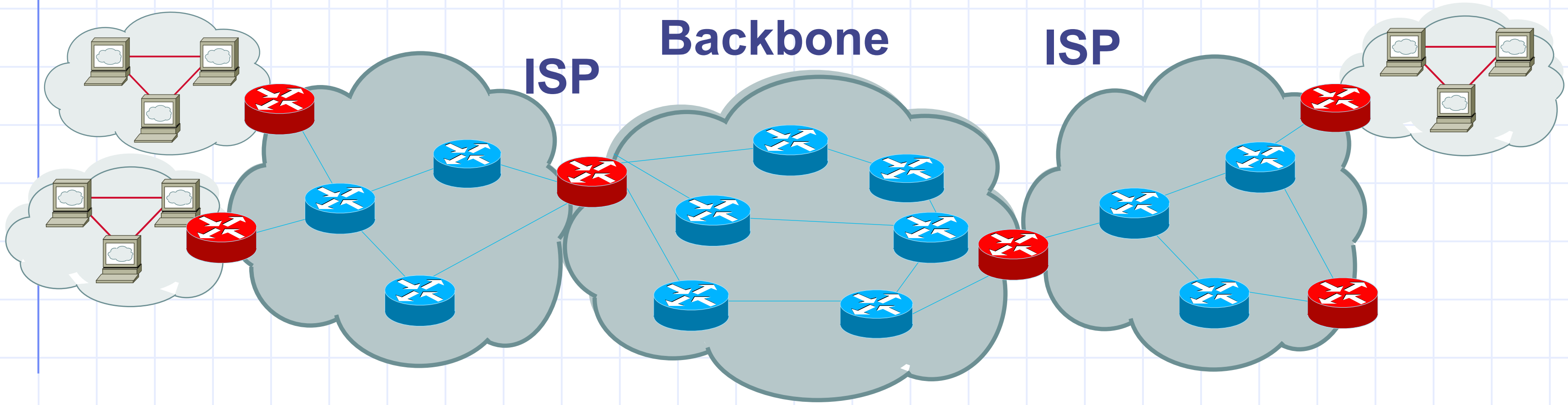
CS 155

Internet Security: How the Internet works and some basic vulnerabilities

Dan Boneh

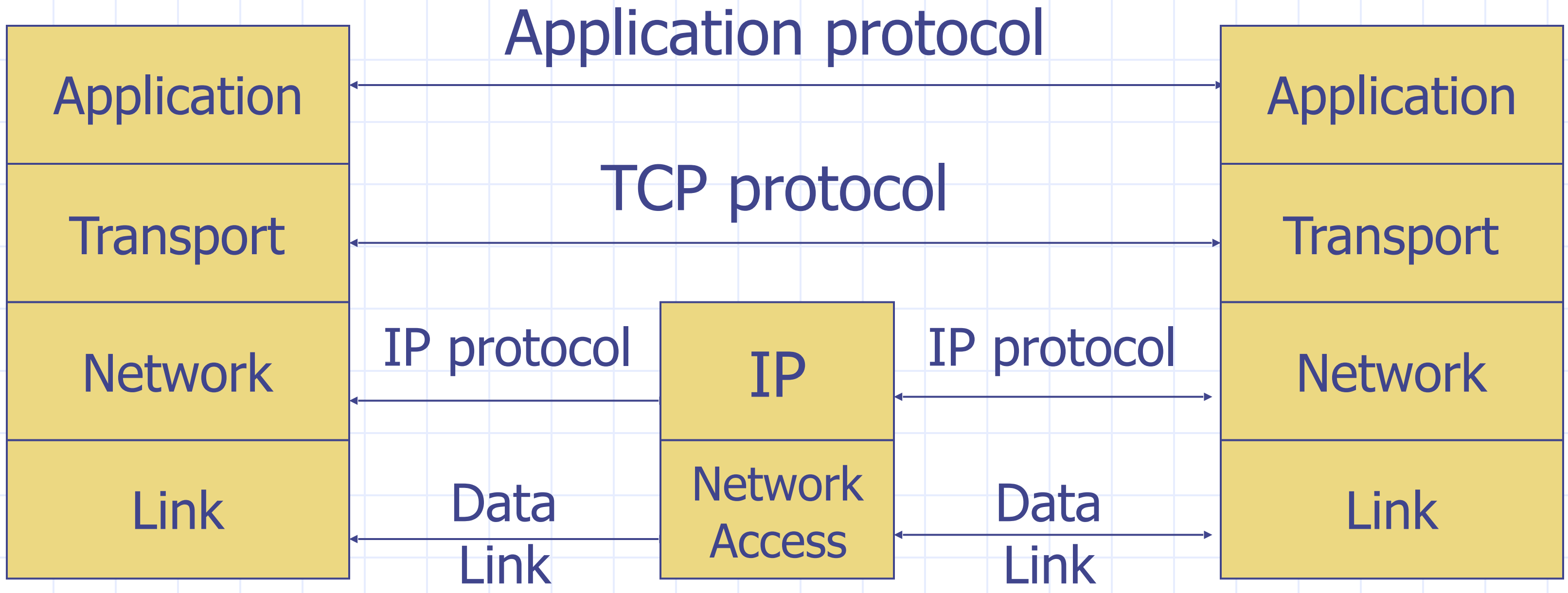
Acknowledgments: Lecture slides are from the Computer Security course taught by Dan Boneh and John Mitchell at Stanford University. When slides are obtained from other sources, a a reference will be noted on the bottom of that slide. A full list of references is provided on the last slide.

Internet Infrastructure

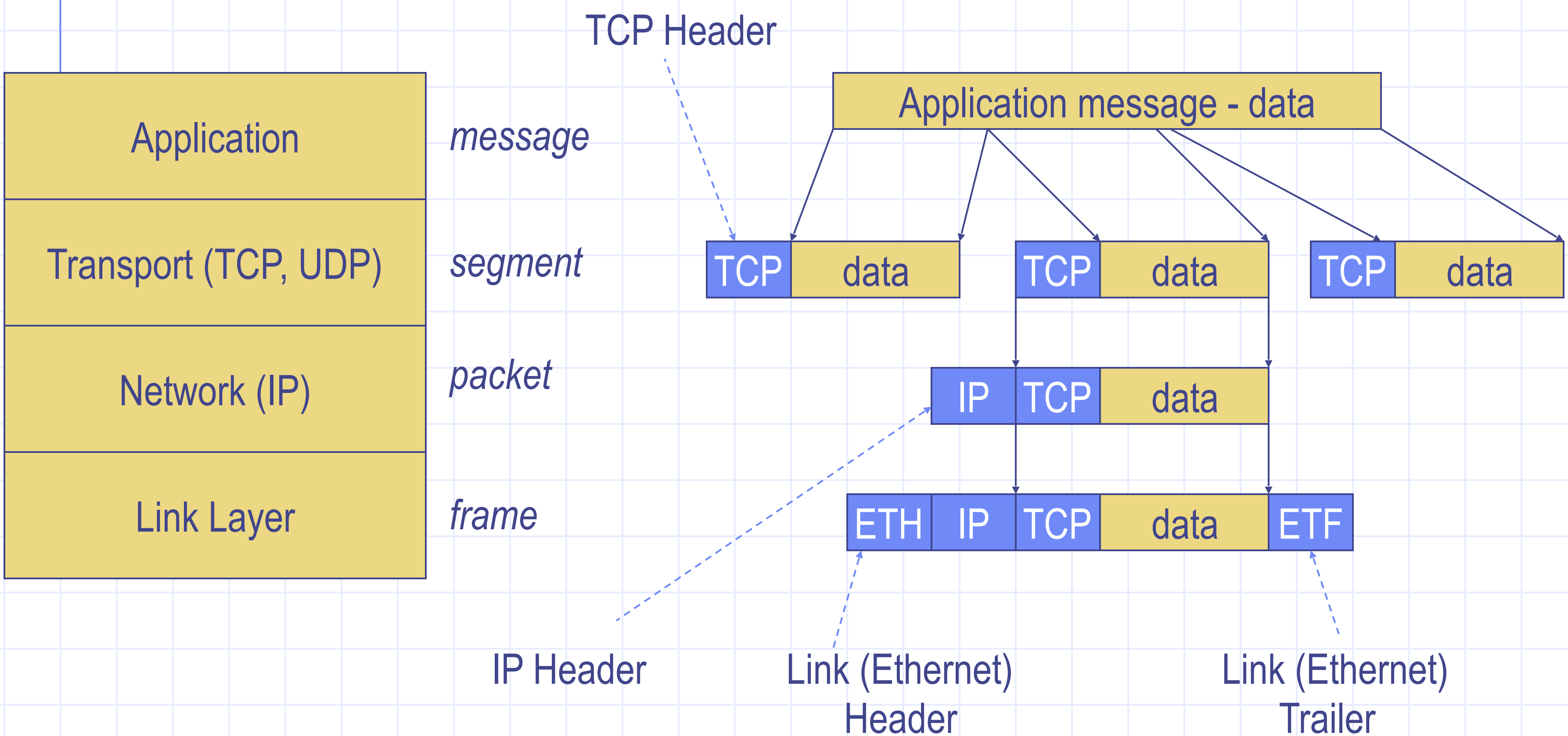


- Local and interdomain routing
  - TCP/IP for routing and messaging
  - BGP for routing announcements
- Domain Name System
  - Find IP address from symbolic name (www.cs.stanford.edu)

TCP Protocol Stack



Data Formats

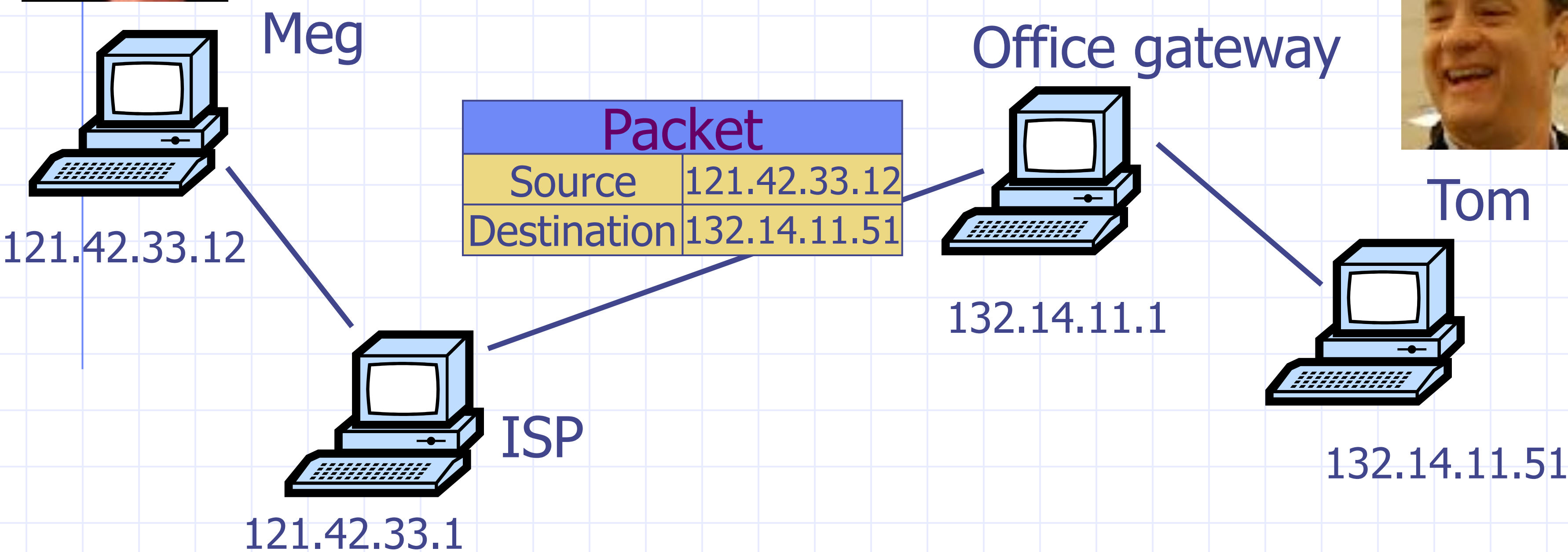


Internet Protocol

- Connectionless
  - Unreliable
  - Best effort
- Notes:
  - src and dest **ports** not parts of IP hdr

Version	Header Length
Type of Service	
Total Length	
Identification	
Flags	Fragment Offset
Time to Live	
Protocol	
Header Checksum	
Source Address of Originating Host	
Destination Address of Target Host	
Options	
Padding	
IP Data	

IP Routing



Packet	
Source	121.42.33.12
Destination	132.14.11.51

- Typical route uses several hops
- IP: no ordering or delivery guarantees

IP Protocol Functions (Summary)

- Routing
  - IP host knows location of router (gateway)
  - IP gateway must know route to other networks
- Fragmentation and reassembly
  - If max-packet-size less than the user-data-size
- Error reporting
  - ICMP packet to source if packet is dropped
- TTL field: decremented after every hop
  - Packet dropped if TTL=0. Prevents infinite loops.

Problem: no src IP authentication

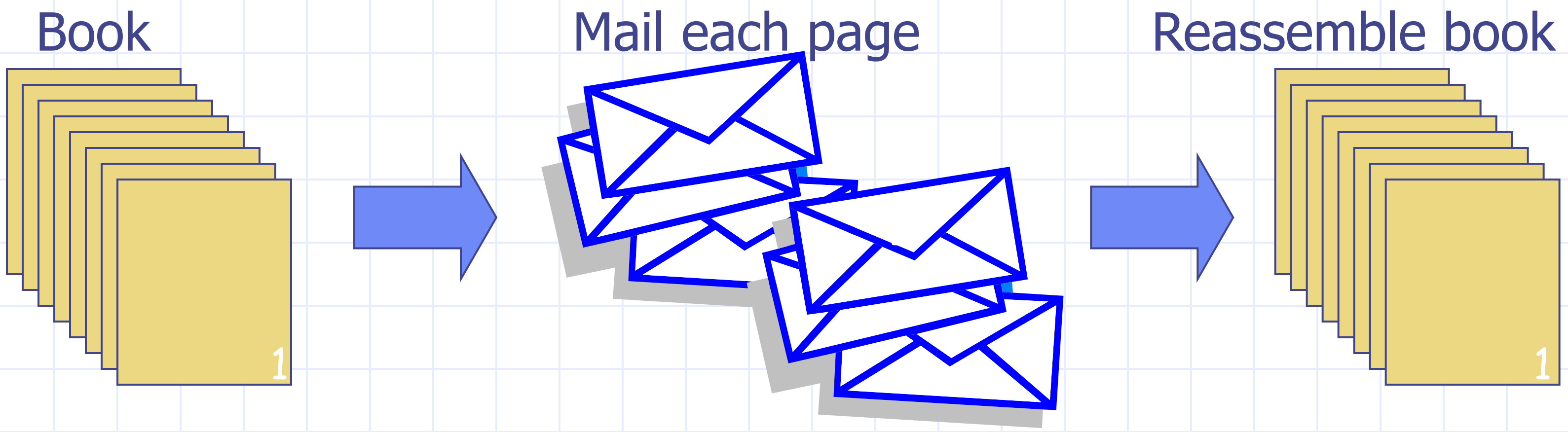
- Client is trusted to embed correct source IP
  - Easy to override using raw sockets
  - **Libnet**: a library for formatting raw packets with arbitrary IP headers

- Anyone who owns their machine can send packets with arbitrary source IP
  - … response will be sent back to forged source IP

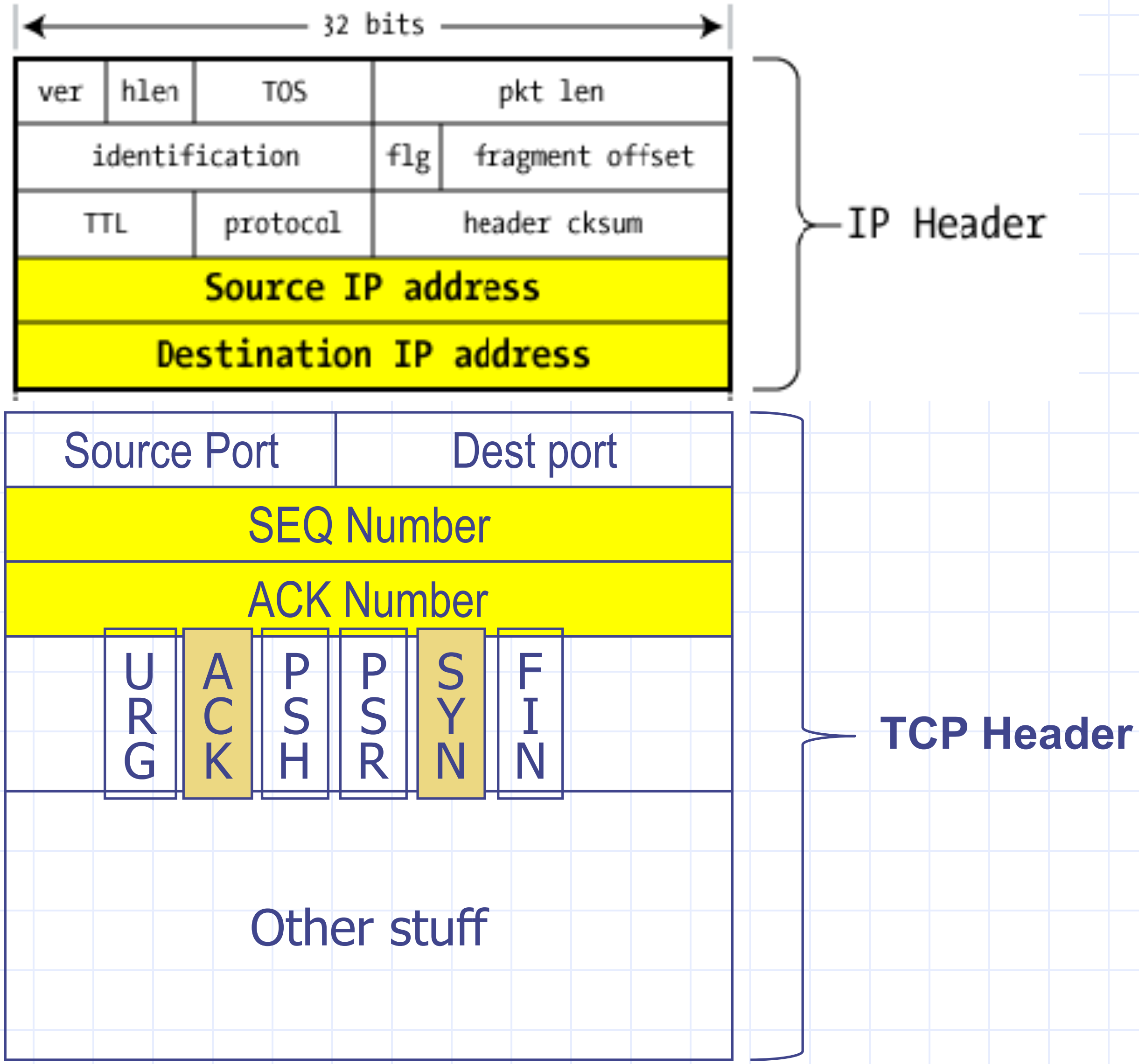
- Implications: (solutions in DDoS lecture)
  - Anonymous DoS attacks;
  - Anonymous infection attacks (e.g. slammer worm)

Transmission Control Protocol

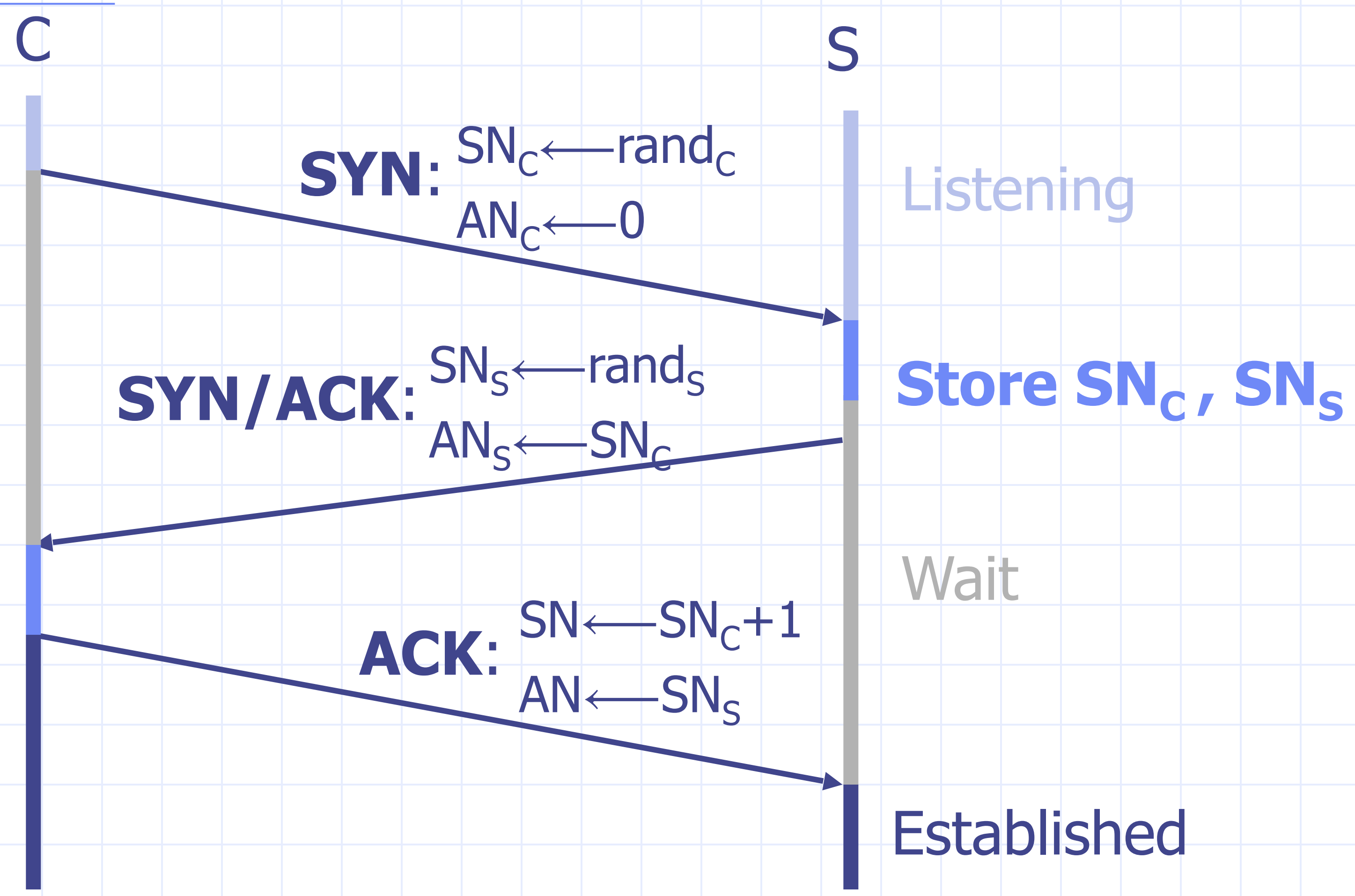
- Connection-oriented, preserves order
  - Sender
    - Break data into packets
    - Attach packet numbers
  - Receiver
    - Acknowledge receipt; lost packets are resent
    - Reassemble packets in correct order



TCP Header (protocol=6)



Review: TCP Handshake



Received packets with SN too far out of window are dropped

Basic Security Problems

1. Network packets pass by untrusted hosts
   - Eavesdropping, packet sniffing
   - Especially easy when attacker controls a machine close to victim (e.g. WiFi routers)

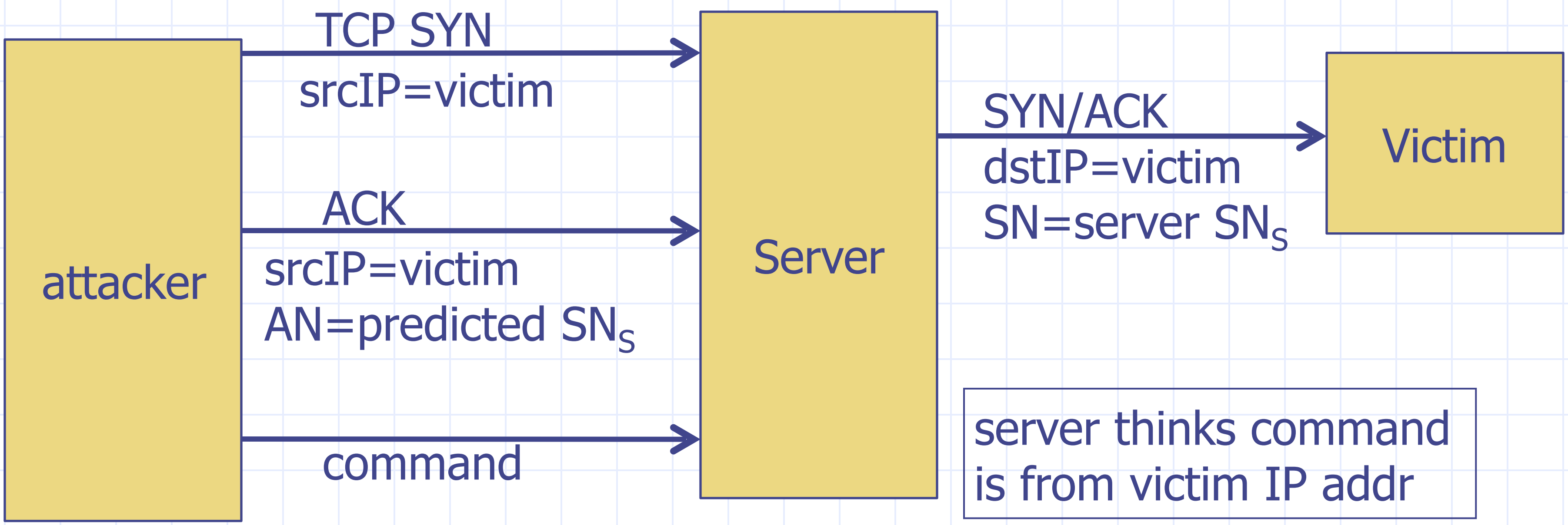
2. TCP state easily obtained by eavesdropping
   - Enables spoofing and session hijacking

3. Denial of Service (DoS) vulnerabilities
   - DDoS lecture

Why random initial sequence numbers?

Suppose initial seq. numbers (SN_C , SN_S) are predictable:

- Attacker can create TCP session on behalf of forged source IP
- Breaks IP-based authentication (e.g. SPF, /etc/hosts)
  - Random seq. num. do not prevent attack, but make it harder



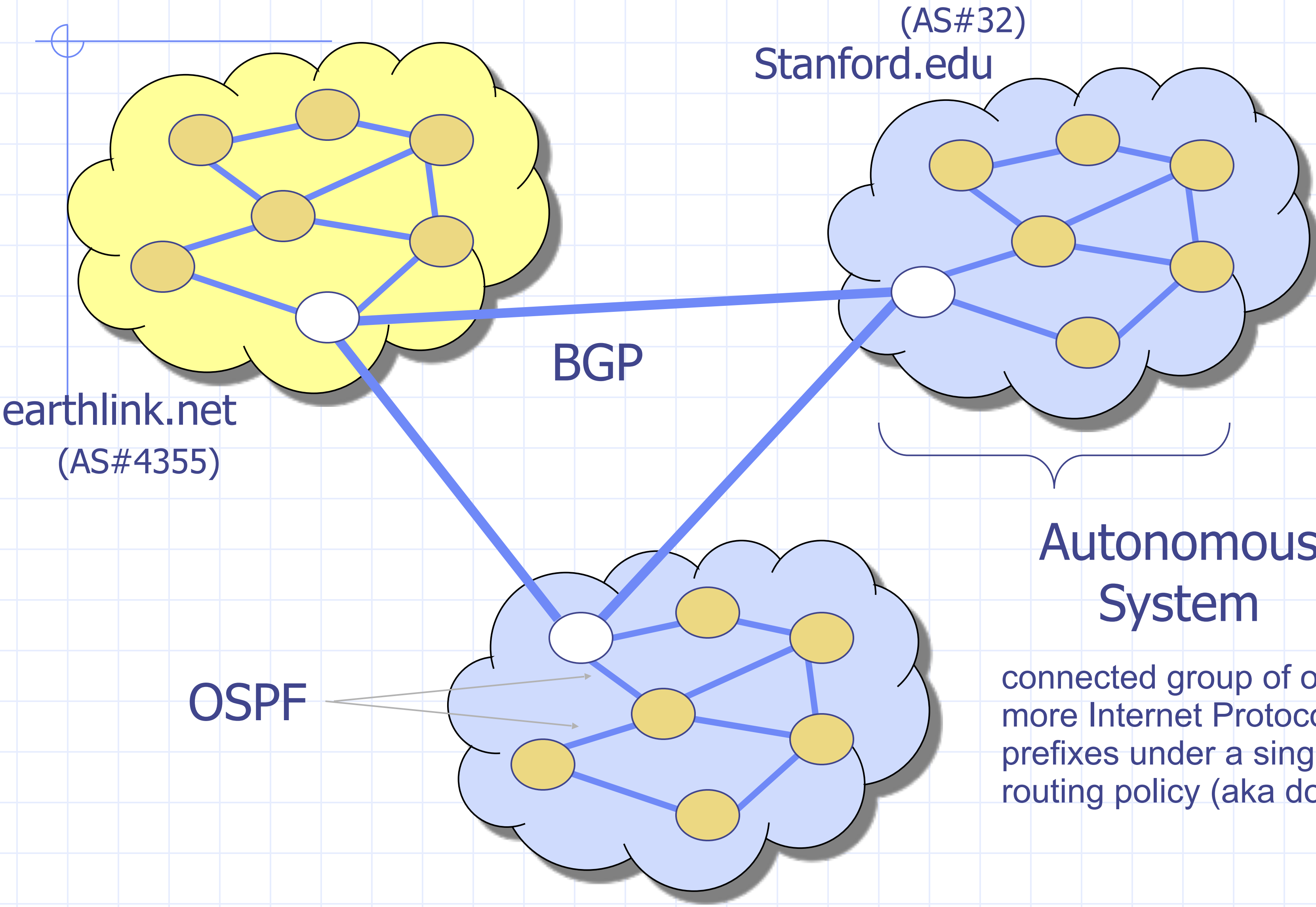
Example DoS vulnerability: Reset

- Attacker sends a Reset packet to an open socket
  - If correct SN_S then connection will close $\Rightarrow$ DoS
  - Naively, success prob. is $1/2^{32}$ (32-bit seq. #'s).
    - … but, many systems allow for a large window of acceptable seq. #'s. Much higher success probability.
  - Attacker can flood with RST packets until one works
- Most effective against long lived connections, e.g. BGP

Routing Security

ARP, OSPF, BGP

Interdomain Routing



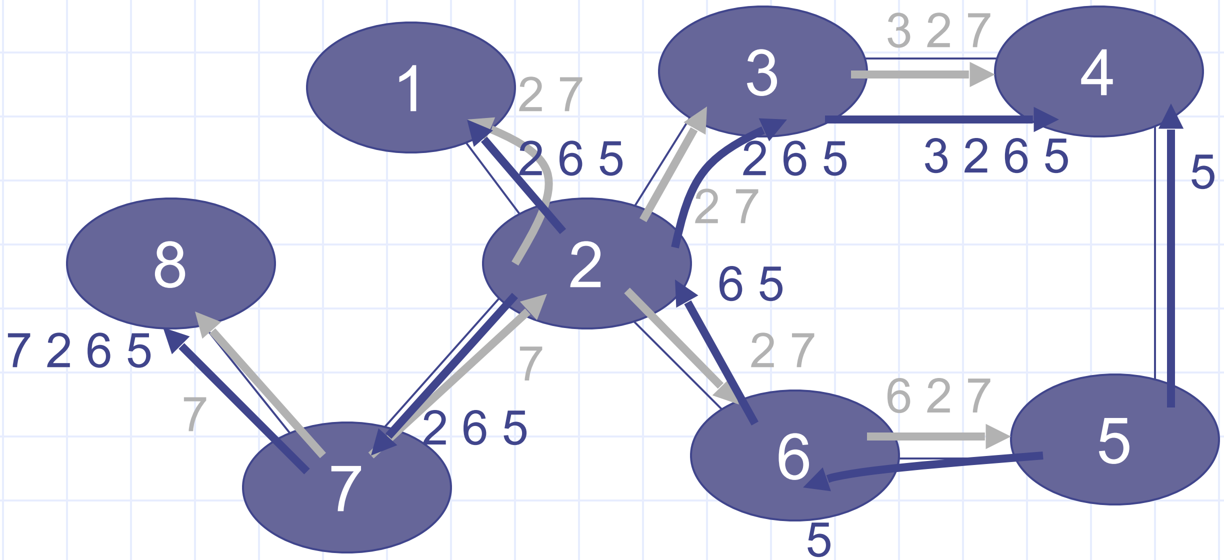
connected group of one or more Internet Protocol prefixes under a single routing policy (aka domain)

Routing Protocols

- ARP (addr resolution protocol): IP addr $\longrightarrow$ eth addr
  Security issues: (local network attacks)
  - Node A can confuse gateway into sending it traffic for Node B
  - By proxying traffic, node A can read/inject packets into B's session (e.g. WiFi networks)
- OSPF: used for routing within an AS
- BGP: routing between Autonomous Systems
  Security issues: unauthenticated route updates
  - Anyone can cause entire Internet to send traffic for a victim IP to attacker's address
    - Example: Youtube-Pakistan mishap (see DDoS lecture)
  - Anyone can hijack route to victim (next slides)

BGP example

[D. Wetherall]



Security Issues

BGP path attestations are un-authenticated

- Anyone can inject advertisements for arbitrary routes
- Advertisement will propagate everywhere
- Used for DoS, spam, and eavesdropping (details in DDoS lecture)
- Often a result of human error

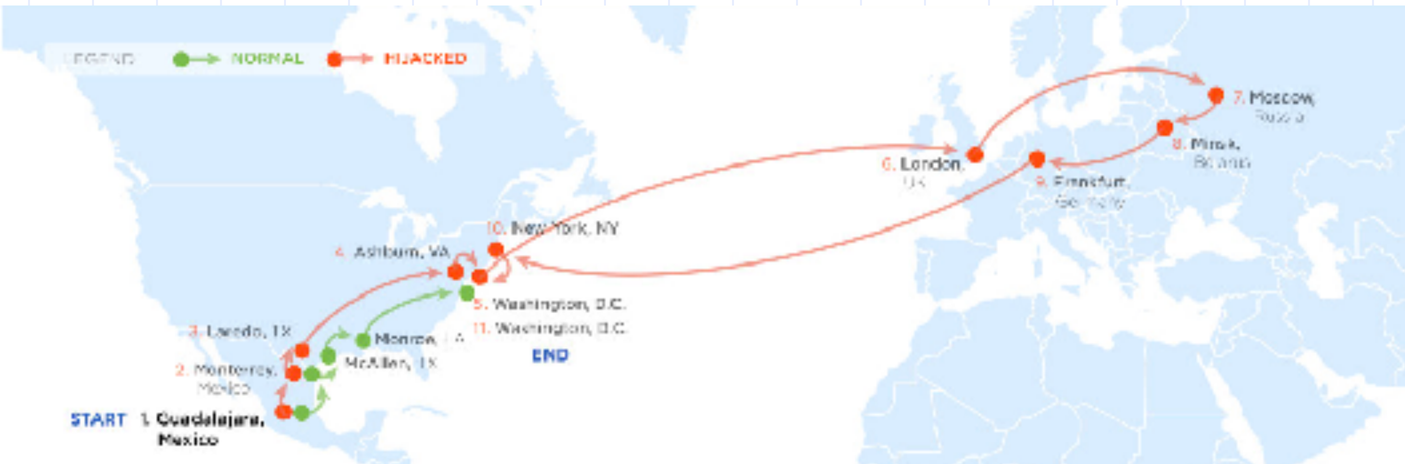
Solutions:

- RPKI: AS obtains a certificate (ROA) from regional authority (RIR) and attaches ROA to path advertisement.
  Advertisements without a valid ROA are ignored.
  Defends against a malicious AS (but not a network attacker)

- SBGP: sign every hop of a path advertisement

Example path hijack (source: Renesys 2013)

Feb 2013: Guadalajara ⟶ Washington DC via Belarus



route
in effect
for several
hours

Normally: Alestra (Mexico) ⟶ PCCW (Texas) ⟶ Qwest (DC)

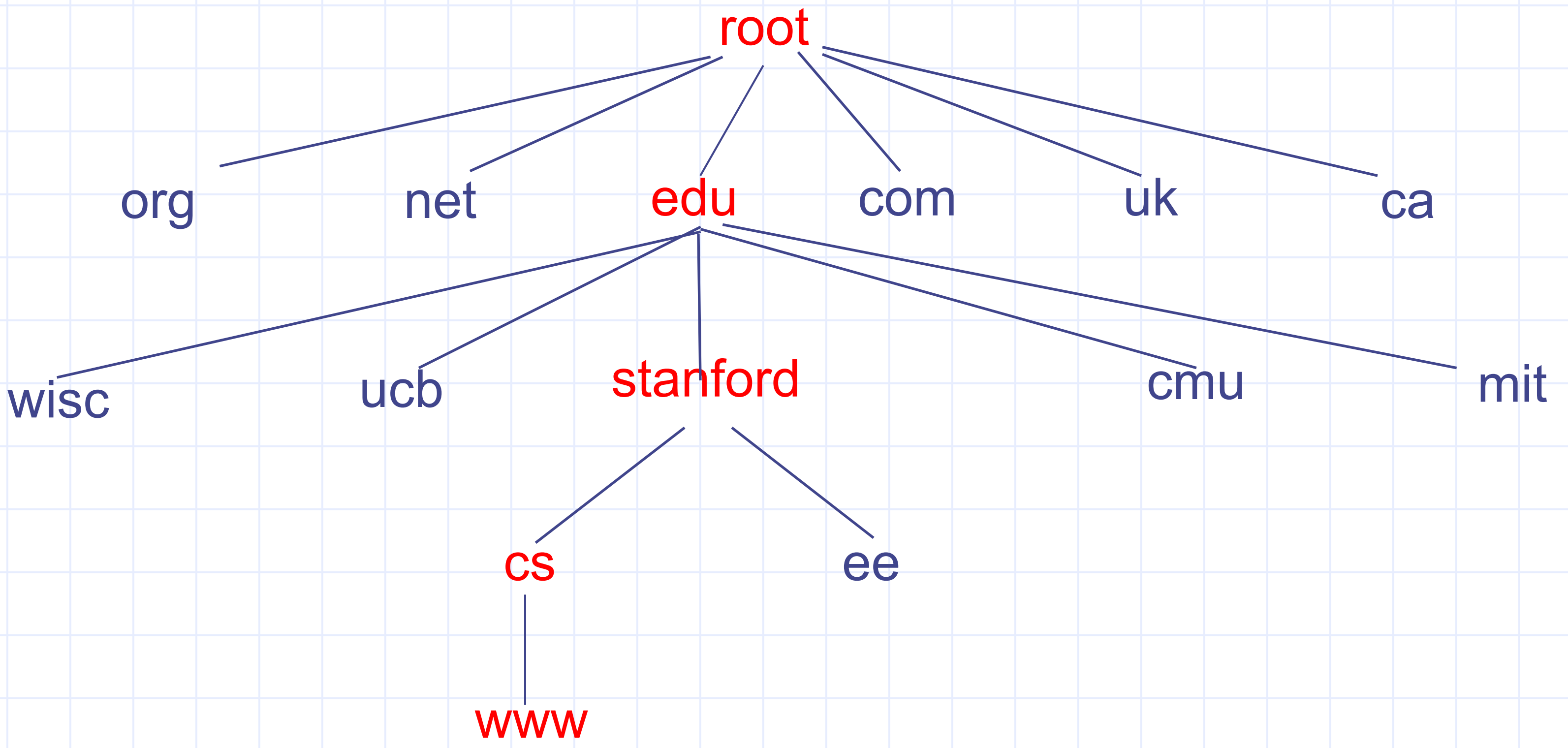
Reverse route (DC ⟶ Guadalajara) is unaffected:

- Person browsing the Web in DC cannot tell by traceroute that HTTP responses are routed through Moscow

Domain Name System

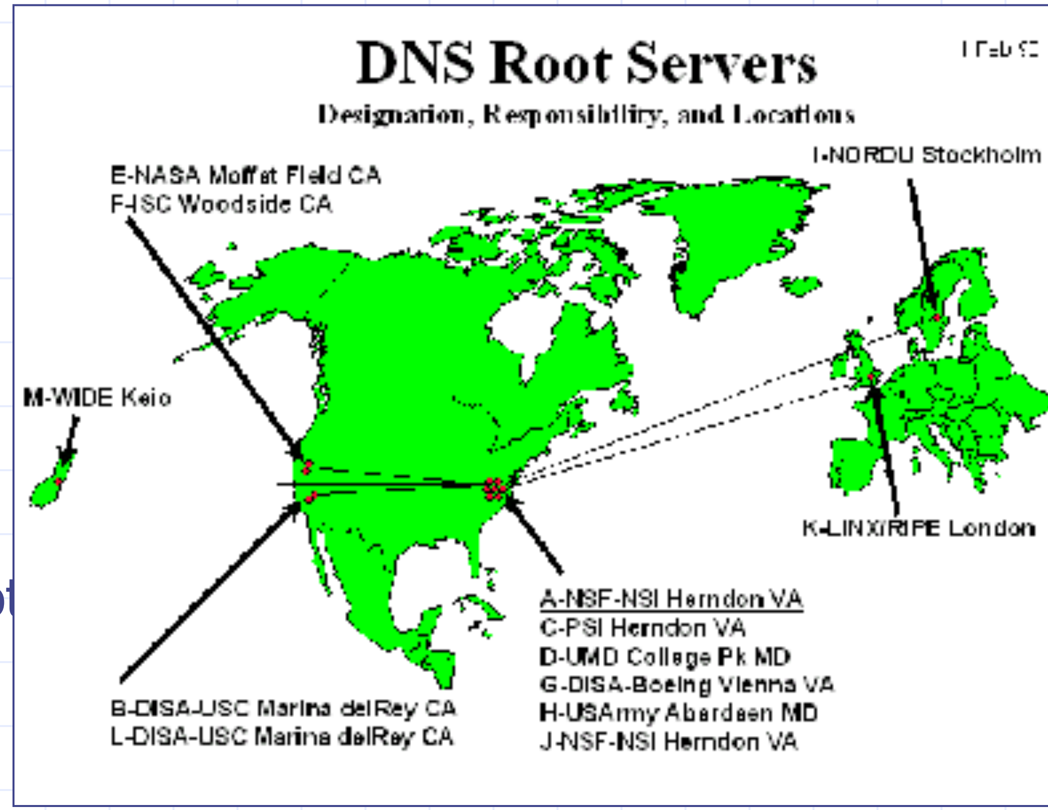
Domain Name System

- Hierarchical Name Space

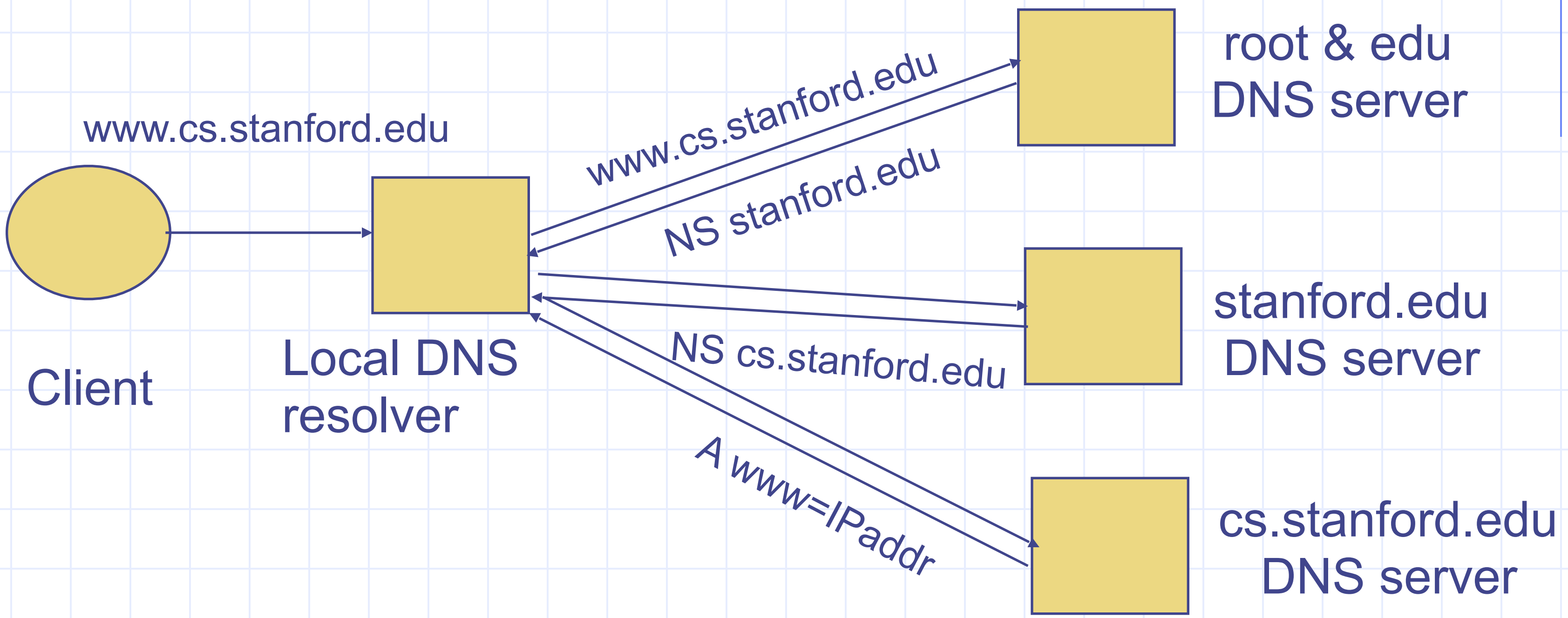


DNS Root Name Servers

- Hierarchical service
  - Root name servers for top-level domains
  - Authoritative name servers for subdomains
  - Local name resolvers contact authoritative servers when they do not know a name



DNS Lookup Example



DNS record types (partial list):

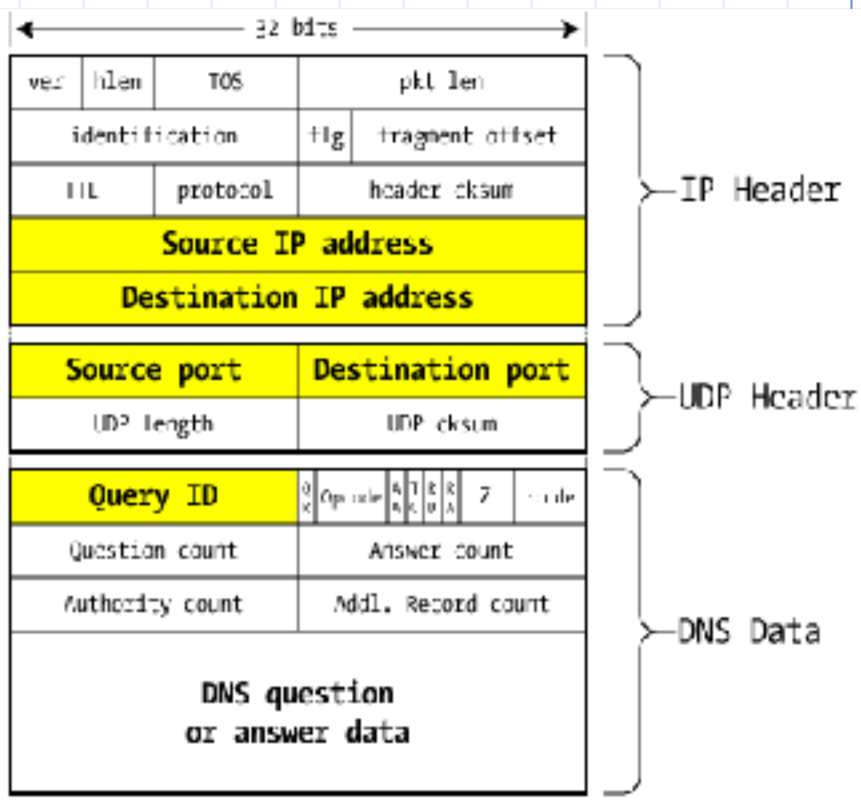
- NS: name server (points to other server)
- A: address record (contains IP address)
- MX: address in charge of handling email
- TXT: generic text (e.g. used to distribute site public keys (DKIM))

Caching

- DNS responses are cached
  - Quick response for repeated translations
  - Note: NS records for domains also cached
- DNS negative queries are cached
  - Save time for nonexistent sites, e.g. misspelling
- Cached data periodically times out
  - Lifetime (TTL) of data controlled by owner of data
  - TTL passed with every record

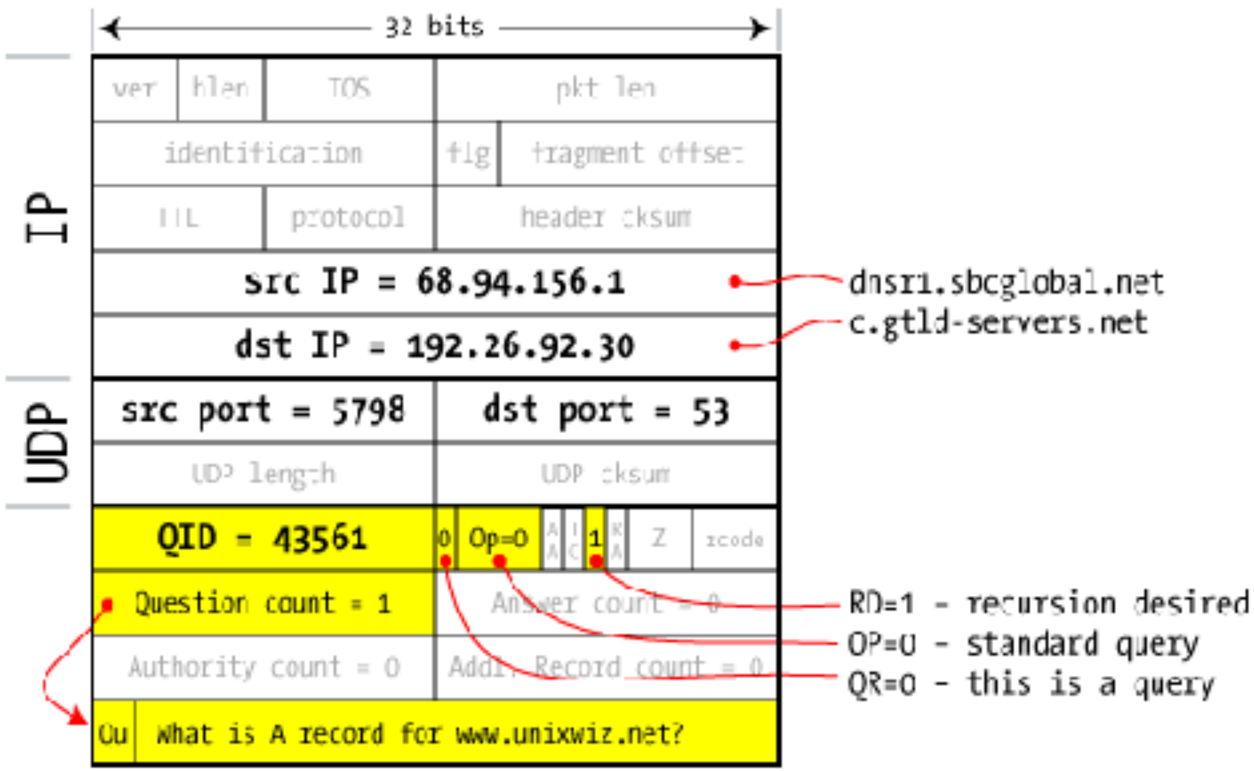
DNS Packet

- Query ID:
  - 16 bit random value
  - Links response to query



(from Steve Friedl)

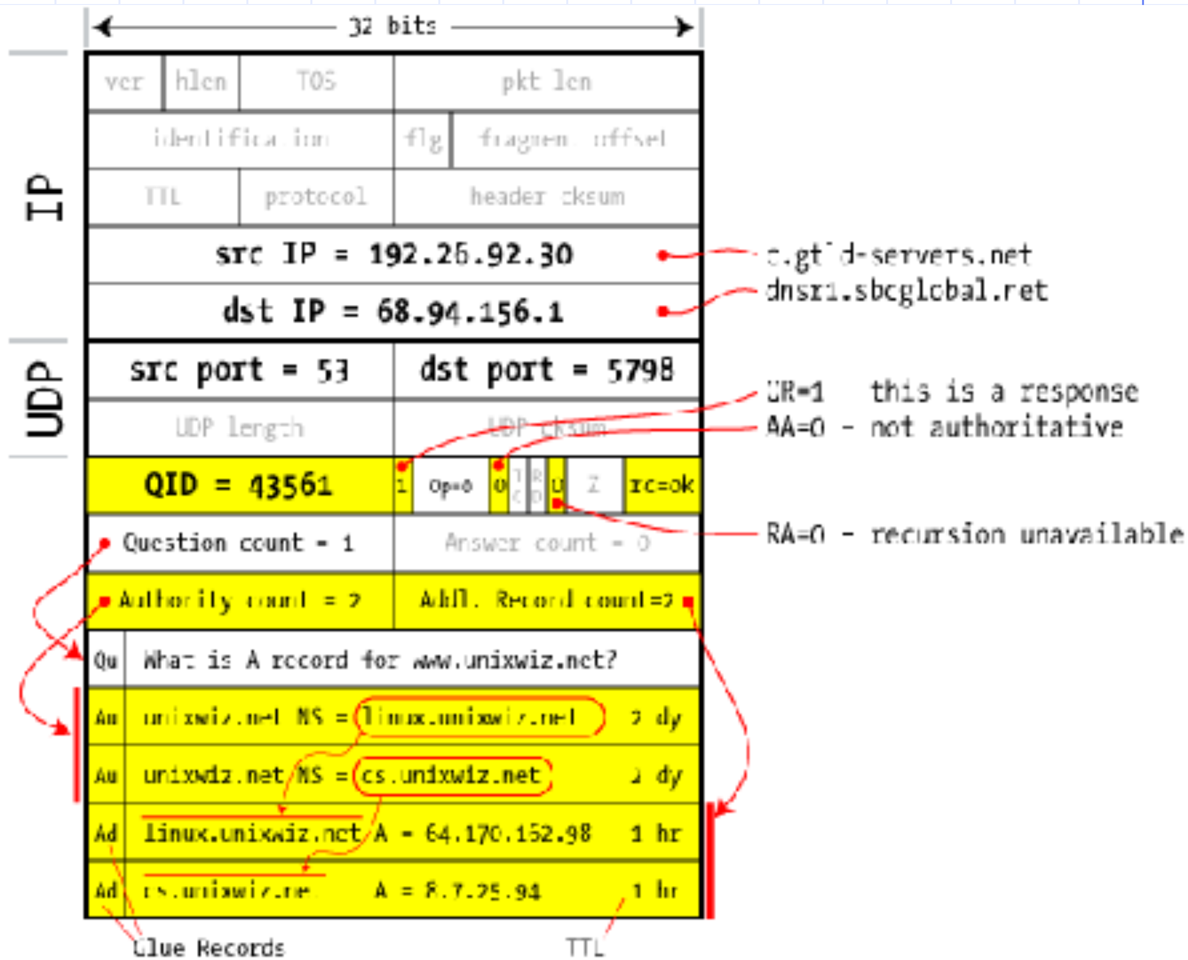
Resolver to NS request



Response to resolver

Response contains IP addr of next NS server (called "glue")

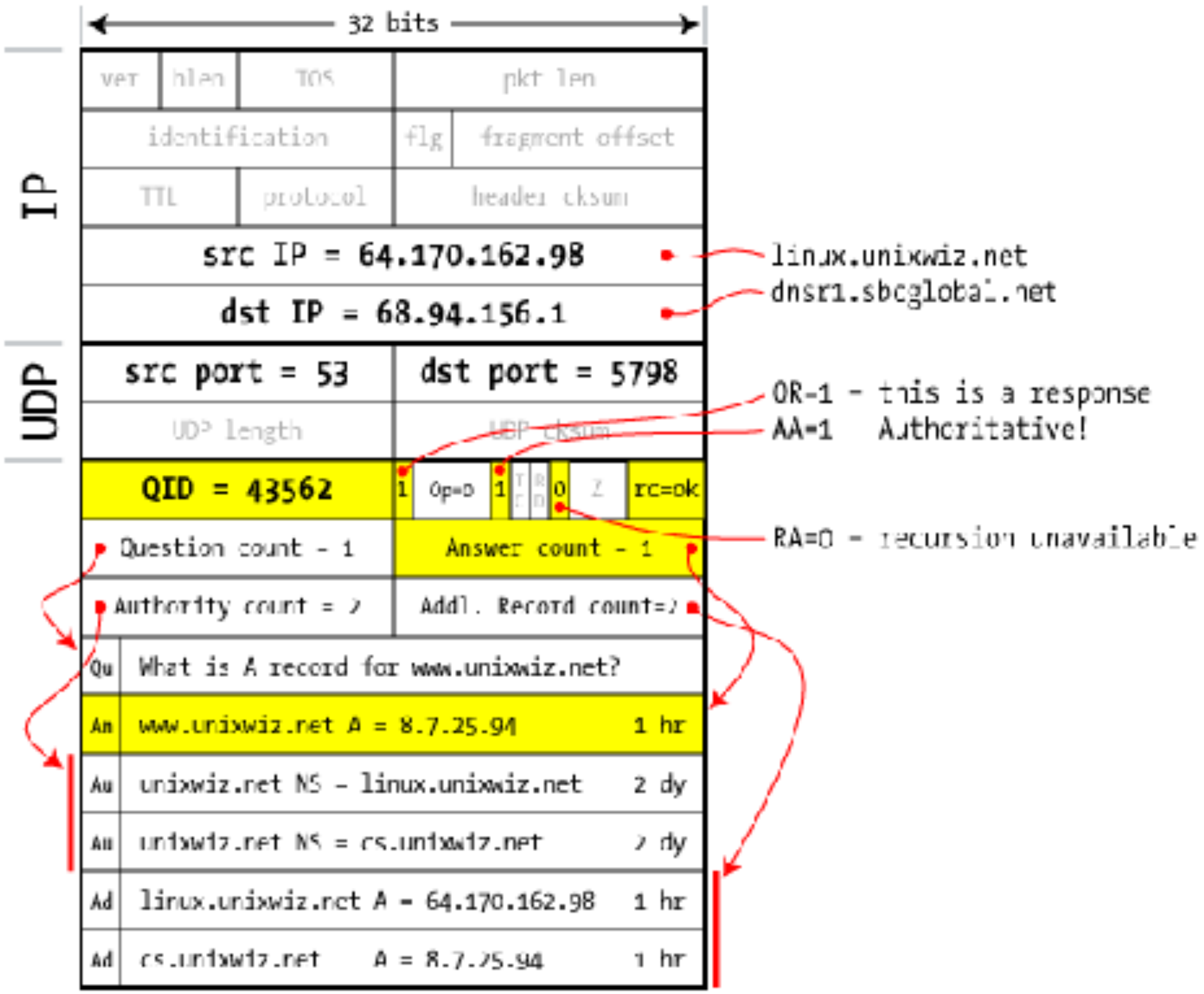
Response ignored if unrecognized QueryID



Authoritative response to resolver

<u>bailiwick checking:</u>
response is cached if it is within the same domain of query (i.e. **a.com** cannot set NS for **b.com**)

final answer



Basic DNS Vulnerabilities

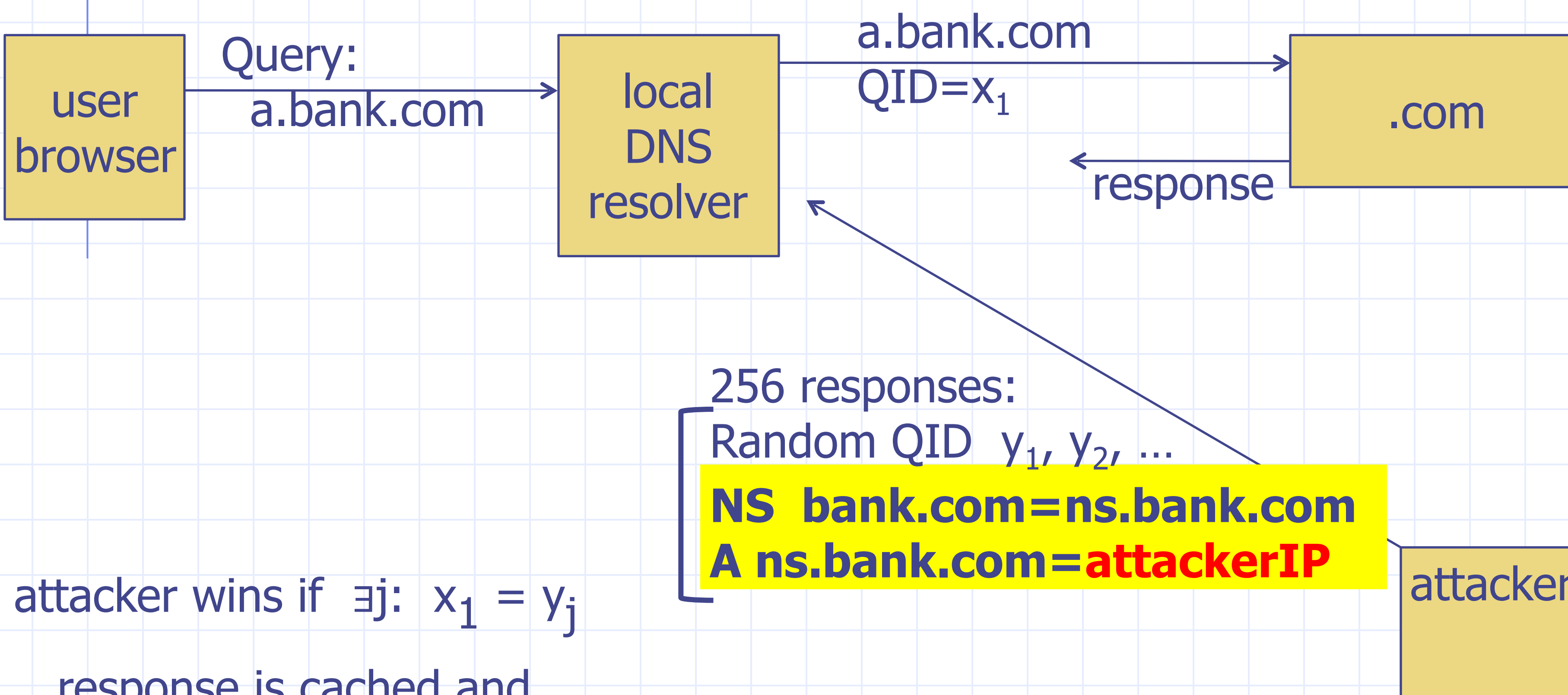
- Users/hosts trust the host-address mapping provided by DNS:
  - Used as basis for many security policies:

    Browser same origin policy,   URL address bar

- Obvious problems
  - Interception of requests or compromise of DNS servers can result in incorrect or malicious responses
    - e.g.:   malicious access point in a Cafe
  - Solution – authenticated requests/responses
    - Provided by DNSsec   …   but few use DNSsec

DNS cache poisoning (a la Kaminsky'08)

- Victim machine visits attacker's web site, downloads Javascript

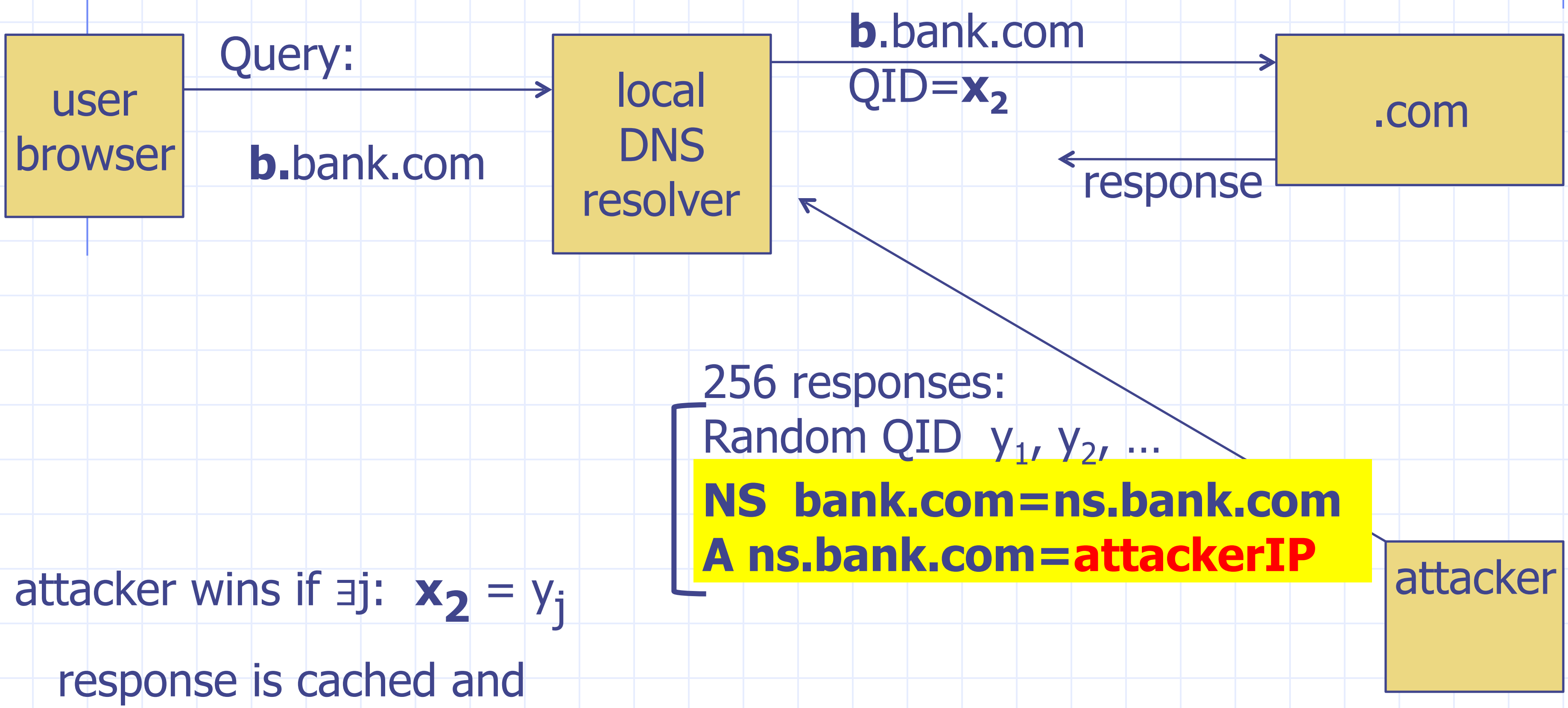


attacker wins if $\exists j$: $x_1 = y_j$

response is cached and attacker owns bank.com

If at first you don't succeed …

- Victim machine visits attacker's web site, downloads Javascript



attacker wins if ∃j: $\mathbf{x_2} = y_j$

response is cached and
attacker owns bank.com

success after ≈ 256 tries (few minutes)

Defenses

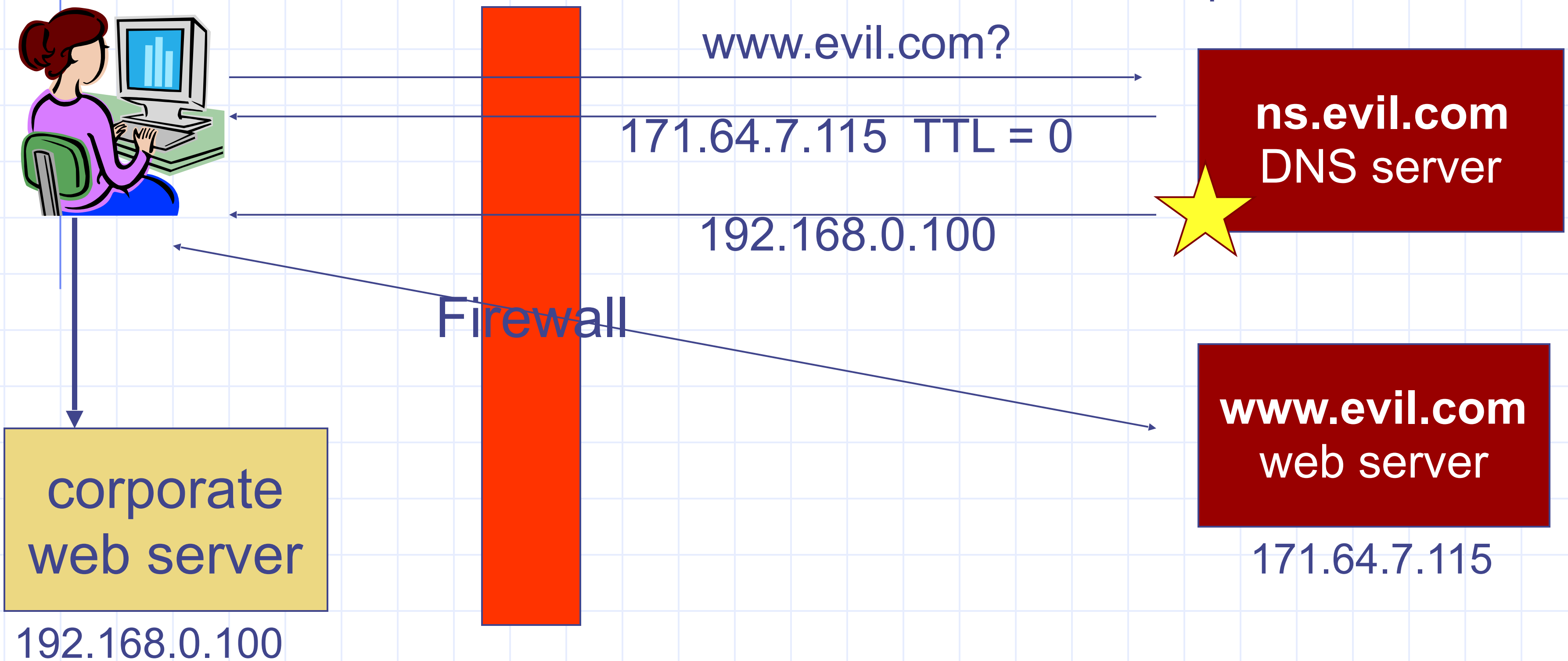
- Increase Query ID size.   How?
- Randomize src port, additional 11 bits
    - Now attack takes several hours
- Ask every DNS query twice:
    - Attacker has to guess QueryID correctly twice (32 bits)
    - … but Apparently DNS system cannot handle the load

[DWF'96, R'01]

DNS Rebinding Attack

<iframe src="**http://www.evil.com**">

DNS-SEC cannot stop this attack



Read permitted: it's the "same origin"

DNS Rebinding Defenses

- Browser mitigation: DNS Pinning
  - Refuse to switch to a new IP
  - Interacts poorly with proxies, VPN, dynamic DNS, …
  - Not consistently implemented in any browser
- Server-side defenses
  - Check Host header for unrecognized domains
  - Authenticate users with something other than IP
- Firewall defenses
  - External names can’t resolve to internal addresses
  - Protects browsers inside the organization

Summary

- Core protocols not designed for security
  - Eavesdropping, Packet injection, Route stealing, DNS poisoning
  - Patched over time to prevent basic attacks

    (e.g. random TCP SN)

- More secure variants exist (next lecture) :

  IP $\longrightarrow$ IPsec

  DNS $\longrightarrow$ DNSsec

  BGP $\longrightarrow$ SBGP