Networking Security: Part 3
CMSC 23200, Spring 2026, Lecture 10

Grant Ho and David Cash

University of Chicago, 04/23/2026
(Slides adapted from Blase Ur and Vern Paxson)

Logistics

- Assignment 4 Due Tonight by 11:59pm
  - Both Part (a): Crypto II + Part (b): Networking due then

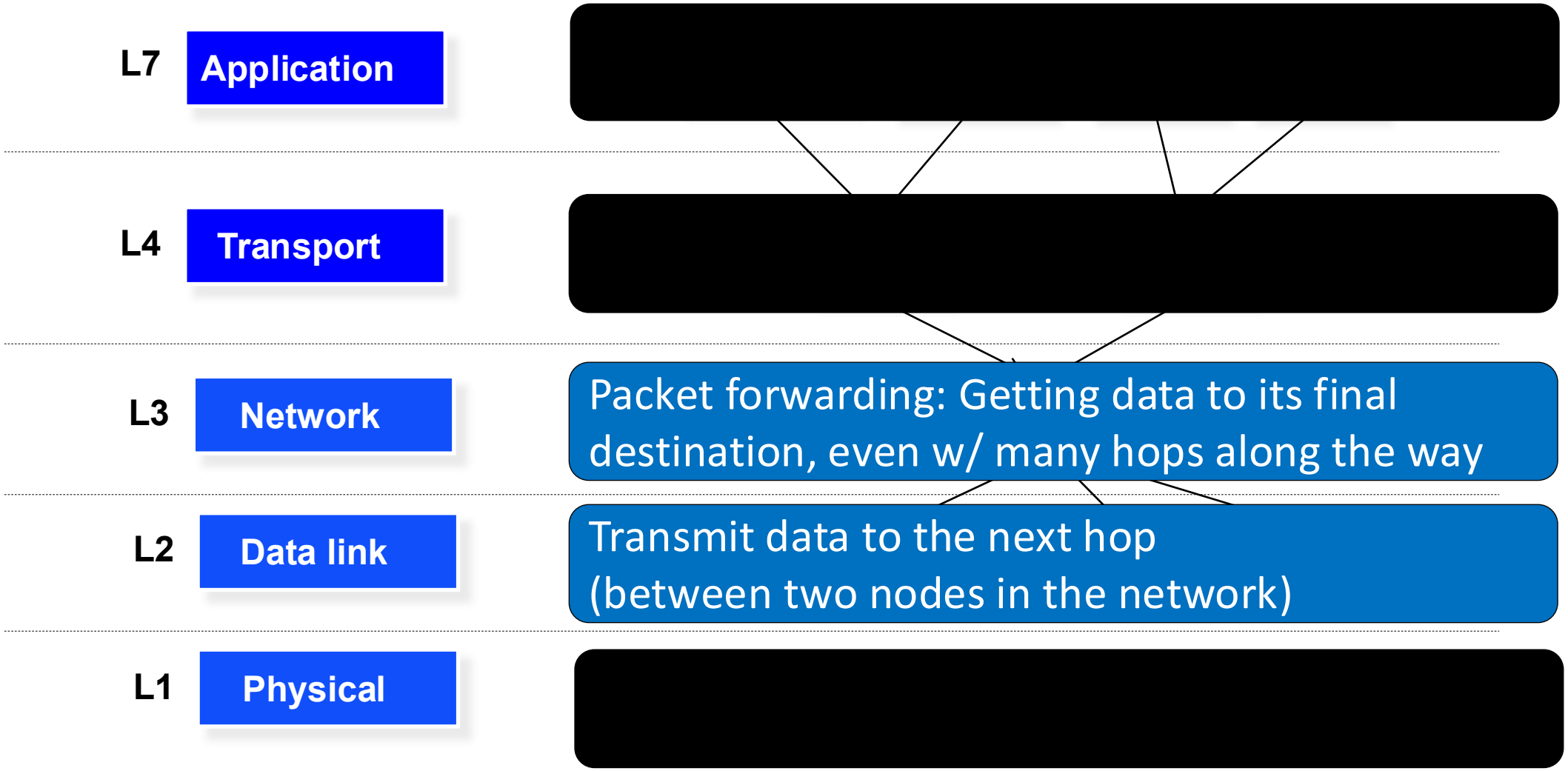
- Office hours next week (Apr 27 – 30)
  - Instructors: 11 – 12pm on Tues as usual
  - TA's: Monday at 11:30am, Wed at 2pm, Thurs at 11:30am

  - No assignment due: meant to help answer any conceptual questions (Instructors) or questions about assignment/discussion sections (TA + Instructors)

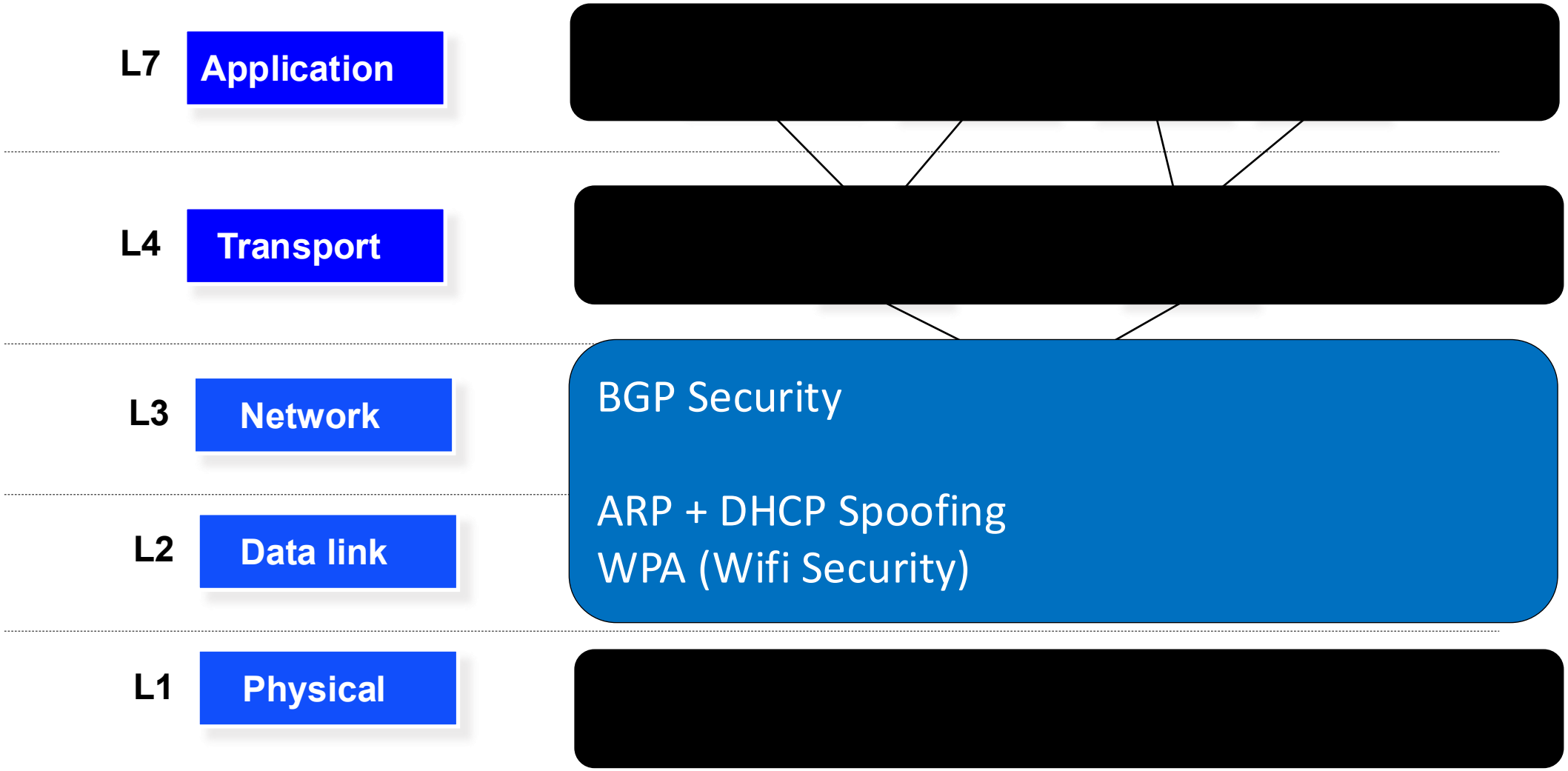
Outline

- TCP/UDP Security
- DNS Security
- Denial of Service (Availability Attacks)
- Network Scanning & Firewalls

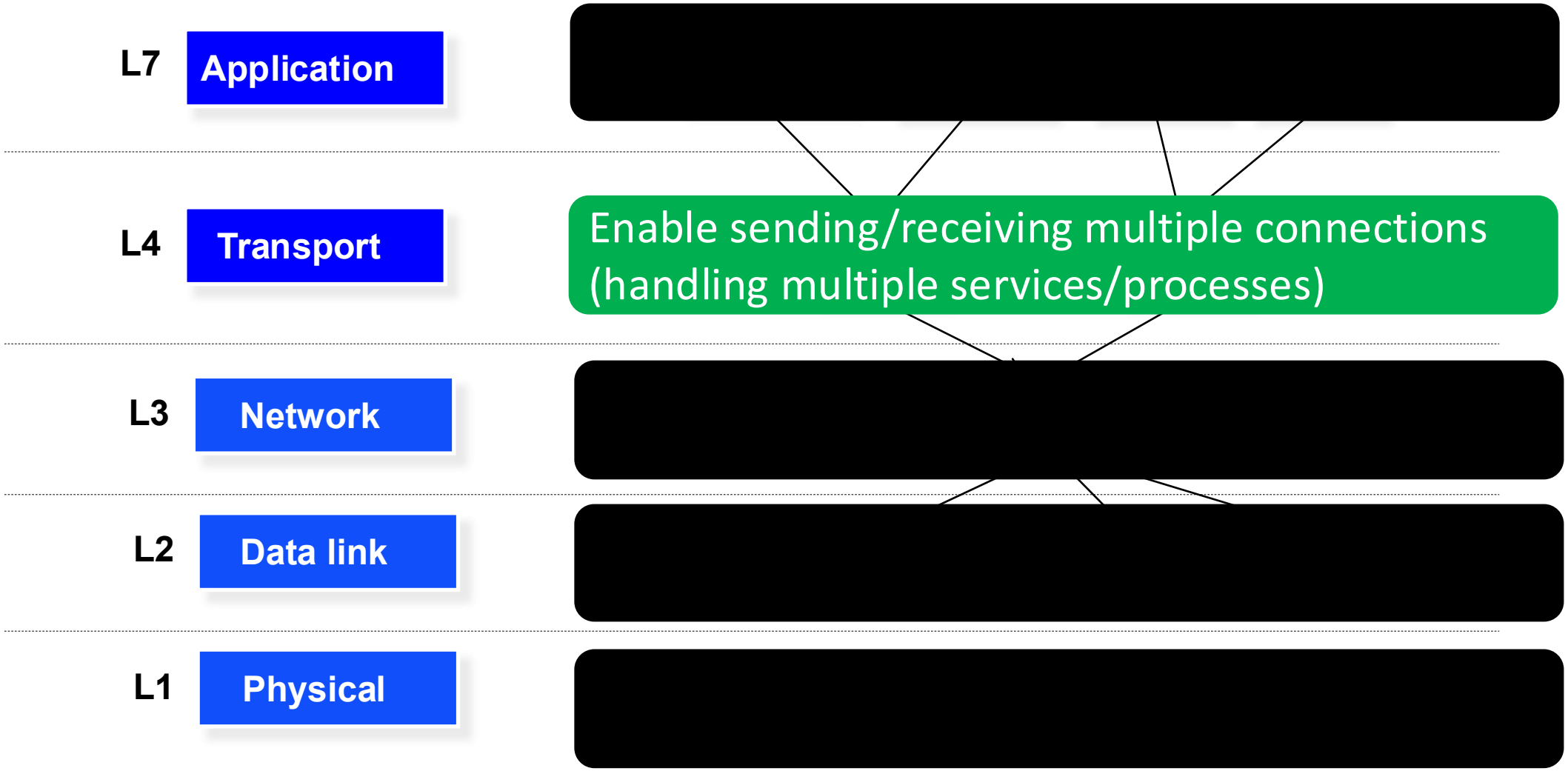
Last Class: Focused on Layer 2/3



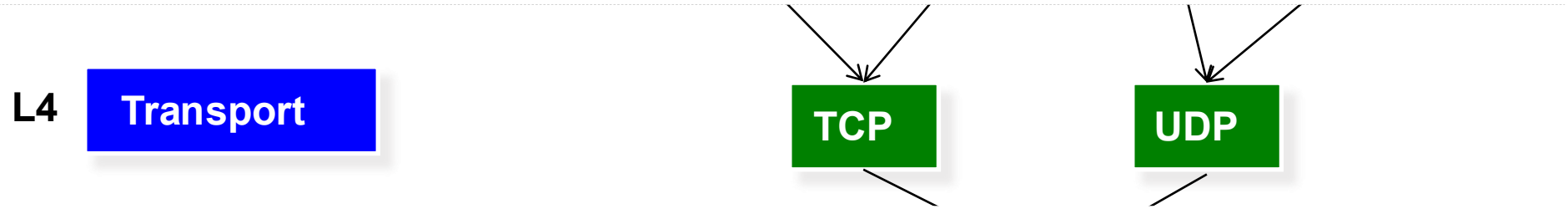
Last Class: Focused on Layer 2/3



Security at Higher Network Layers



Transport Layer Goals



1. Handle multiple connections streams (ports) &
2. Get ALL of the data to its destination

UDP: User Datagram Protocol

UDP: Simple transport protocol that adds ports to support traffic multiplexing (multiple simultaneous connections)

UDP datagram header + payload

Source Port (16 bits)	**Destination Port (16 bits)**
Length (16 bits)	**Checksum (16 bits)**
Payload: Application Data (variable length)	

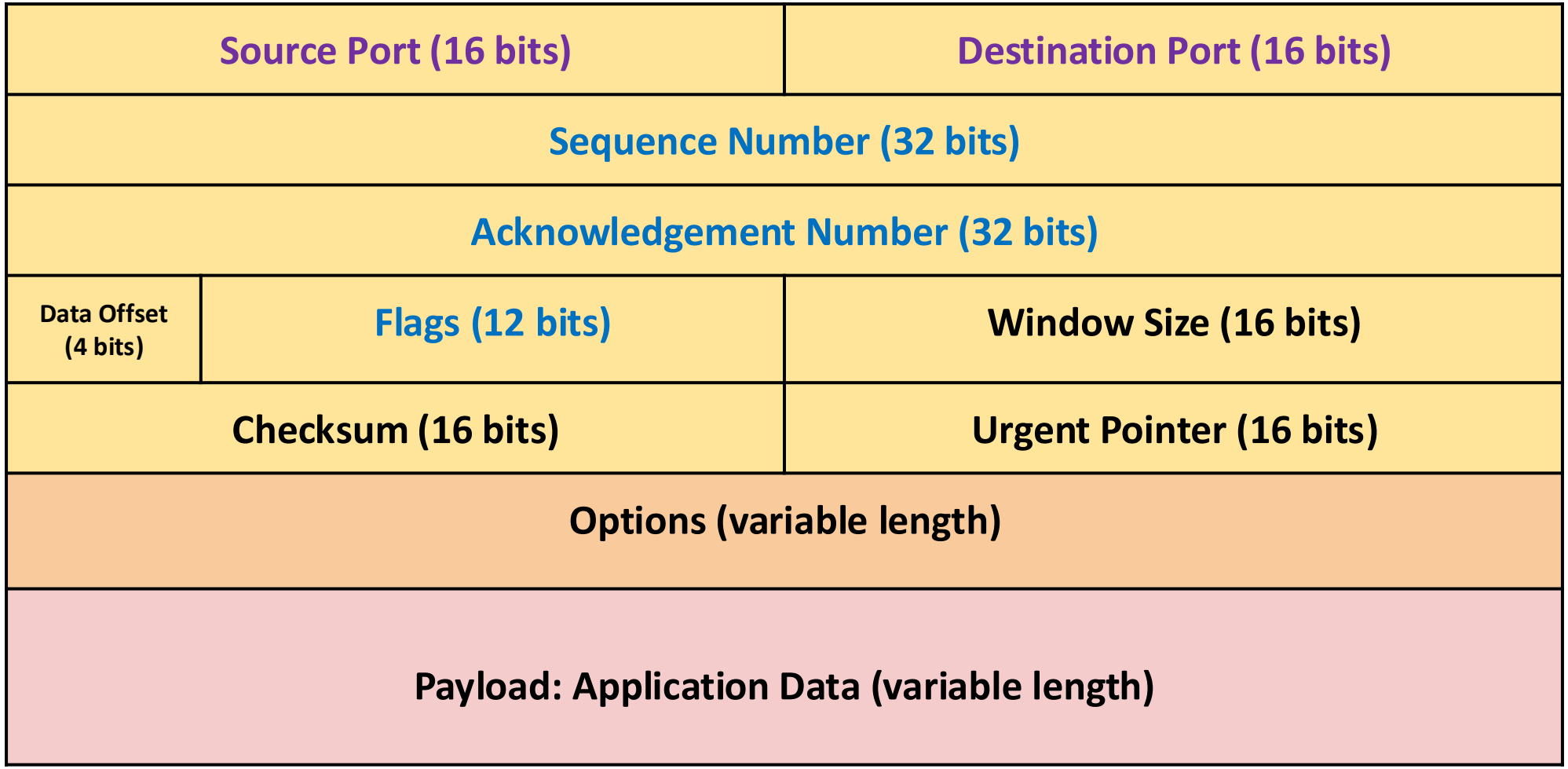
TCP: Reliable Data Streams

- Routing on the Internet is full of hazards: packets being dropped, re-ordered, and duplicated
  - Faulty router, shark nibbling on underwater cable, router power outage, etc.

- Most applications want a stream of bytes delivered reliably and in-order between different hosts

- Transmission Control Protocol (TCP) provides reliable + in-order byte streams, along with other services (e.g., congestion control)
  - Uses a combination of ***sequence numbers*** and ***flags*** to ensure reliable & controlled connections

TCP Packet Structure



TCP Flags

SYN

- Indicates the beginning of the connection

ACK

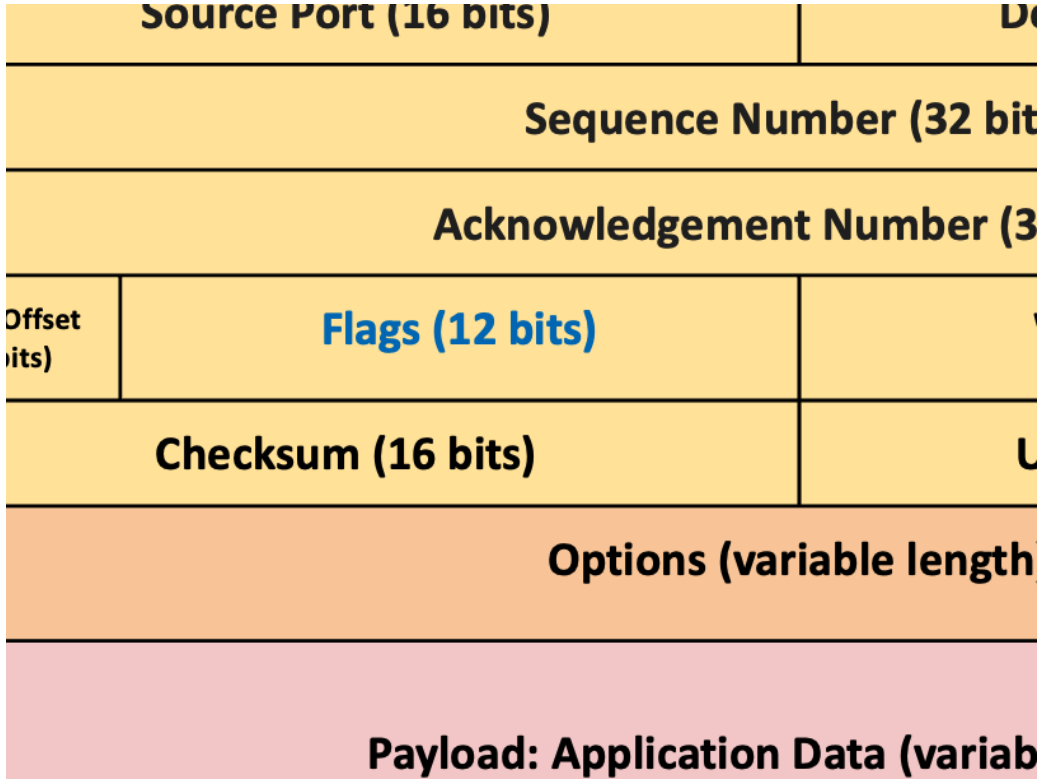
- Acknowledges the receipt of something (in the ack number)
- Pretty much always set except the very first packet

FIN

- One way to end the connection
- Requires an acknowledgement
- No longer sending packets, but will continue to receive

RST

- One way to end a connection
- Does not require an acknowledgement
- No longer sending or receiving packets



TCP Sequence Numbers

- Two data streams in a TCP session, one in each direction
- Bytes in data stream numbered with a 32-bit sequence number
- Every packet has sequence number that indicates where data belongs
- Receiver sends acknowledgement number that indicates data received

H	e	l	l	o		s	e	r	v	e	r
50	51	52	53	54	55	56	57	58	59	60	61

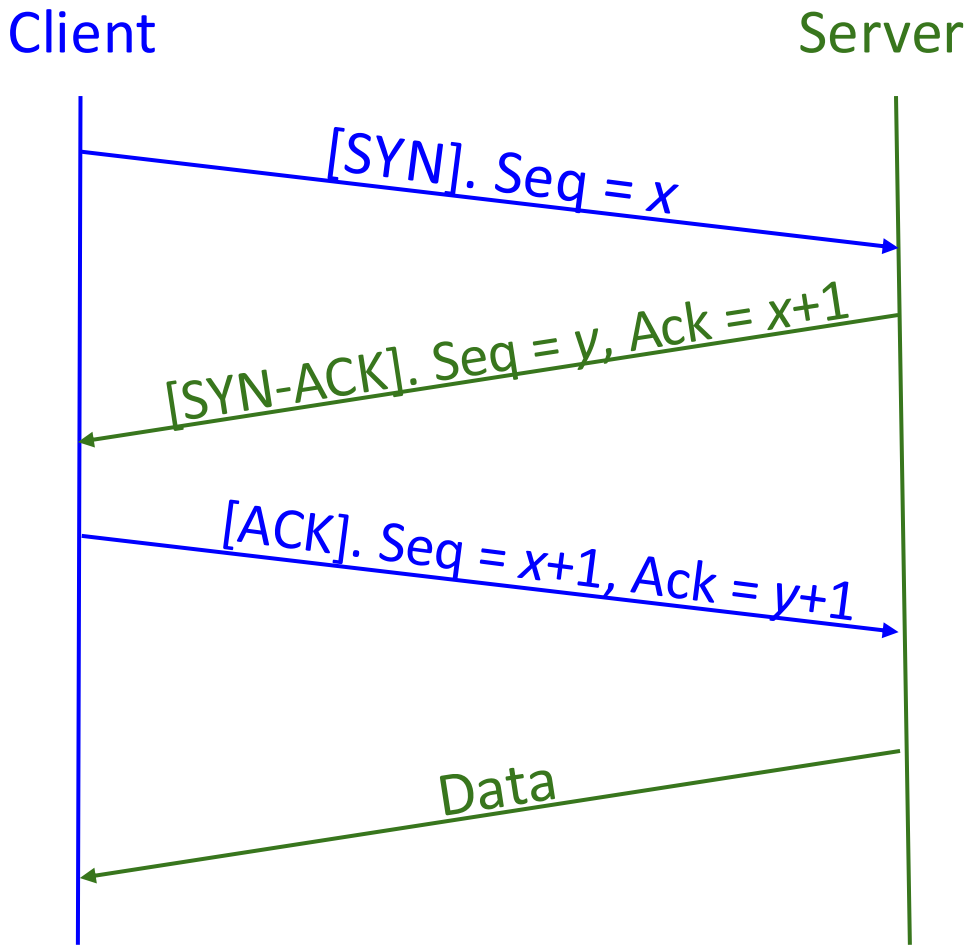
H	e	l	l	o		c	l	i	e	n	t
25	26	27	28	29	30	31	32	33	34	35	36

Example: Bytes from the client are numbered starting at 50.

Example: Bytes from the server are numbered starting at 25.

TCP Setup: 3-Way Handshake

1. Client chooses an random initial sequence number (ISN) *x*: sends a SYN (synchronize) packet to the server
2. Server chooses an random ISN *y* for bytes it will send and responds with a SYN-ACK packet
3. Client responds with an ACK packet
4. Once both hosts have synchronized sequence numbers, the connection is “established”



TCP: Sending and Receiving Data (After Setup)

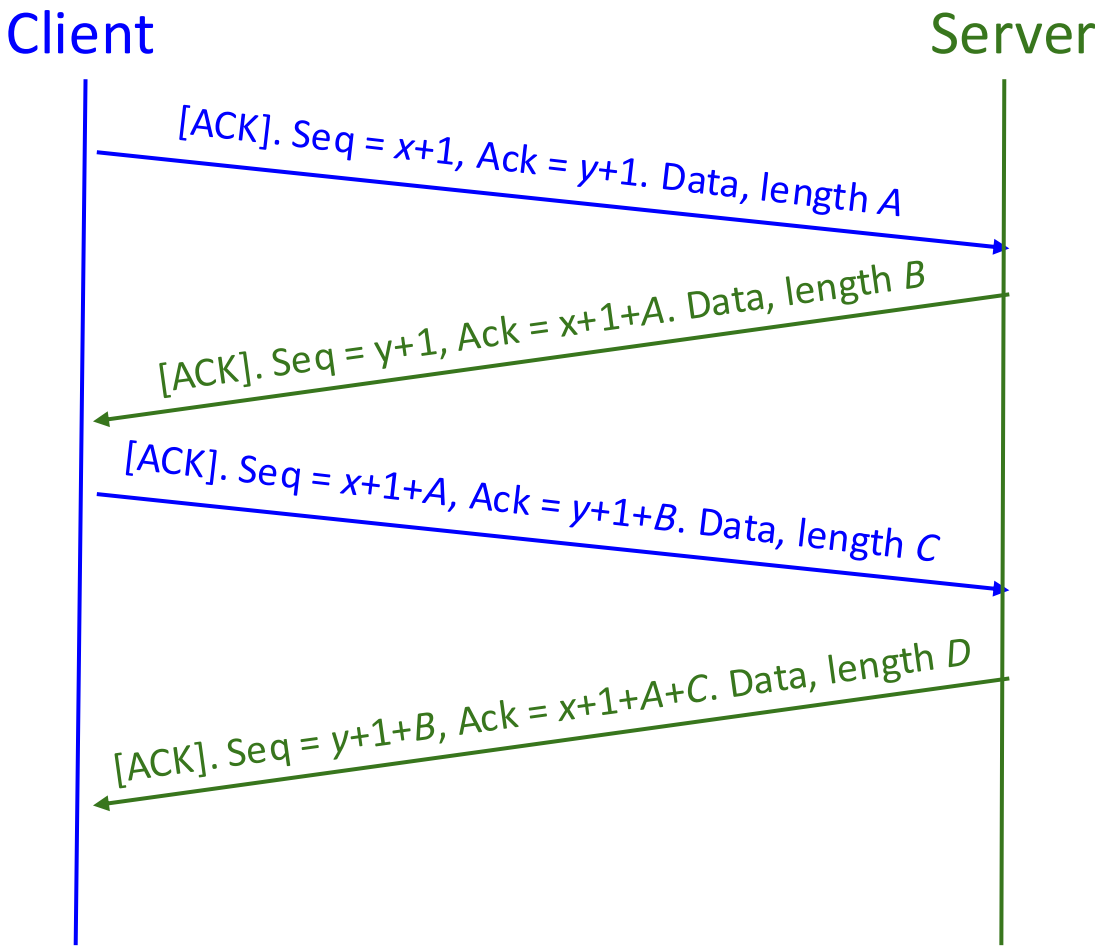
TCP headers use seq #'s to provide ordering and reliable delivery

A packet's sequence number = the sequence number of the first byte of its current data

- Byte *i* of the byte stream is represented by sequence number *x* + *i* (client) or *y* + *i* (server)

A packet's ACK number = the sequence number of the byte it expects to receive next

- Equal to (sequence number) + (length of data) for the last received packet



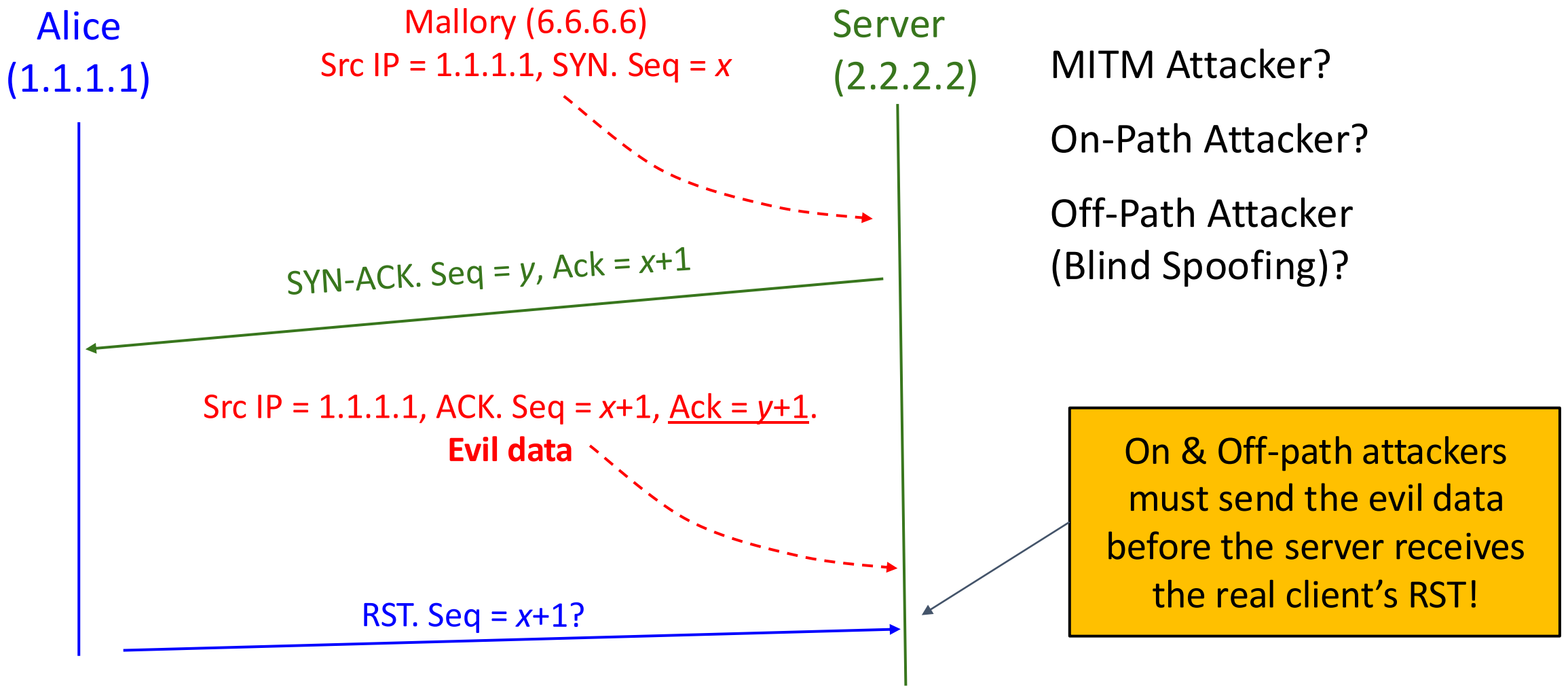
TCP: Retransmission (Reliable Delivery)

- If a packet is dropped (lost in transit):
  - Recipient won't send an ACK, so the sender will not receive the ACK
  - The sender repeatedly tries to send the packet again until it receives the ACK

- If a packet is received by recipient, but the ACK is dropped:
  - Sender won't receive an ACK, so they try to send packet again
  - The recipient ignores the duplicate data and sends the ACK again

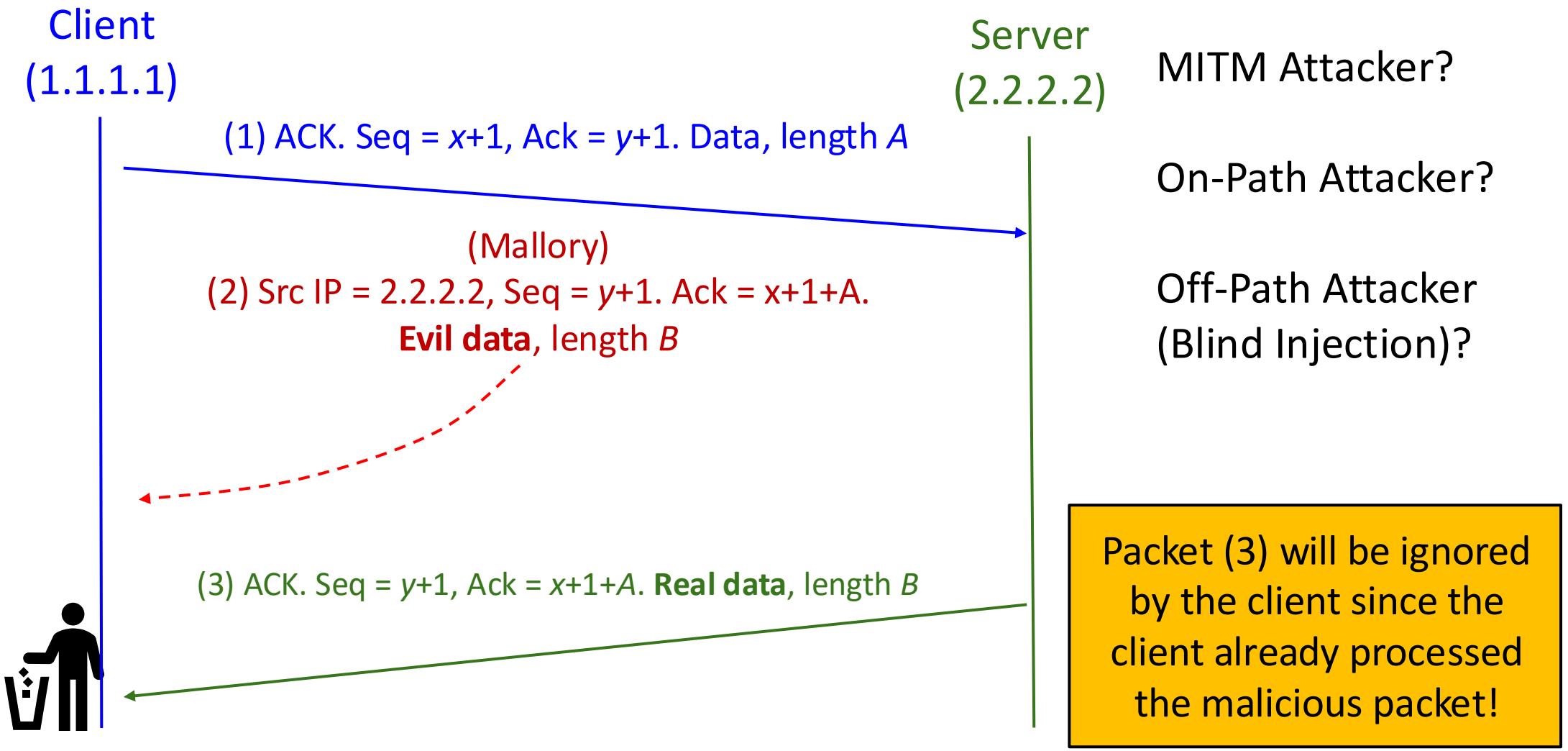
Attacks: TCP Spoofing

(Spoof connection to appear to come from someone else [Alice])



Attacks: TCP Data Injection

(Tampering with an existing session to modify / inject data into a connection)



TCP & UDP Attacks

- TCP provides no confidentiality or integrity
  - Instead, we rely on higher layers (like TLS) to prevent those kind of attacks

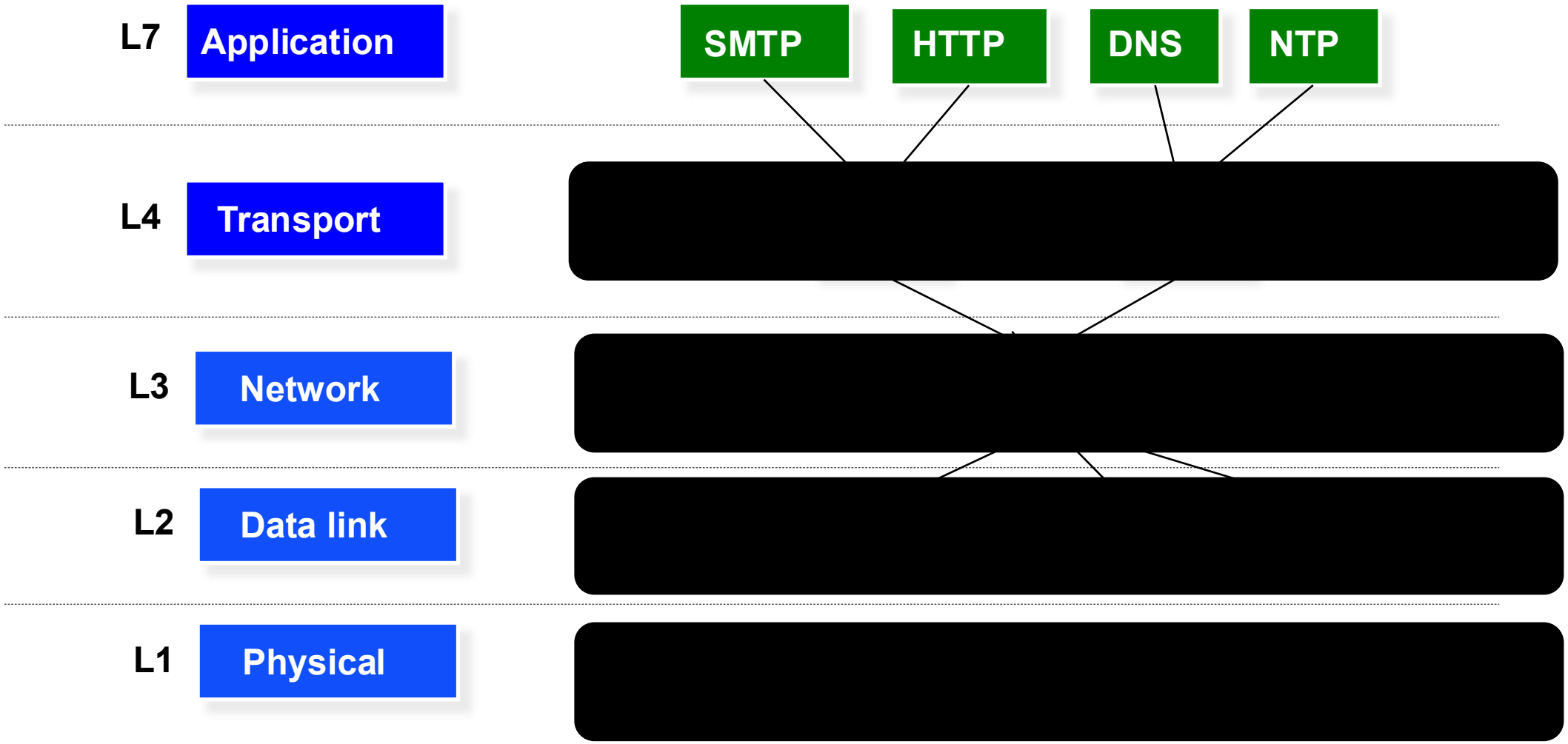
- Defense against off-path attackers rely on choosing random sequence numbers
  - Bad randomness can lead to trivial off-path attacks:
    TCP sequence numbers used to be based on the system clock!

- UDP: Attacks even easier! No sequence numbers to guess/forge.

Outline

- TCP/UDP Security
- DNS Security
- Denial of Service (Availability Attacks)
- Network Scanning & Firewalls

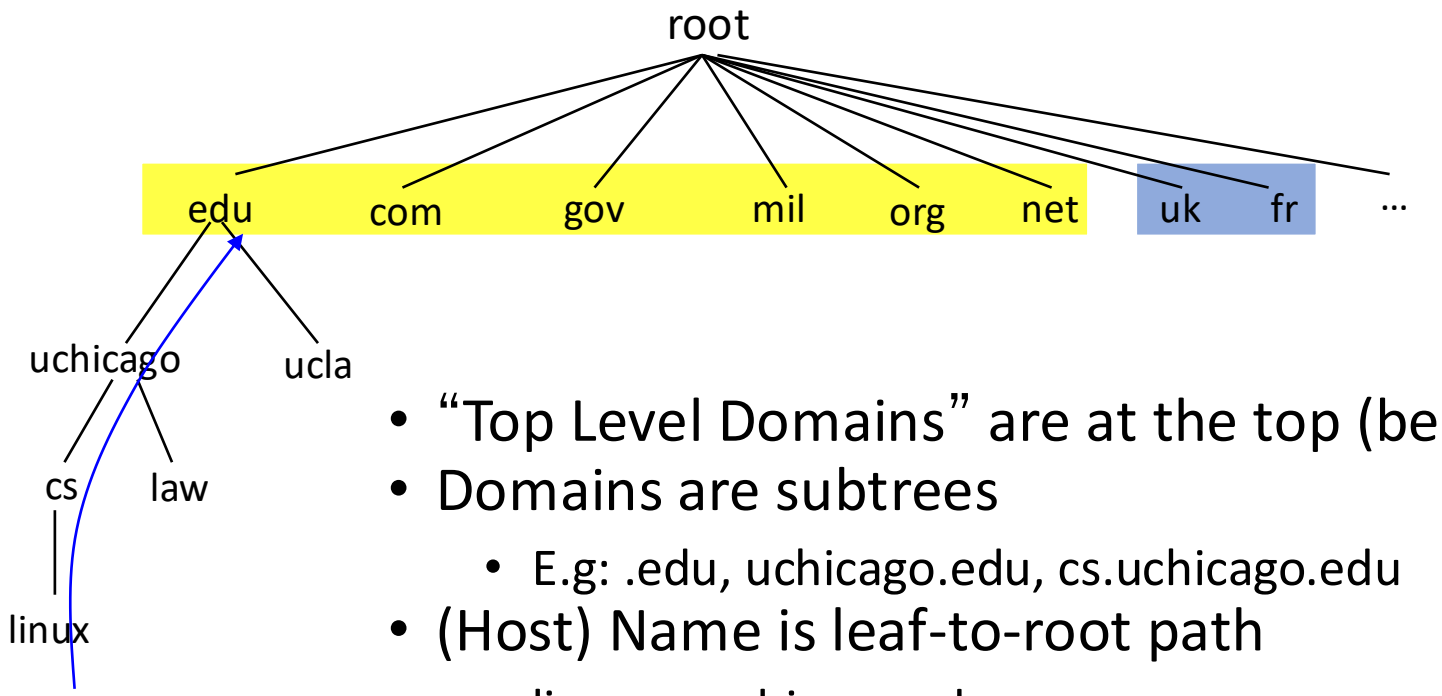
Attacks at Layer 7 (DNS)



DNS (Domain Name System)

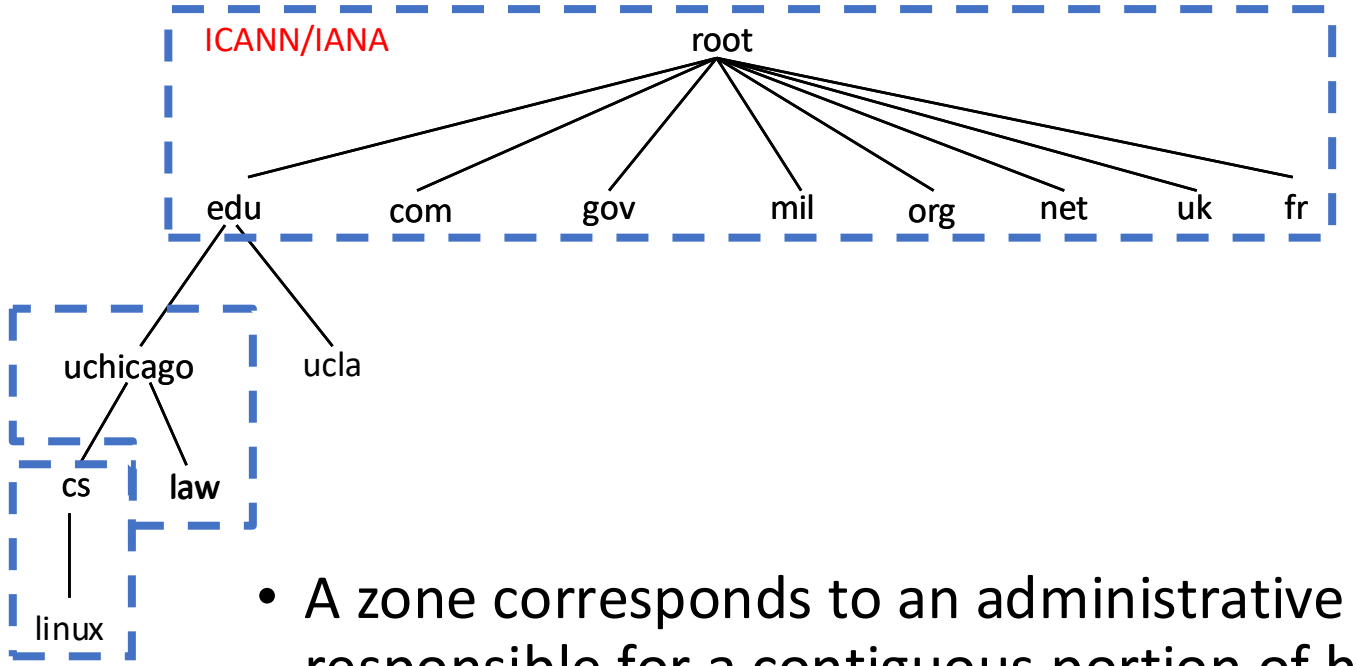
- Host (machine) IP addresses: e.g., *128.135.11.239*
  - A number used by protocols (e.g., BGP)
- Host names: e.g., *super.cs.uchicago.edu*
  - Usable by humans
- Domain Name System (DNS): how we map from hostnames to IP addresses

Hierarchical Namespace



- "Top Level Domains" are at the top (below root)
- Domains are subtrees
  - E.g: .edu, uchicago.edu, cs.uchicago.edu
- (Host) Name is leaf-to-root path
  - linux.cs.uchicago.edu
- Name collisions trivially avoided!
  - each domain's responsibility

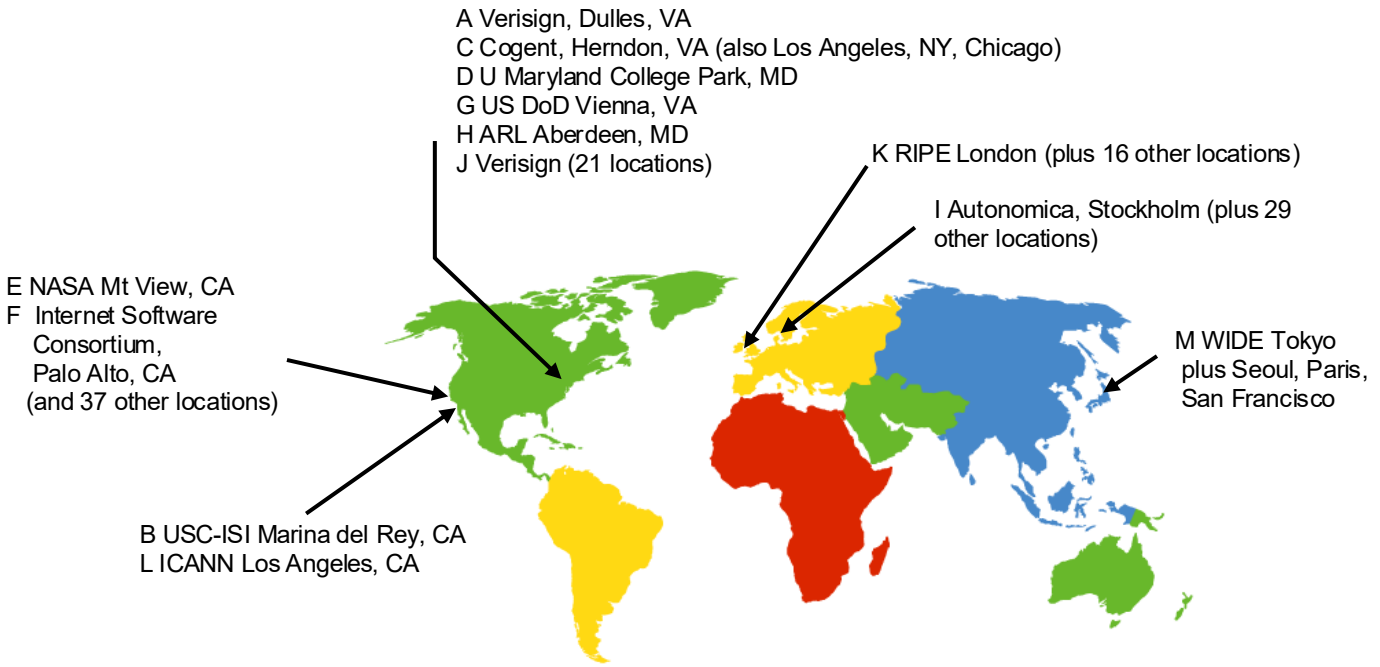
Hierarchical Administration



- A zone corresponds to an administrative authority responsible for a contiguous portion of hierarchy
- UChicago controls law.uchicago.edu and *.cs.uchicago.edu while CS controls *.cs.uchicago.edu

DNS Root Servers

- 13 root servers (labeled A-M; see http://www.root-servers.org/)
- All replicated via anycast



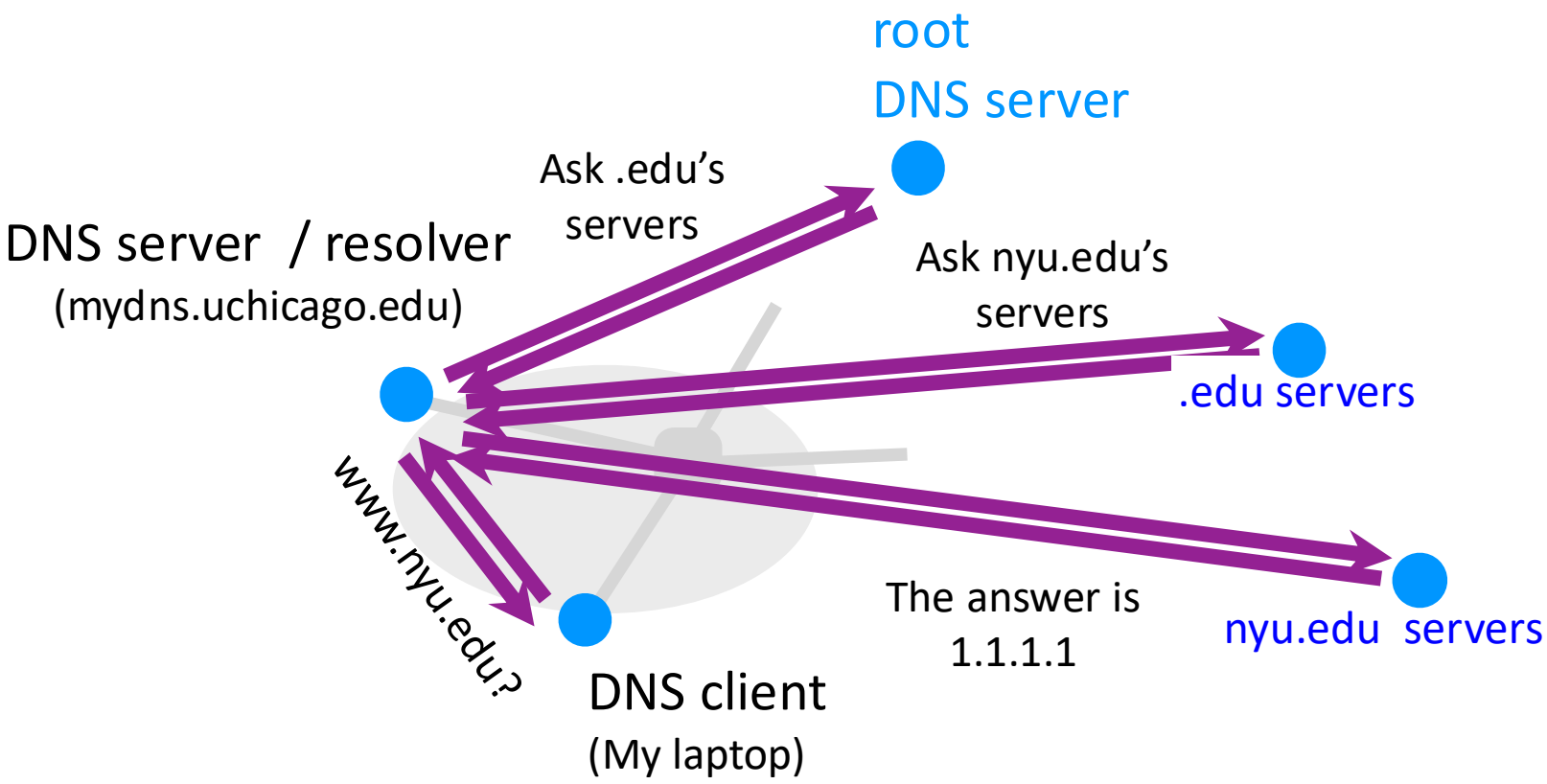
DNS Records

- 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)

Registering a Domain

- Example: you want “blaseur.com”
- Register blaseur.com at registrar (e.g., Dreamhost)
  - Provide registrar with names and IP addresses of your authoritative name server(s)
  - Registrar inserts your name server’s info into the .com TLD server
- You store resource records in your domain’s DNS (name)server
  - e.g., type A record for www.blaseur.com
  - e.g., type MX record for blaseur.com

DNS Query (Lookup)



DNS Packet Format: UDP Header

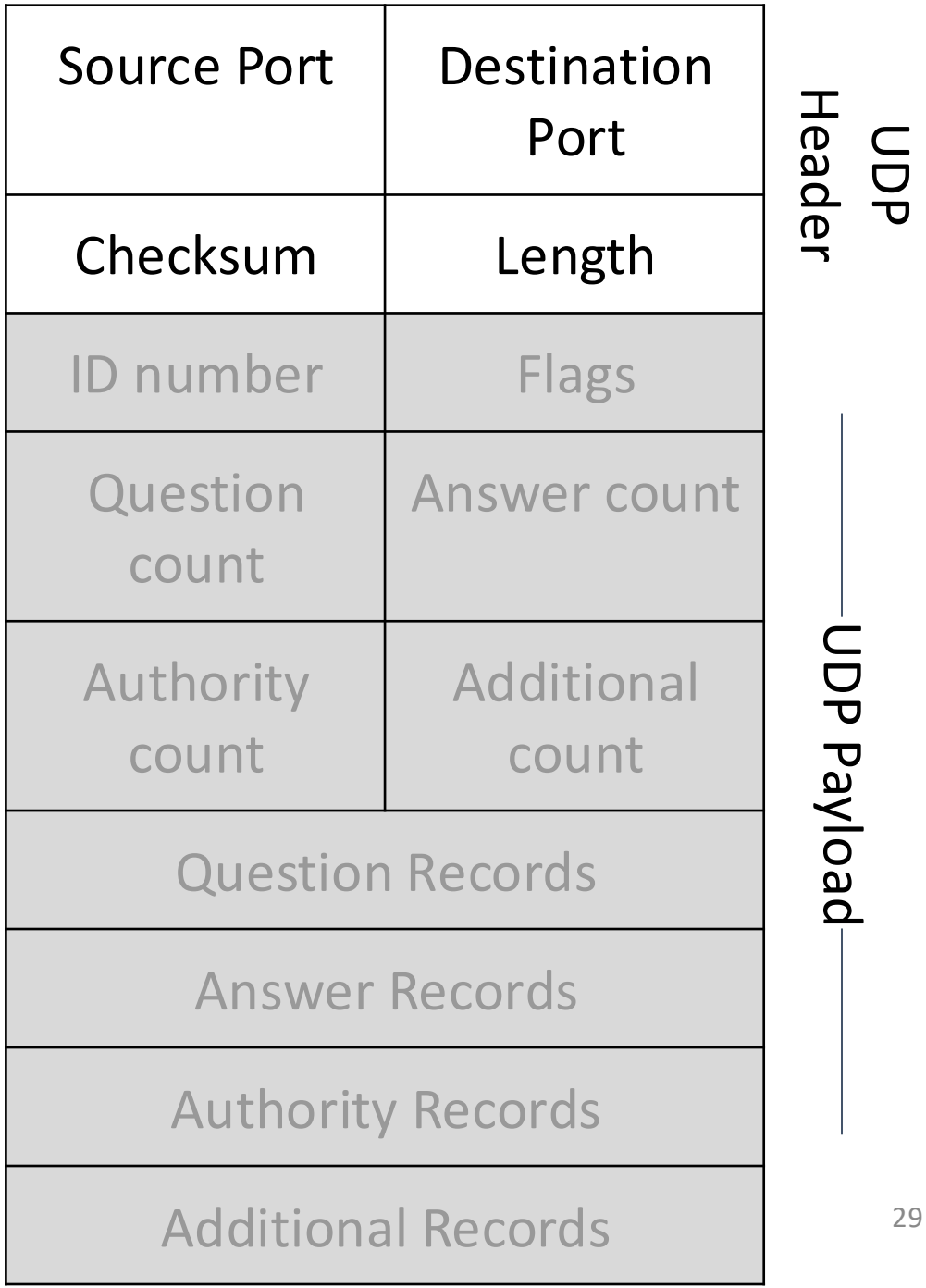
DNS is designed to be lightweight and fast: uses UDP for transport protocol

Source port (16 bits): Chosen by the client

- Can be randomized for security, as we'll see later

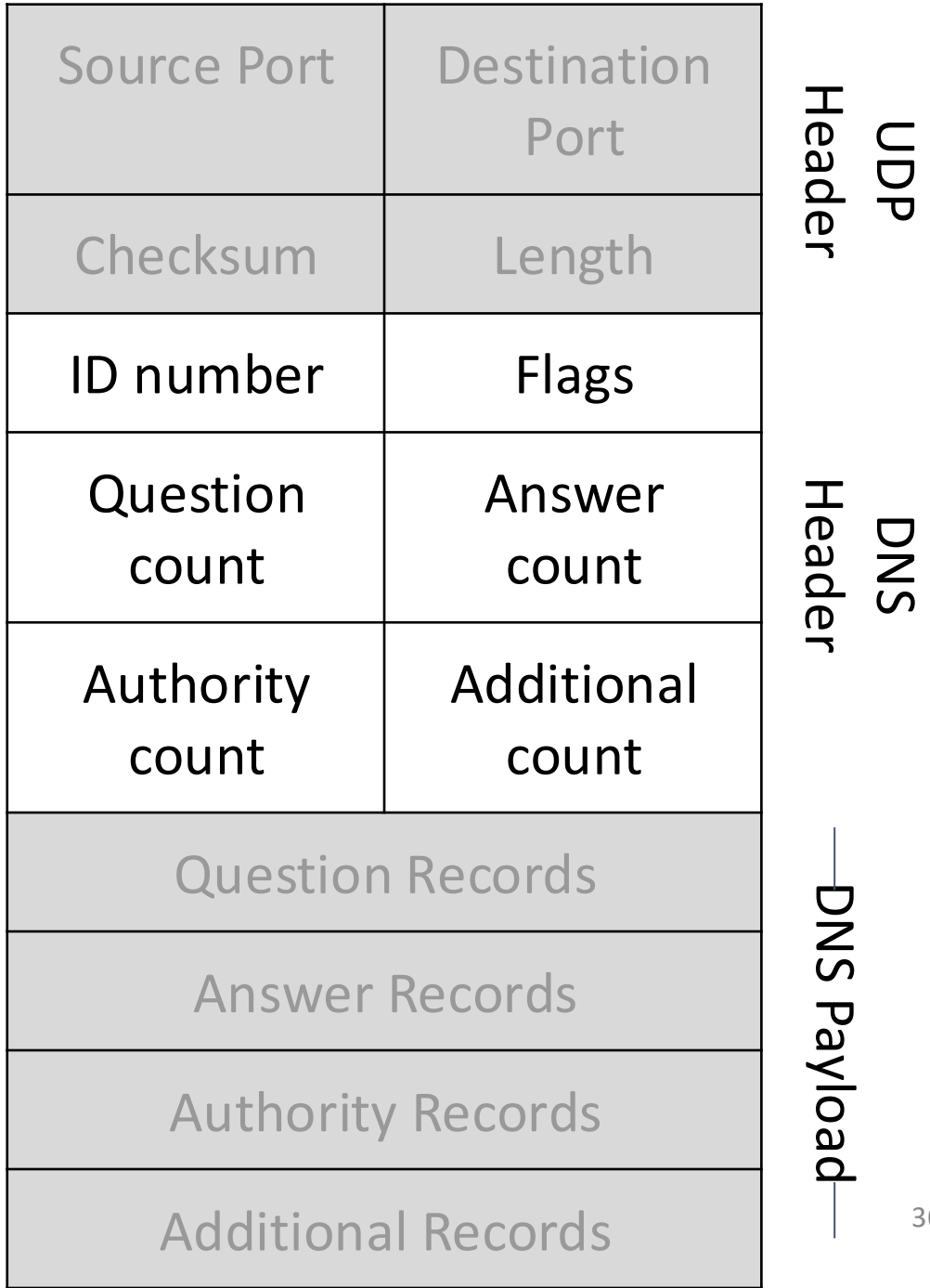
Destination port (16 bits): Usually 53

- DNS name servers answer requests on port 53



DNS Packet Format: DNS Header

- **ID number** (16 bits): Used to associate queries with responses
  - Client picks an ID number in the query
  - Name server uses the same ID number in the response
  - Should be random for security, as we'll see later



DNS Packet Format: DNS Payload

- The DNS payload contains a variable number of **resource records** (**RRs**)
- RRs are sorted into four sections
  - Question section
    (What is the query?)
  - Answer section
    (What is the final answer?)
  - Authority section
    (Which name servers should I ask for more info?
  - Additional section
    (What are the IP addresses of name servers I should ask?)

Example: DNS Request Packet

Lookup for "www.unixwiz.net"

Request to the ".net" TLD nameserver

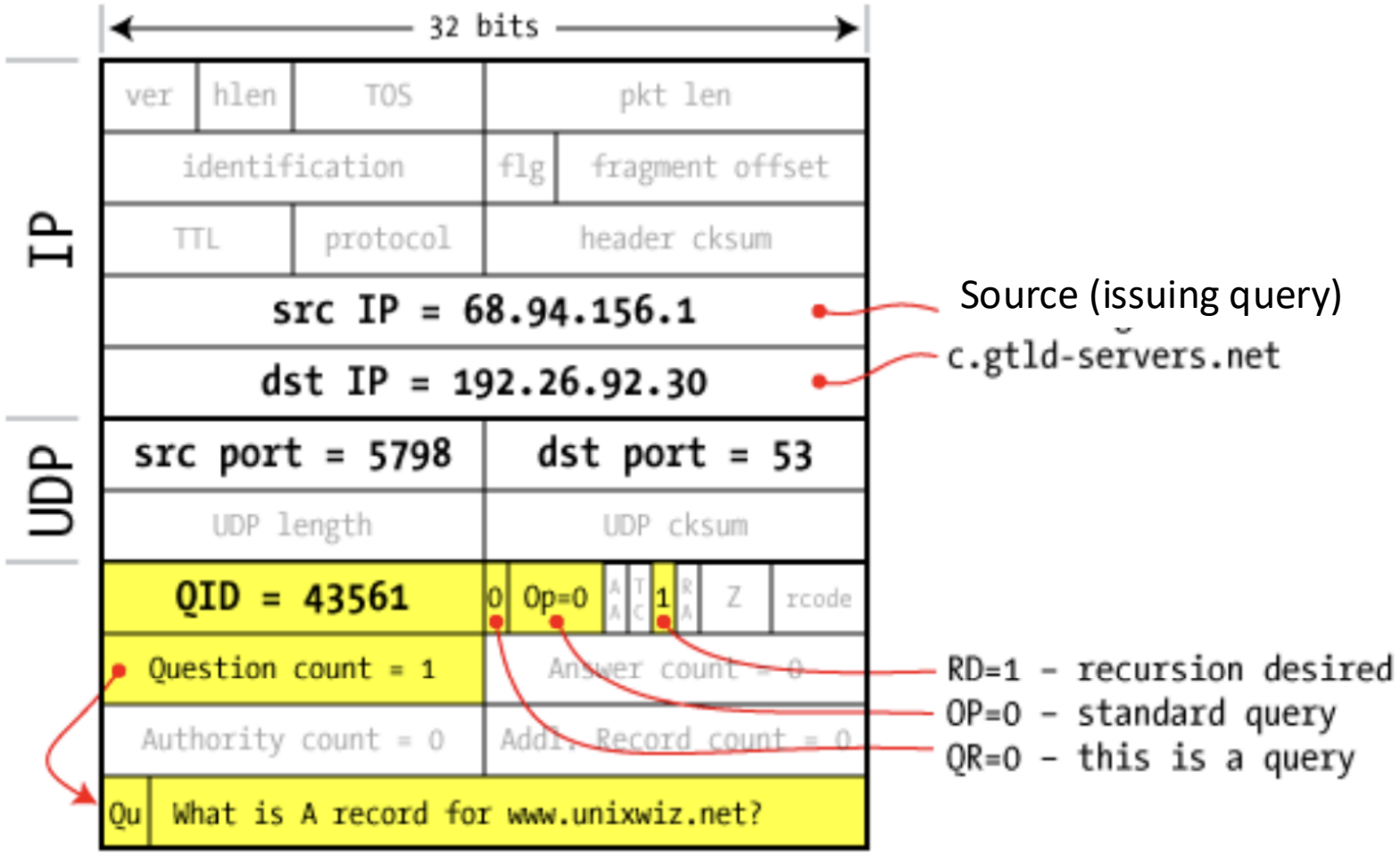


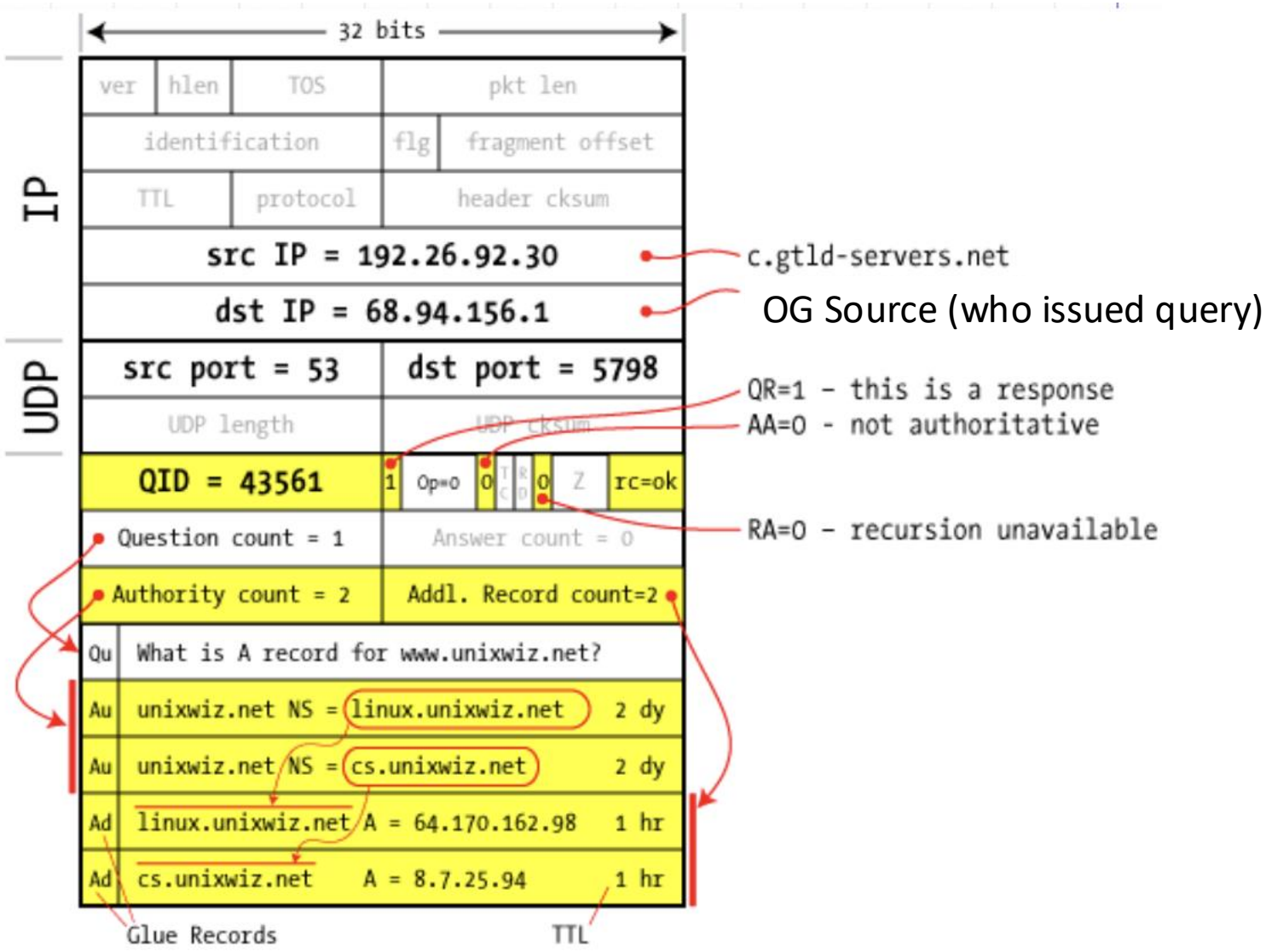
Image from http://unixwiz.net/techtips/iguide-kaminsky-dns-vuln.html

Example: Response Packet

Response by the ".net" TLD nameserver to our local DNS resolver

Authority Section: Who are the name servers you should talk to next?

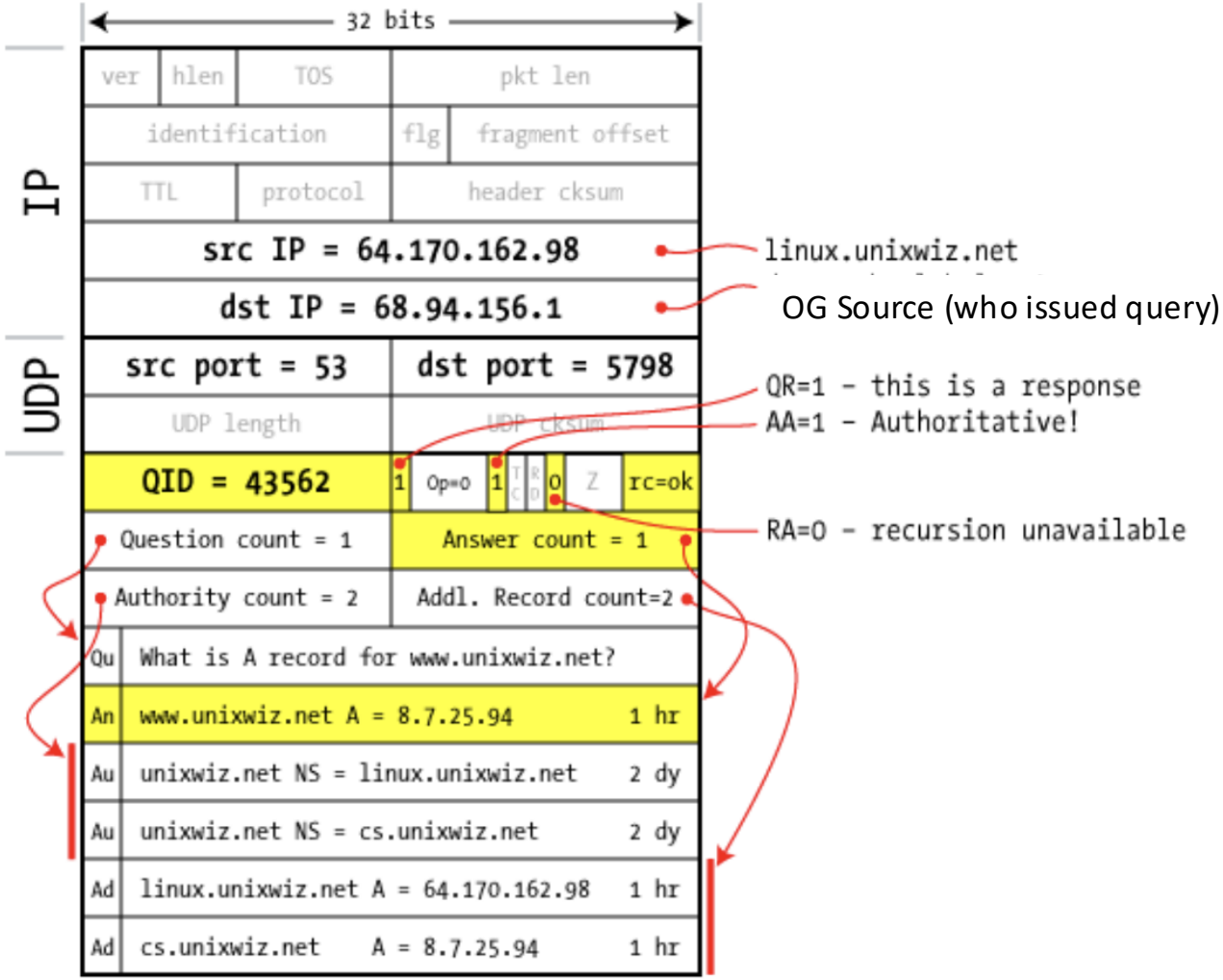
Additional Section [Glue Records]: What are their IP Addresses so you can go ask them?



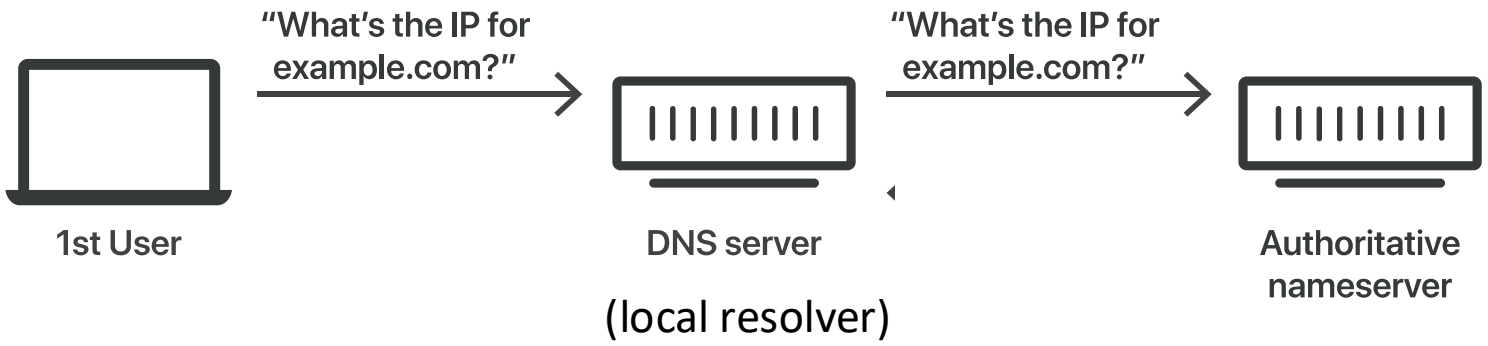
Example: Authoritative Response

Response by the "unixwiz.net" nameservers to our local DNS resolver

Answer Section: What is the IP address for the queried domain: "www.unixwiz.net"?



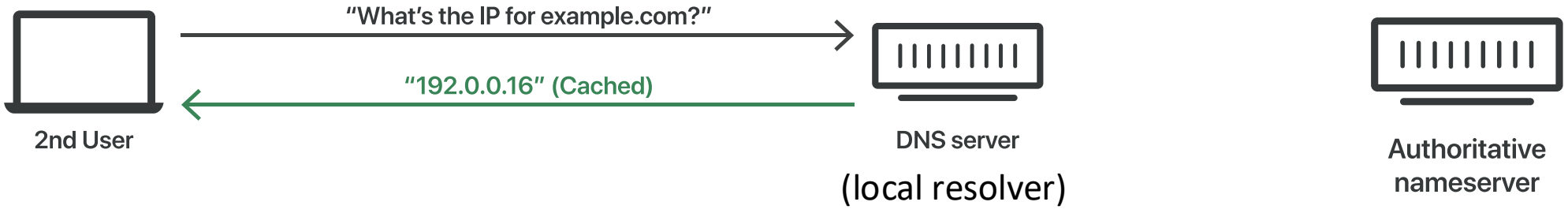
DNS (Uncached)



example.com
IP address: 192.0.0.16

Images from https://www.cloudflare.com/learning/dns/dns-cache-poisoning/

DNS (Cached, Benign)

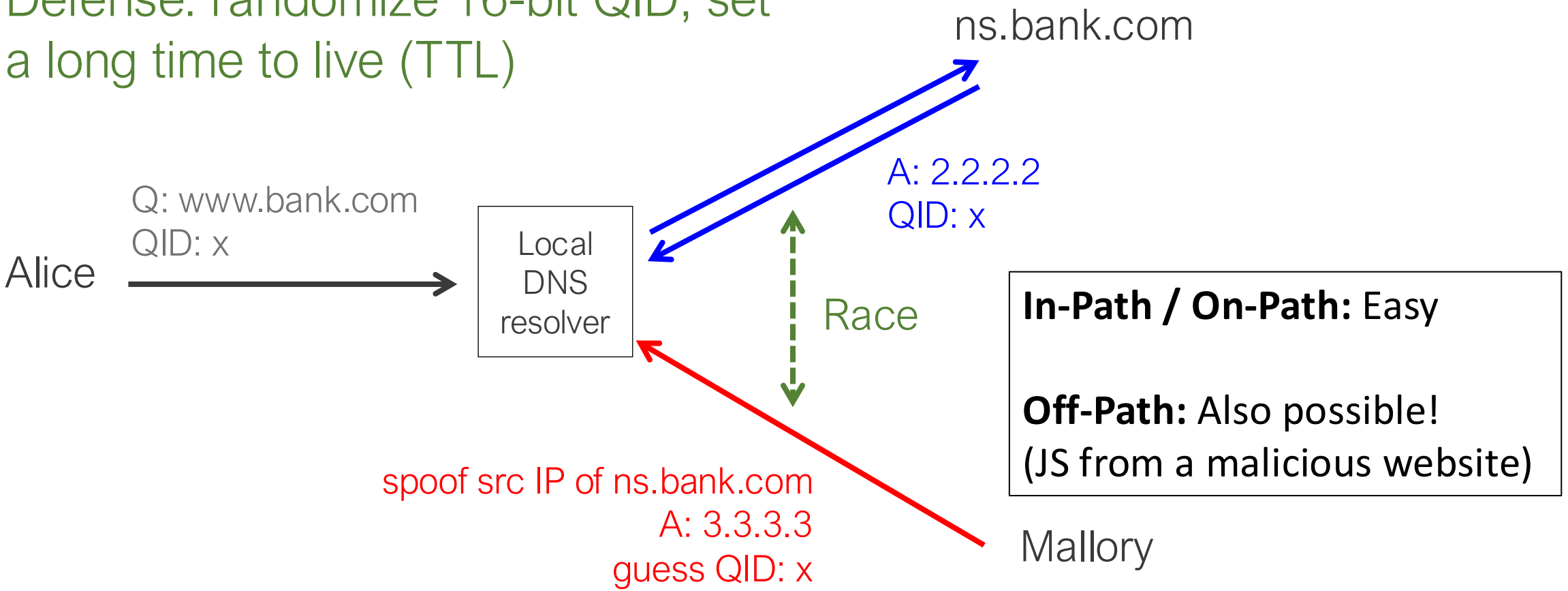


example.com
IP address: 192.0.0.16

Images from https://www.cloudflare.com/learning/dns/dns-cache-poisoning/

Attack: DNS Spoofing (DNS A Records: Single Name -> IP address)

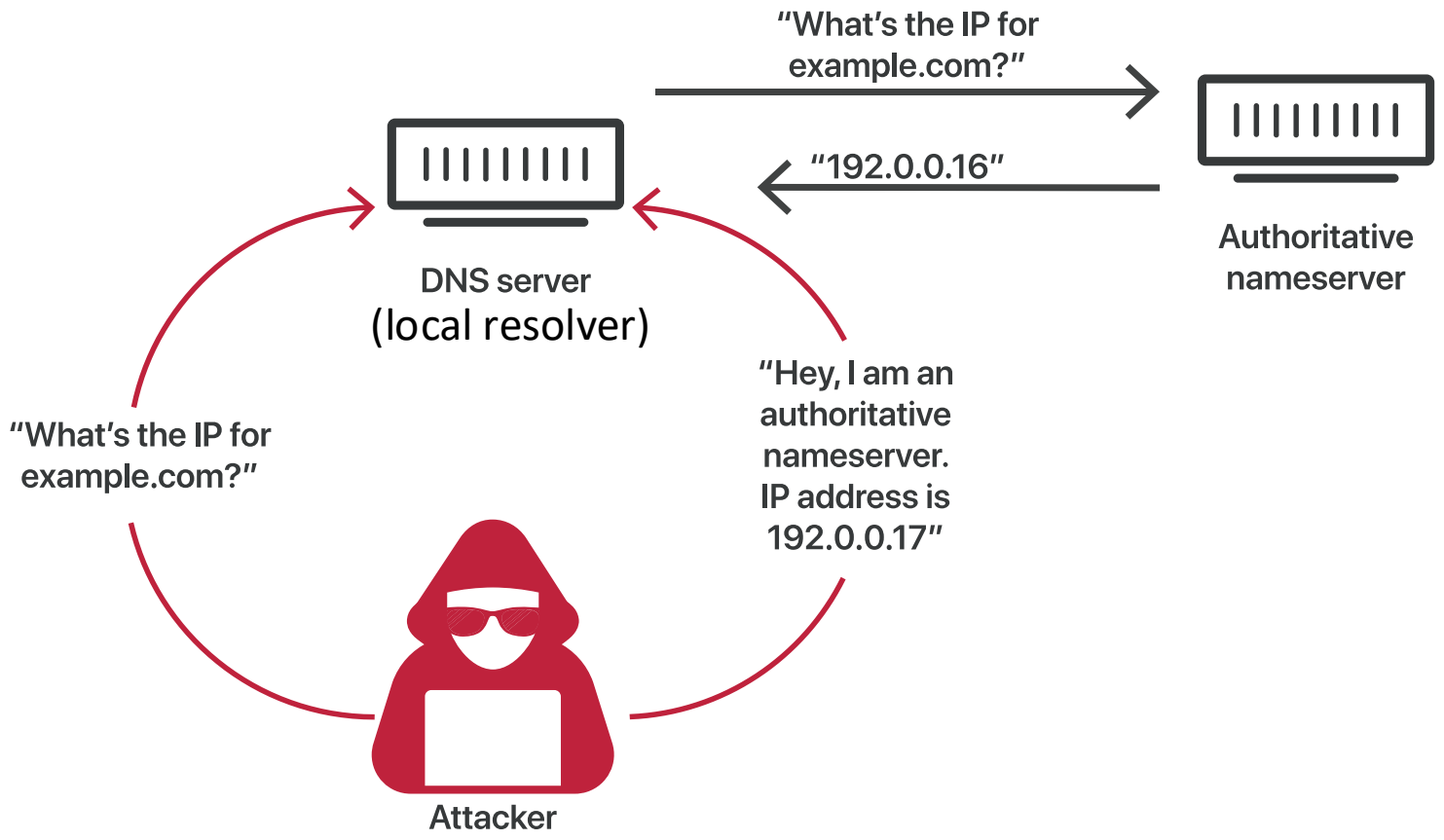
Defense: randomize 16-bit QID, set a long time to live (TTL)



In-Path / On-Path: Easy

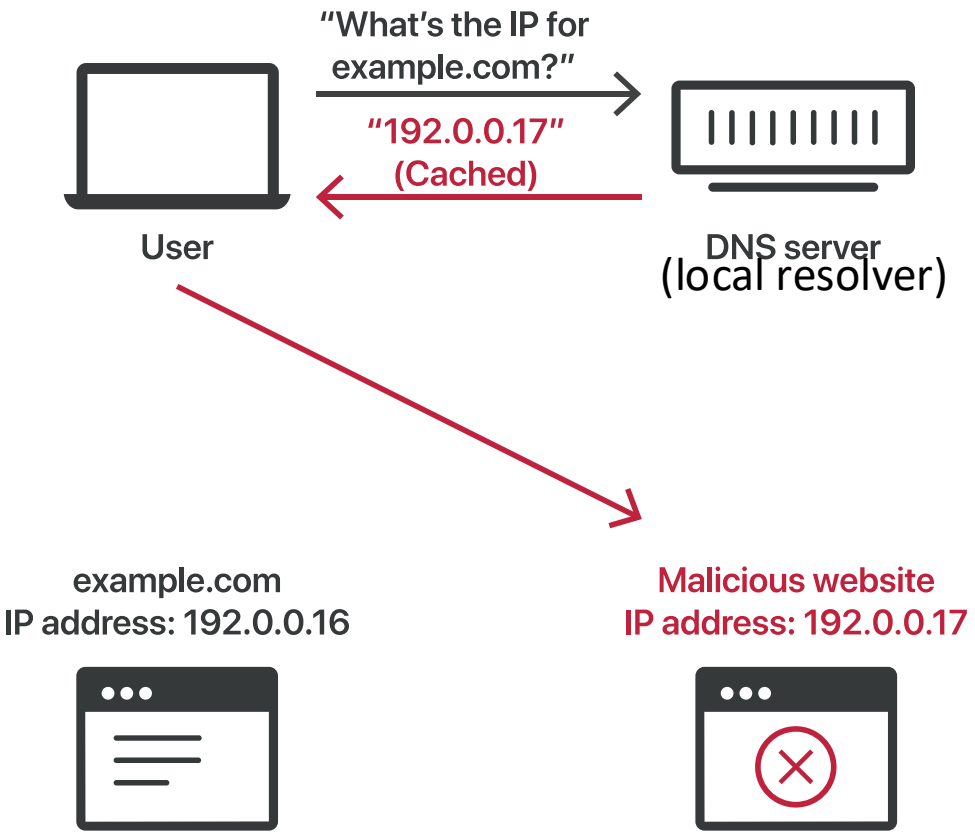
Off-Path: Also possible!
(JS from a malicious website)

DNS Cache Poisoning Attack (DNS A Record)



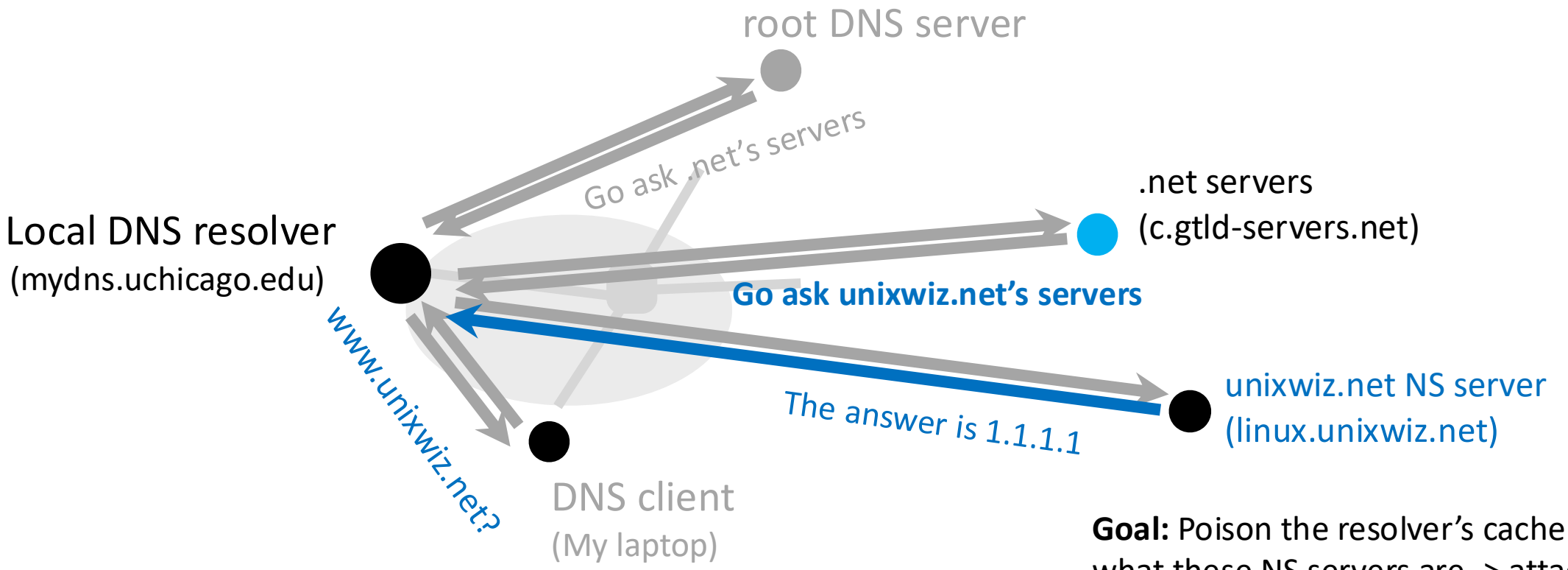
Images from https://www.cloudflare.com/learning/dns/dns-cache-poisoning/

Attack: DNS Cache Poisoning (Poisoning One DNS A Record)



Images from https://www.cloudflare.com/learning/dns/dns-cache-poisoning/

Cache Poisoning: DNS NS Records



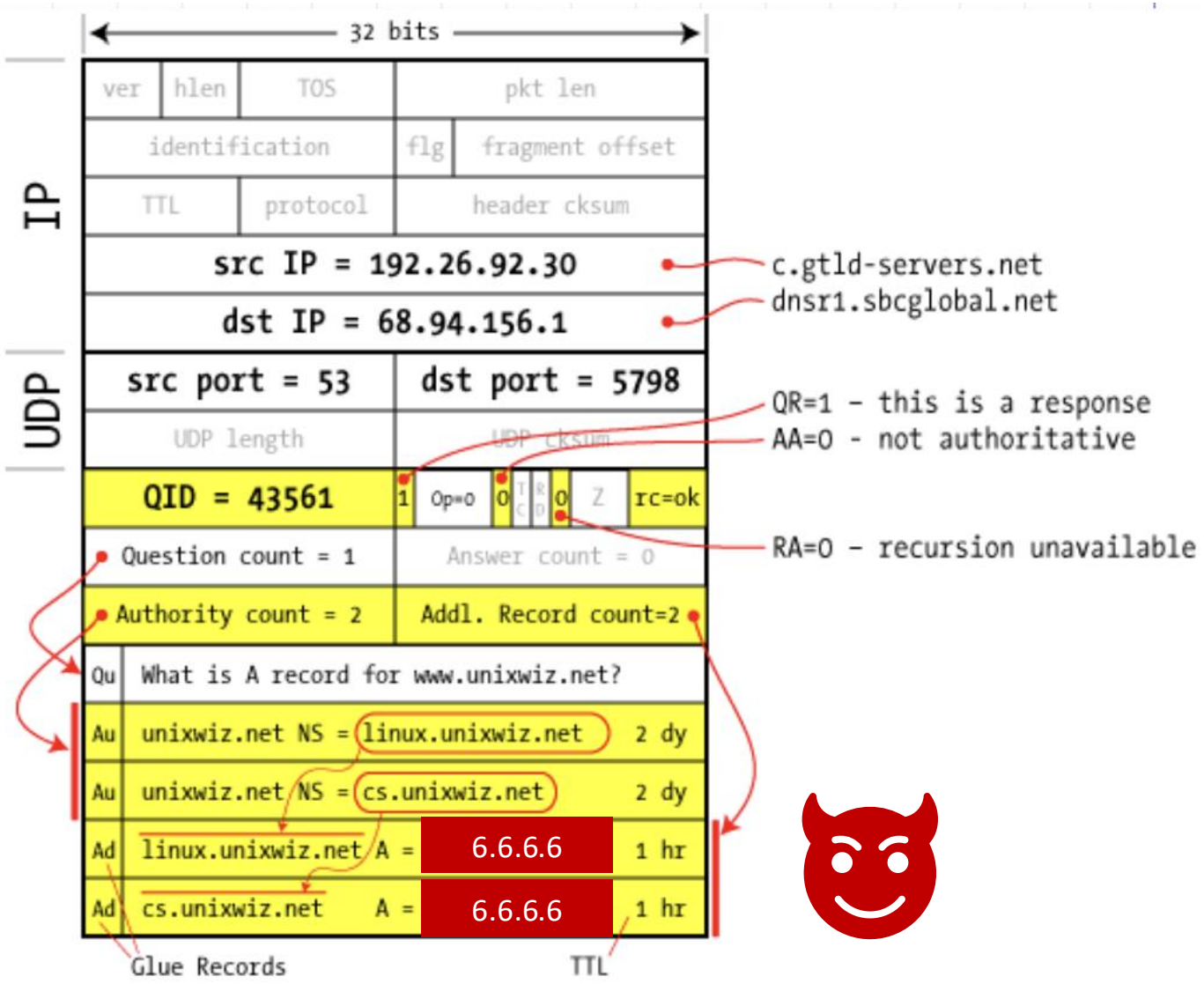
Goal: Poison the resolver's cache of what these NS servers are -> attacker's servers.
(Can control all subdomain IP mappings)

Response Packet w/ NS Info

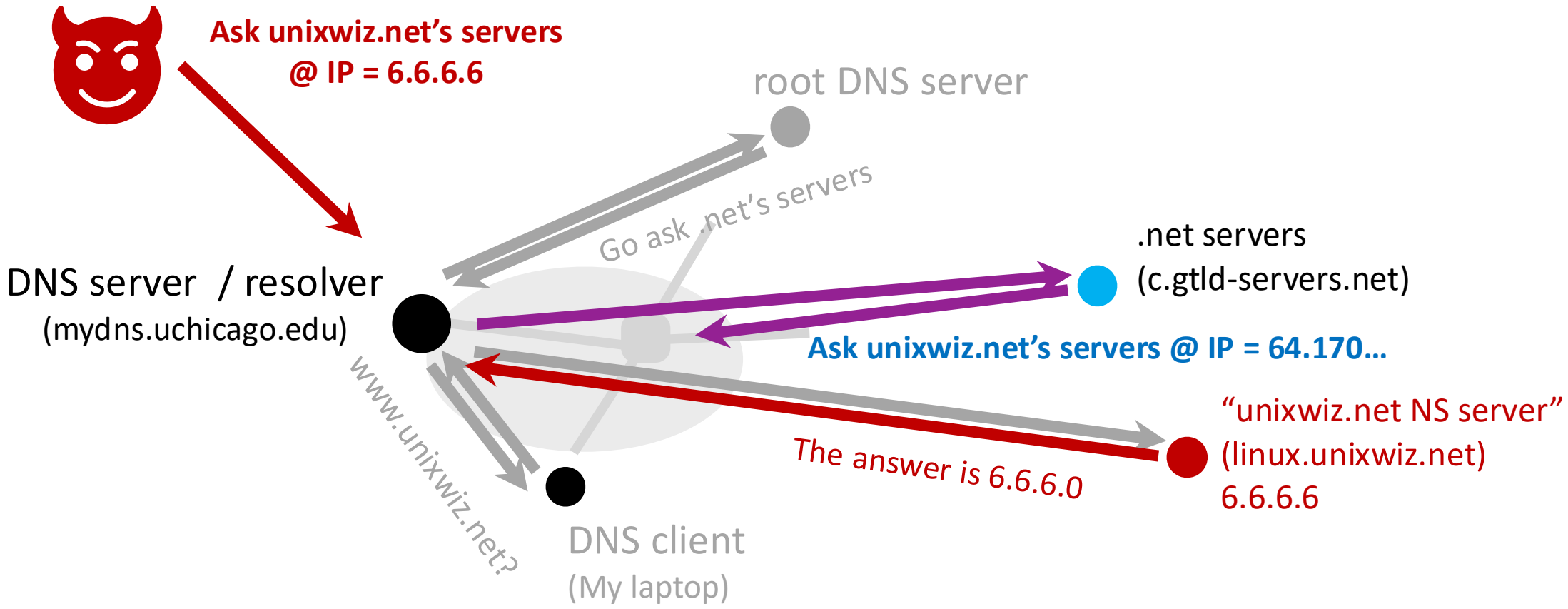
Response by the ".net" TLD nameserver to our local DNS resolver

Authority Section: Who are the name servers you should talk to next?

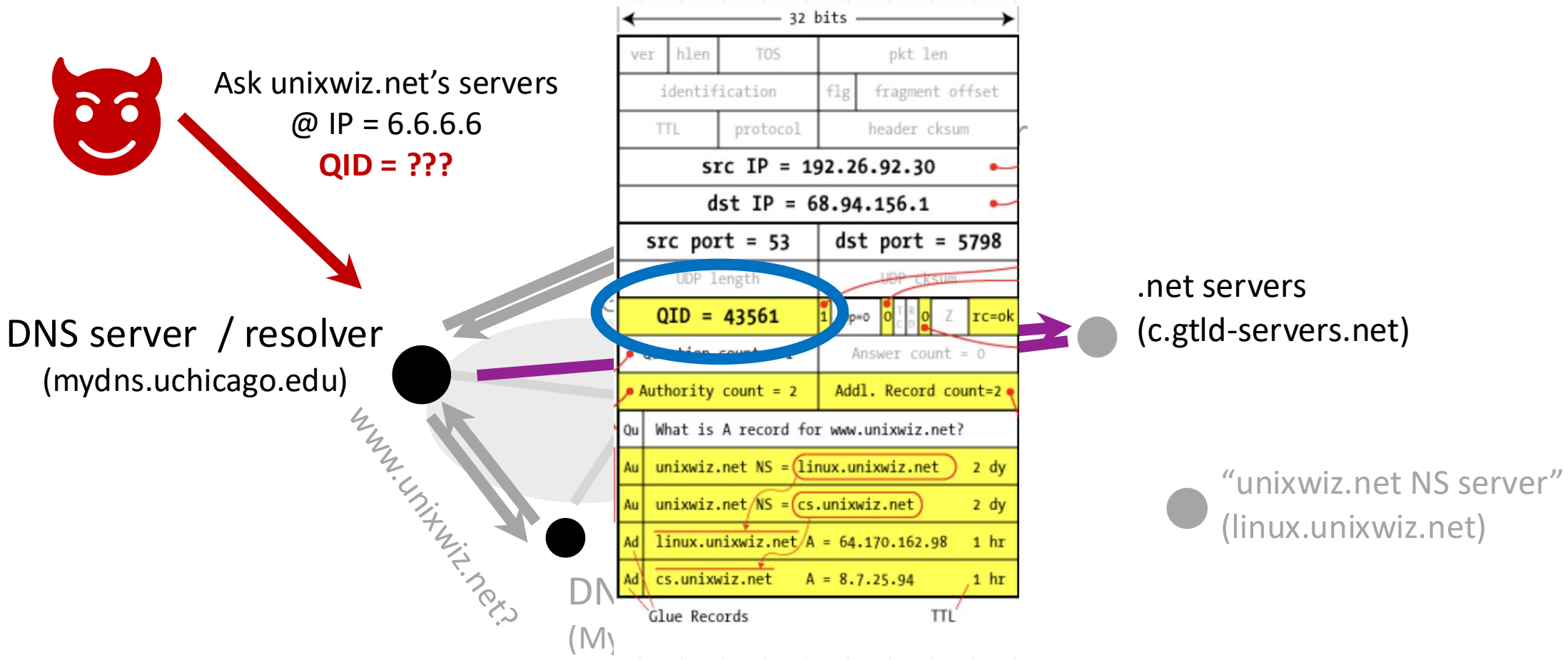
Additional Section [Glue Records]: What are their IP Addresses so you can go ask them?



DNS: Poisoning Authority (NS) Records



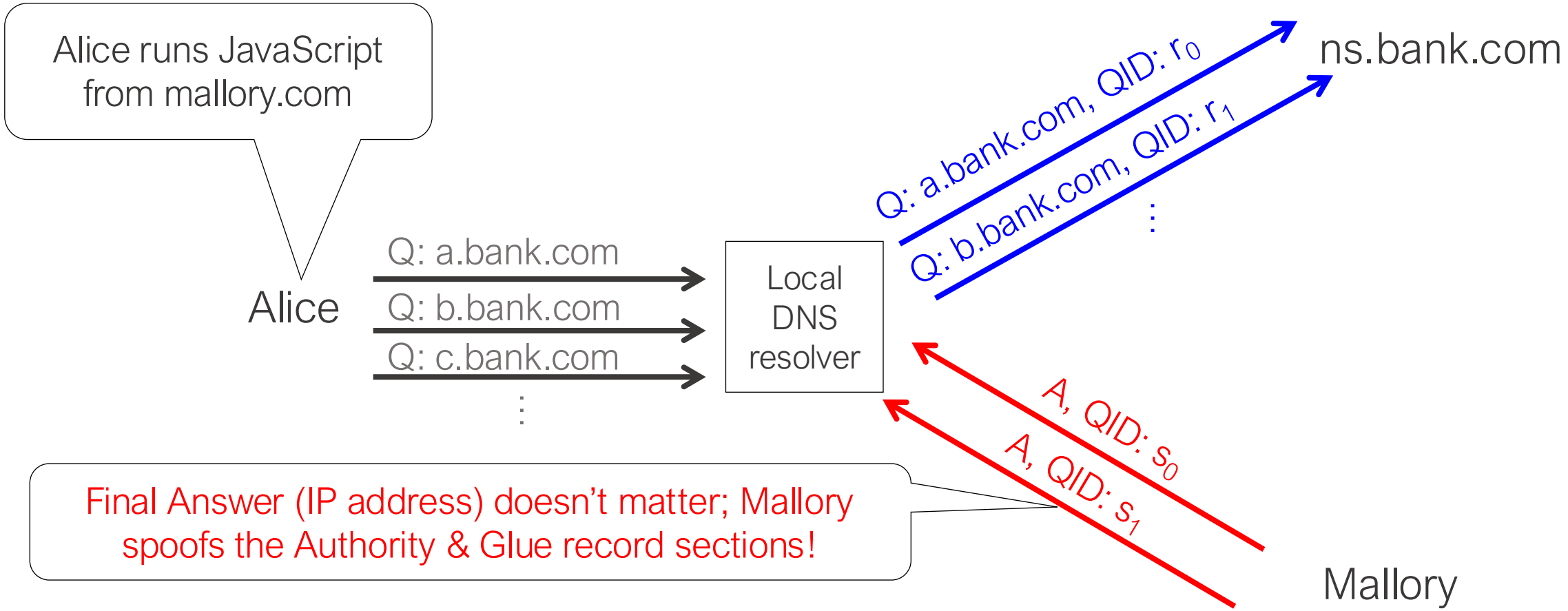
Kaminsky Attack (2008)



Challenge: Attacker needs to guess the correct Query ID.

Can an off-path attacker make this attack work?

Kaminsky Attack (2008)



Mallory wins if any $r_i = s_j$

See http://unixwiz.net/techtips/iguide-kaminsky-dns-vuln.html for details

Kaminsky Attack (2008)

- **Key Idea:** attacker forces DNS resolver to issue many queries by using many fake subdomains (e.g., z123123.bank.com)
  - Only needs to guess the query ID correctly for one of the queried subdomains (QID: 16 bits = only ~65,000 possible values)
  - Attaches a poisoned authority & glue record [NS info] in their reply
  - Once poisoning succeeds: all un-cached subdomain lookups will ask the attacker's server instead of the domain's real nameserver

- **Defense:** Randomize both the query ID and source port (16 -> 32 bits)
  - Billions of possible values: very low probability of winning the race even with many guesses at a time

General DNS Security: DNSSEC

- DNS responses signed
- Higher levels vouch for lower levels
  — e.g., root vouches for .edu, .edu vouches for .uchicago, …
- Root public key published

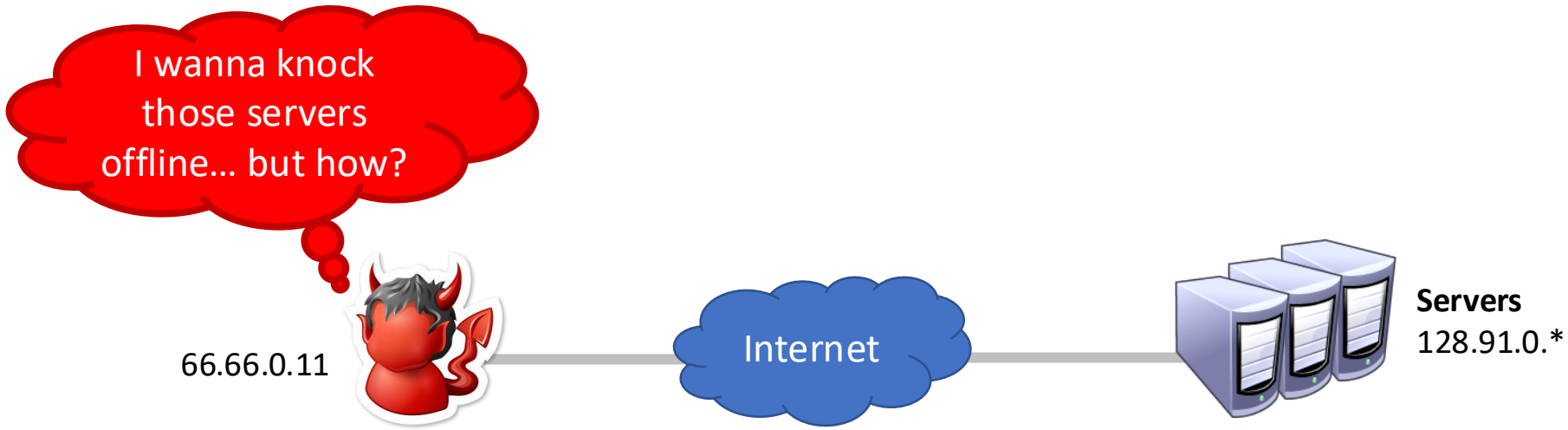
- Most people don’t use DNSSEC and never will: Use TLS instead

Outline

- TCP/UDP Security
- DNS Security
- Denial of Service (Availability Attacks)
- Network Scanning & Firewalls

Denial of Service (DoS) Attacks

- **Threat Model:** Active attacker who can freely send packets to target
- **Goal:** Prevent users from being able to access a target: specific computer, service, or piece of data (Disrupt Availability)



Attacker Motivations for DoS

- Showing off / entertainment / ego
- Competitive advantage
  - Maybe commercial, maybe just to win
- Vendetta / denial-of-money
- Extortion
- Impair defenses
- Political statements / manipulation
- Warfare

Botnets Beat Spartan Laser on *Halo 3*

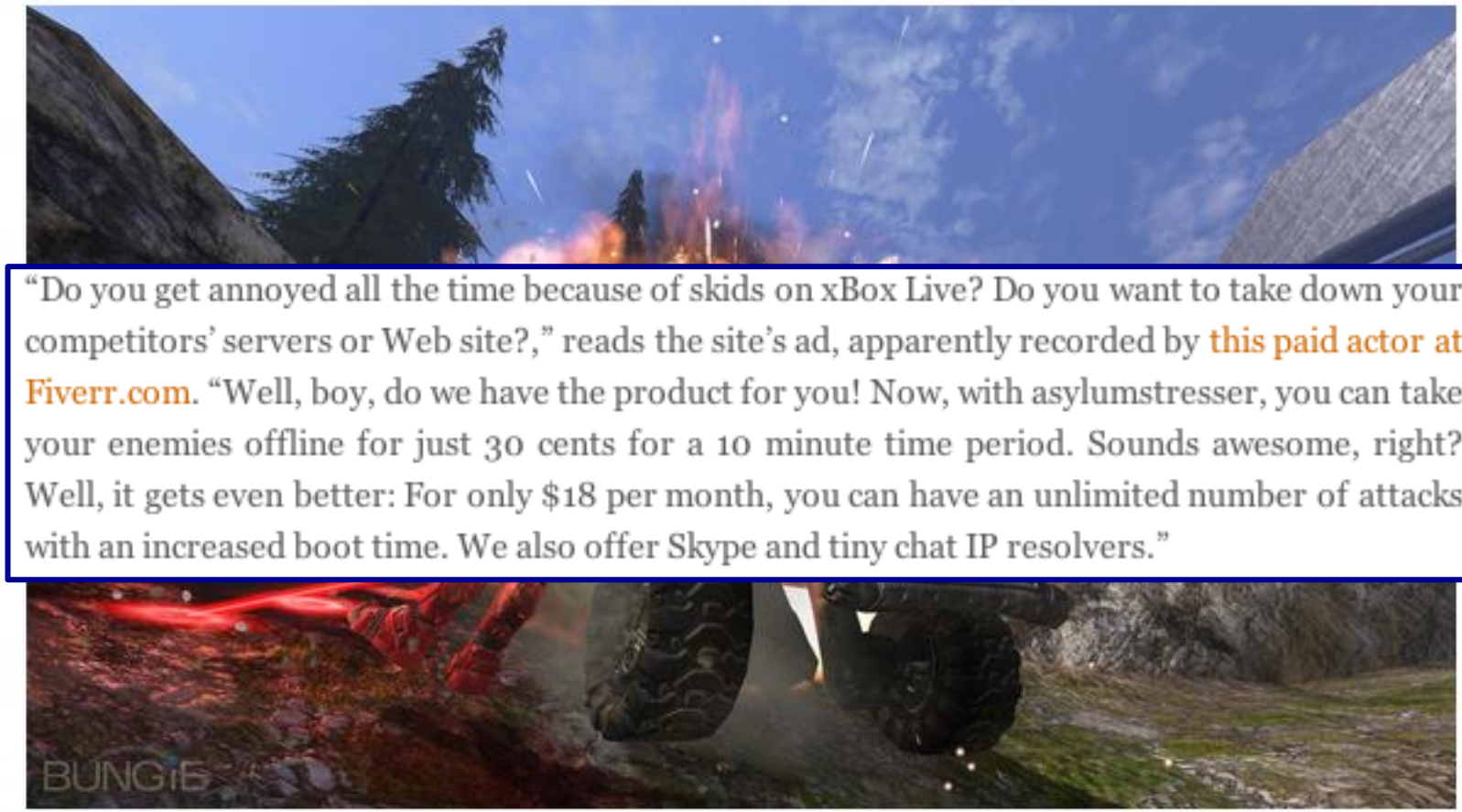
By Kevin Poulsen ✉ February 4, 2009 | 12:13 pm | Categories: Cybarmageddon!

What's the most powerful weapon you can wield when playing *Halo 3* online?

I know. You can control the entire map with a battle rifle and a couple of sticky grenades. But that teeny-bopper you just pwned has you beat with the tiny botnet he leased with his allowance money.

Botnets Beat Spartan Laser on *Halo 3*

By Kevin Poulsen ✉ February 4, 2009 | 12:13 pm | Categories: Cybarmageddon!



"Do you get annoyed all the time because of skids on xBox Live? Do you want to take down your competitors' servers or Web site?," reads the site's ad, apparently recorded by this paid actor at Fiverr.com. "Well, boy, do we have the product for you! Now, with asylumstresser, you can take your enemies offline for just 30 cents for a 10 minute time period. Sounds awesome, right? Well, it gets even better: For only $18 per month, you can have an unlimited number of attacks with an increased boot time. We also offer Skype and tiny chat IP resolvers."

What's the most powerful weapon you can wield when playing *Halo 3* online?

I know. You can control the entire map with a battle rifle and a couple of sticky grenades. But that teeny-bopper you just pwned has you beat with the tiny botnet he leased with his allowance money.

Extortion via DDoS on the rise

By Denise Pappalardo and Ellen Messmer, Network World, 05/16/05

Criminals are increasingly targeting corporations with distributed denial-of-service attacks designed not to disrupt business networks but to extort thousands of dollars from the companies.

Ivan Maksakov, Alexander Petrov and Denis Stepanov were accused of receiving $4 million from firms that they threatened with cyberattacks.

The trio concentrated on U.K. Internet gambling sites, according to the prosecution. One bookmaker, which refused to pay a demand for $10,000, was attacked and brought offline--which reportedly cost it more than $200,000 a day in lost business.


DDOS EM'
WHEN IN DOUBT KNOCK EM' OUT!
Features
Resolvers
Skype
Steam
Cloudflare
IP Tools
Geolocation
IP Logger
Host to IP
MAX BOOT TIME OF
3600
UDP, SSYN, RUDY, UDP-LAG, ARME, GET, POST

Denial of Service (DoS): Availability

Two main DoS Strategies:

1. Exploit program flaws (e.g., bug that crashes the target)
2. Exhaust the target's resources (CPU, memory, bandwidth, etc.)

Often very easy to perform… but difficult to mitigate ☹

DoS from program flaws = fairly straightforward

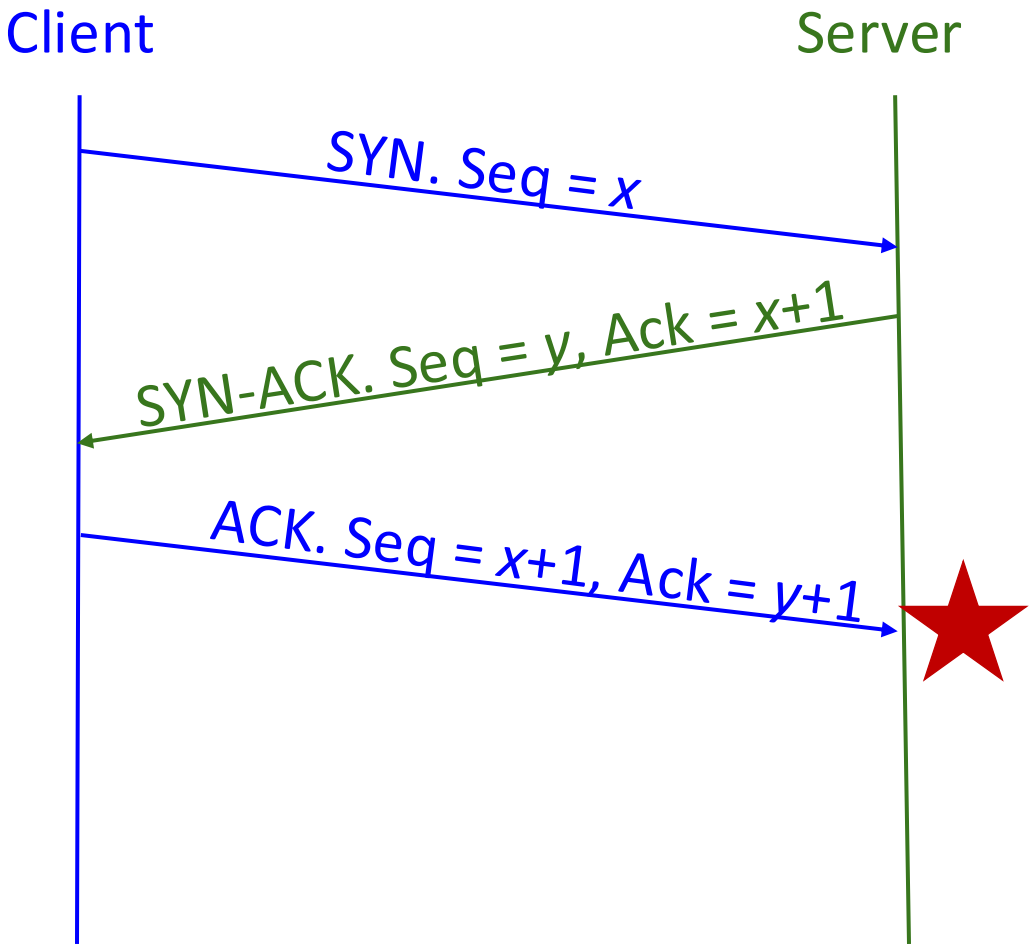
- Most attacks we'll discuss focus on resource exhaustion

DoS Attack Parameters

- Asymmetric Attack:
  - Attacker either generates a much larger cost at the target, or has much more resources (e.g., bandwidth) than the target

- What kind of packets does the attacker send to the victim?
  - Minimize effort and risk of detection for attacker
  - While also maximizing damage to the target

TCP SYN Flooding



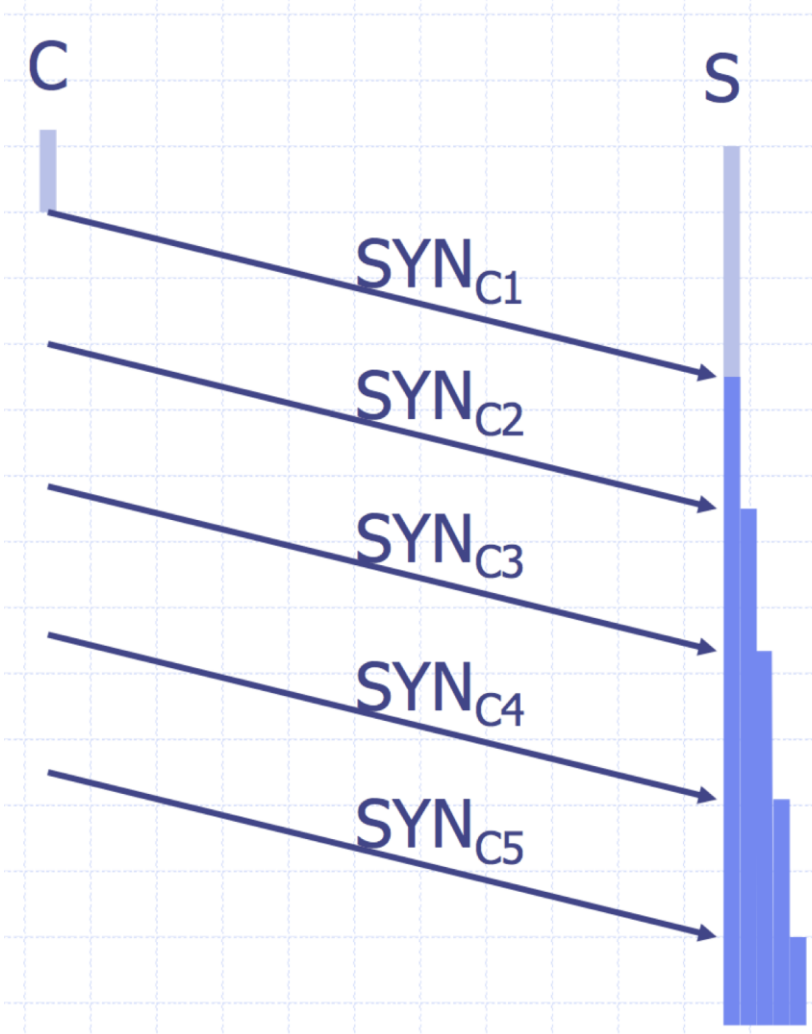
Server stores state during TCP handshake:

- Allocates memory to validate that client's ACK number is correct

Attack: Flood the target with SYN packets

- Exhausts available memory for target: no more connections
- Asymmetry: Easy to Spoof many SYN packets & attacker doesn't need state

TCP SYN Flooding



Single machine:

- SYN Packets with **random source IP addresses**
- Fills up backlog queue on server
- No further connections possible

SYN Flooding Defenses

- **Core Problem:** Server commits resources without confirming client's identity or requiring them to commit resources

- **Defense Approach #1:** Overprovision
  - Have lots of servers with lots of memory
  - Drawbacks: expensive + target server might not be able to acquire sufficient resources vs. motivated attacker

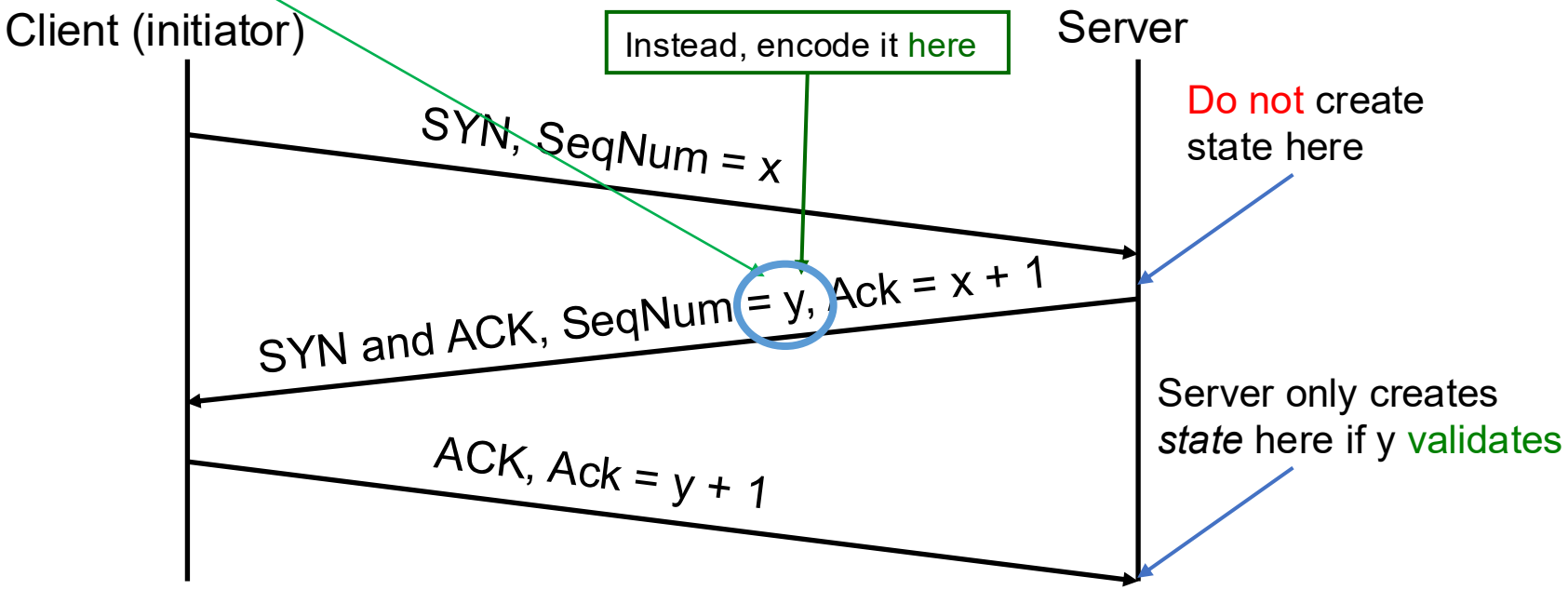
SYN Flooding Defenses

- **Approach #2:** Detect & Filter
  - Server can try to identify packets that are SYN Flooding & ignore them
  - Drawbacks: hard to identify them
    - Only have src IP address in packets
    - But the attacker can spoof these src IP addresses!

- **Approach #3:** Change the ACK validation so the server doesn't have to store state!
  - Practical Defense: SYN cookies

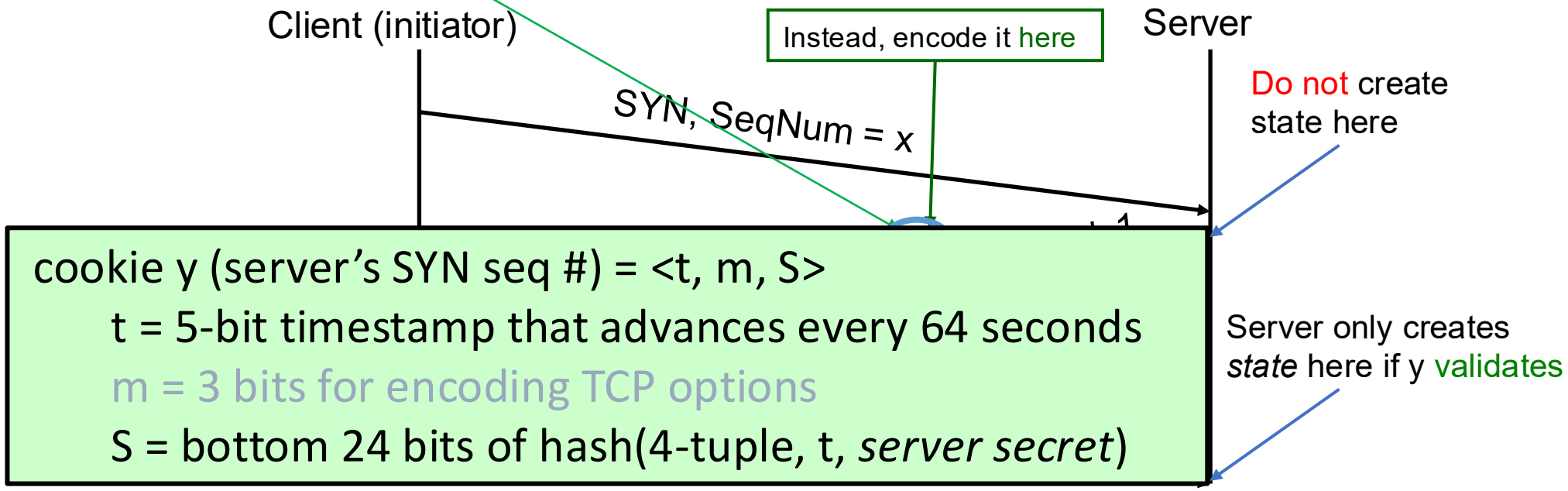
Practical Defense: *SYN Cookies*

- Server: when SYN arrives, encode critical state entirely within SYN-ACK's sequence # y !
  - y = *encoding* of necessary state, using server secret
- When ACK of SYN-ACK arrives, server only creates state ***if*** value of **y** from it agrees w/ secret



Practical Defense: *SYN Cookies*

- Server: when SYN arrives, encode critical state entirely within SYN-ACK's sequence # y !
  - y = *encoding* of necessary state, using server secret
- When ACK of SYN-ACK arrives, server only creates state ***if*** value of **y** from it agrees w/ secret



Reflection & Amplification Attacks

SYN Flooding: exhaust *memory* of server

Network DoS: exhaust *network bandwidth* of server / client

- Amplification Attacks: Exploit asymmetry in protocols, where a network request packet generates much greater response traffic
- Reflection Attacks: Use third-party machines (not controlled by attacker) to flood the target
- Amplification + Reflection often used together

Ping (ICMP) Protocol

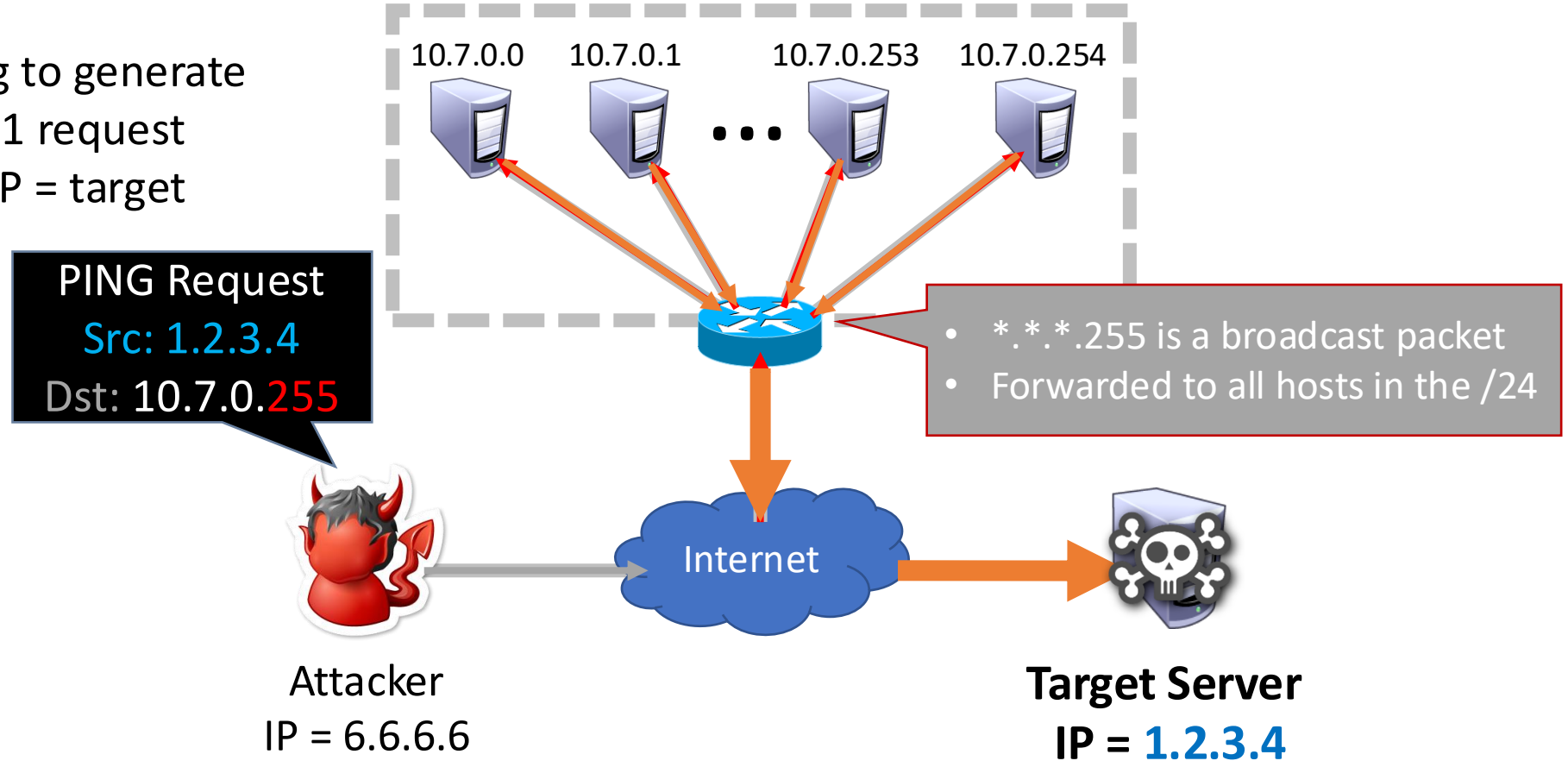
- Essential, low-level network utility (status/liveliness check)
- Sends a “ping” ICMP message to a host on the internet

```
$ ping 66.66.0.255
PING 66.66.0.255 (66.66.0.255) 56(84) bytes of data.
64 bytes from 66.66.0.255: icmp_seq=1 ttl=58 time=41.2 ms
```

- Destination host is supposed to respond with a “pong” indicating that it can receive packets
- By default, ping messages are 56 bytes long (+ some header bytes)

The Smurf Attack: ICMP Flooding

- Abuses broadcasting to generate many responses for 1 request
- Attacker spoofs src IP = target



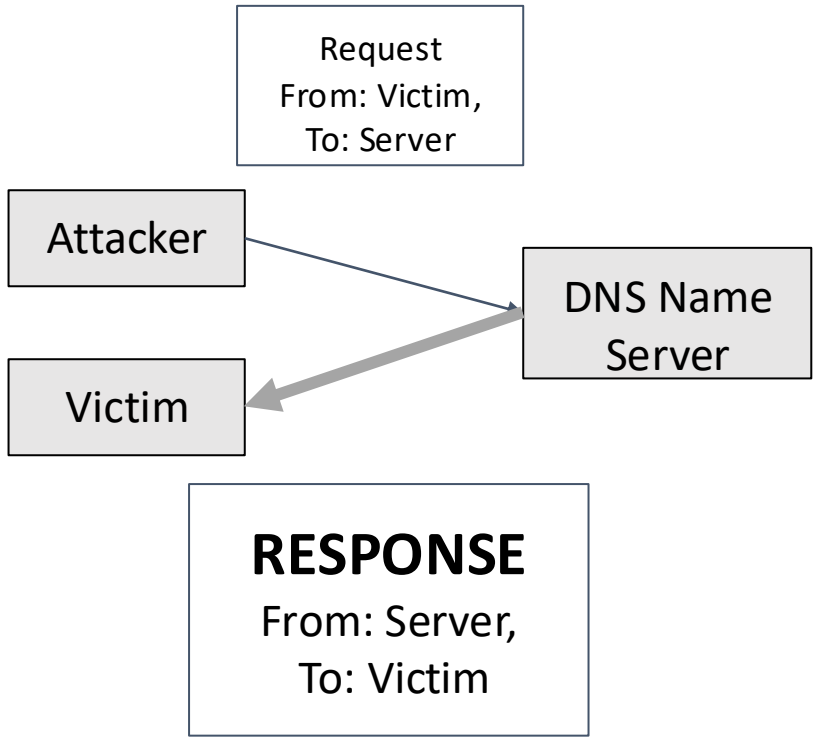
Why Does Smurfing Work?

1. Internet Control Message Protocol (ICMP) does not include authentication
   - Receivers accept messages without verifying the source
   - Enables attackers to spoof the src IP addr of messages
2. Attacker benefits from an amplification factor

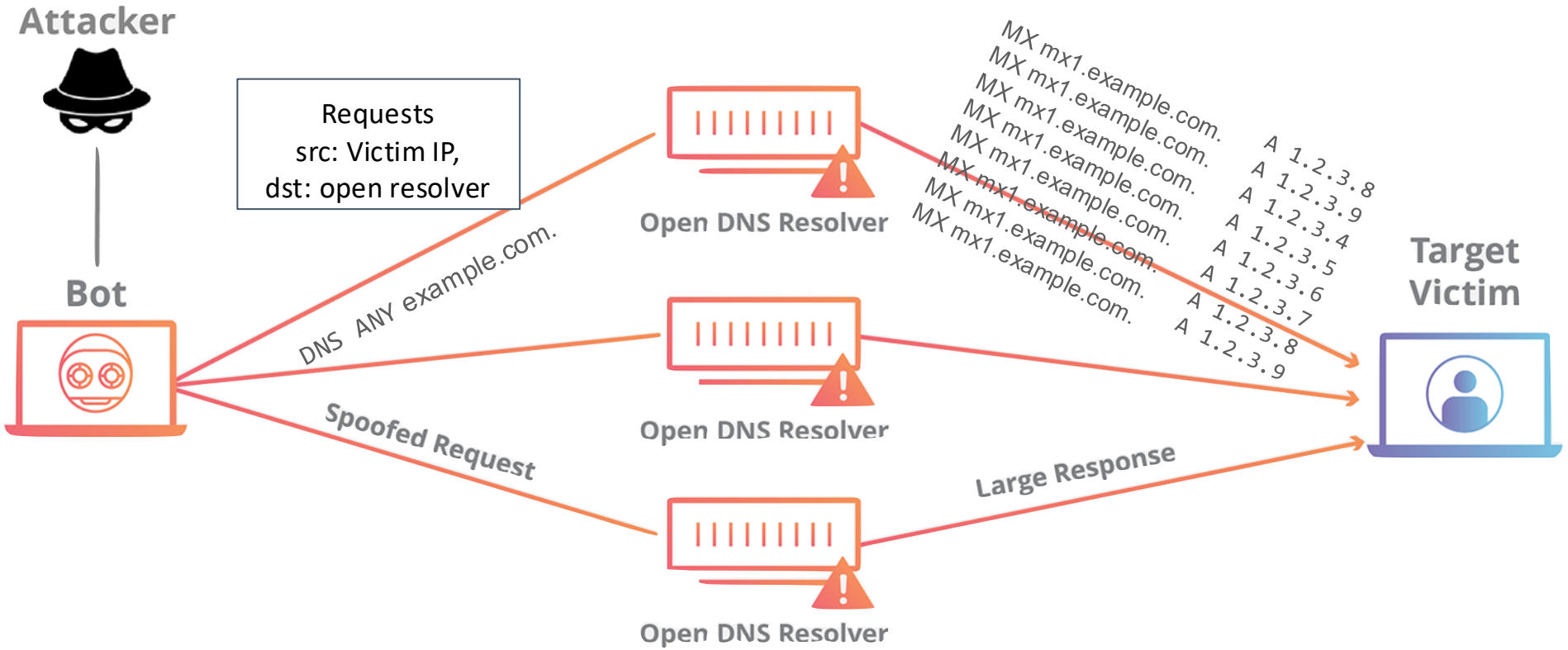
$$amp\ factor = \frac{total\ response\ size}{request\ size}$$

UDP Amplification & Reflection

- Some protocols / commands generate large responses for a single (small) request
  - DNS: Query type "*ANY*" returns all records server has about a domain
  - NTP: MONLIST returns list of last 600 clients who asked for the time recently
- Attack: Spoof requests from target machine's src IP address to other services
  - Typically use UDP-based protocols: Why?



DNS Reflection (+ Amplification) Attack



Spoof DNS requests from victim src IP addr to many **open** DNS resolvers

- Open resolvers accept requests from any client, e.g. 8.8.8.8, 8.8.4.4, 1.1.1.1, 1.0.0.1
- February 2014 – 25 million open DNS resolvers on the internet

Image: Cloudflare

Preventing Spoofing: Ingress & Egress Filtering

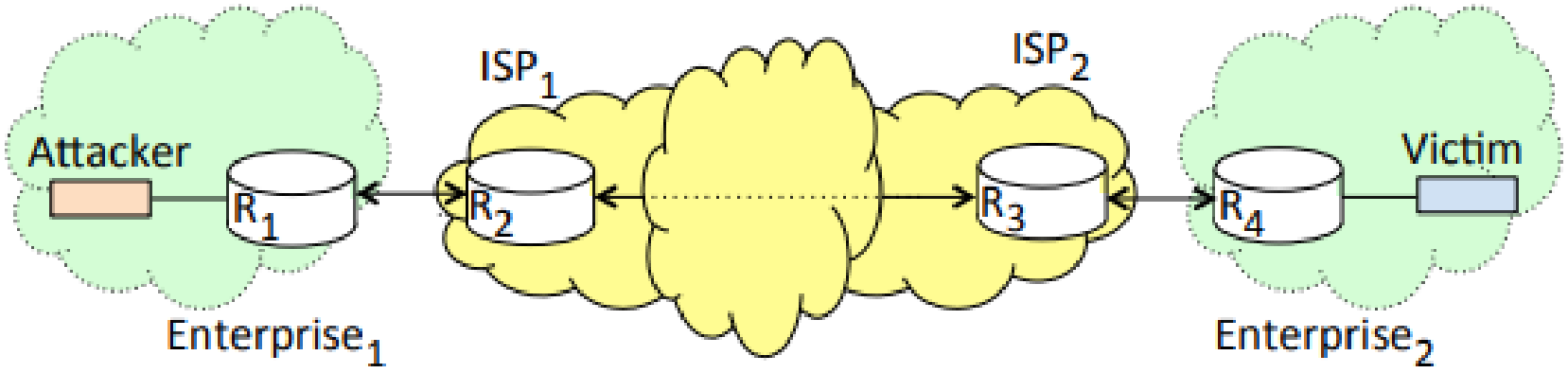
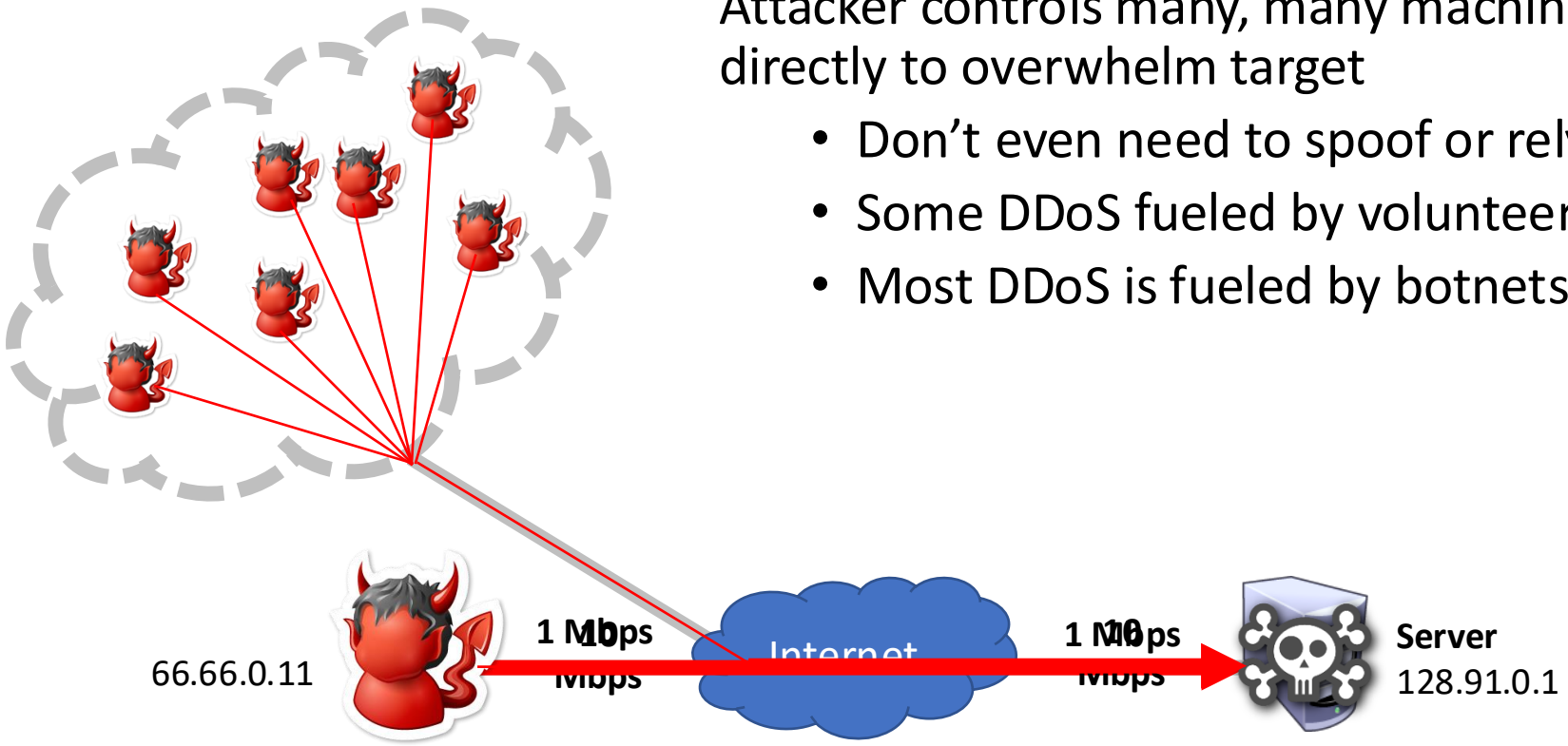


Figure 11.6: Ingress and egress filtering. An attacker may use a spoofed source IP address in traffic sent to a victim. ISP_1 does ingress filtering at R_2 for traffic entering from $Enterprise_1$. $Enterprise_1$ does egress filtering at R_1 for traffic leaving to ISP_1. For firewall rules to implement ingress and egress filtering, see Table 10.1 in Section 10.1.

- Networks know which IP addresses belong to them and
- ISPs/ASNs know which IP addresses they've given to sub-networks

Distributed Denial of Service (DDoS) Attacks



Attacker controls many, many machines and uses them directly to overwhelm target

- Don’t even need to spoof or rely on UDP protocols
- Some DDoS fueled by volunteers (e.g. Anonymous)
- Most DDoS is fueled by botnets (e.g., Mirari)

THE WALL STREET JOURNAL.

October 21, 2016

Cyberattack Knocks Out Access to Websites

Popular sites such as Twitter, Netflix and PayPal were unreachable for part of the day

twitter

amazon web services™

PayPal

NETFLIX

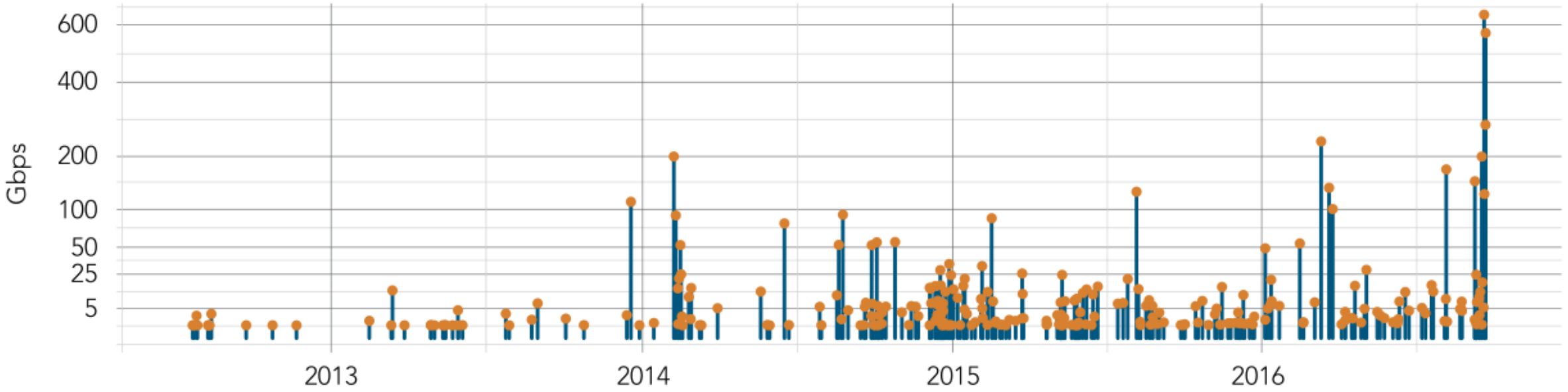
SOUNDCLOUD

Spotify

GitHub

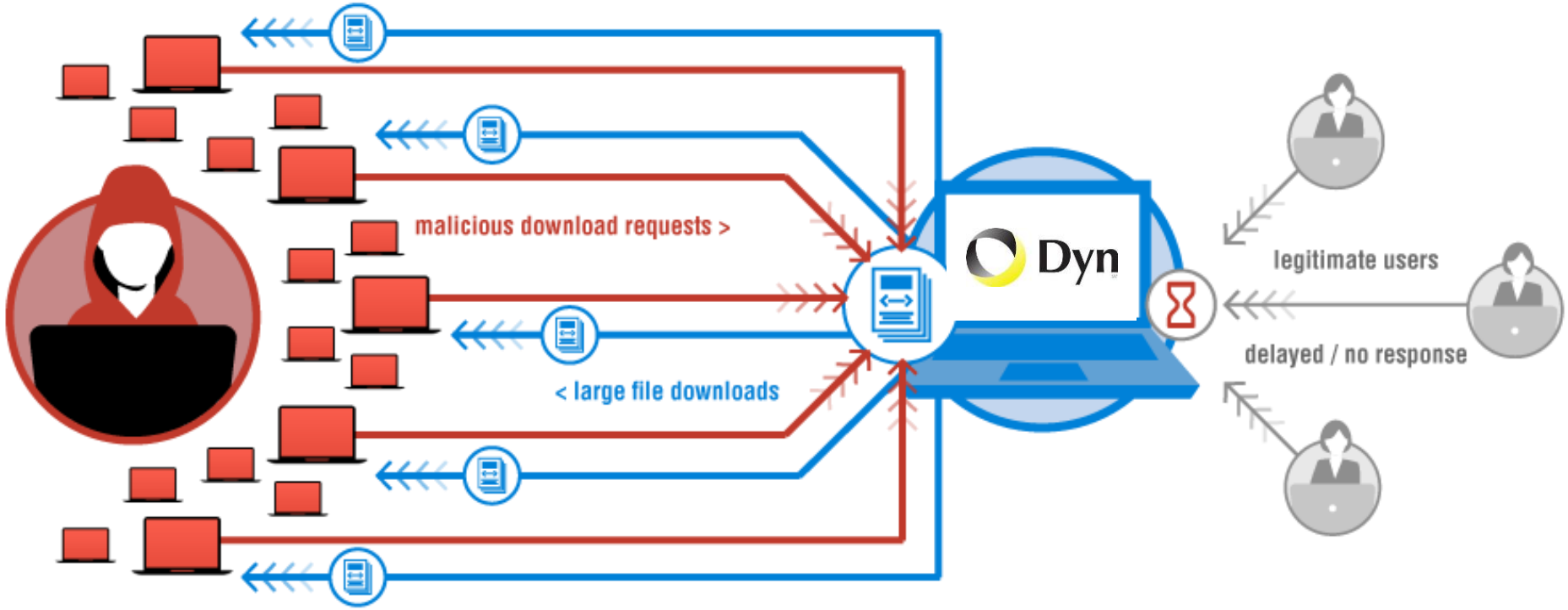
reddit

New York Times



"The magnitude of the attacks seen during the final week were significantly larger than the majority of attacks Akamai sees on a regular basis. […] In fact, while the attack on September 20 was the largest attack ever mitigated by Akamai, the attack on September 22 would have qualified for the record at any other time, peaking at 555 Gbps."

Source: 2017 Akamai State of the Internet



> “We are still working on analyzing the data but the estimate at the time of this report is up to 100,000 malicious endpoints. […] There have been some reports of a magnitude in the 1.2 Tbps range; at this time we are unable to verify that claim.”

Image: Verisign

A Botnet of IoT Devices (Mirai)

~~≈ 200K Hosts~~

200K IoT devices

Not Amplification.

Flood with SYN, ACK, UDP, and GRE packets

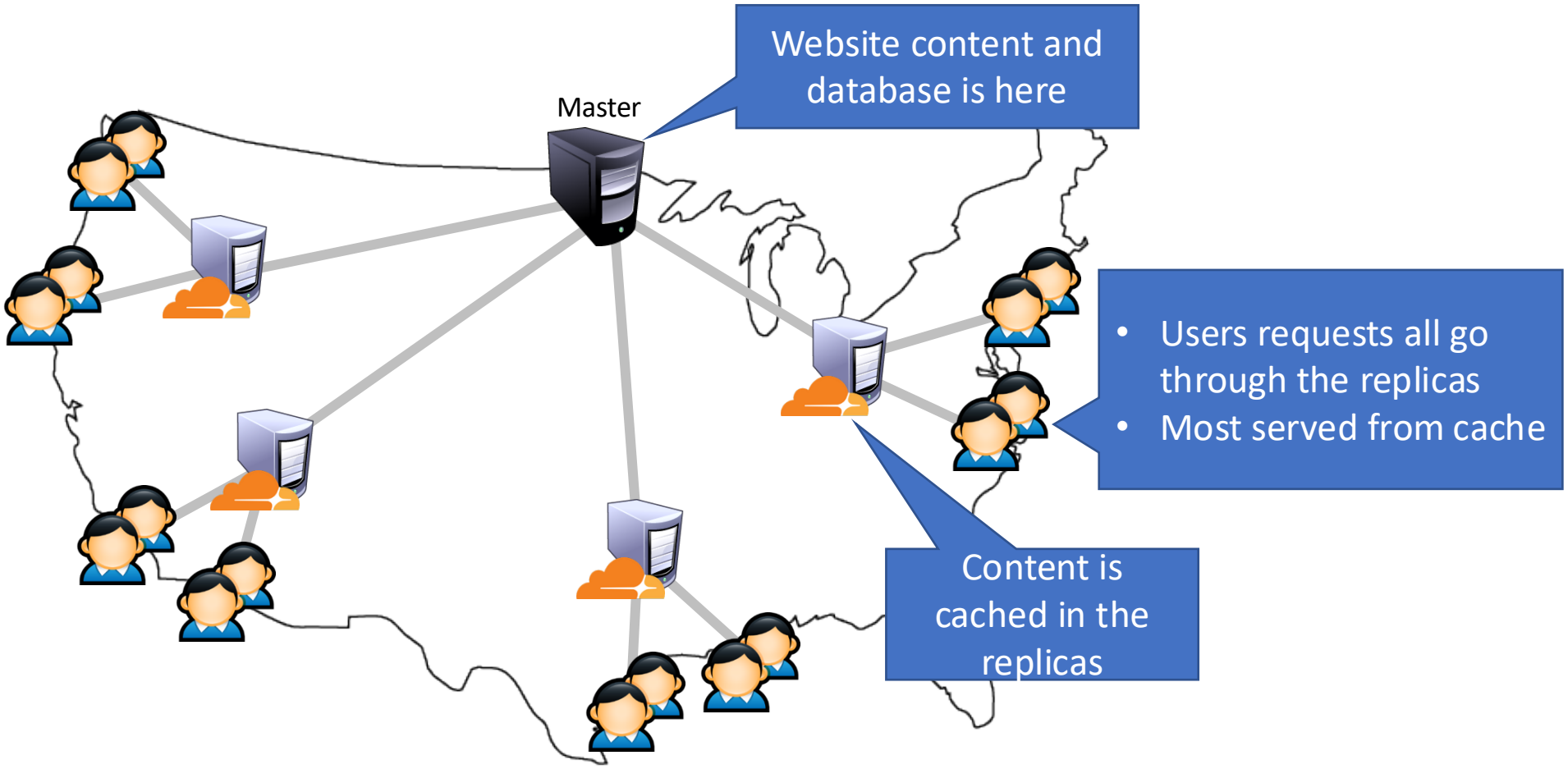
Infamous DDoS Attacks

When	Against Who	Size	How
March 2013	Spamhaus	120 Gbps	Botnet + DNS reflection
February 2014	Cloudflare	400 Gbps	Botnet + NTP reflection
September 2016	Krebs	620 Gbps	Mirai
October 2016	Dyn (major DNS provider)	1.2 Tbps	Mirai
March 2018	Github	1.35 Tbps	Botnet + memcached reflection

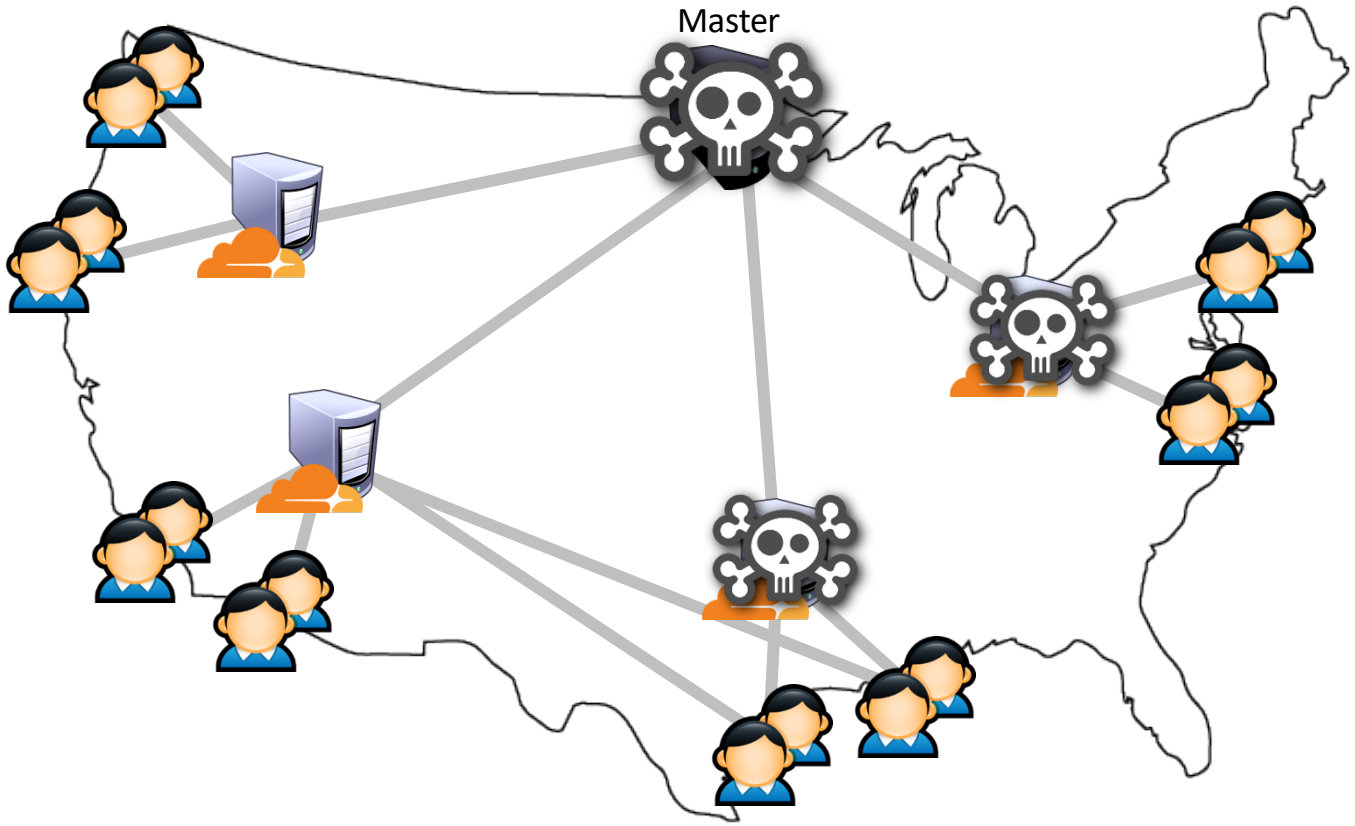
Content Delivery Networks (CDNs)

- CDNs help companies scale-up their websites
  - Cache customer content on many replica servers
  - Users access the website via the replicas
- Examples: Akamai, Cloudflare, Rackspace, Amazon Cloudfront, etc.
- Side-benefit: DDoS protection
  - CDNs have many servers, and a huge amount of bandwidth
  - Difficult to knock all the replicas offline
  - Difficult to saturate all available bandwidth
  - No direct access to the master server
- Cloudflare: 15 Tbps of bandwidth over 149 data centers

Content Delivery Networks (CDNs)

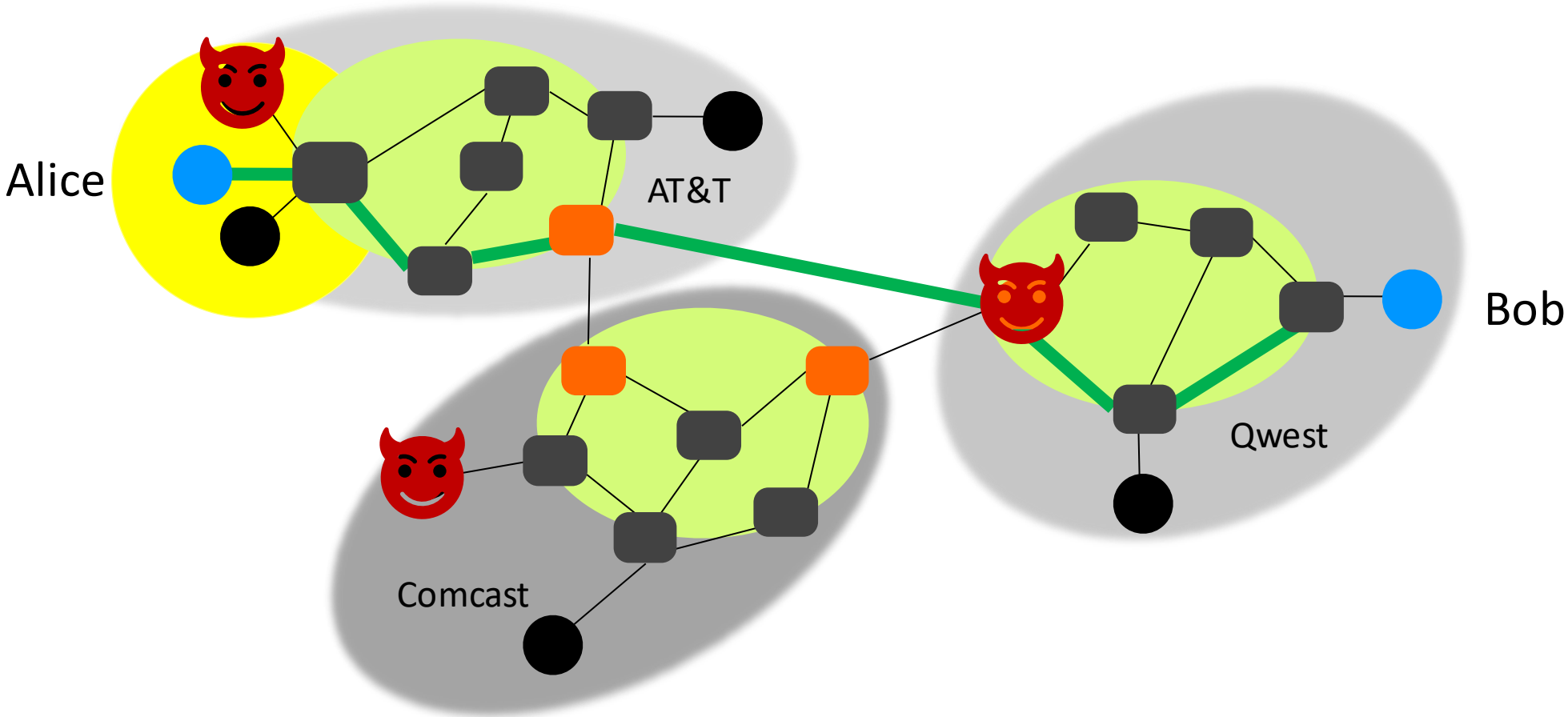


DDoS Defense via CDNs



- What if you DDoS the master replica?
  - Cached copies in the CDN still available
  - Easy to do ingress filtering at the master
- What if you DDoS the replicas?
  - Difficult to kill them all
  - Dynamic DNS can redirect users to live replicas

Networking Attacks Recap



Outline

- TCP/UDP Security
- DNS Security
- Denial of Service (Availability Attacks)
- Network Scanning & Firewalls

Network Scanning

- Goals: Identify information about hosts on a network
  - Which IP addresses have assigned machines?
  - What services do those machines offer (SSH, HTTP, DNS, etc.)?
  - Is there a machine with known vulnerabilities at a particular IP address?

- Useful technique for both attackers & defenders

Network Scanning Tools: Traceroute

- ping (ICMP): check if host is responsive
- traceroute — hops between me and host
  - Sends repeated ICMP reqs w/ increasing TTL

```
thor Wed Oct 24(12:51am)[~]:-> traceroute www.slack.com
traceroute to www.slack.com (52.85.115.213), 64 hops max, 52 byte packets
 1  v11router (128.135.11.1)  1.265 ms  0.788 ms  0.778 ms
 2  a06-021-100-to-d19-07-200.p2p.uchicago.net (10.5.1.186)  1.292 ms  0.749 ms  0.833 ms
 3  d19-07-200-to-h01-391-300.p2p.uchicago.net (10.5.1.46)  2.124 ms  2.435 ms  2.072 ms
 4  192.170.192.34 (192.170.192.34)  0.755 ms
    192.170.192.32 (192.170.192.32)  0.810 ms  0.701 ms
 5  192.170.192.36 (192.170.192.36)  0.887 ms  0.918 ms  0.877 ms
 6  r-equinix-isp-ae2-2213.wiscnet.net (216.56.50.45)  1.625 ms  1.803 ms  1.866 ms
 7  * * *
 8  * * *
 9  * * *
10  * * *
11  178.236.3.103 (178.236.3.103)  4.516 ms  4.326 ms  4.320 ms
12  * * *
13  * * *
14  * * *
15  server-52-85-115-213.ind6.r.cloudfront.net (52.85.115.213)  4.554 ms  4.398 ms  4.757 ms
thor Wed Oct 24(12:52am)[~]:->
```

Port Scanning

- What services are running on a server? Nmap

```
linux3 Wed Oct 24(12:54am)[~]:-> nmap www.cs.uchicago.edu

Starting Nmap 7.01 ( https://nmap.org ) at 2018-10-24 00:55 CDT
Nmap scan report for www.cs.uchicago.edu (34.203.108.171)
Host is up (0.019s latency).
Other addresses for www.cs.uchicago.edu (not scanned): 54.164.17.80 54.85.61.218
rDNS record for 34.203.108.171: ec2-34-203-108-171.compute-1.amazonaws.com
Not shown: 998 filtered ports
PORT    STATE SERVICE
80/tcp  open  http
443/tcp open  https

Nmap done: 1 IP address (1 host up) scanned in 4.99 seconds
linux3 Wed Oct 24(12:55am)[~]:->
```

- 5 se

SYN Scanning

Send only a SYN : only needs application to run TCP

Responses:

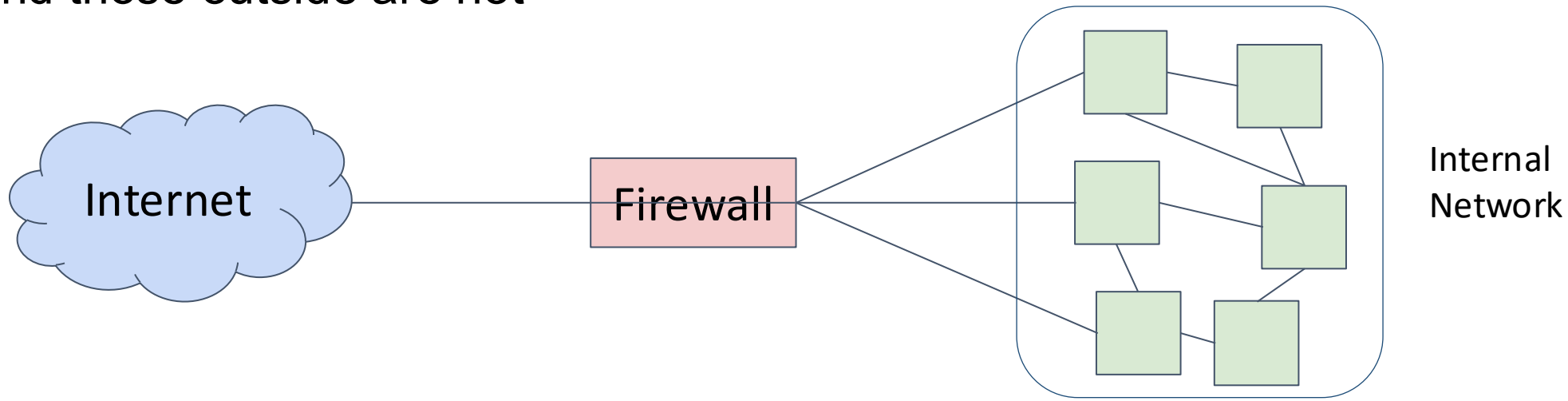
- SYN-ACK — port open
- RST — port closed
- Nothing — filtered (e.g., firewall)

Firewalls

- How do you protect a set of systems against external attack?
  - Example: A company network with many servers and employee computers
- Observation: More network services = more risk
  - Each available service creates more opportunities for vulnerabilities
  - Turning off all network services is often infeasible (printing, SSH, etc.)
- Observation: More networked machines = more risk
  - What if you have to secure hundreds of systems?
  - What if the systems have different hardware, operating systems, and users?
  - What if there are some systems in the network that you aren't aware of?
- Instead of securing individual machines, we want to secure the entire network!

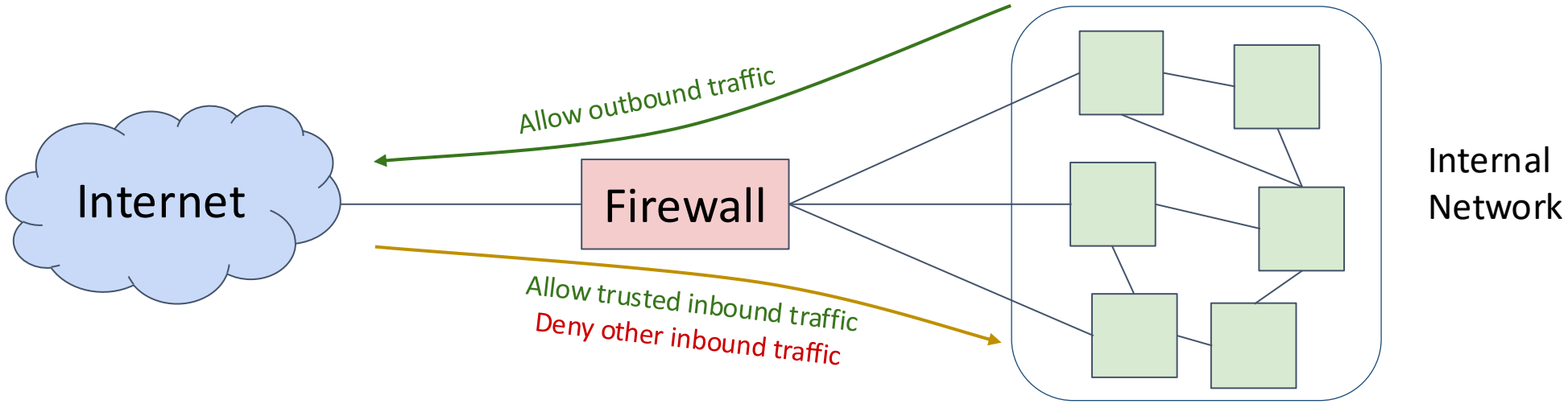
Firewalls and Security Policies

- Idea: Create single point of access in & out of network (chokepoint), with a monitor
  - “Ensure complete mediation”
  - Any traffic that could affect vulnerable systems must pass through the firewall
- Network access is controlled by a **policy** (based on threat model)
  - Defines what traffic is allowed to exit the network (**outbound policy**)
  - Defines what traffic is allowed to enter the network (**inbound policy**)
  - Traditional threat model: assume machines “inside” the network are trusted, and those outside are not



Firewalls and Security Policies

- What's the policy of a standard home network?
  - Outbound policy: Allow outbound traffic
    - Users inside the network can connect to any service
  - Inbound policy: Only some traffic is able to enter the network
    - Allow inbound traffic in response an outbound connection
    - Allow inbound traffic to certain, trusted services (e.g. SSH)
    - Deny all other inbound traffic



Course In-Context So Far

- Systems Security
  - OS-Level Access Control & Privilege Separation
  - Software Security

- Networking Security
  - Three types of attackers & attacks at each network layer
  - Availability: Attacks & Defenses

- Crypto
  - Math & Theory-based tools that allow us to achieve confidentiality, integrity, and authentication