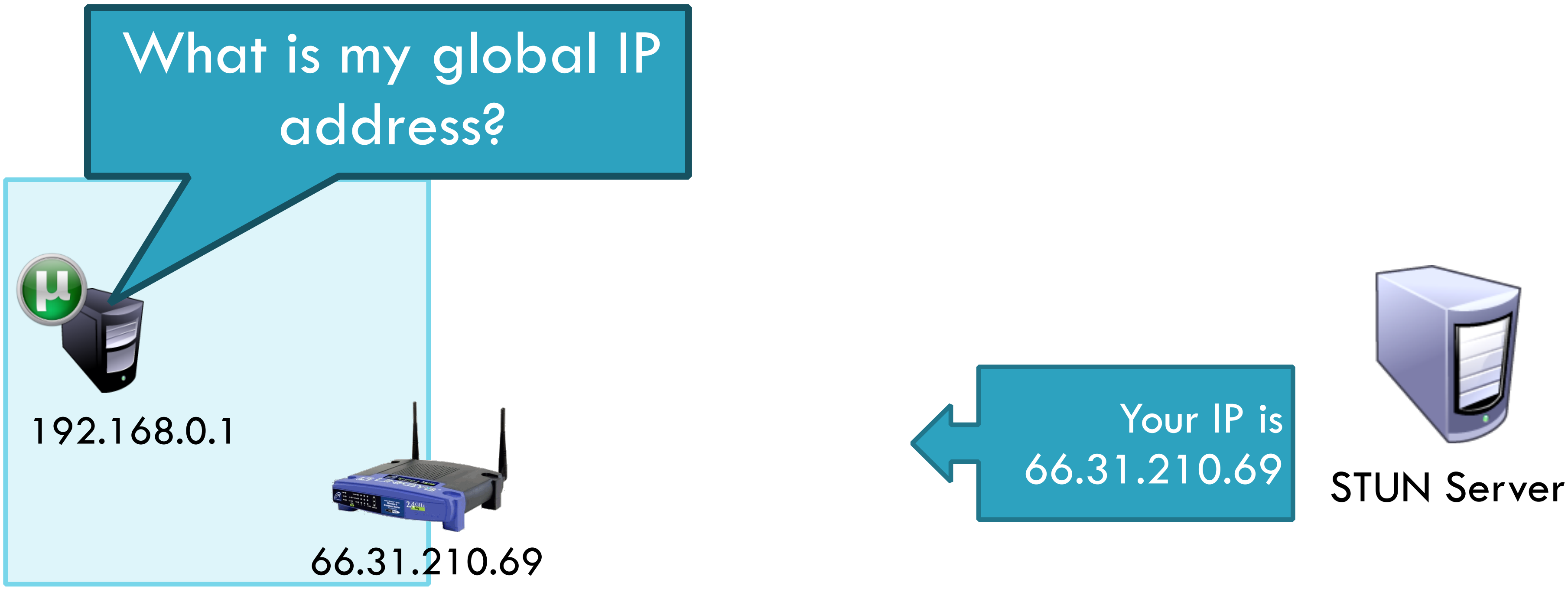
STUN

- **S**ession **T**raversal **U**tilities for **N**AT
  - Use a third-party to echo your global IP address
  - Also used to probe for symmetric NATs/firewalls
    - i.e. are external ports open or closed?



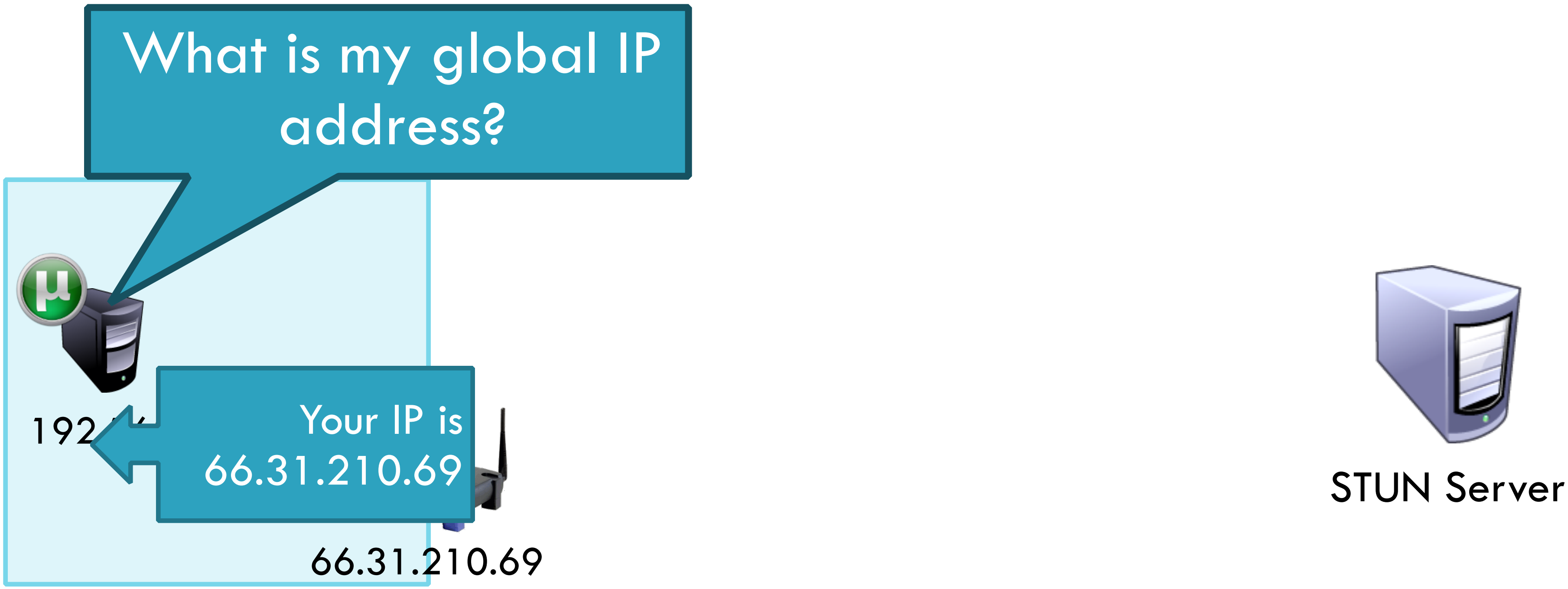
STUN

- **S**ession **T**raversal **U**tilities for **N**AT
  - Use a third-party to echo your global IP address
  - Also used to probe for symmetric NATs/firewalls
    - i.e. are external ports open or closed?



STUN

- **S**ession **T**raversal **U**tilities for **N**AT
  - Use a third-party to echo your global IP address
  - Also used to probe for symmetric NATs/firewalls
    - i.e. are external ports open or closed?

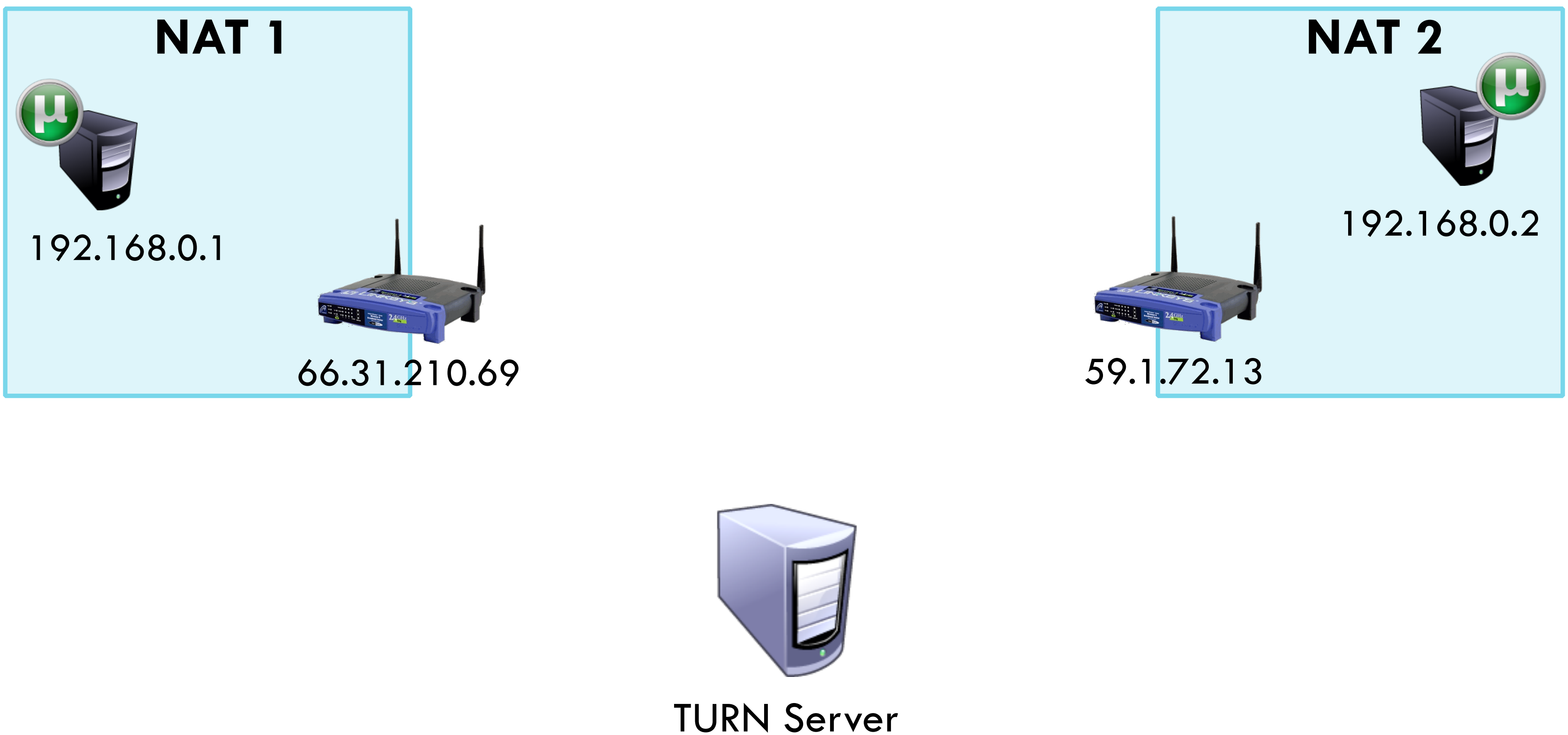


Problems With STUN

- Only useful in certain situations
  - One peer is behind a symmetric NAT
  - Both peers are behind partial NATs
- Not useful when both peers are fully behind full NATs

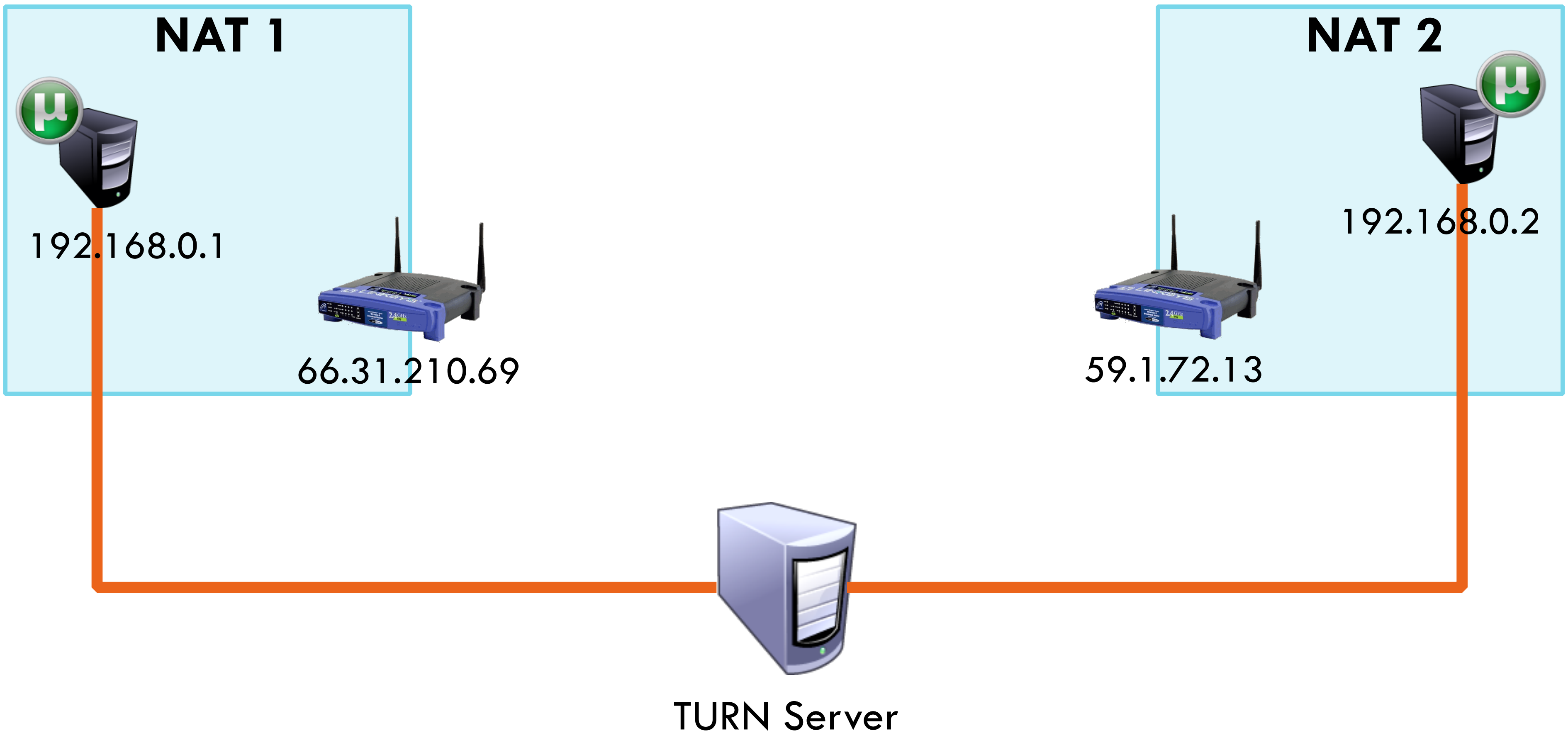
TURN

- Traversal Using Relays around NAT



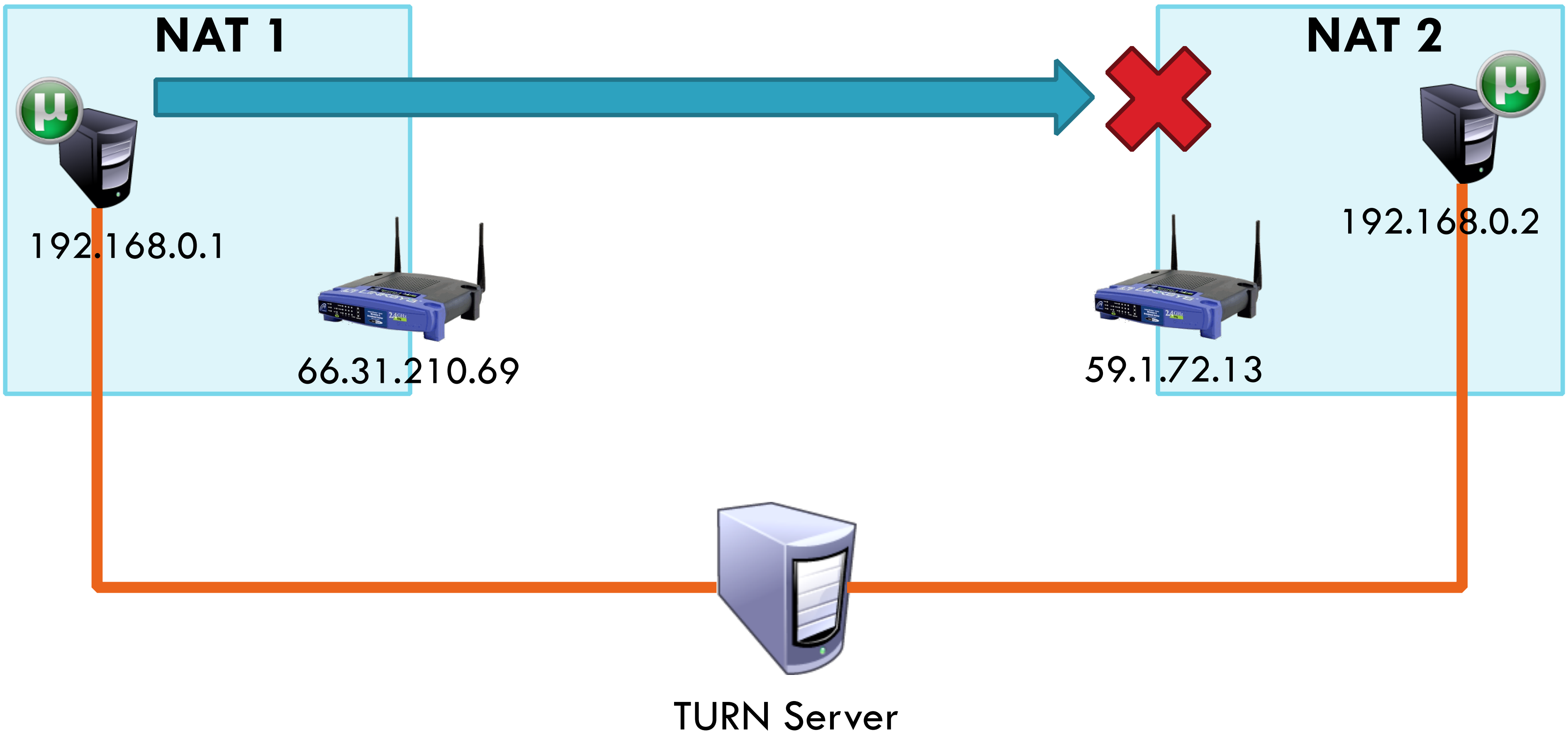
TURN

- **T**raversal **U**sing **R**elays around **N**AT



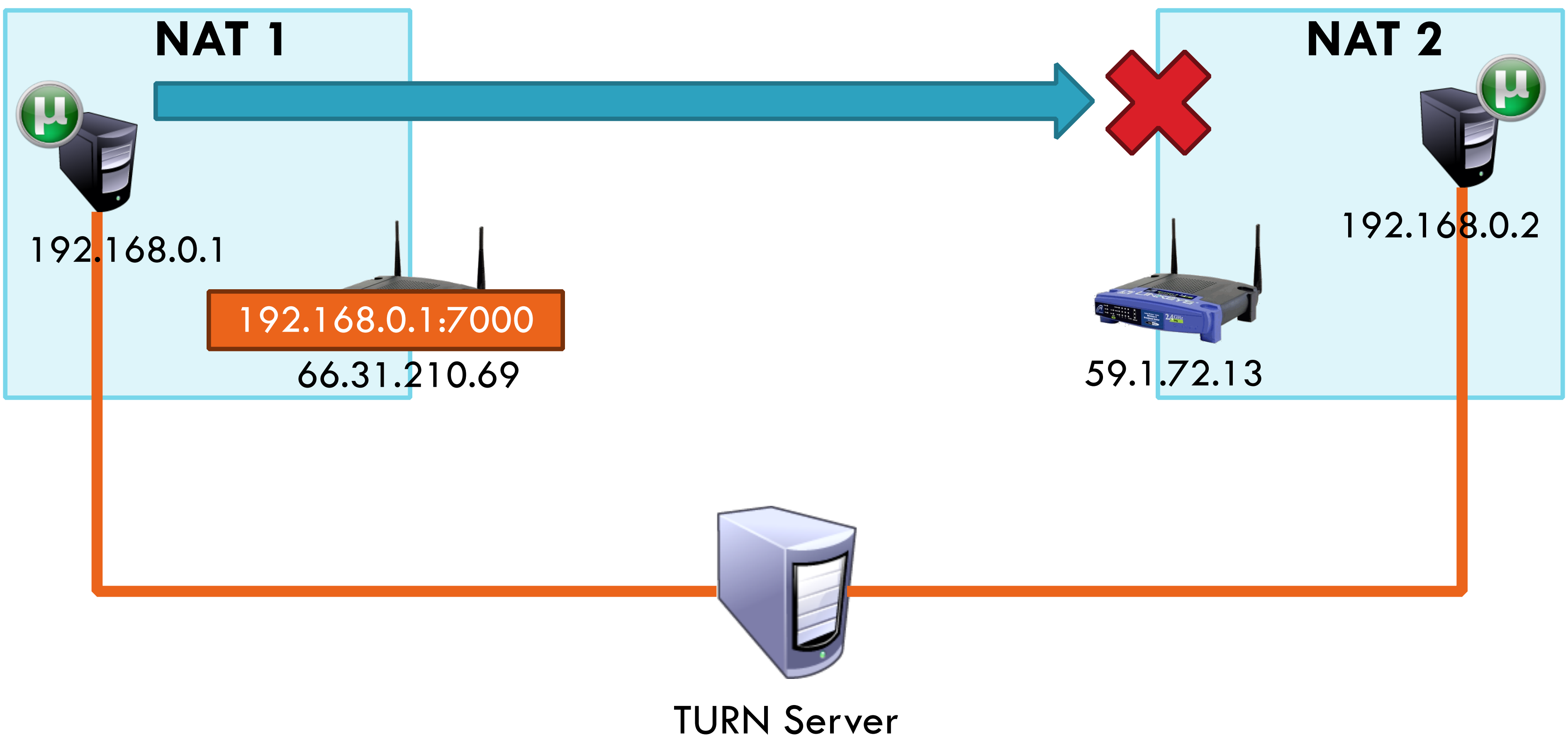
TURN

- Traversal Using Relays around NAT



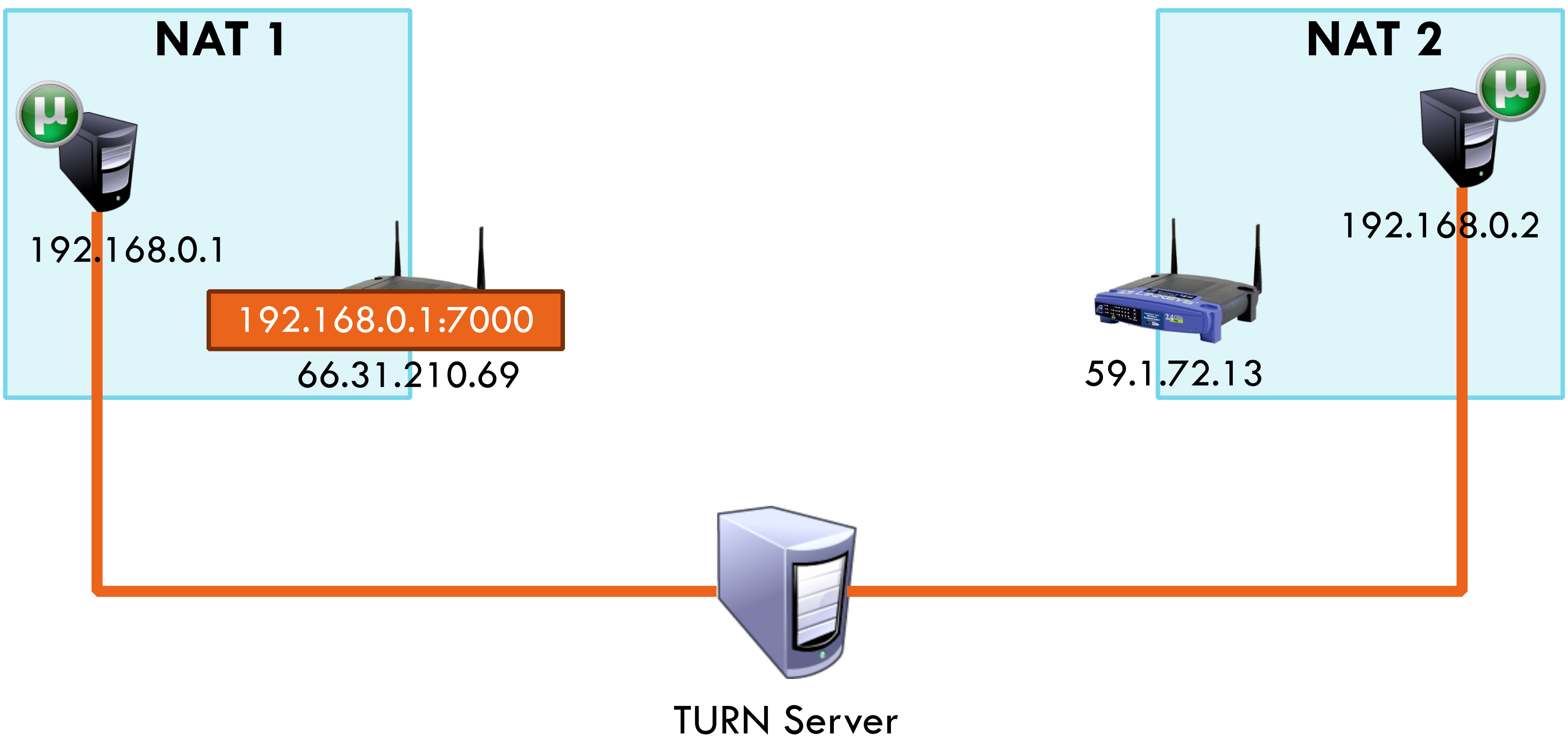
TURN

- **T**raversal **U**sing **R**elays around **N**AT



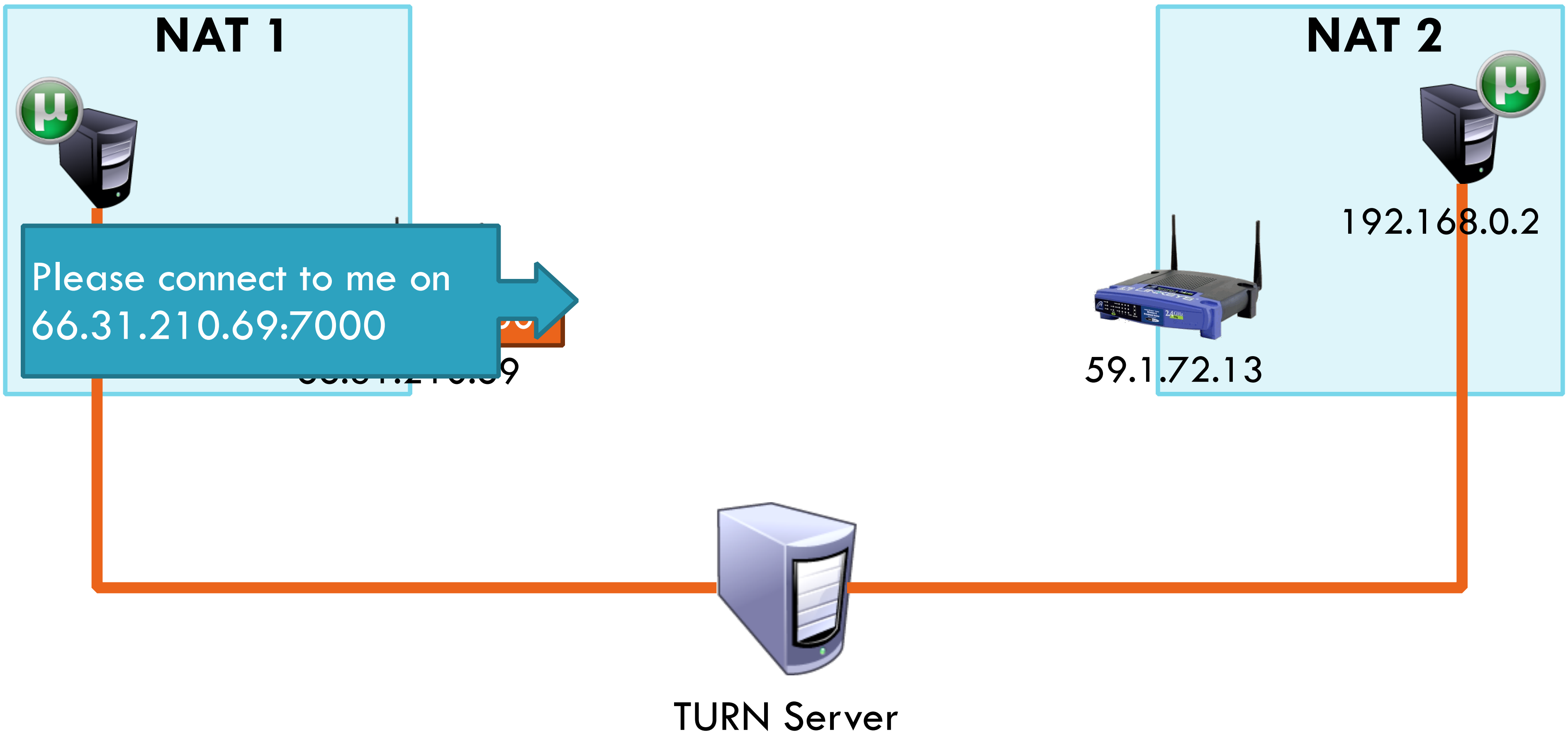
TURN

- **T**raversal **U**sing **R**elays around **N**AT



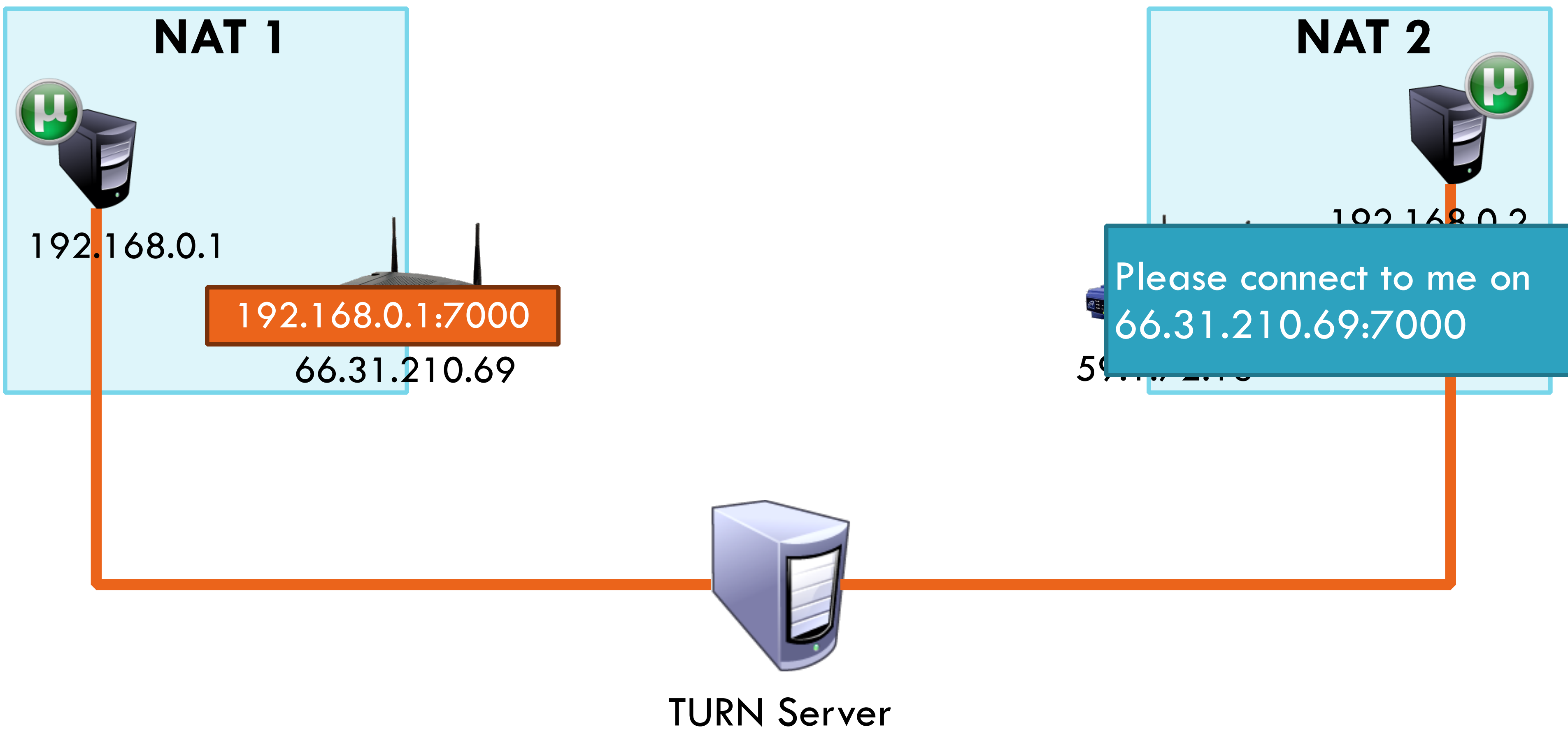
TURN

- Traversal **U**sing **R**elays around **NAT**



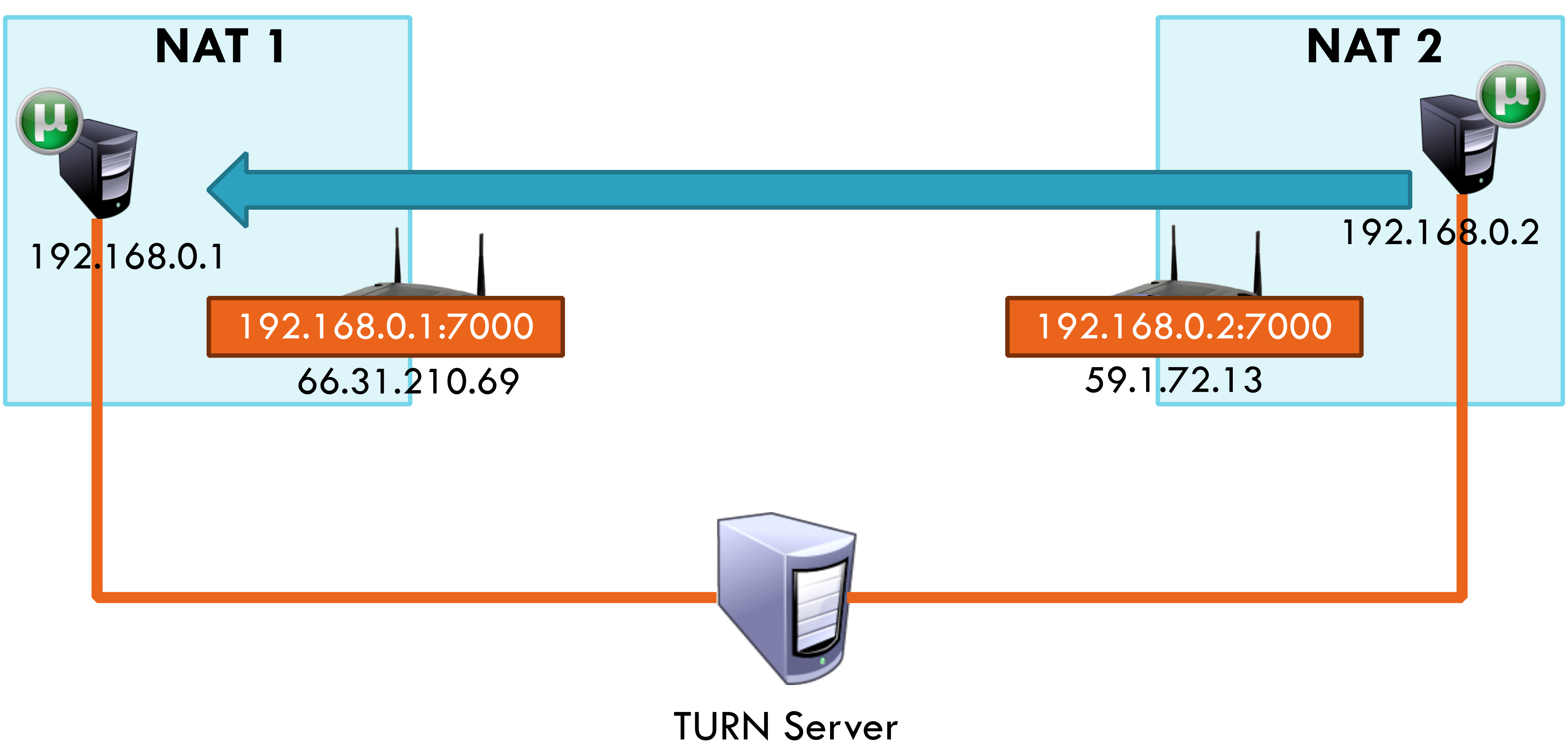
TURN

- **T**raversal **U**sing **R**elays around **N**AT



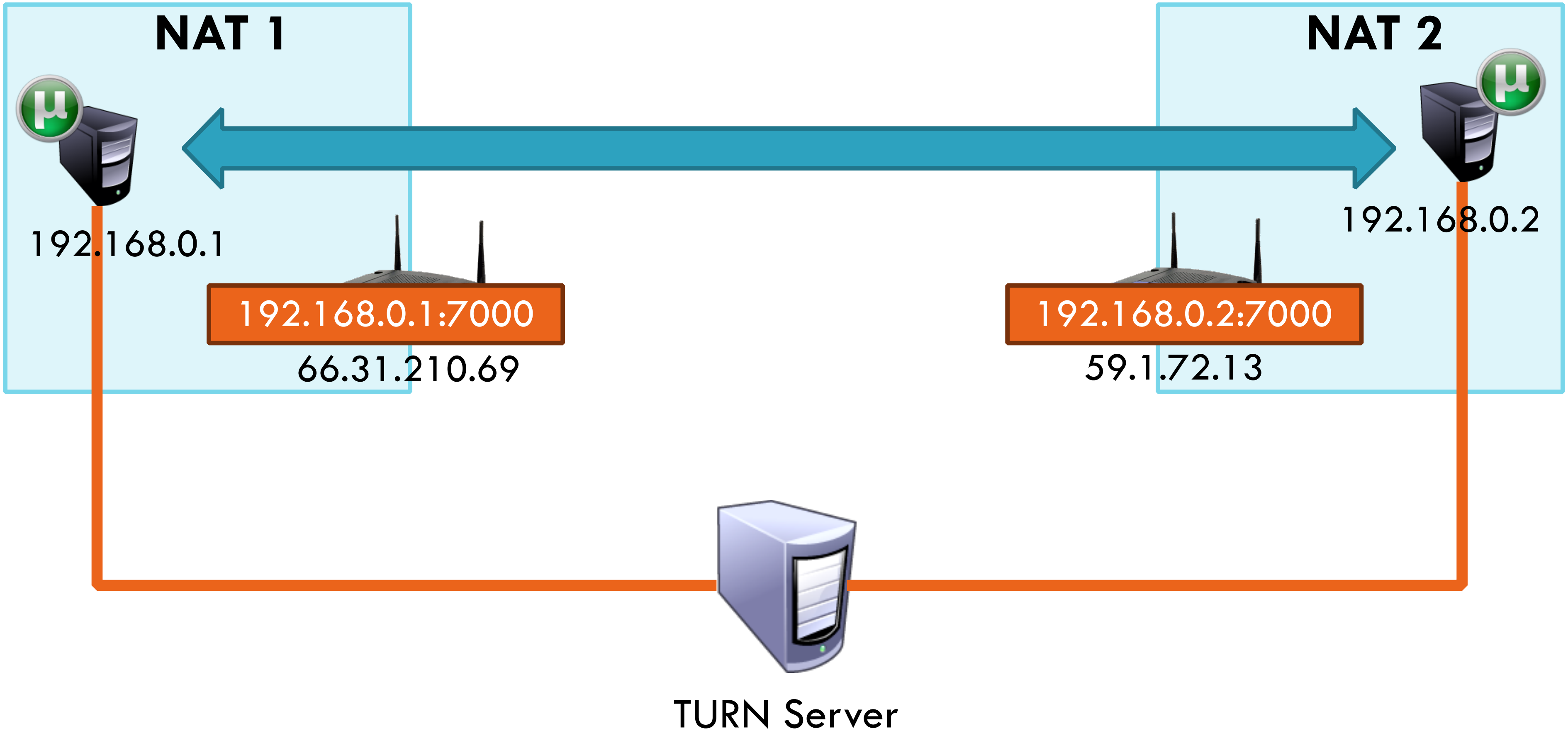
TURN

- **T**raversal **U**sing **R**elays around **N**AT



TURN

- **T**raversal **U**sing **R**elays around **N**AT



Outline

- DNS
- NAT
- Other middleboxes

Firewall

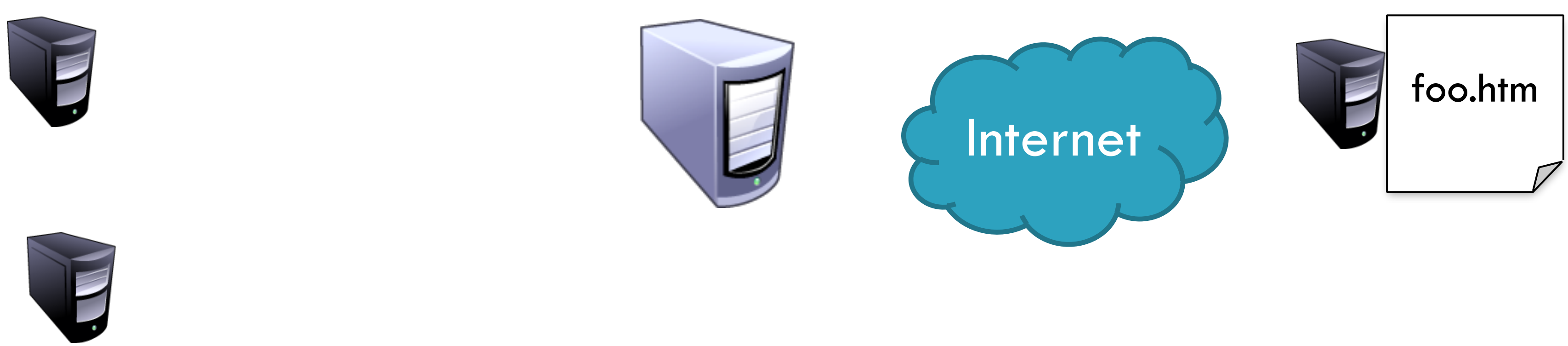
- A device that blocks traffic according to a set of rules
  - Why?
  - Services with vulnerabilities turned on by default
  - ISP policy forbidding certain traffic due to ToS
- Typically specified using a 5-tuple
  - E.g., block outbound SMTP; block inbound SQL server reqs
- GFC (Great Firewall of China)
  - Known to block based on IP, filter DNS requests, etc

Web caching

- ISP installs cache near network edge that caches copies of Web pages
  - Why?
  - **Performance:** Content is closer to clients, TCP will perform better with lower RTTs
  - **Cost:** "free" for the ISP to serve from inside the network
- Limitations
  - Much of today's content is not static (why does this matter?)
  - Content ownership
  - Potential privacy issues
  - Long tail of content popularity

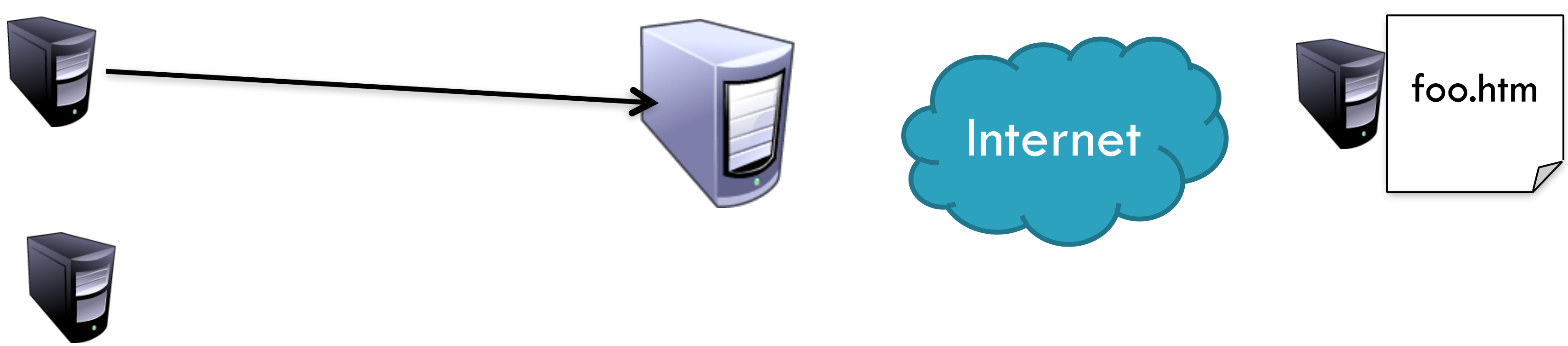
Web caching

- ISP installs cache near network edge that caches copies of Web pages
  - Why?
  - **Performance:** Content is closer to clients, TCP will perform better with lower RTTs
  - **Cost:** "free" for the ISP to serve from inside the network



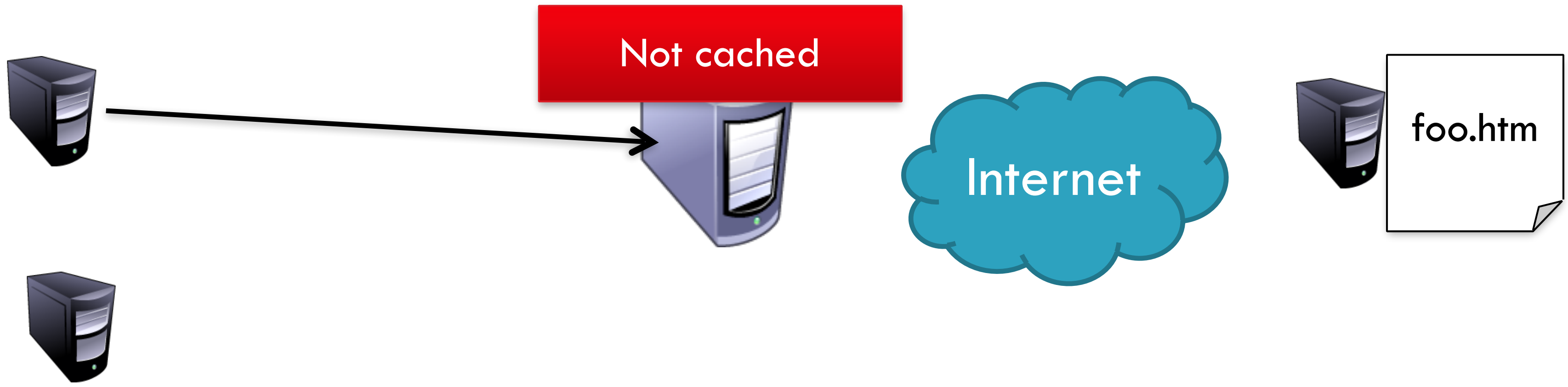
Web caching

- ISP installs cache near network edge that caches copies of Web pages
  - Why?
  - **Performance:** Content is closer to clients, TCP will perform better with lower RTTs
  - **Cost:** “free” for the ISP to serve from inside the network



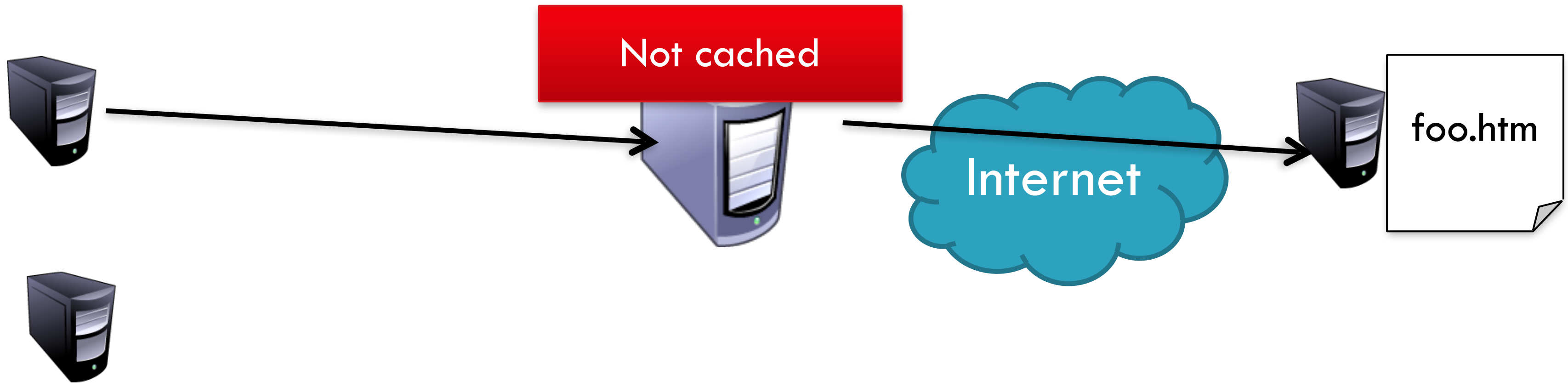
Web caching

- ISP installs cache near network edge that caches copies of Web pages
  - Why?
  - **Performance:** Content is closer to clients, TCP will perform better with lower RTTs
  - **Cost:** "free" for the ISP to serve from inside the network



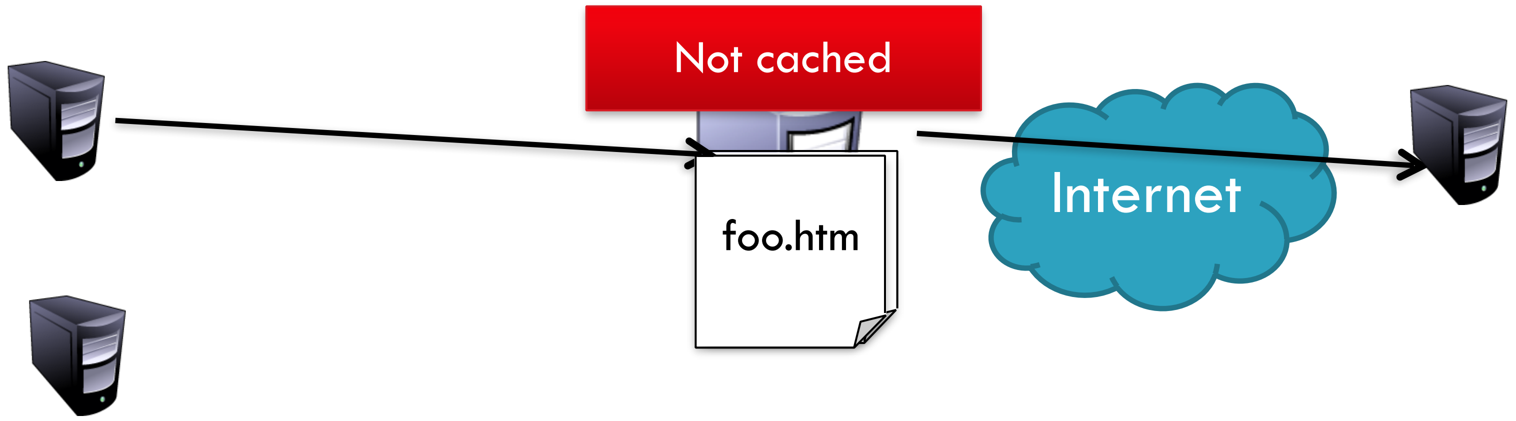
Web caching

- ISP installs cache near network edge that caches copies of Web pages
  - Why?
  - **Performance:** Content is closer to clients, TCP will perform better with lower RTTs
  - **Cost:** “free” for the ISP to serve from inside the network



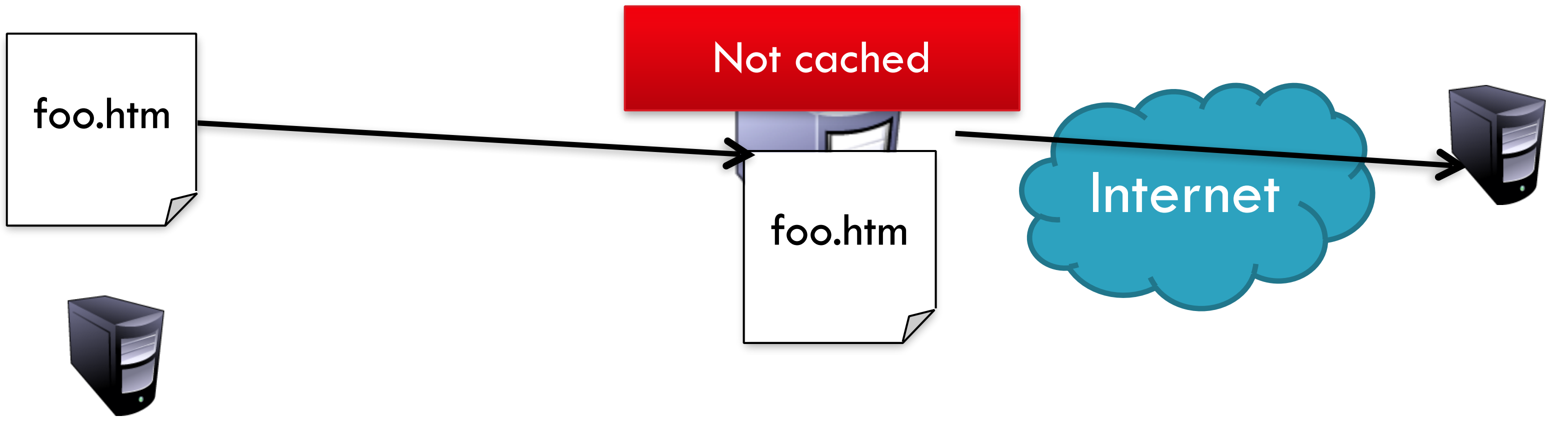
Web caching

- ISP installs cache near network edge that caches copies of Web pages
  - Why?
  - **Performance:** Content is closer to clients, TCP will perform better with lower RTTs
  - **Cost:** “free” for the ISP to serve from inside the network



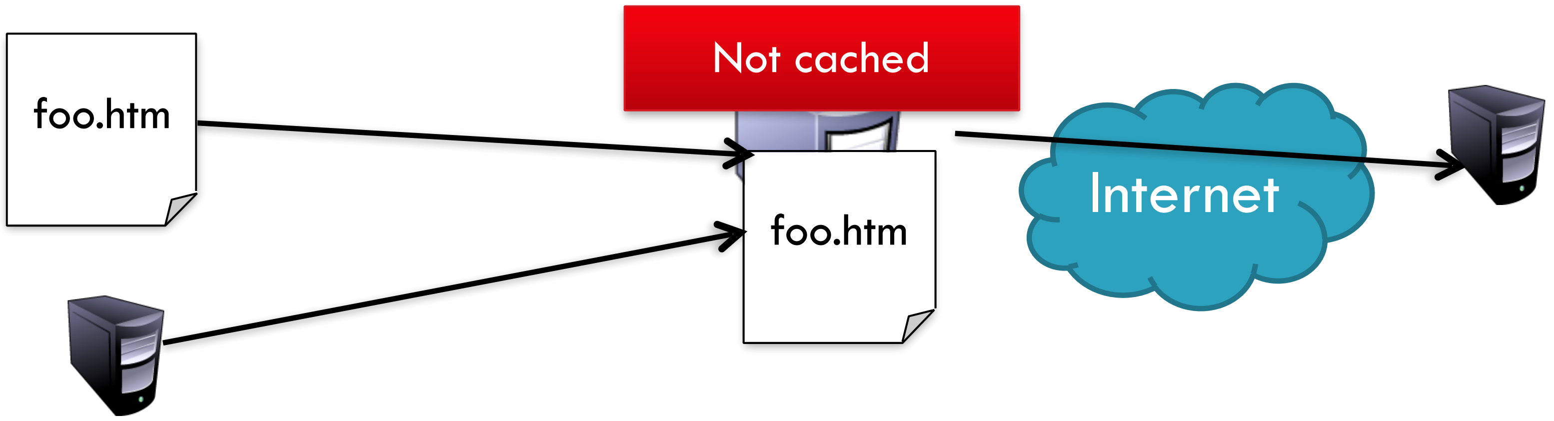
Web caching

- ISP installs cache near network edge that caches copies of Web pages
  - Why?
  - **Performance:** Content is closer to clients, TCP will perform better with lower RTTs
  - **Cost:** "free" for the ISP to serve from inside the network



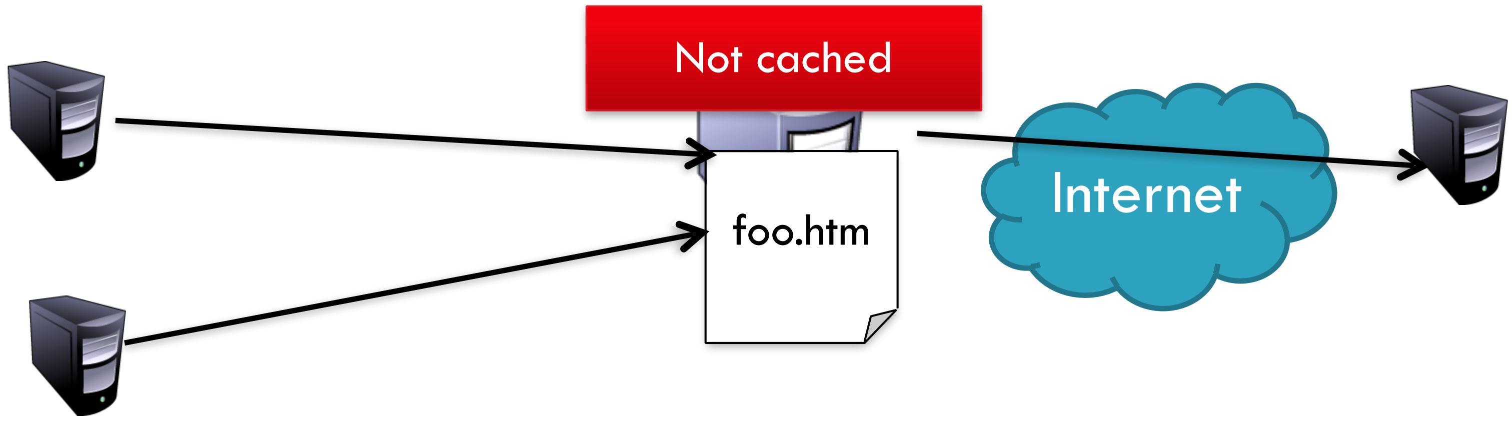
Web caching

- ISP installs cache near network edge that caches copies of Web pages
  - Why?
  - **Performance:** Content is closer to clients, TCP will perform better with lower RTTs
  - **Cost:** "free" for the ISP to serve from inside the network



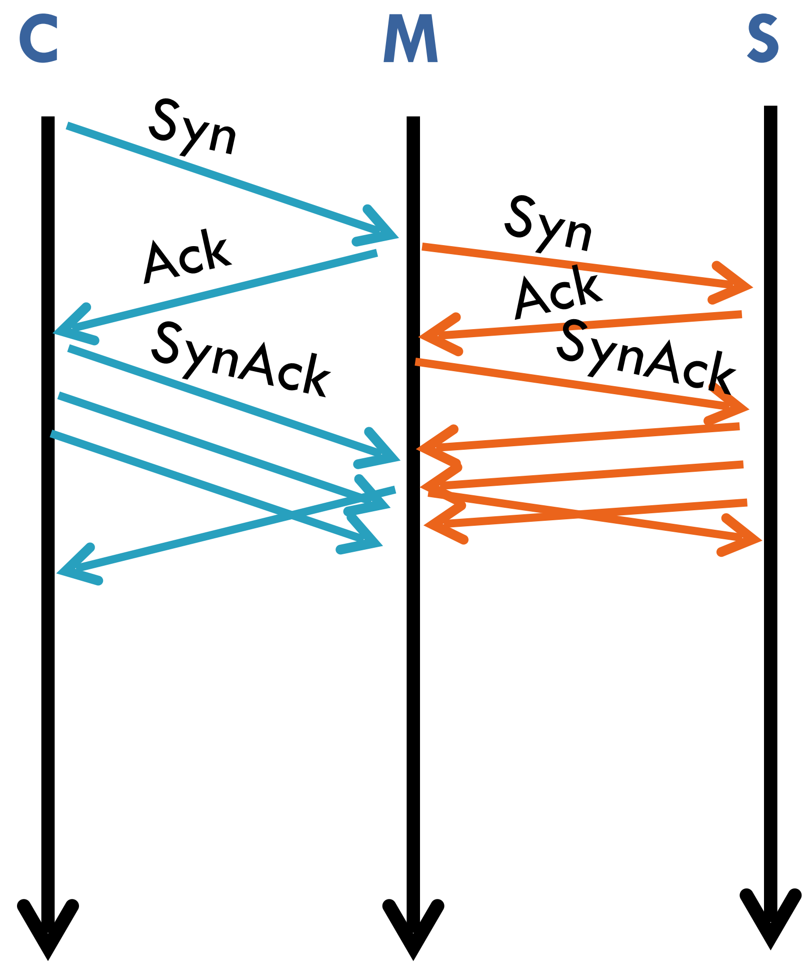
Web caching

- ISP installs cache near network edge that caches copies of Web pages
  - Why?
  - **Performance:** Content is closer to clients, TCP will perform better with lower RTTs
  - **Cost:** "free" for the ISP to serve from inside the network



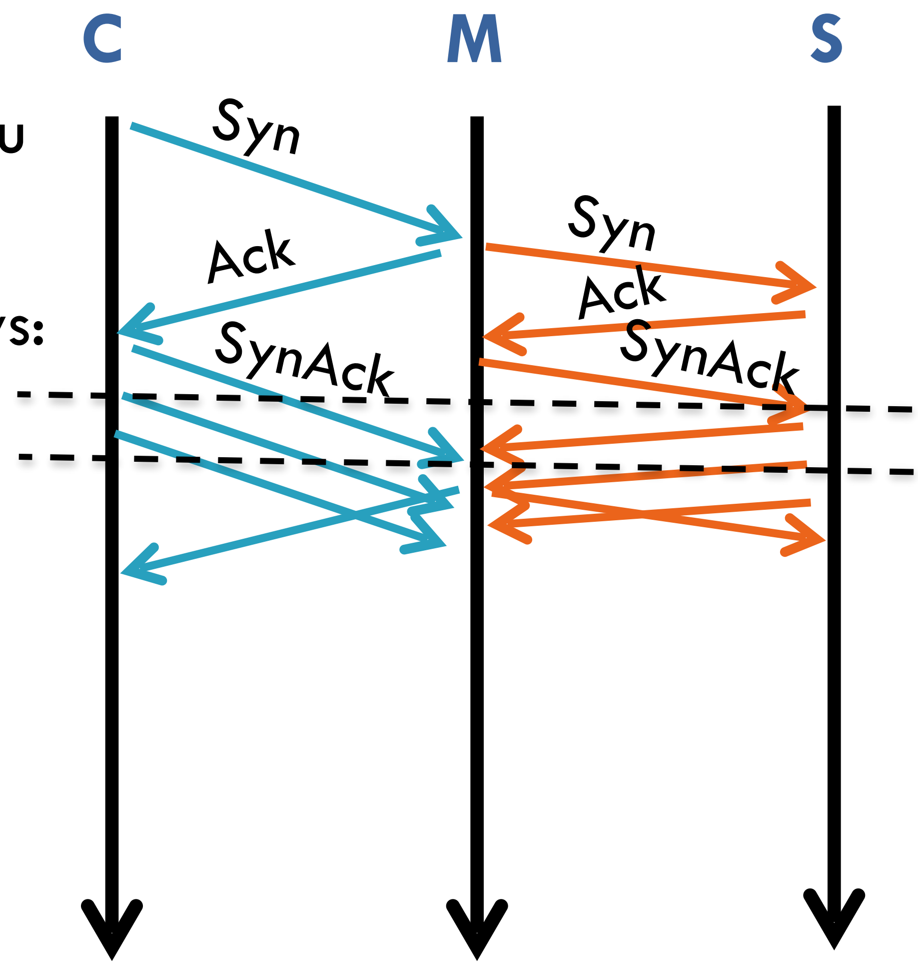
Proxying

- Non-split connections
  - Like NAT, but IP address is no longer the one assigned to you
- Split connections
  - Middlebox maintains two flows: C-M and M-S
  - Can be done transparently
    - How?



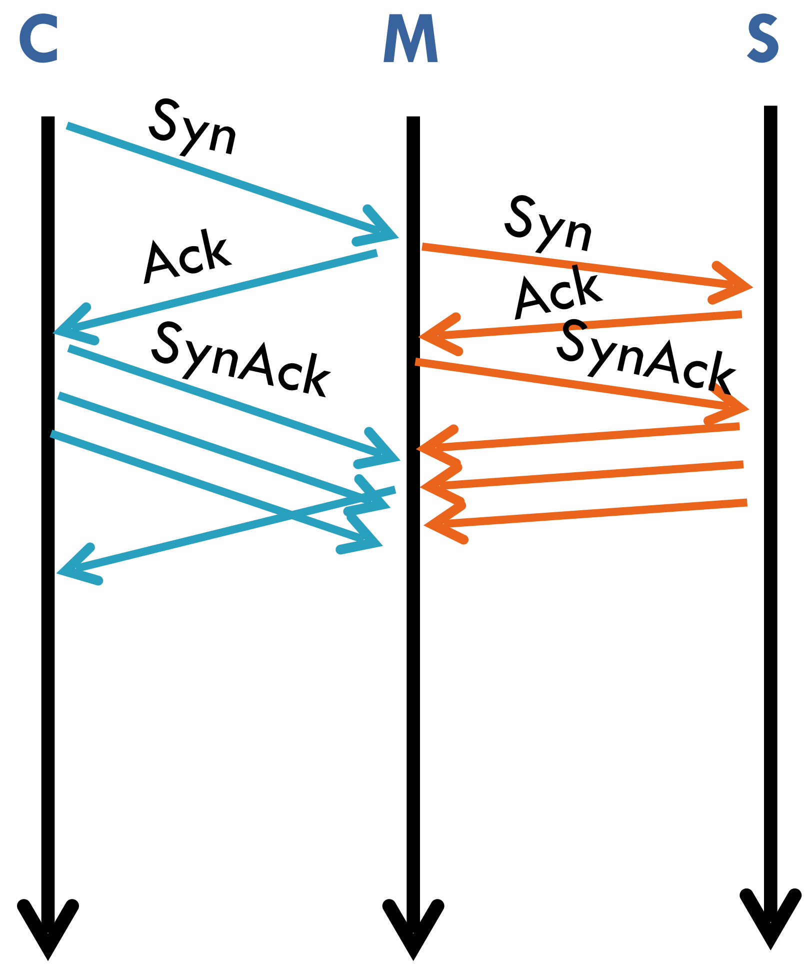
Proxying

- Non-split connections
  - Like NAT, but IP address is no longer the one assigned to you
- Split connections
  - Middlebox maintains two flows: C-M and M-S
  - Can be done transparently
    - How?



Proxying

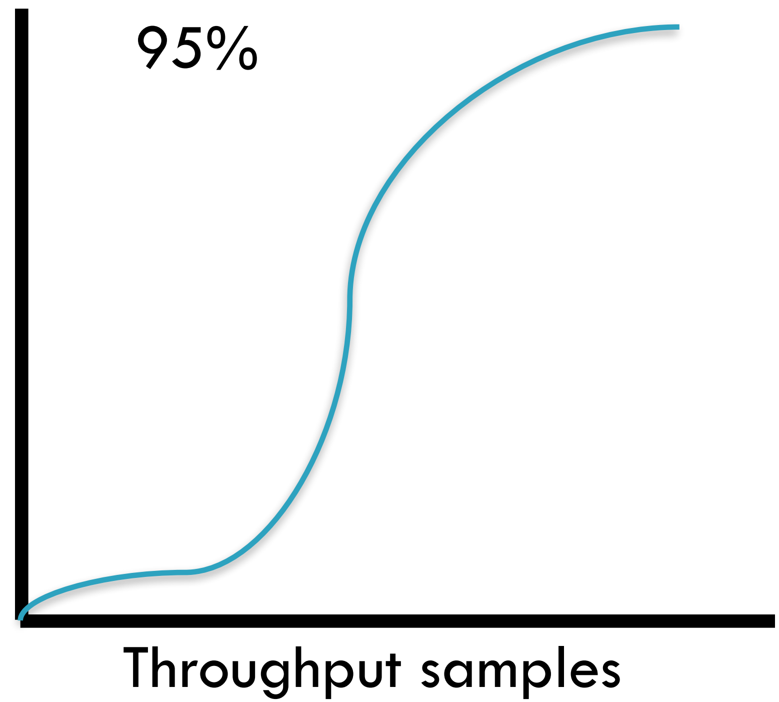
- Advantages
  - RTT is lower on each end
  - Can use different MTUs
  - Particularly useful in cell ntwks
- Disadvantages
  - Extra delay can be bad for small flows
  - Buffering/state makes it potentially costly



Shaping

- ISPs are often charged according to 95% model
  - Internet usage is very "peaky", e.g., at 5pm, or when House of Cards season 2 is released

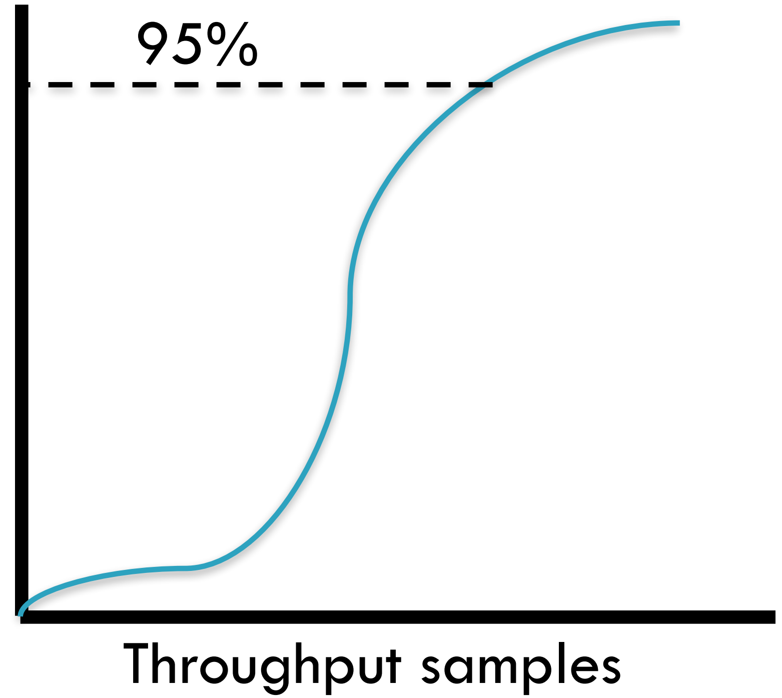
- To control costs, ISPs such as Rogers **shape** client traffic
  - Time-of day
  - Traffic type
- Common implementations
  - Token Bucket (see next deck)
  - RSTs



Shaping

- ISPs are often charged according to 95% model
  - Internet usage is very "peaky", e.g., at 5pm, or when House of Cards season 2 is released

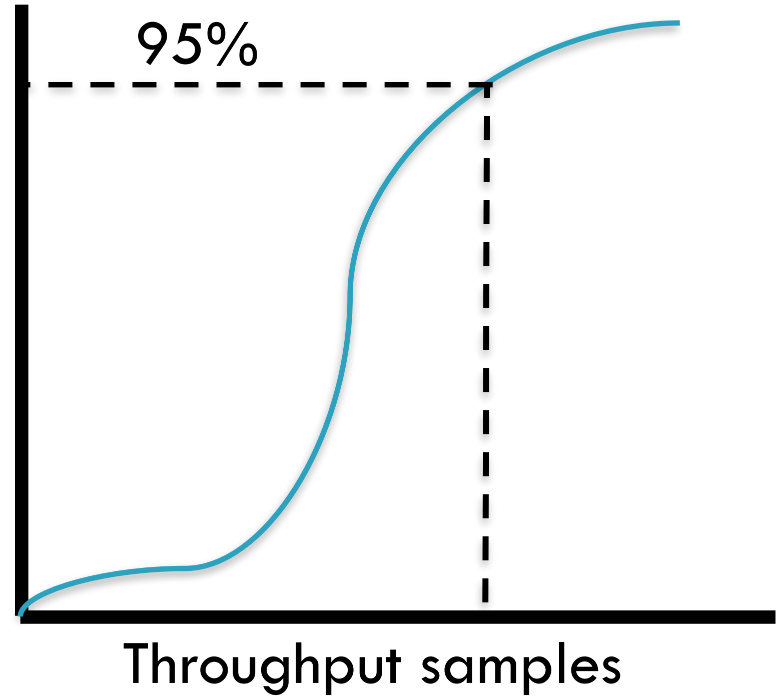
- To control costs, ISPs such as Rogers **shape** client traffic
  - Time-of day
  - Traffic type
- Common implementations
  - Token Bucket (see next deck)
  - RSTs



Shaping

- ISPs are often charged according to 95% model
  - Internet usage is very "peaky", e.g., at 5pm, or when House of Cards season 2 is released

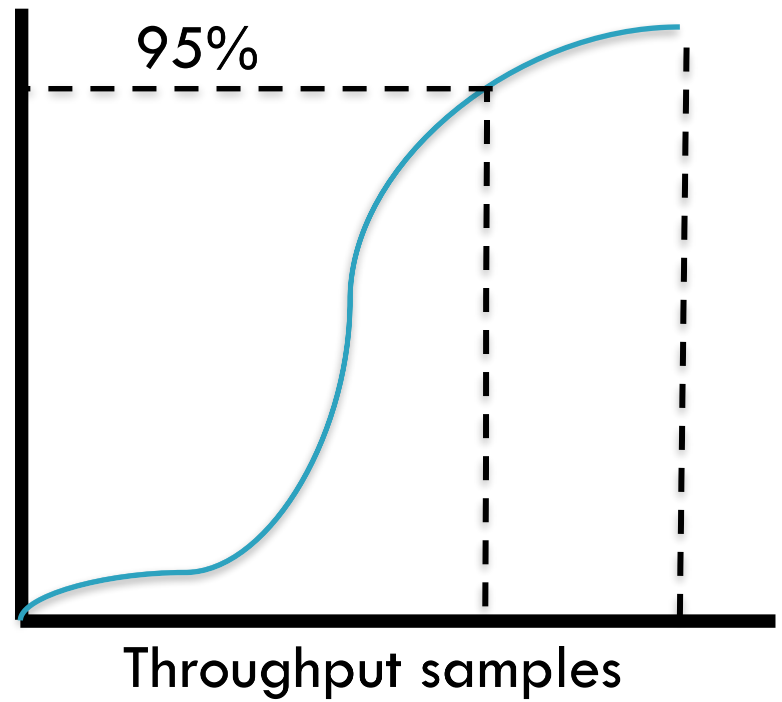
- To control costs, ISPs such as Rogers **shape** client traffic
  - Time-of day
  - Traffic type
- Common implementations
  - Token Bucket (see next deck)
  - RSTs



Shaping

- ISPs are often charged according to 95% model
  - Internet usage is very "peaky", e.g., at 5pm, or when House of Cards season 2 is released

- To control costs, ISPs such as Rogers **shape** client traffic
  - Time-of day
  - Traffic type
- Common implementations
  - Token Bucket (see next deck)
  - RSTs



Shaping

- ISPs are often charged according to 95% model
  - Internet usage is very “peaky”, e.g., at 5pm, or when House of Cards season 2 is released
- To control costs, ISPs such as Rogers **shape** client traffic
  - Time-of day
  - Traffic type
- Common implementations
  - Token Bucket (see next deck)
  - RSTs