Network Security (Cont.)

CMSC 23200, Spring 2025, Lecture 8

Grant Ho

University of Chicago, 04/17/2025
(Slides adapted from Blasé Ur, Peyrin Kao, and Zakir Durumeric)

Logistics

- **Final Exam:** Wed, May 28 @ 10am for BOTH SECTIONS

- Assignment 3 (Crypto) due tonight by 11:59pm

- Discussion Section resumes next week (04/23)

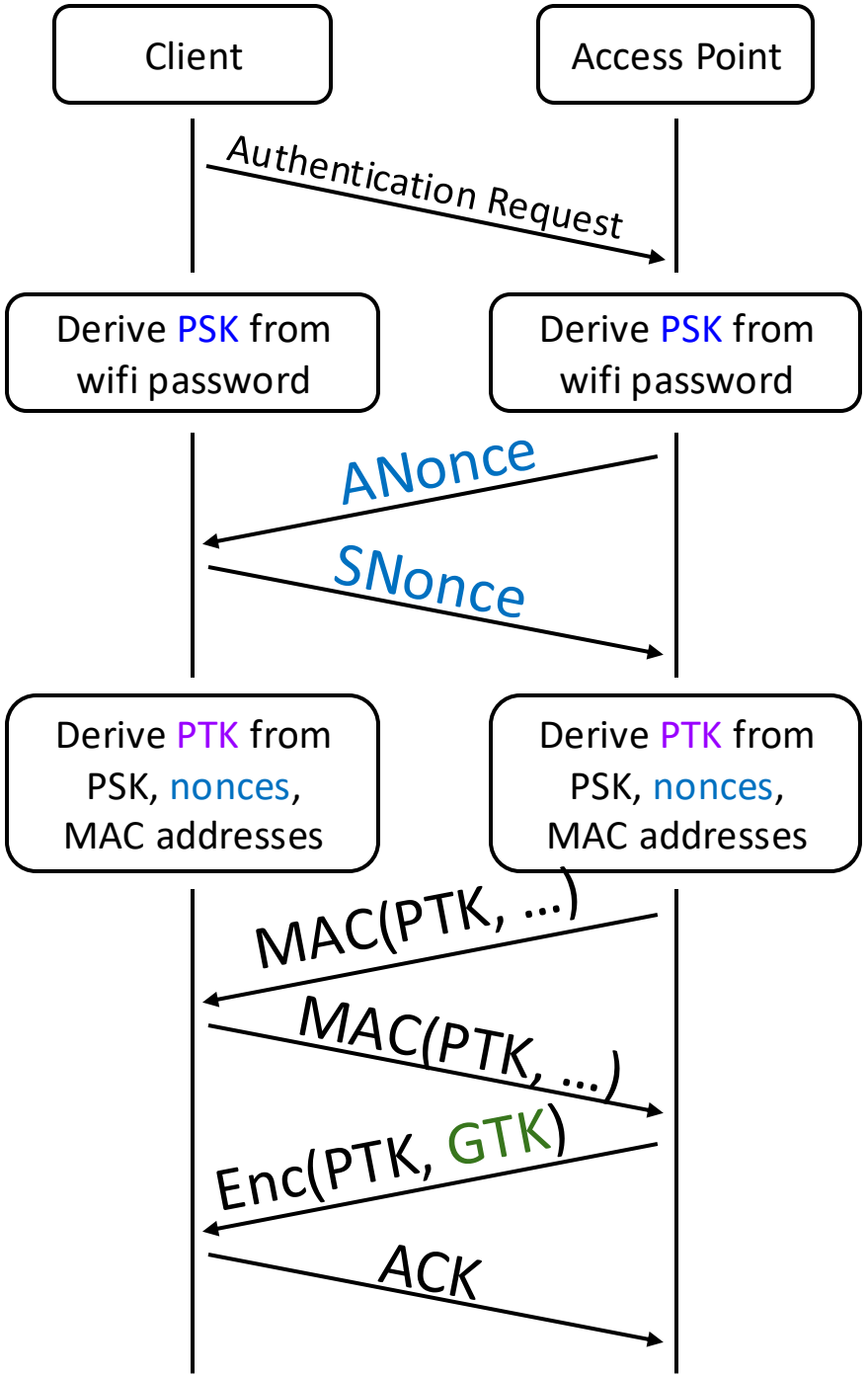
- No Assignment due next week.
  Assignment 4 released next Friday (4/25)

Outline

- Wrap-Up: Layer 2 (Wi-Fi)
- Layer 3 (BGP) Security
- TCP & UDP Attacks
- DNS Security

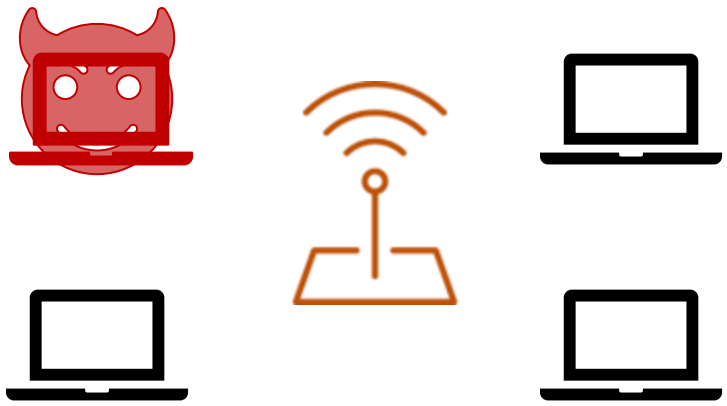
Recall: Wi-Fi and WPA2

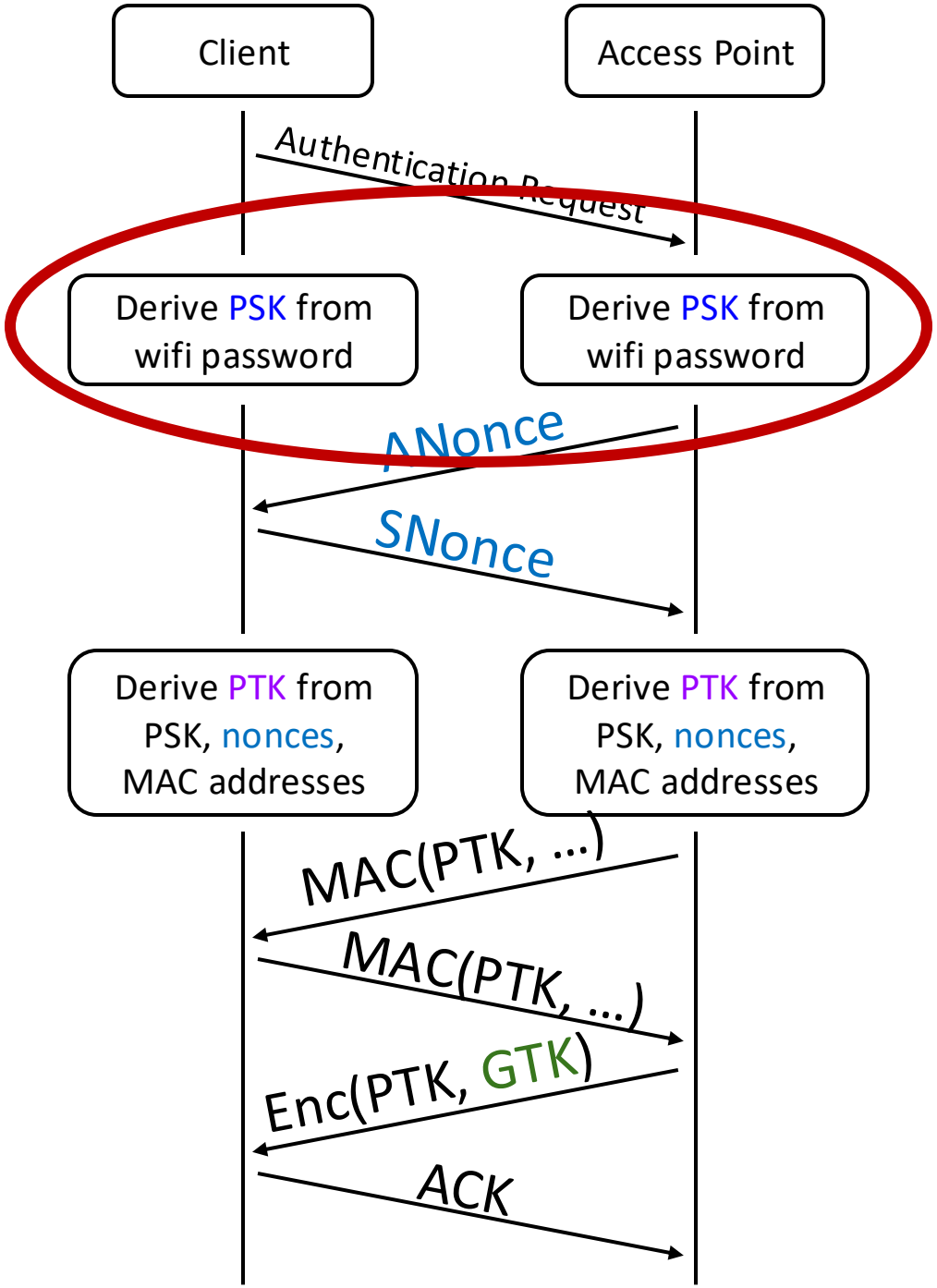
- **Wi-Fi:** Layer 2 Broadcast protocol for transmitting data between local machines
- **WPA2:** Protocol for securing data sent over Wi-Fi by (1) establishing a shared symmetric key & (2) performing symmetric crypto on Wi-Fi packet payloads



WPA Weakness

- **Problem:** Everyone uses the same shared Wi-Fi password
- **Therefore:** an attacker can generate the symmetric keys (PTK) for everyone they see joining the network

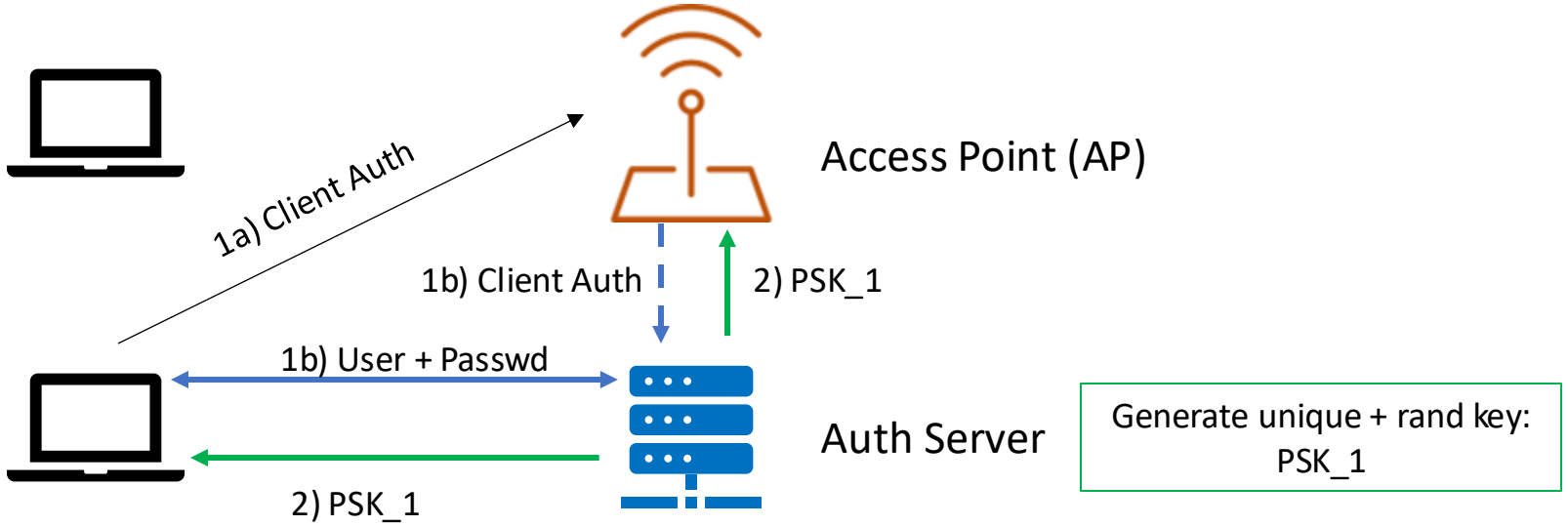




WPA Enterprise

Solution: Generate unique Wi-Fi session keys based on user accounts

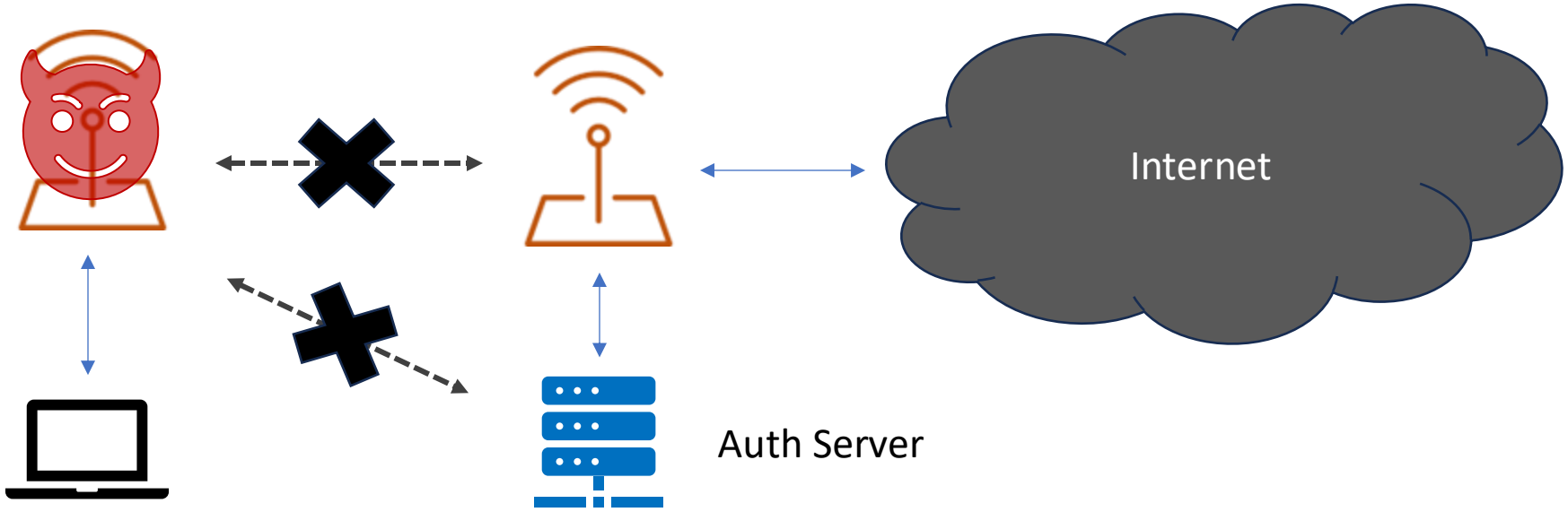
- Add additional authentication server (w/ a certificate verifying identity)
- AP configured to have a secure channel w/ Auth Server during setup
- Users now authenticate to the auth server when connecting to Wi-Fi
- If auth succeeds, auth server sends random key (PSK) to user & AP



WPA-Enterprise Attacks

WPA Enterprise defends against:

- **Rogue AP attack [if unauthorized user]**: The APs must authenticate themselves, which the attacker can't do (so attacker & user have no network access!)



WPA-Enterprise Attacks

WPA Enterprise defends against:

- **Rogue AP attack [if unauthorized user]**: The APs must authenticate themselves to the auth server, which the attacker can't do
- **Other Wi-Fi users:** Every user has unique PSK & derives unique session keys (PTK's)
- **Brute-force attack**: The generated PSK is too long & random to brute-force

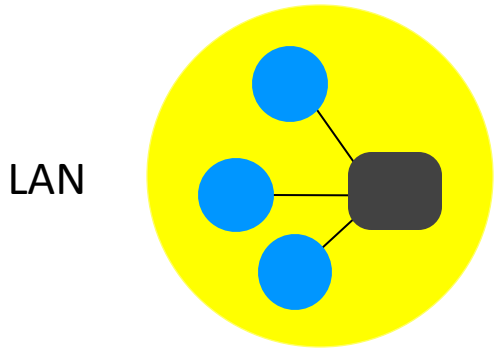
However, still vulnerable to (1) ARP or DHCP spoofing, and
(2) Rogue AP's that have network access

- If a user connects to malicious AP with network access, then the malicious AP can just relay & MITM the user's network connections.

Outline

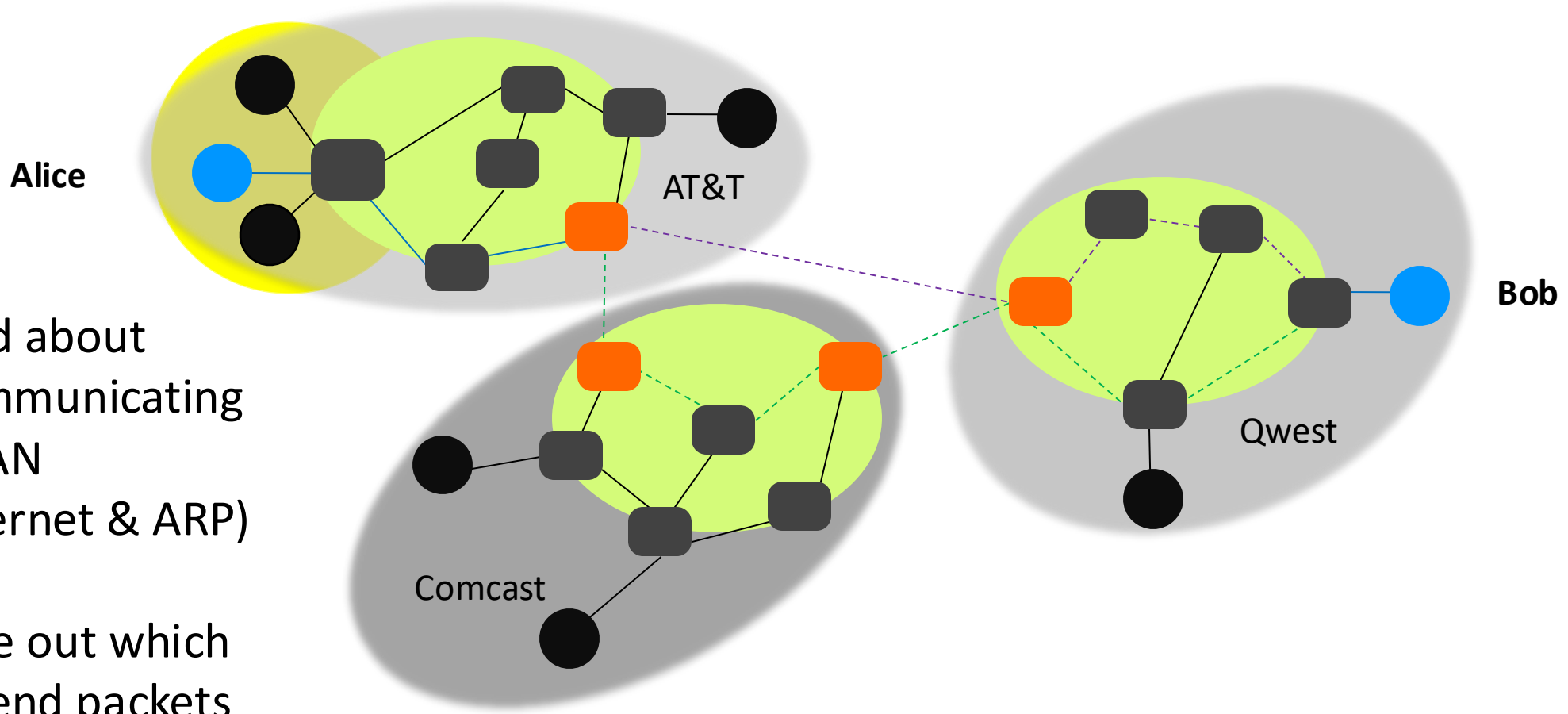
- Wrap-Up: Layer 2 (Wi-Fi)
- Layer 3 (BGP) Security
- TCP & UDP Attacks
- DNS Security

BGP: Routing Across the Internet



Previously: talked about
protocols for communicating
within a single LAN
(e.g., Wi-Fi / Ethernet & ARP)

BGP: Routing Across the Internet

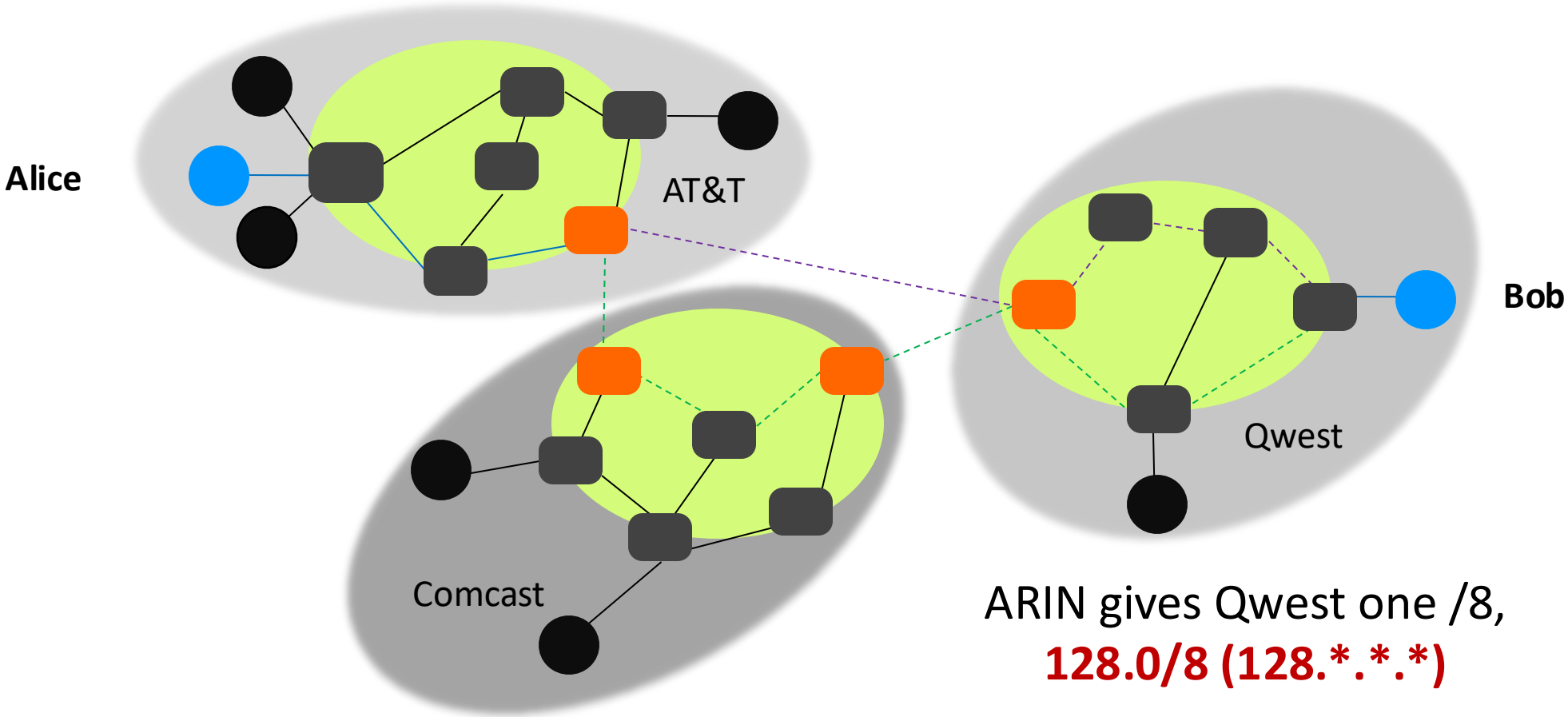


Previously: talked about protocols for communicating within a single LAN (e.g., Wi-Fi / Ethernet & ARP)

How do we figure out which route (path) to send packets across the Internet?

Internet = Collection of Autonomous Systems (AS)

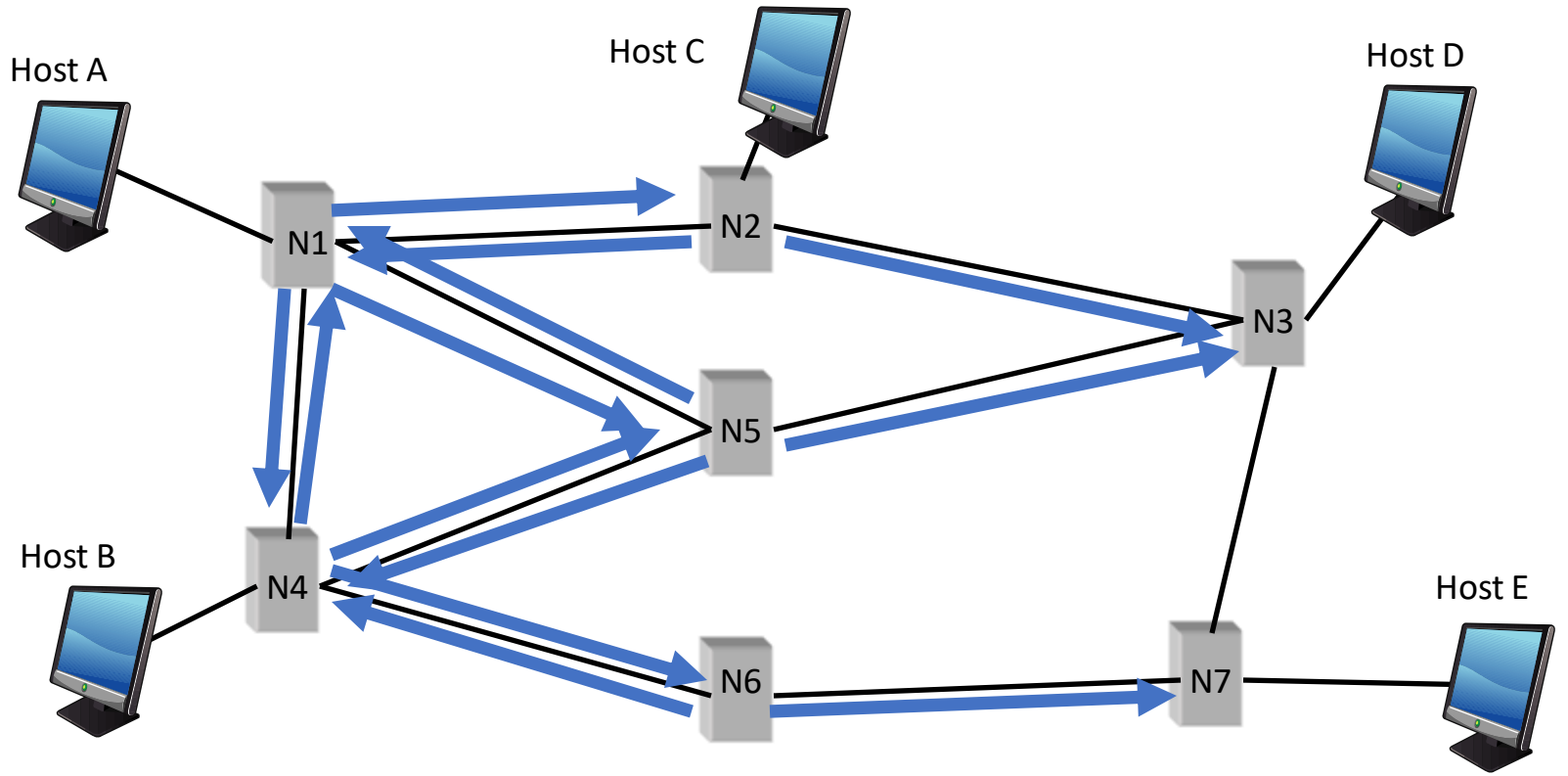
- AS: Collection of IP prefixes controlled by a single administrative entity
- 100,000+ Autonomous Systems (March 2021)



ARIN gives Qwest one /8,
128.0/8 (128.*.*.*)

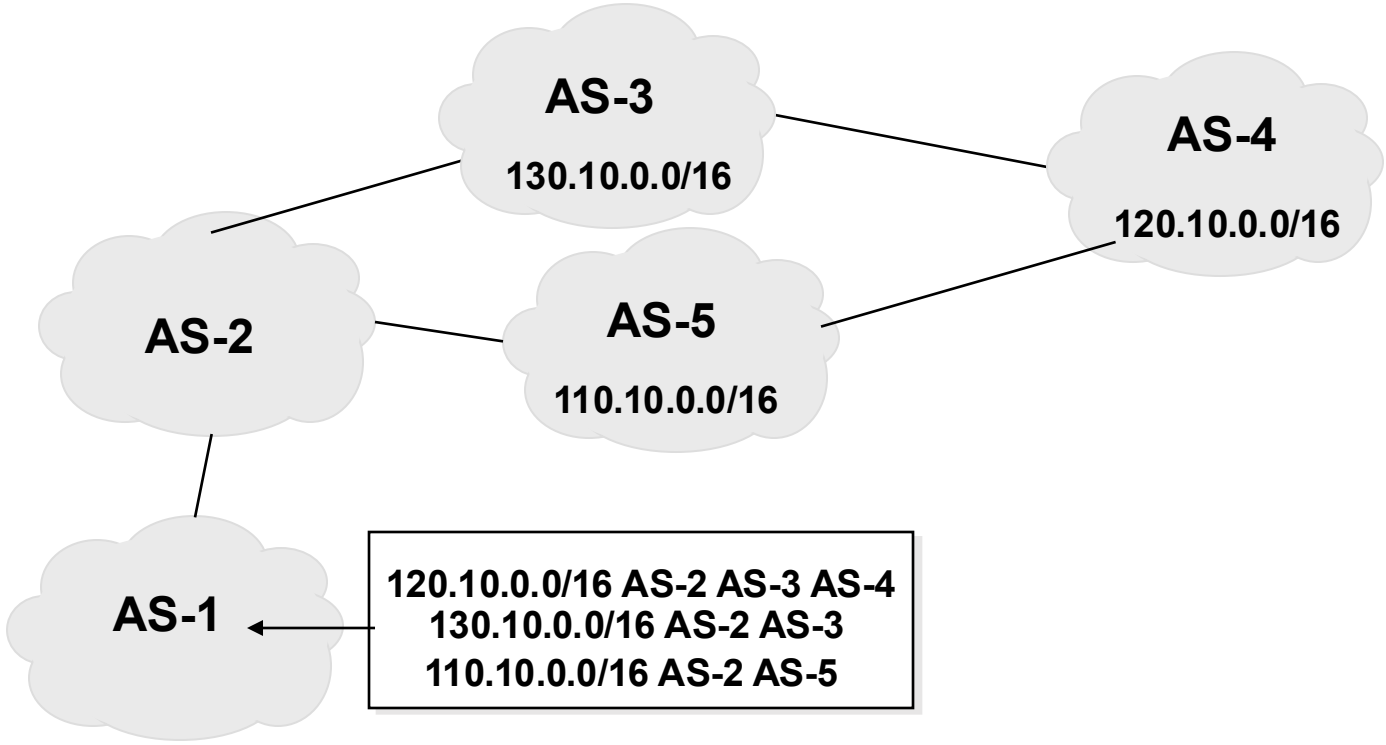
BGP: Border Gateway Protocol

- BGP: Finding viable paths (routes) across the Internet (between multiple networks)
  - Each router maintains a global routing table (paths they know)
  - Routers announce their routing tables to their neighbors if an update occurs



BGP: Discovering Viable Routes

- BGP provides potential routing paths: sequence of AS's a route traverses
- Actual route (for dest IP addr) determined by AS border routers.
  e.g., *possible* choice: route with fewest # of AS's



Attacks on BGP: Prefix Hijacking

- Attacker (malicious BGP router) can advertise a more desirable route even if the route isn't real
  - Just lie during BGP advertising & have it propagate

- Goal 1: Route traffic through networks you control so that you can observe the traffic
- Goal 2: Send lots of traffic to someone you don't like (denial of service)

BGP Prefix Hijacking

4/25/2019
02:30 PM

Marc Laliberte
Commentary

Connect Directly

0 COMMENTS
COMMENT NOW

Login

100% 0%

How a Nigerian ISP Accidentally Hijacked the Internet

For 74 minutes, traffic destined for Google and Cloudflare services was routed through Russia and into the largest system of censorship in the world, China's Great Firewall.

On November 12, 2018, a small ISP in Nigeria made a mistake while updating its network infrastructure that highlights a critical flaw in the fabric of the Internet. The mistake effectively brought down Google — one of the largest tech companies in the world — for 74 minutes.

To understand what happened, we need to cover the basics of how Internet routing works. When I type, for example, HypotheticalDomain.com into my browser and hit enter, my computer creates a web request and sends it to Hypothtetical.Domain.com servers. These servers likely reside in a different state or country than I do. Therefore, my Internet service provider (ISP) must determine how to route my web browser's request to the server across the Internet. To maintain their routing tables, ISPs and Internet backbone companies use a protocol called Border Gateway Protocol (BGP).

https://www.darkreading.com/cloud/how-a-nigerian-isp-accidentally-hijacked-the-internet/a/d-id/1334482

BGP Defenses

Some attempts to mitigate Prefix Hijacking (S-BGP / BGP-Sec)

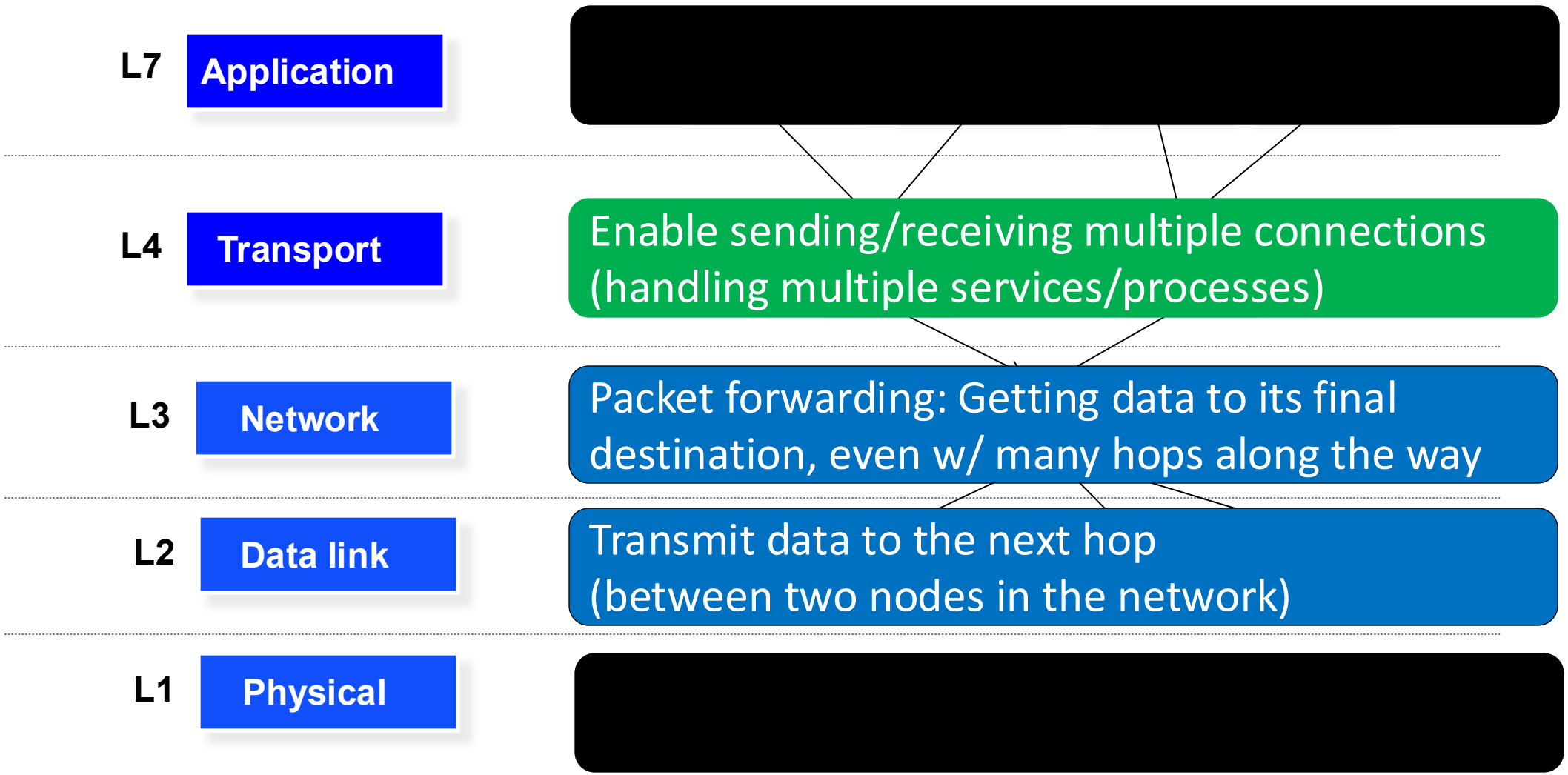
- Idea: Cryptographically sign routes
- Problems: Costly & Slow adoption

Instead: often rely on higher layer's security (e.g., TLS)

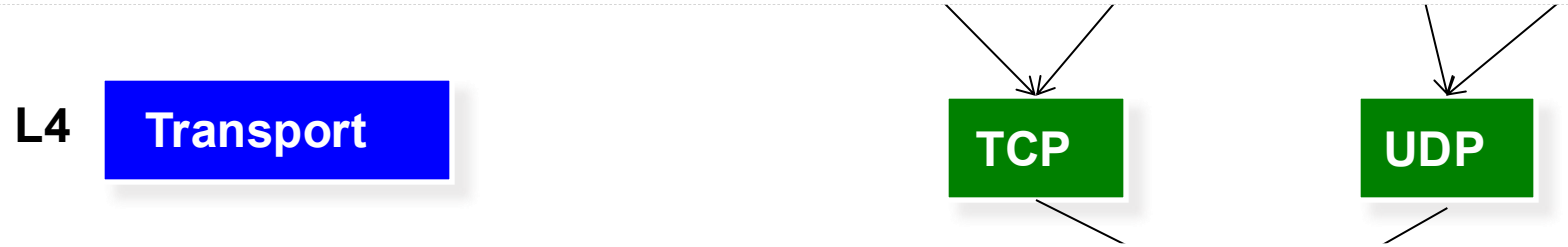
Outline

- Wrap-Up: Layer 2 (Wi-Fi)
- Layer 3 (BGP) Security
- TCP & UDP Attacks
- DNS Security

Network Stack (OSI 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)	

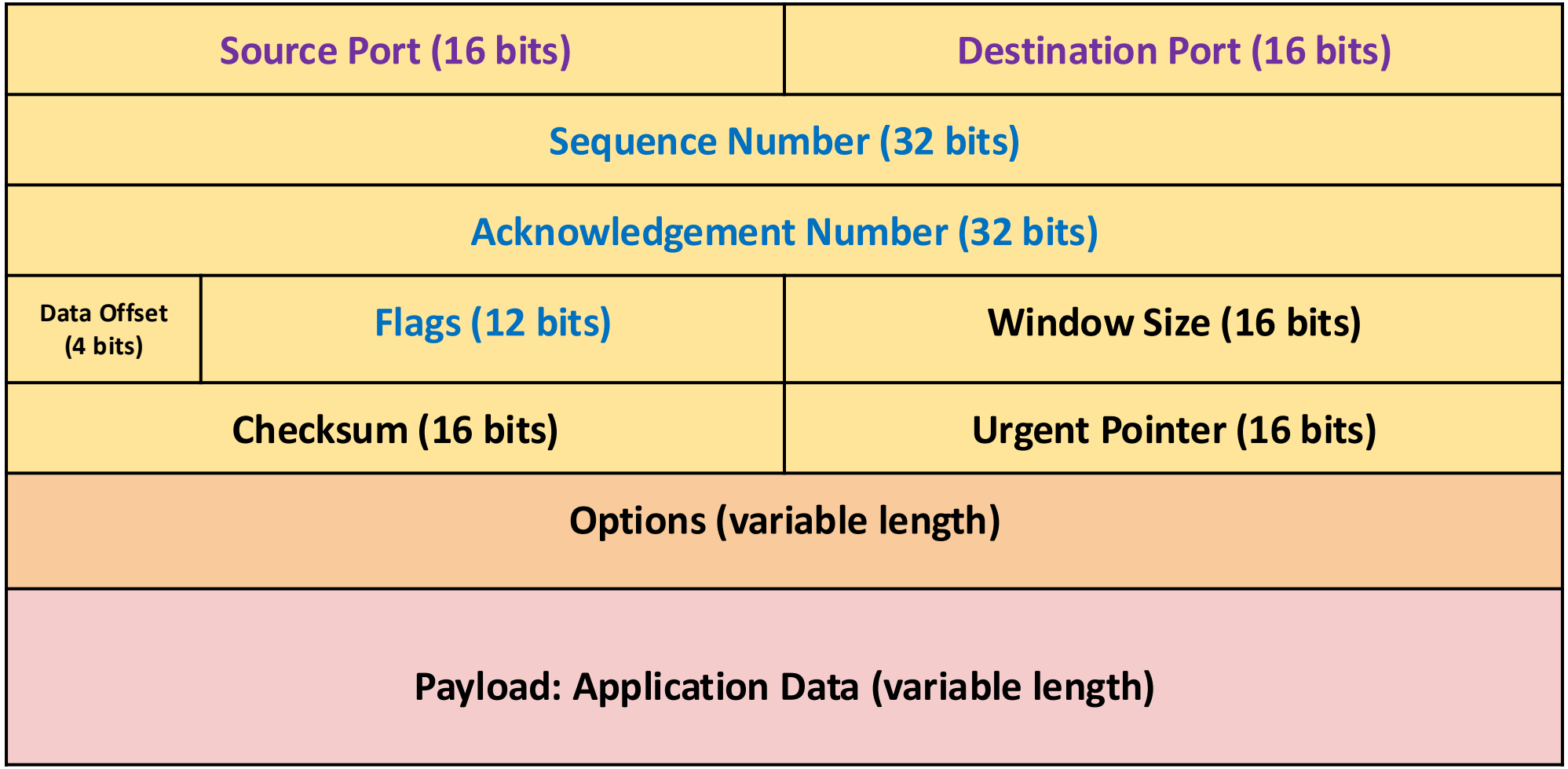
TCP: Reliable Data Streams

- 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 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

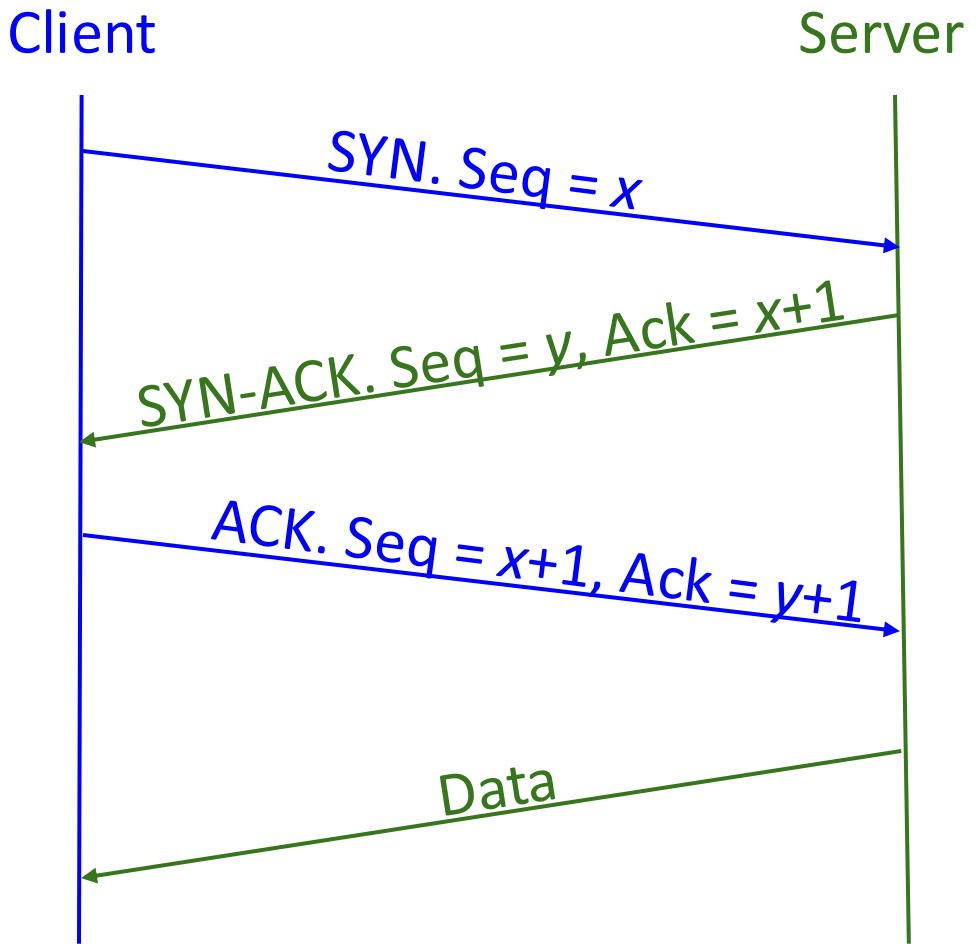
H	e	l	l	o		c	l	i	e	n	t
25	26	27	28	29	30	31	32	33	34	35	36

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

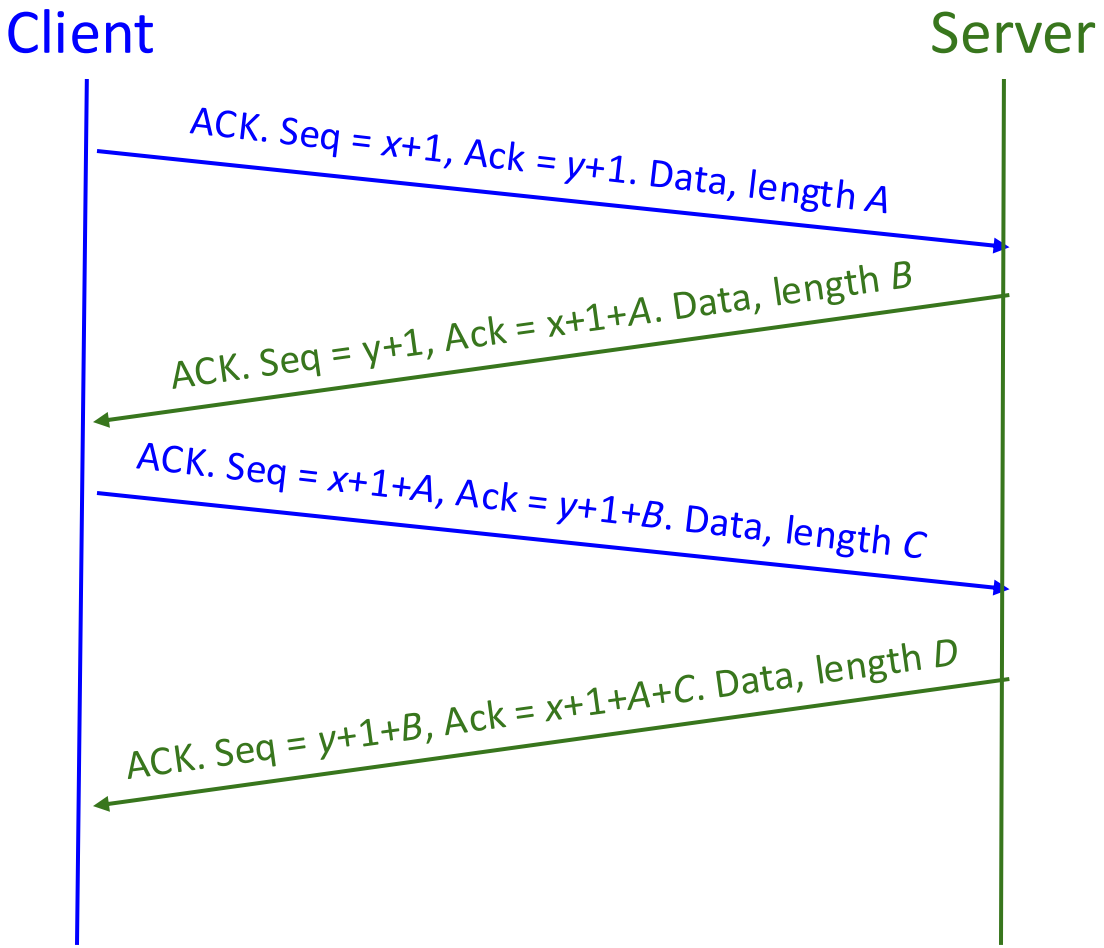
TCP headers use seq #'s to provide ordering and reliable delivery

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

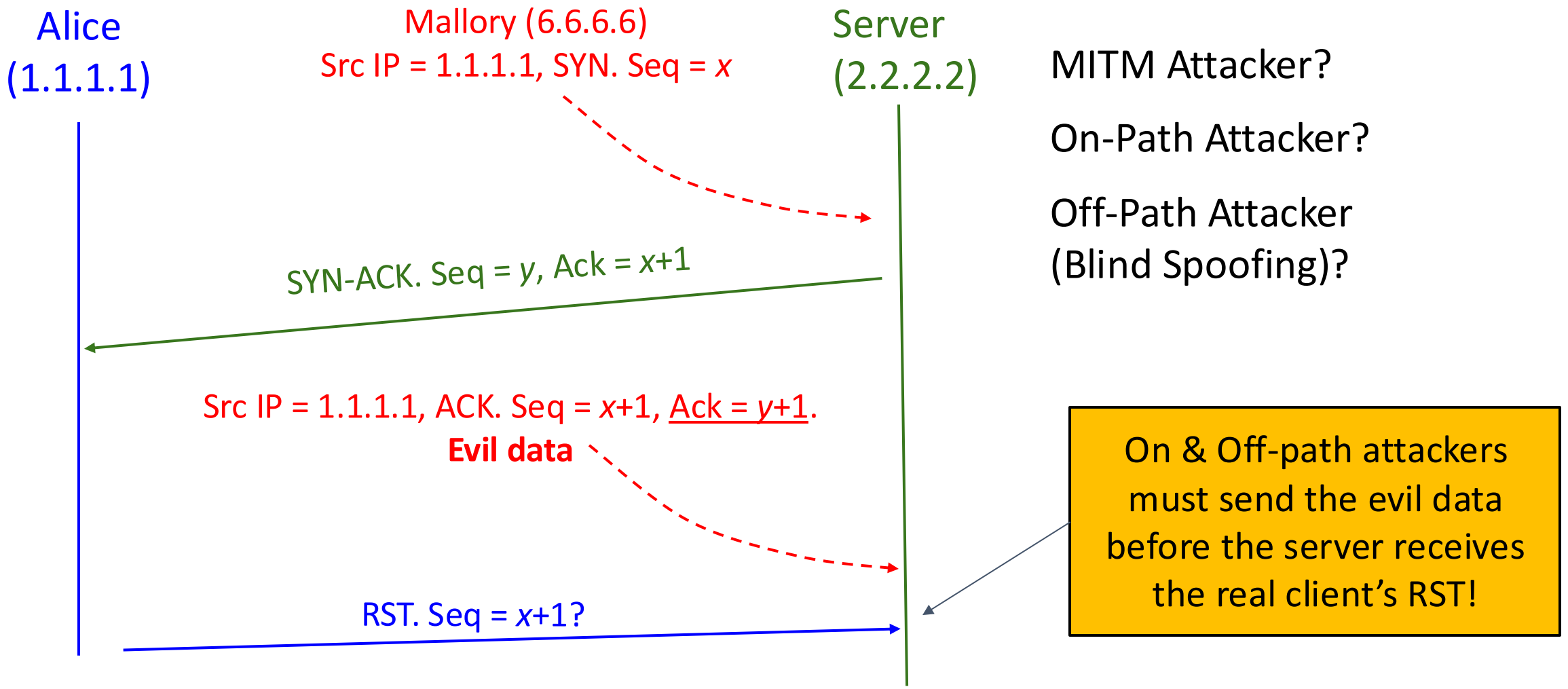
- If a packet is received by recipient, but the ACK is dropped:
  - Sender won't receive an ACK, so they try to send packet again
  - The recipient ignores the duplicate data and sends the ACK again

TCP Flags

- SYN
  - Indicates the beginning of the connection
- ACK
  - Indicates that the user is acknowledging 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

Attacks: TCP Spoofing

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



MITM Attacker?

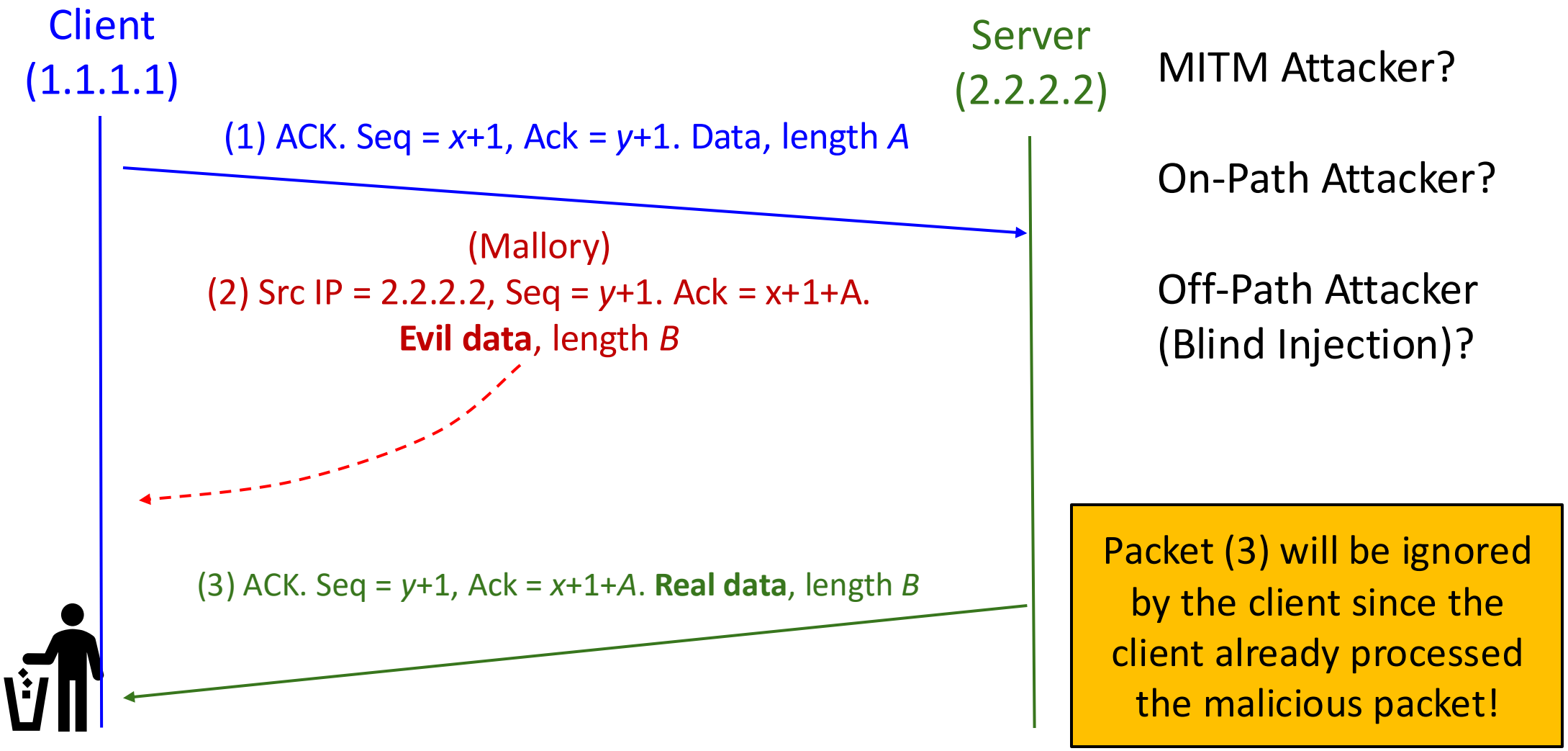
On-Path Attacker?

Off-Path Attacker (Blind Spoofing)?

On & Off-path attackers must send the evil data before the server receives the real client's RST!

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.

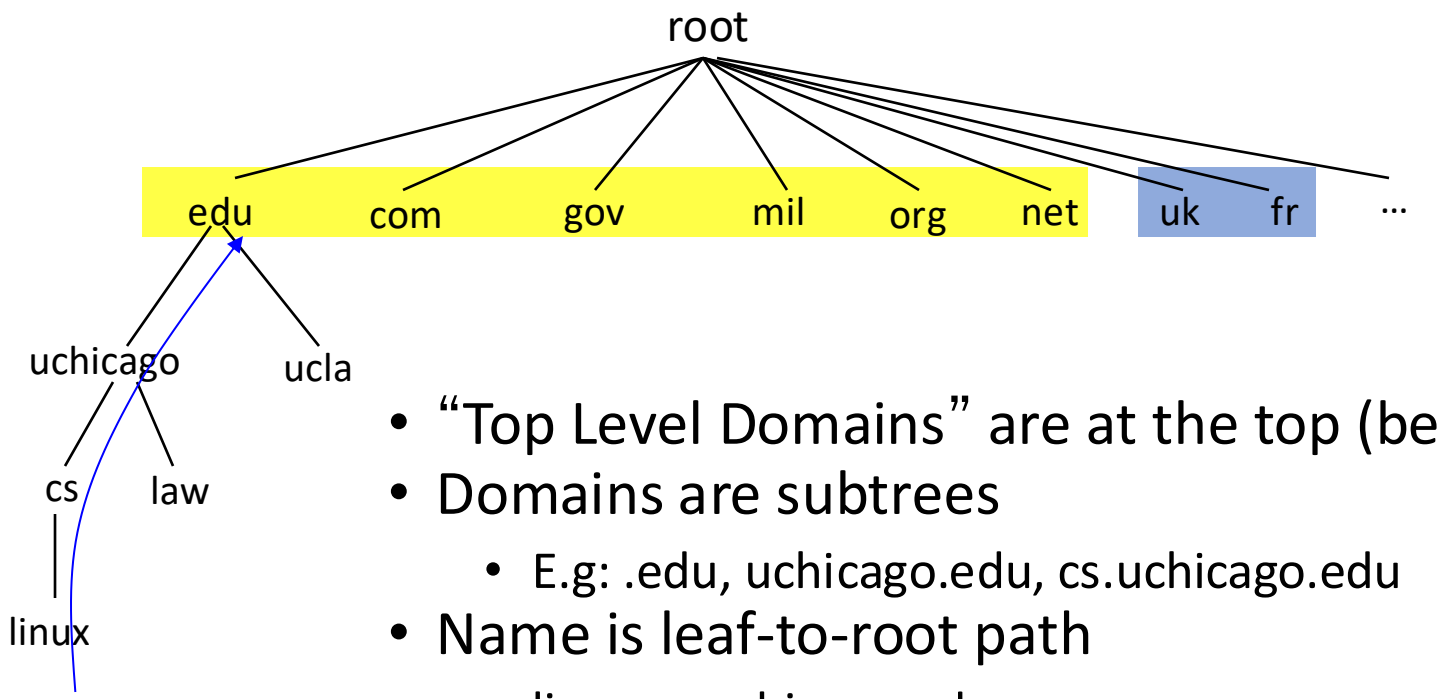
Outline

- Wrap-Up: Layer 2 (Wi-Fi)
- Layer 3 (BGP) Security
- TCP & UDP Attacks
- DNS Security

DNS (Domain Name System)

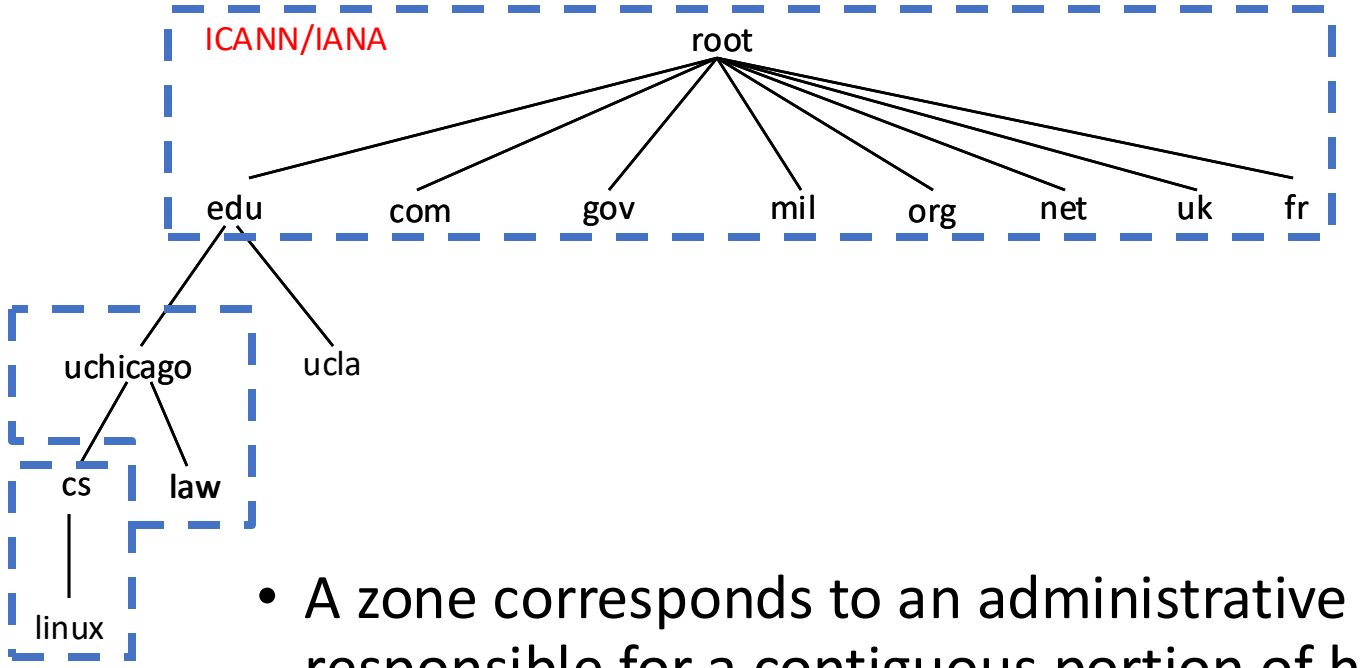
- Host (machine) IP addresses: e.g., *128.135.11.239*
  - A number used by protocols (e.g., BGP)
- Host names: e.g., *super.cs.uchicago.edu*
  - Usable by humans
- Domain Name System (DNS): how we map from hostnames to IP addresses

Hierarchical Namespace



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

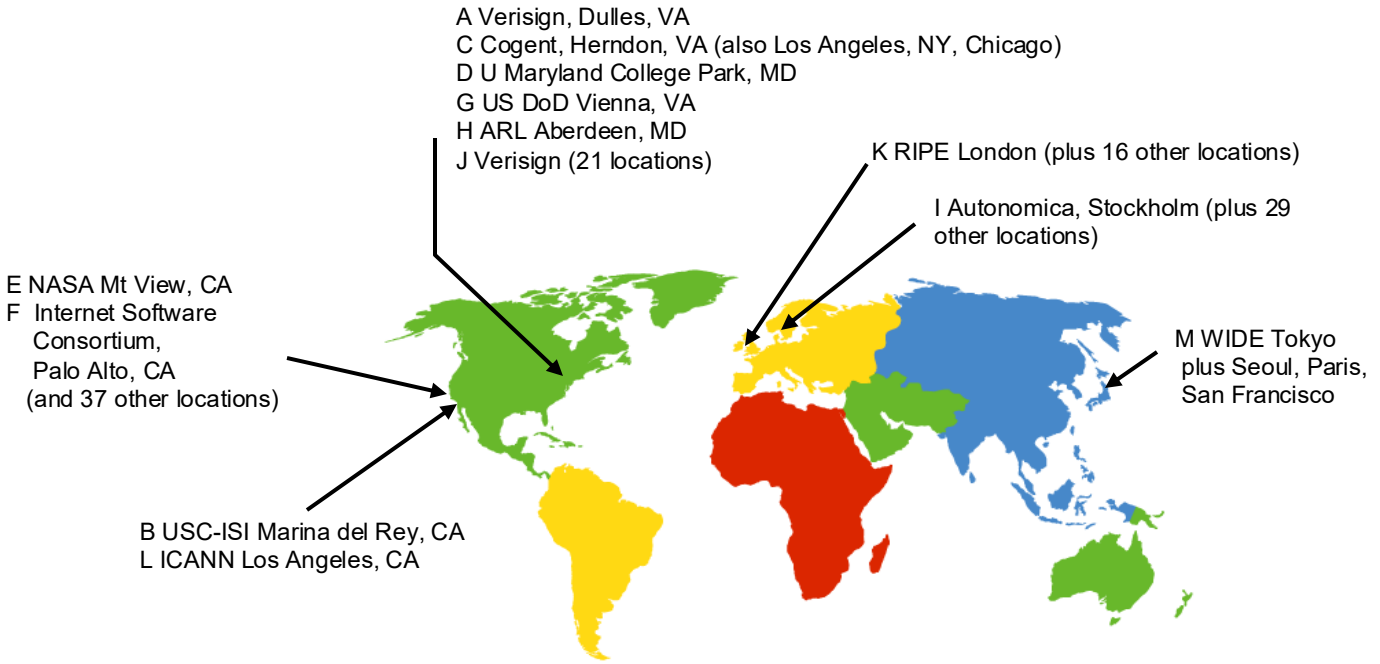
Hierarchical Administration



- 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



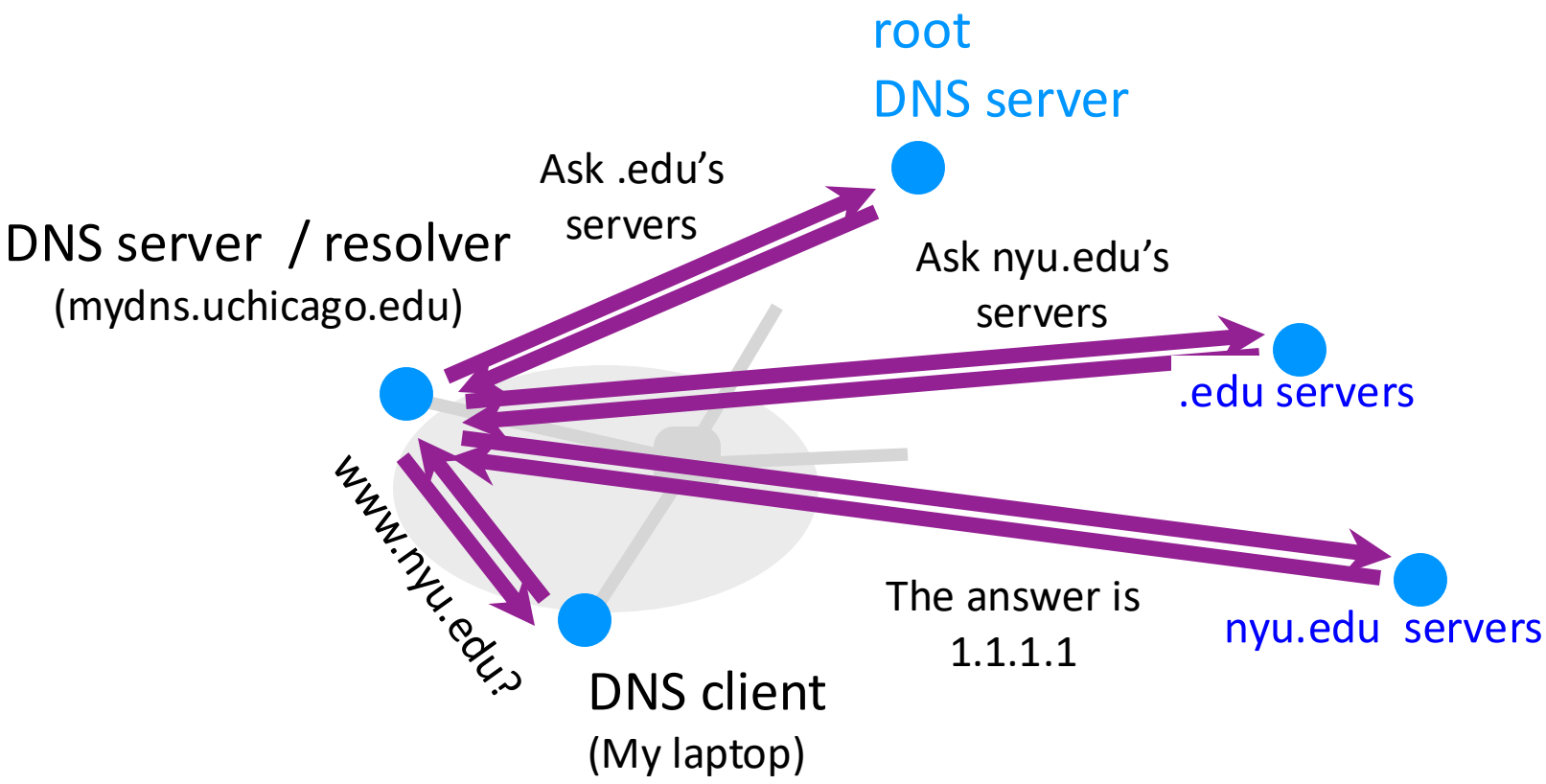
DNS Records

- 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

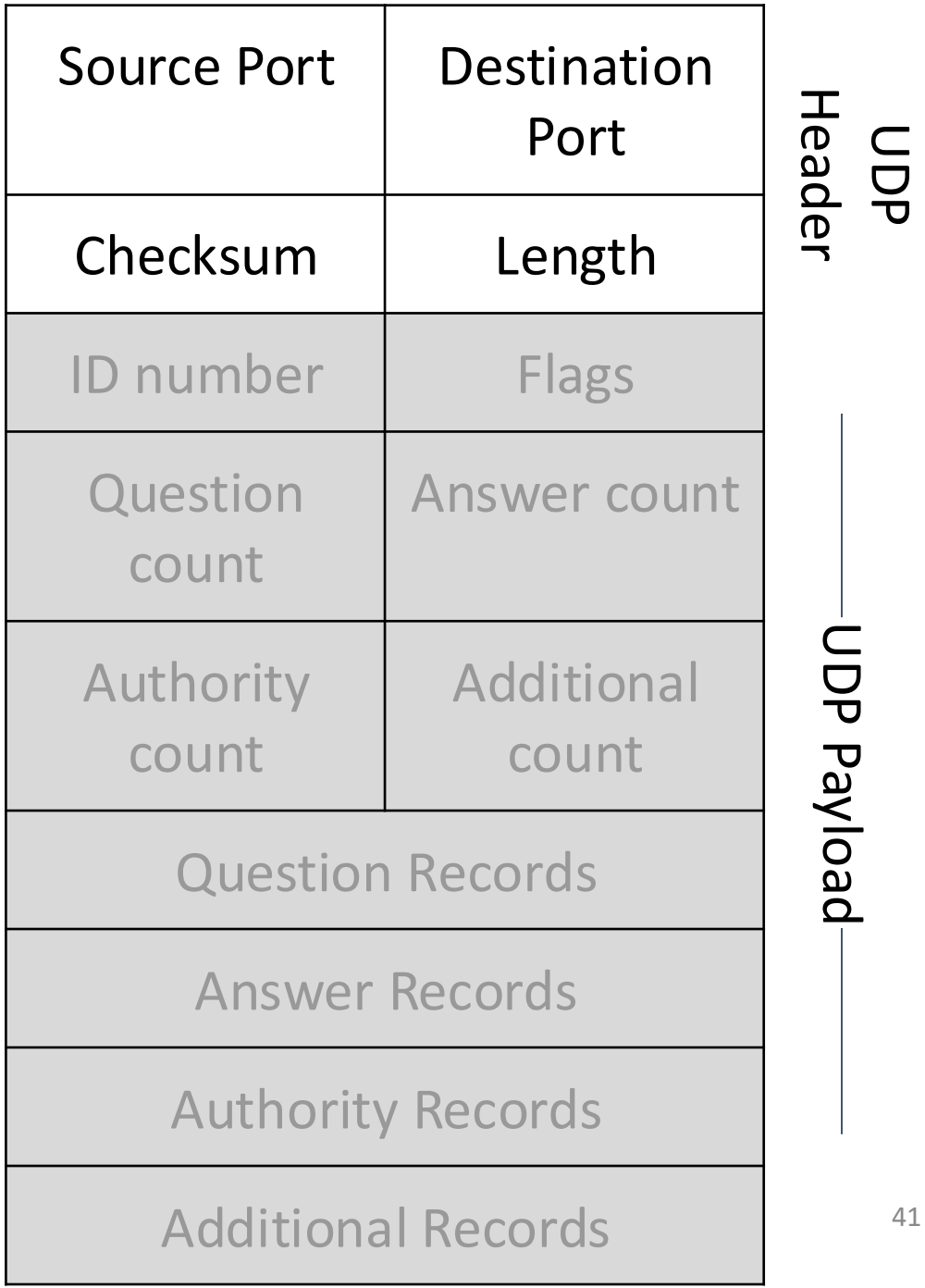
DNS is designed to be lightweight and fast: uses UDP for transport protocol

Source port (16 bits): Chosen by the client

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

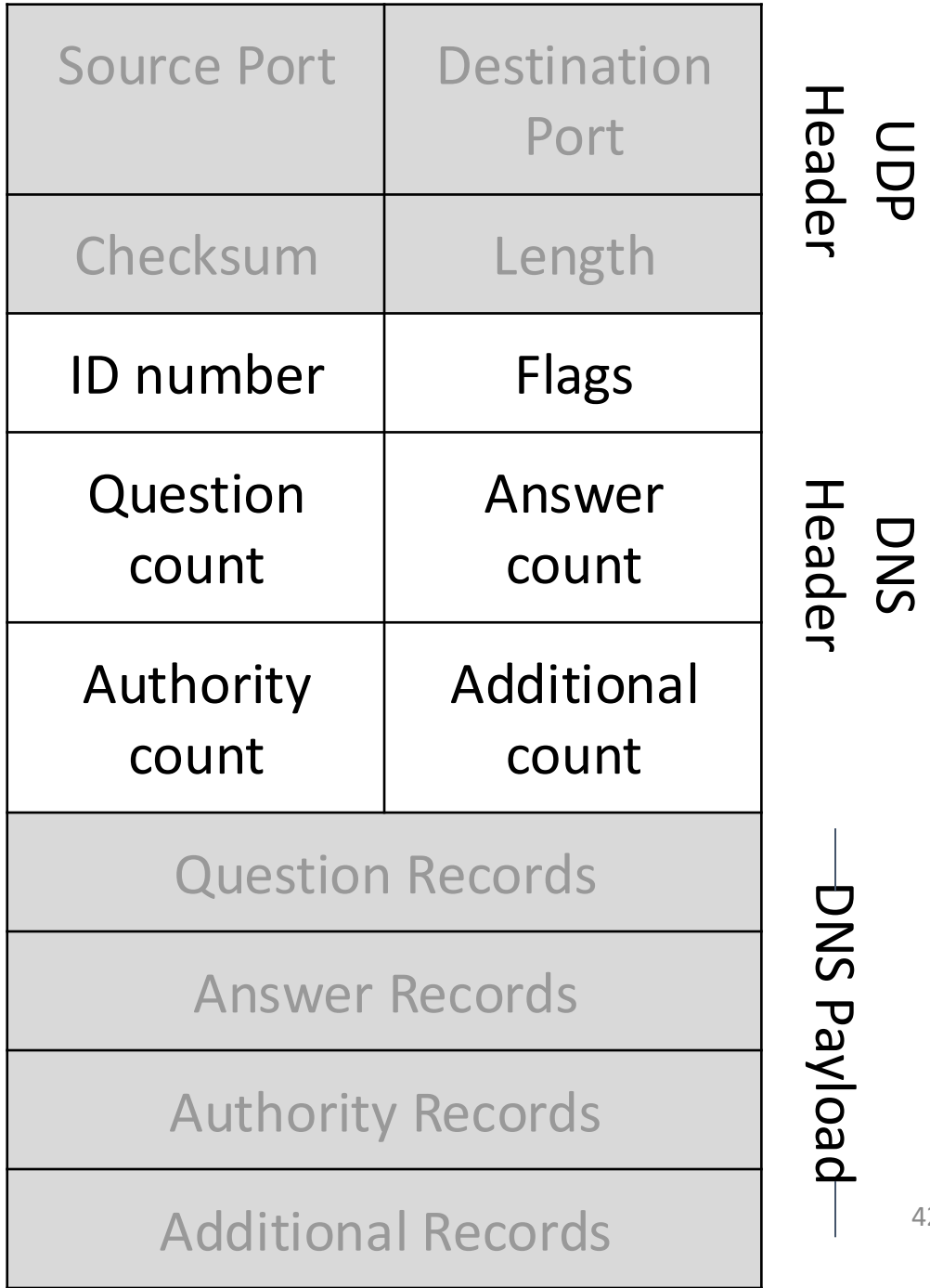
Destination port (16 bits): Usually 53

- 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

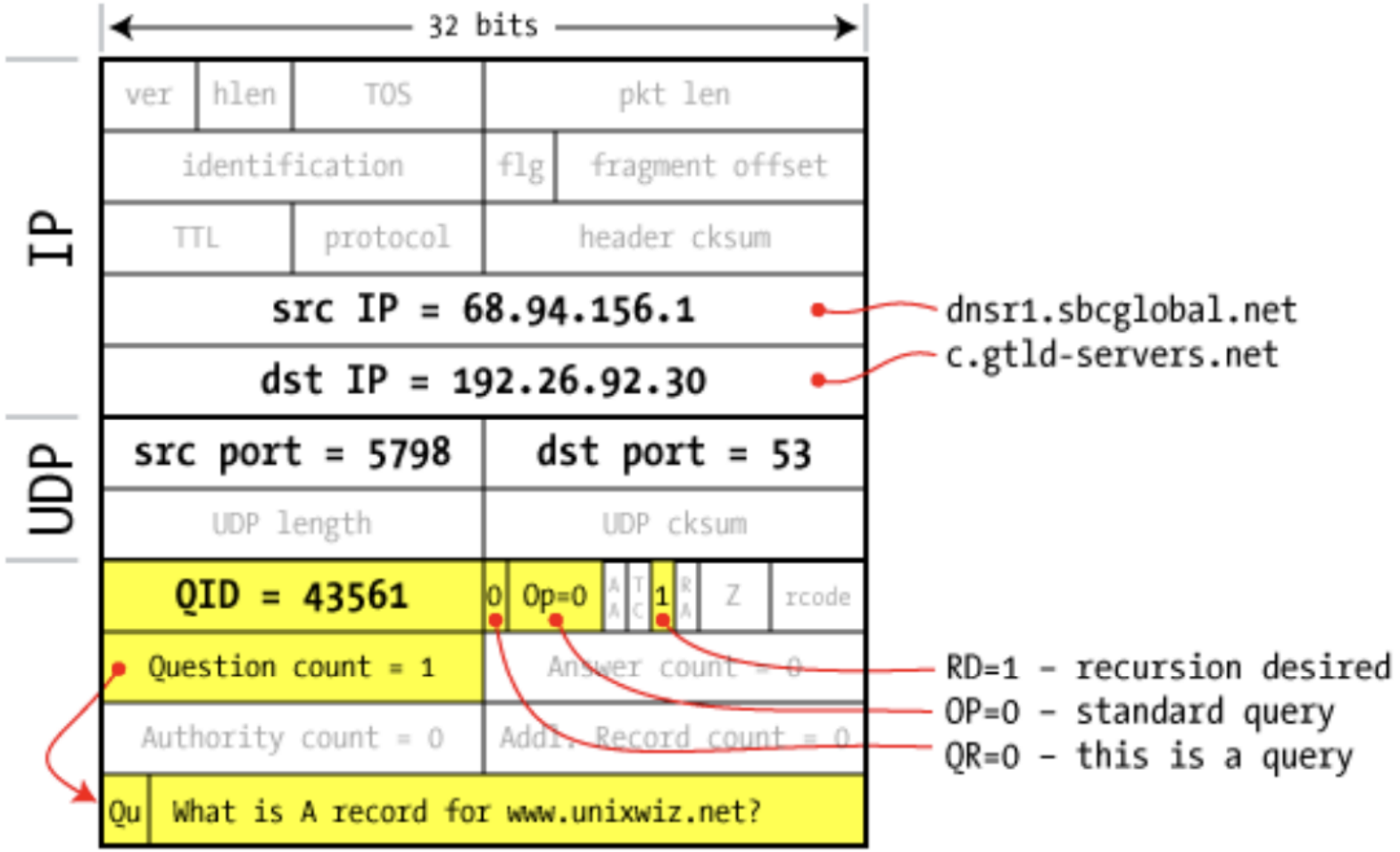


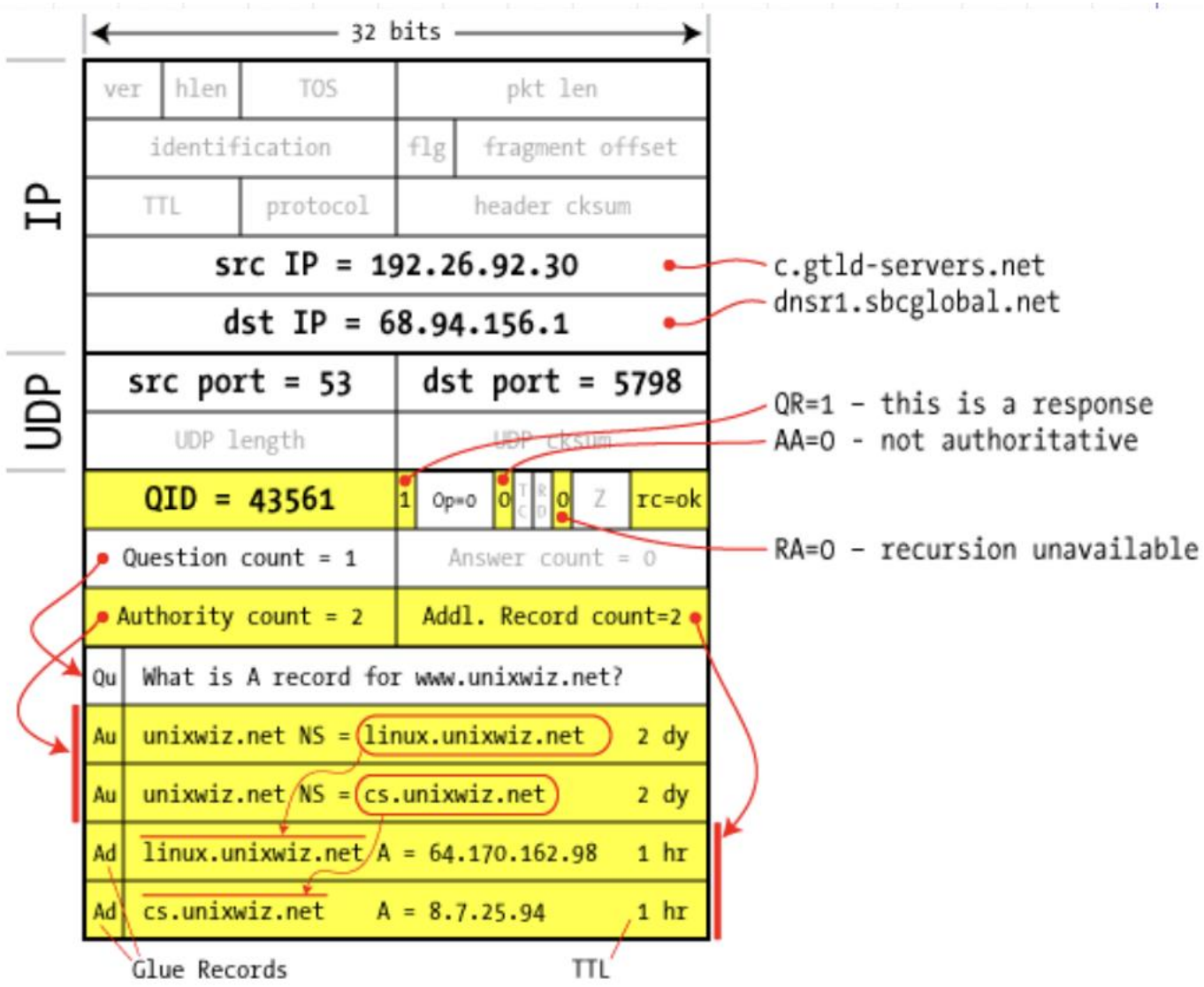
Image from http://unixwiz.net/techtips/iguide-kaminsky-dns-vuln.html

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?

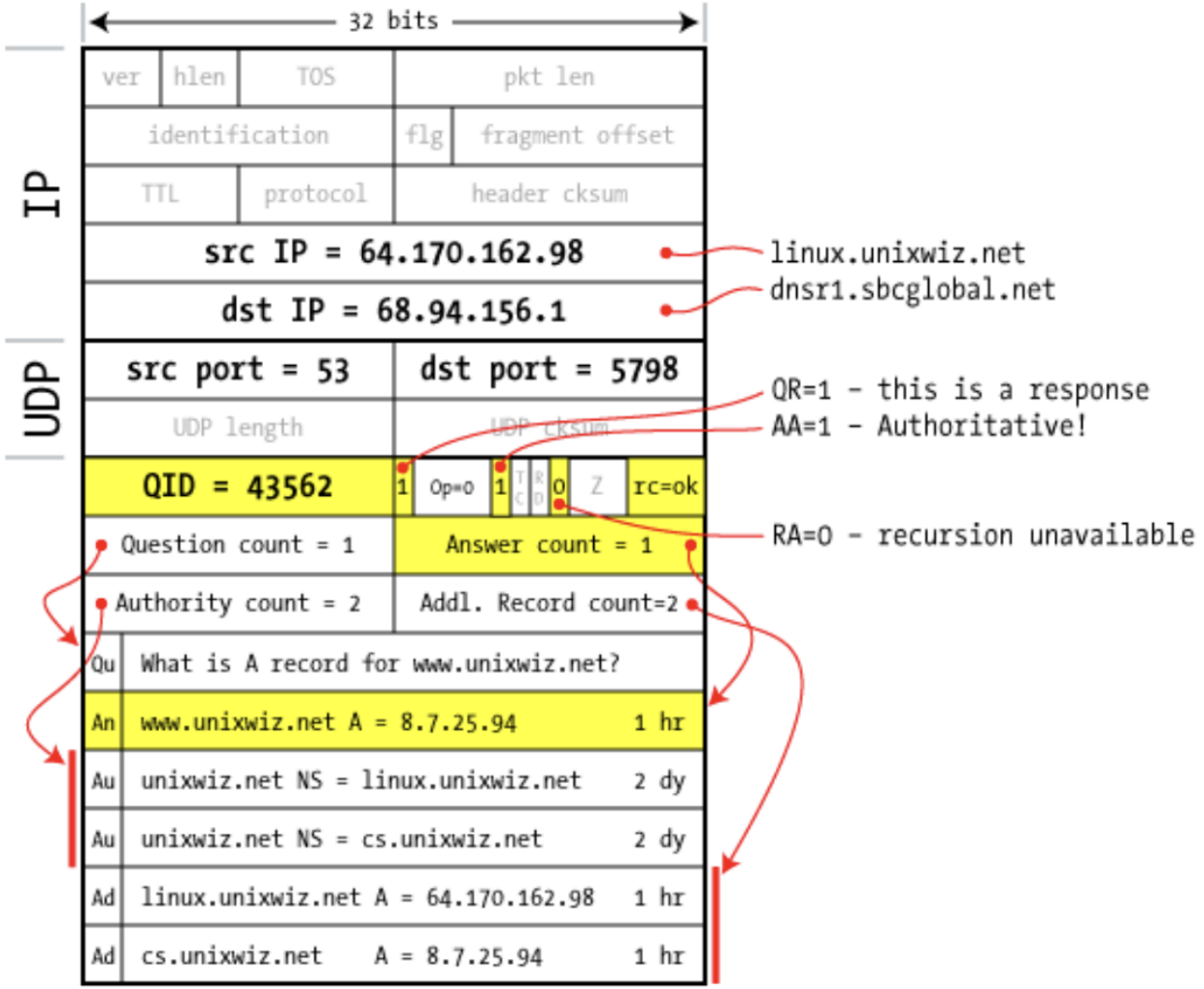
Additional Section [Glue Records]: What are their IP Addresses so you can go ask them?



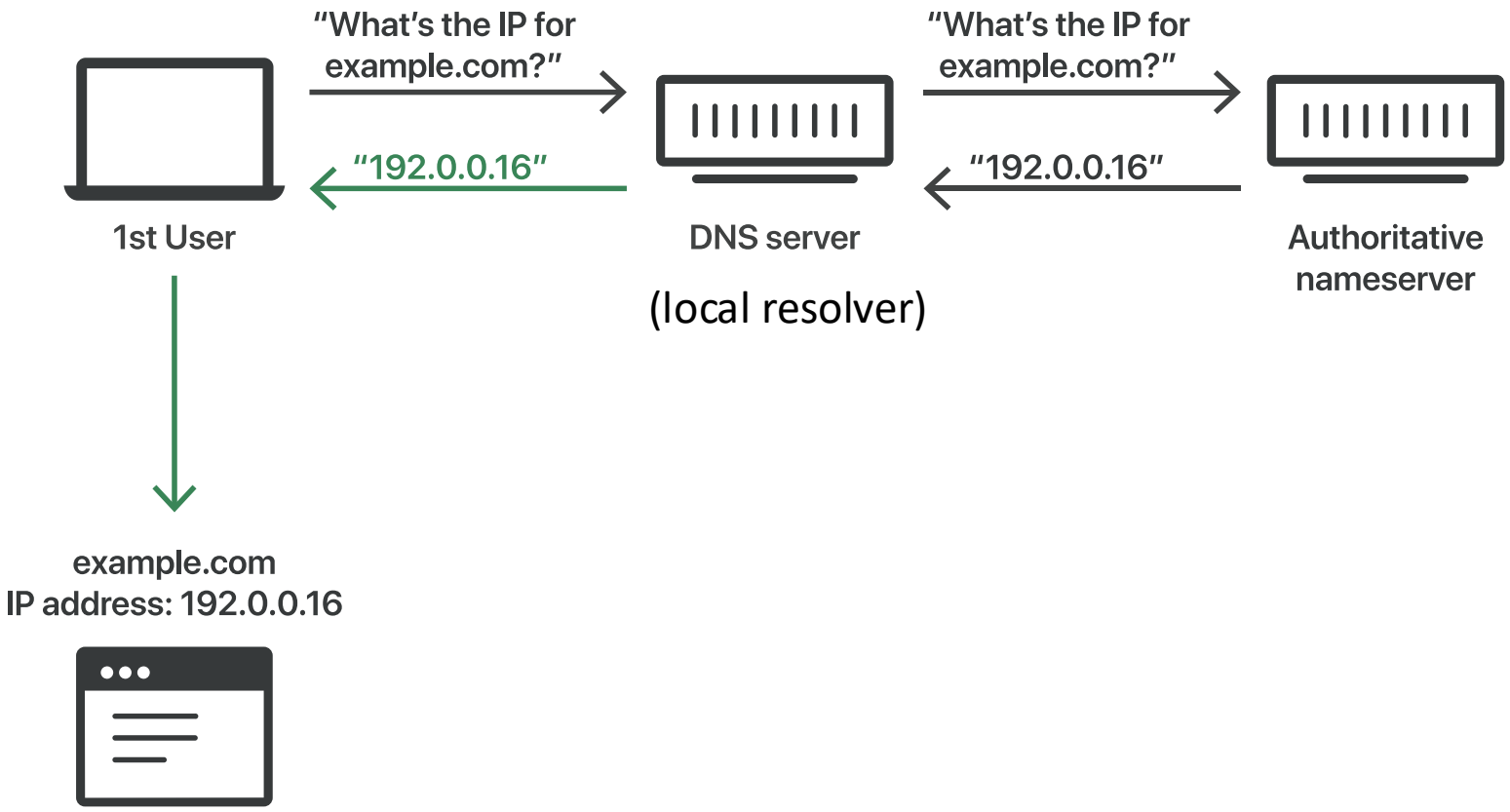
Example: Authoritative Response

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"?

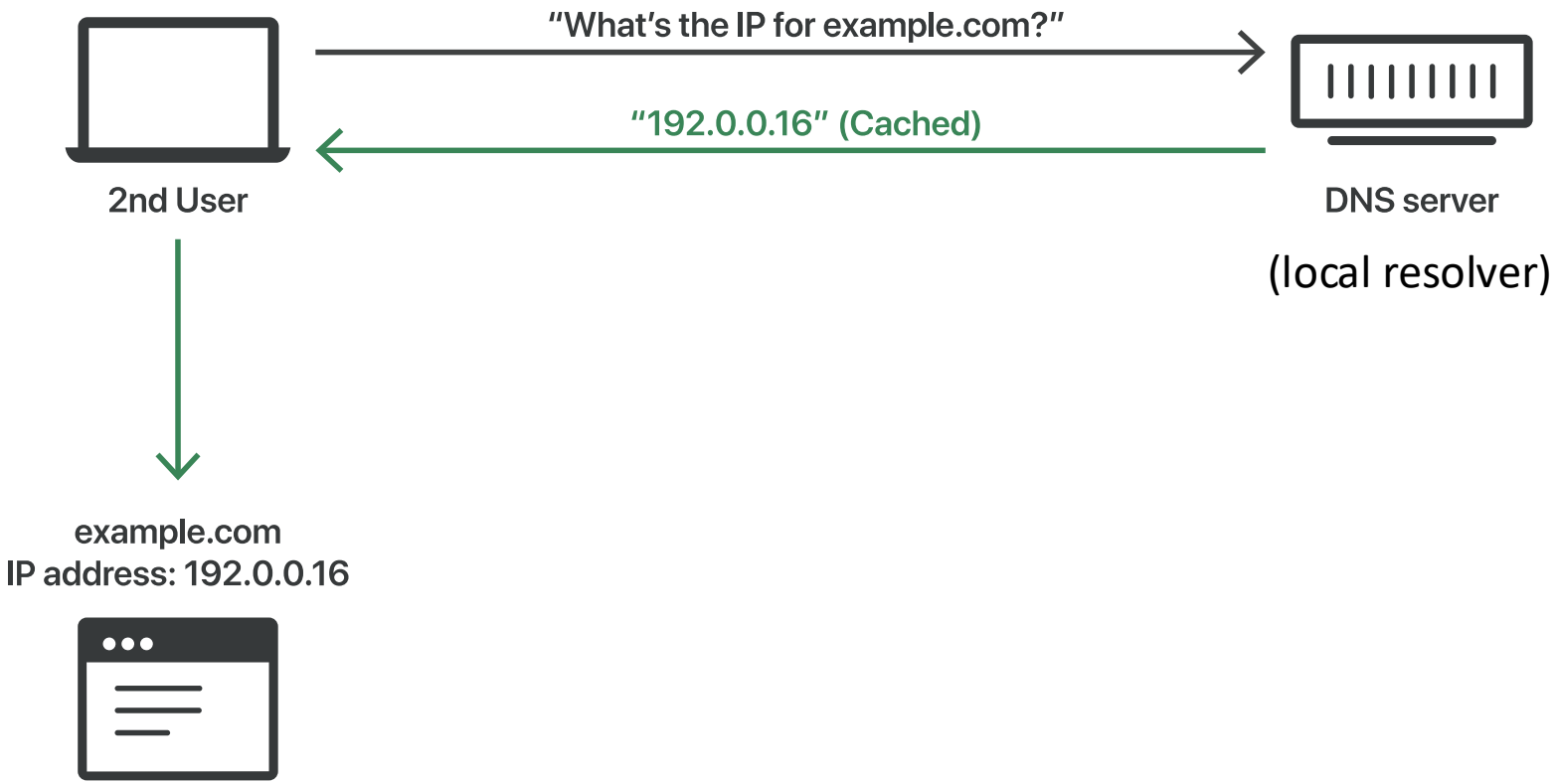


DNS (Uncached)



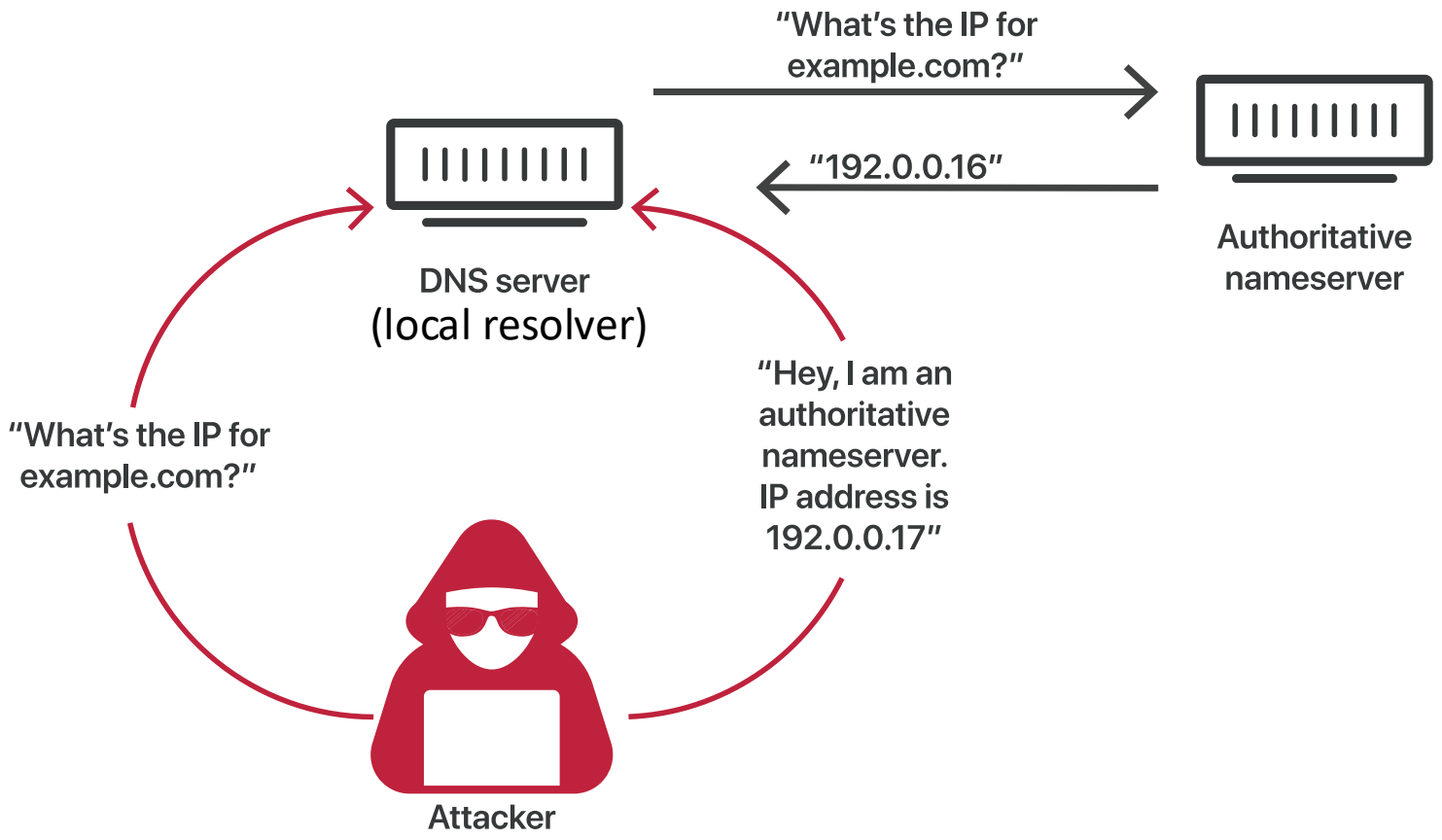
Images from https://www.cloudflare.com/learning/dns/dns-cache-poisoning/

DNS (Cached, Benign)



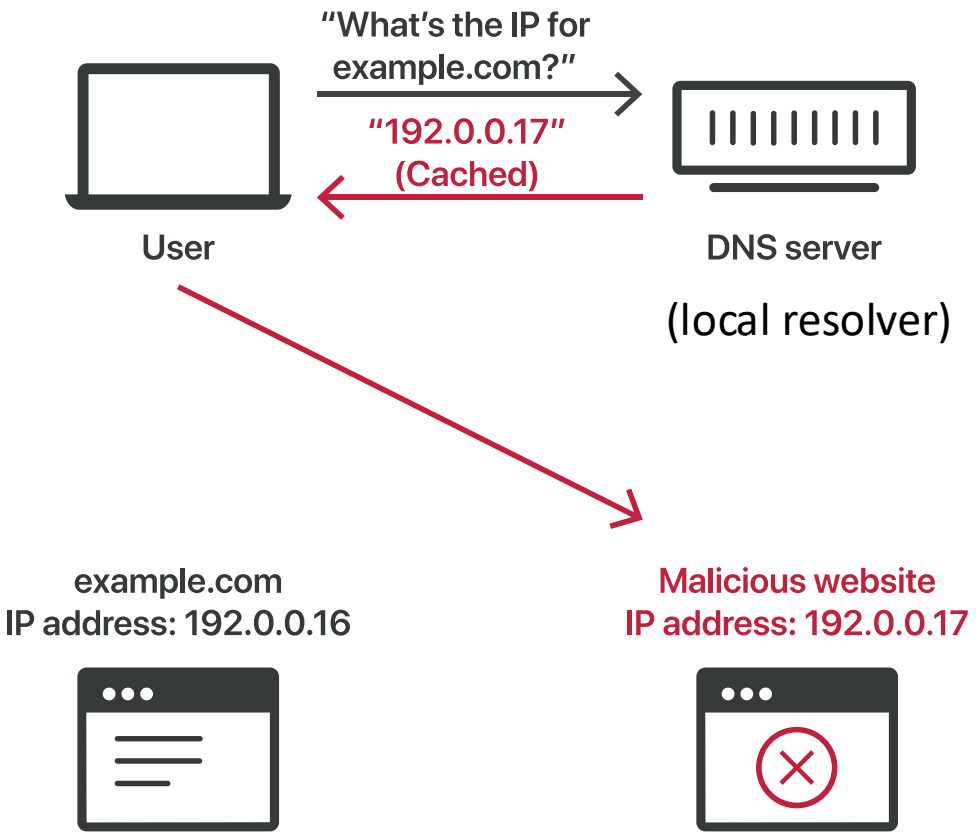
Images from https://www.cloudflare.com/learning/dns/dns-cache-poisoning/

DNS Cache Poisoning Attack



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

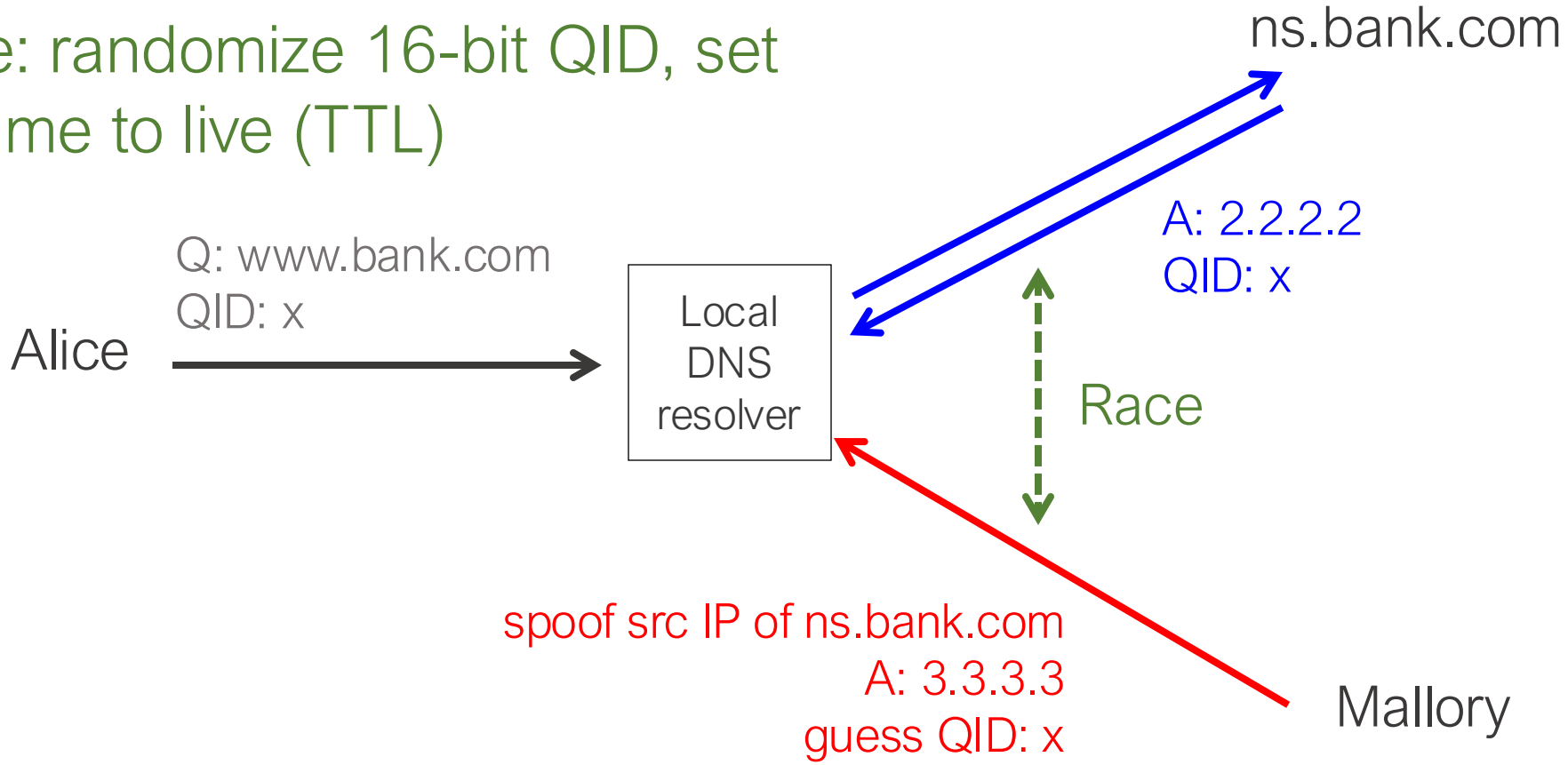
DNS Cache Poisoning Result



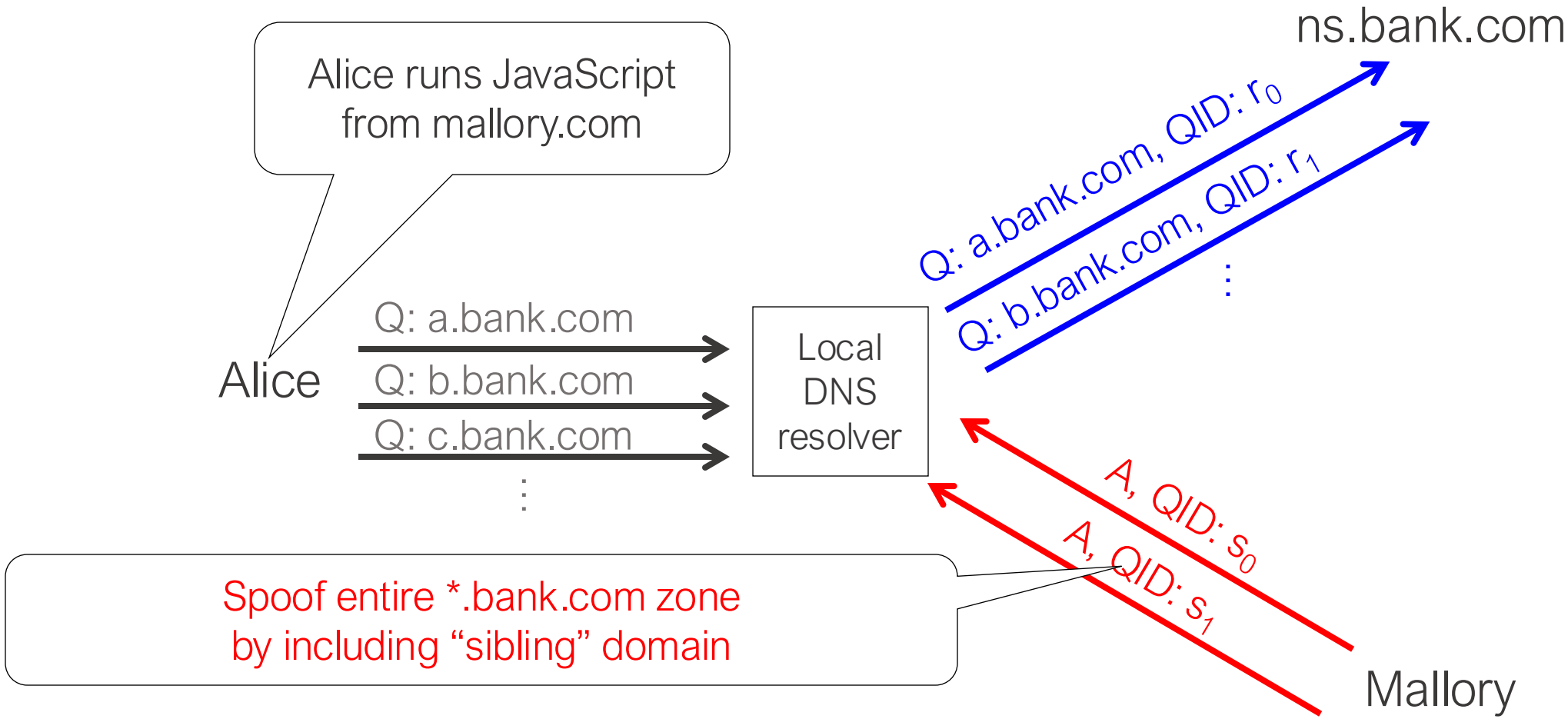
Images from https://www.cloudflare.com/learning/dns/dns-cache-poisoning/

DNS Cache Poisoning

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



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)

- The attacker's fraudulent responses for fake subdomains (e.g., z123123.bank.com) return a response saying that the authoritative nameserver for bank.com = host controlled by the attacker

- Defense: Randomize both the query ID and source port

DNS Defenses: 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 (next week).

The End