CSE 5095 & ECE 6095 – Spring 2016 – Instructor Marten van Dijk

Lecture # 1

Network Security

Saeed Valizadeh
Department of Computer Science and Engineering
University of Connecticut
Email: Valizadeh.mh@engr.uconn.edu

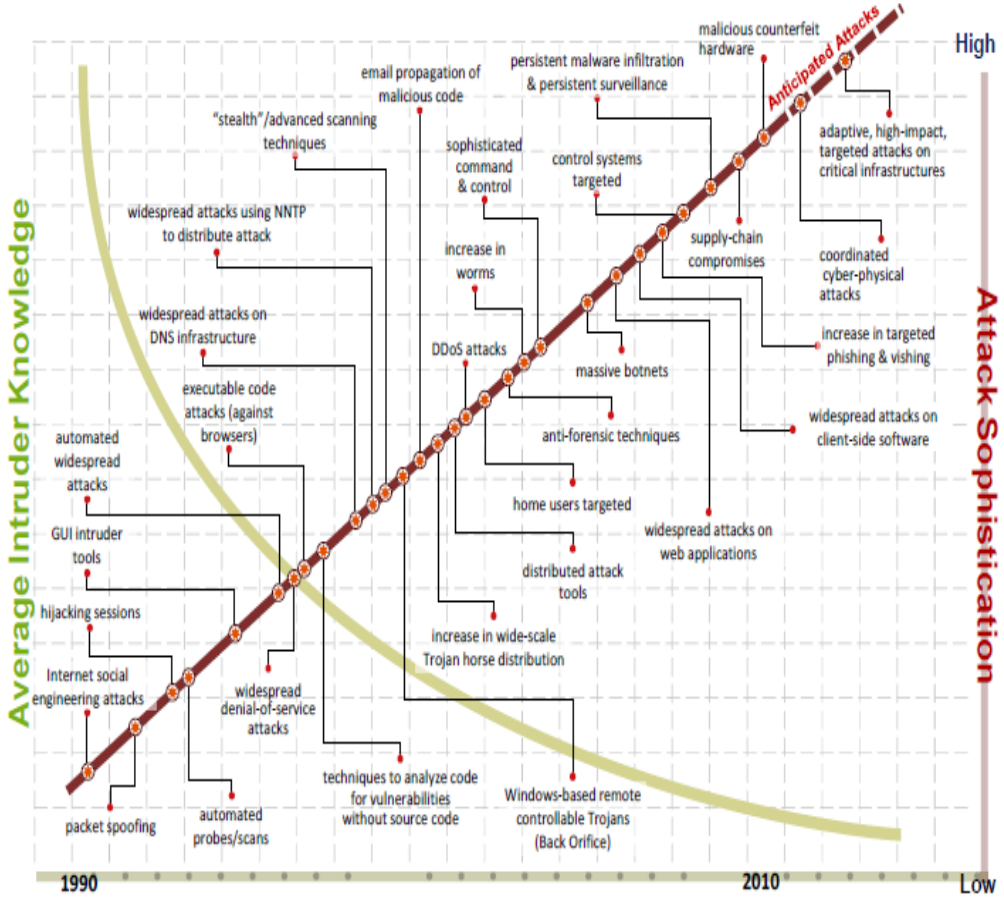
With help from Marten van Dijk

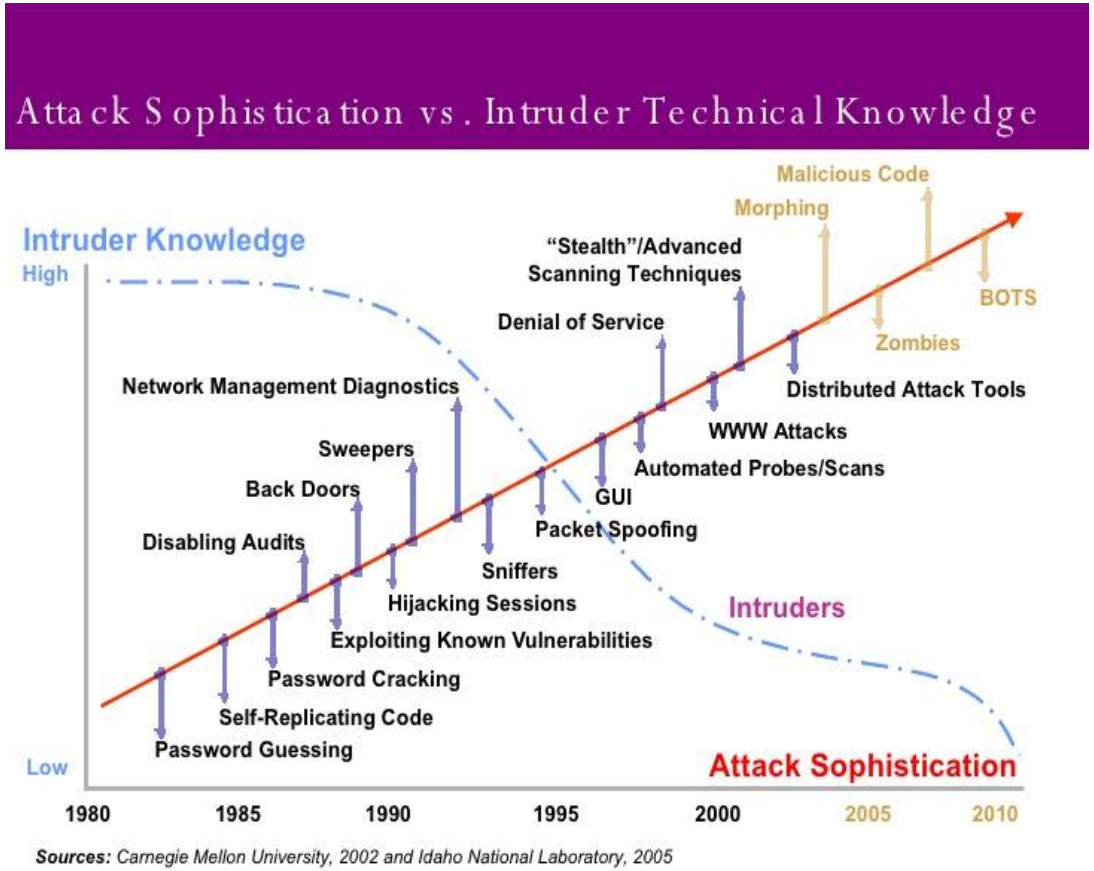
UCONN

Outline

- **Motivation**
- **Part 1** Background: Security Objectives, Basics and Definitions
- **Part 2**
  - 2-1 Internet Security: How the Internet works and some basic vulnerabilities
  - 2-2 Routing Security
  - 2-3 Domain Name System(DNS)
- **Part 3**
  - 3-1 Network Protocol Security
  - 3-2 Standard Network Defenses
    - 3-2-1 Firewalls
    - 3-2-2 Intrusion Detection Systems
- **Part 4**
  - 4-1 Unwanted Traffic: Denial of Service Attacks
  - 4-2 DOS Mitigation
  - 4-3 Dos Defense mechanisms

Motivation





Sources: Carnegie Mellon University, 2002 and Idaho National Laboratory, 2005

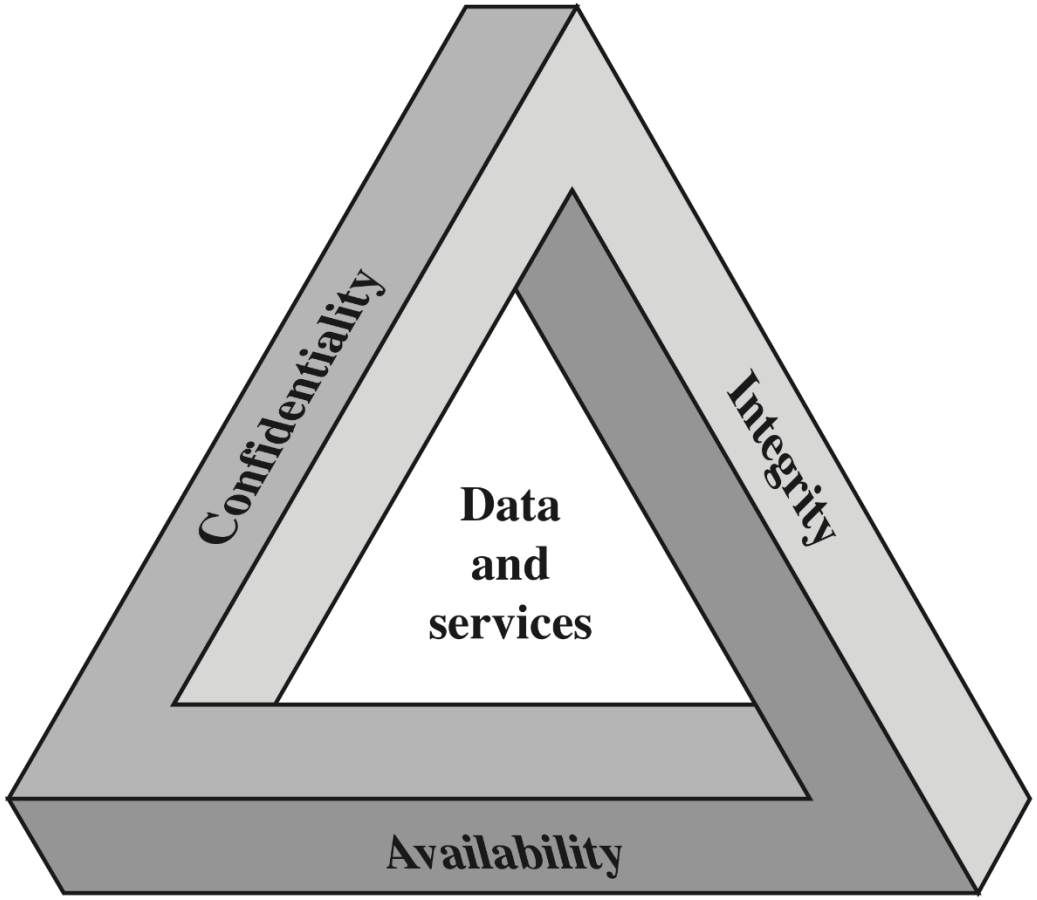
Part 1

Background:

- Security Objectives(CIA, CIA^{+})
- Security Basics
- Basic Definitions

Mostly based on NetSec course at IUST and Some of course from the Internet ☺

Security Objectives/Requirements(CIA Model)



CIA^{+}(CIA + Legitimate Use)

Authenticity

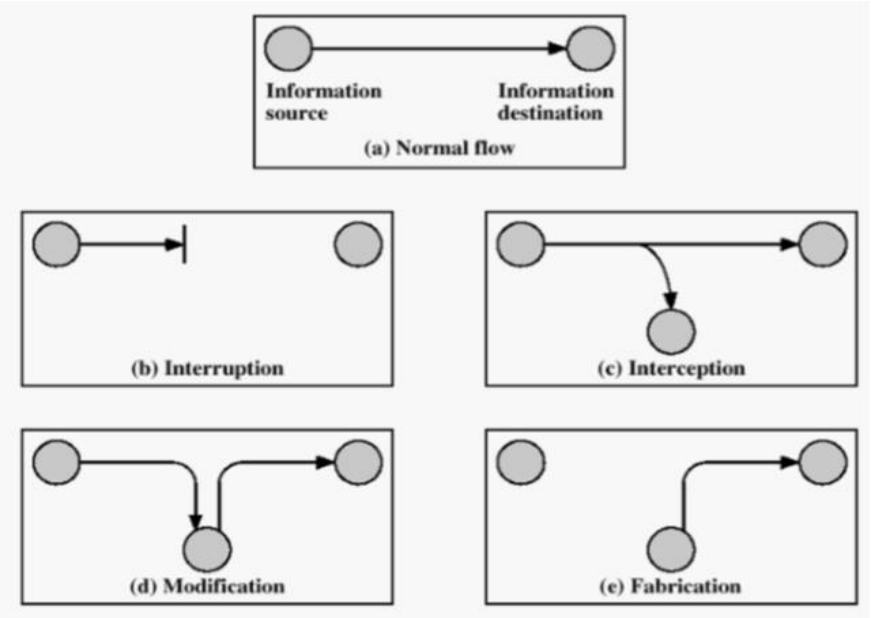
- Verifying that users are who they say they are and that each input arriving at the system came from a trusted source

Accountability

- The security goal that generates the requirement for actions of an entity to be traced uniquely to that entity

General Attacks…

- **Interception:** This is an attack on confidentiality/integrity
- **Modification:** This is an attack on integrity
- **Interruption:** This is an attack on availability
- **Fabrication:** caused by lack of authenticity and ….
- + There are some other important issues:
  - Authenticity, Accountability, Authorization, ….



Security Basics : Vulnerability, Threat and Attack

- **Vulnerability:**
  - A weakness that makes targets susceptible to an attack.

- **Threat:**
  - The expressed potential for the occurrence of a harmful event such as an attack.

- **Attack:**
  - An action taken against a target with the intention of doing harm.

Ref: https://peterhgregory.wordpress.com/2009/03/14/security-basics-definitions-of-threat-attack-and-vulnerability/

Basic Definitions-Policy & Mechanisms

- **Security policy** is a definition of what it means to *be secure* for a system, organization or other entity. For systems, the security policy addresses constraints on functions and flow among them, constraints on access by external systems and adversaries including programs and access to data by people.

- A ***Security Mechanism*** is a method, tool, or procedure for enforcing a security policy.
  - Encipherment, Digital Signature, Access Control, Data Integrity, Authentication Exchange, Traffic padding, Routing Control, Notarization

https://en.wikipedia.org/wiki/Security_policy
http://www.informit.com/articles/article.aspx?p=363728&seqNum=3

Basic Definitions- Security Services

- Defined by:
  - **X.800 and ISO 7498-2**: Security service is a service, provided by a layer of communicating open systems, which ensures adequate security of the systems or of data transfers as defined by ITU-T X.800 Recommendation.
  - **CNSS Instruction No. 4009**: *A capability that supports one, or more, of the security requirements (Confidentiality, Integrity, Availability). Examples of security services are key management, access control, and authentication.*
  - **W3C Web service Glossary**: *A processing or communication service that is provided by a system to give a specific kind of protection to resources, where said resources may reside with said system or reside with other systems, for example, an authentication service or a PKI-based document attribution and authentication service. A security service is a superset of AAA services. Security services typically implement portions of security policies and are implemented via security mechanisms.*

https://en.wikipedia.org/wiki/Security_service_(telecommunication)

Table 1 /X.800

Service	Mechanism							
	Encipherment	Digital signature	Access control	Data integrity	Authentication exchange	Traffic padding	Routing control	Notarization
Peer entity authentication	Y	Y	·	·	Y	·	·	·
Data origin authentication	Y	Y	·	·	·	·	·	·
Access control service	·	·	Y	·	·	·	·	·
Connection confidentiality								
	Y	.	·	·	·	·	Y	·
Connectionless confidentiality	Y	·	·	·	·	·	Y	·
Selective field confidentiality	Y	·	·	·	·	·	·	·
Traffic flow confidentiality	Y	·	·	·	·	Y	Y	·
Connection Integrity with recovery	Y	·	·	Y	·	·	·	·
Connection integritywithout recovery	Y	·	·	Y	·	·	·	·
Selective field connection integrity	Y	·	·	Y	·	·	·	·
Connectionless integrity	Y	Y	·	Y	·	·	·	·
Selective field connectionless integrity	Y	Y	·	Y	·	·	·	·
Non-repudiation. Origin	·	Y	·	Y	·	·	·	Y
Non-repudiation. Delivery		Y	·	Y	·	·	·	Y

Illustration of relationship of security services and mechanisms

Table 2/X.800

What do we want?
To implement a Security Service
Through Security Mechanisms
In a special layer

Service	Layer						
	1	2	3	4	5	6	7*
Peer entity authentication	·	·	Y	Y	·	·	Y
Data origin authentication	·	·	Y	Y	·	·	Y
Access control service	·	·	Y	Y	·	·	Y
Connection confidentiality	Y	Y	Y	Y	·	Y	Y
Connectionless confidentiality	·	Y	Y	Y	·	Y	Y
Selective field confidentiality	·	·	·	·	·	Y	Y
Traffic flow confidentiality	Y	·	Y	·	·	·	Y
Connection Integrity with recovery	·	·	·	Y	·	·	Y
Connection integrity without recovery	·	·	Y	Y	·	·	Y
Selective field connection integrity	·	·	·	·	·	·	Y
Connectionless integrity	·	·	Y	Y	·	·	Y
Selective field connectionless integrity	·	·	·	·	·	·	Y
Non-repudiation Origin	·	·	·	·	·	·	Y
Non-repudiation. Delivery	·	·	·	·	·	·	Y

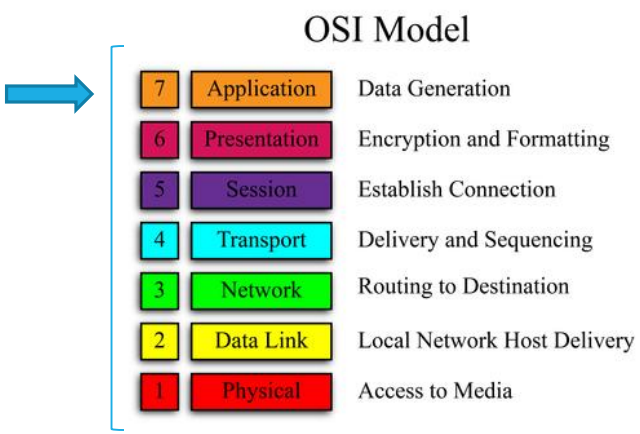


Illustration of the relationship of security services and layers

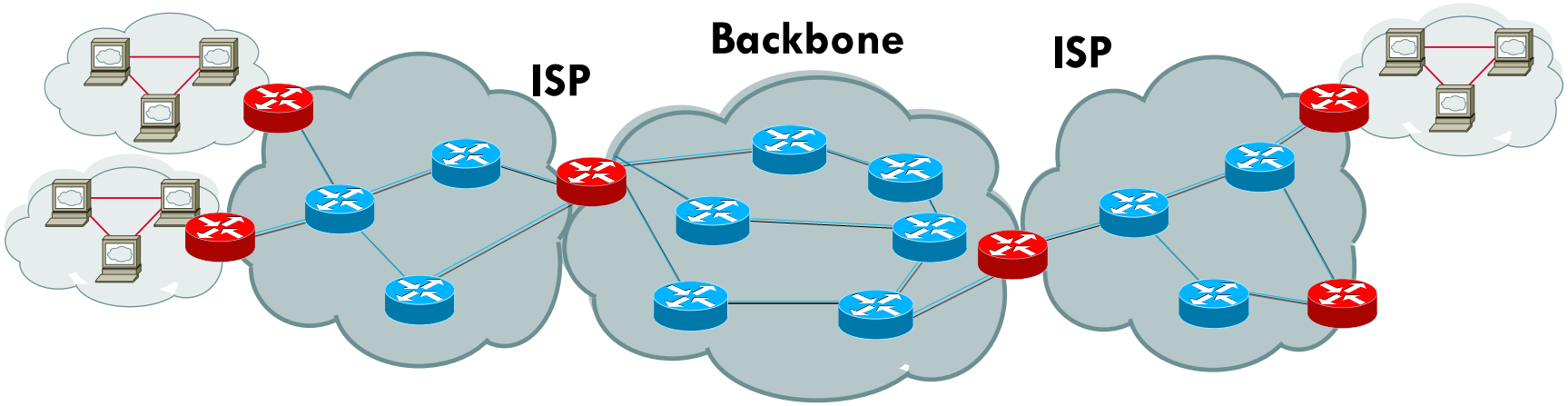
Securing information and communications systems: principles, technologies, and applications Steven Furnell, Sokratis Katsikas, Javier Lopez, Artech House, 2008 - 362 pages

Part 2-1

Internet Security:
How the Internet works and some basic vulnerabilities

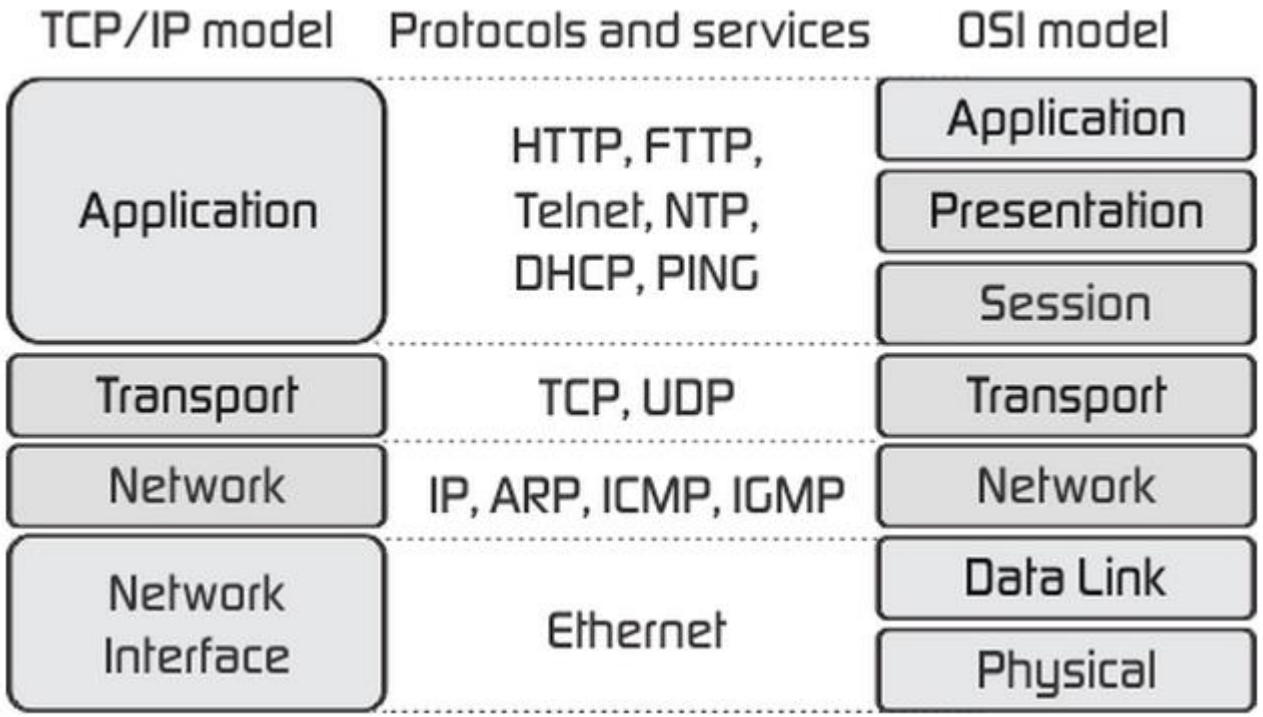
Mostly Based on and extracted from Dan Boneh Lecures on Computer and Network Security, course material at https://crypto.stanford.edu/cs155/
+ of course with help of internet ☺

Internet Infrastructure

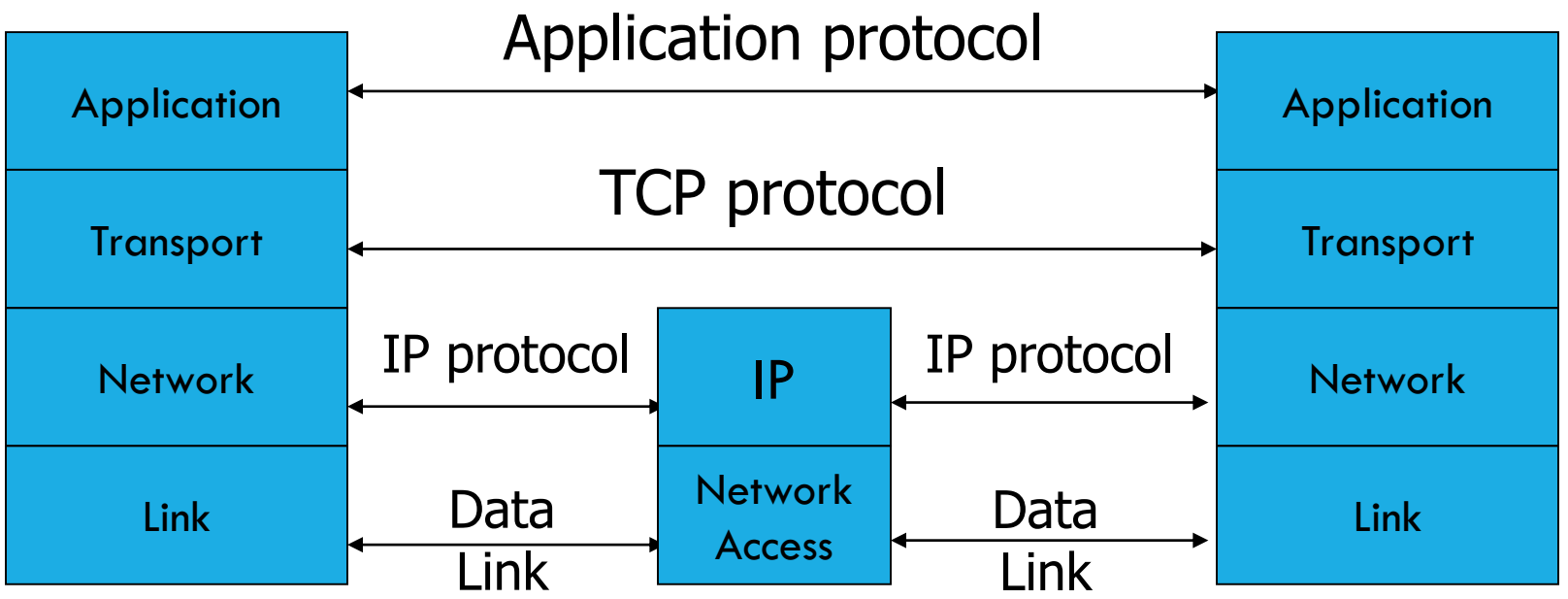


- Local and Inter-domain routing
  - TCP/IP for routing and messaging
  - BGP for routing announcements
- Domain Name System(DNS)
  - Find IP address from symbolic name (www.uconn.edu)

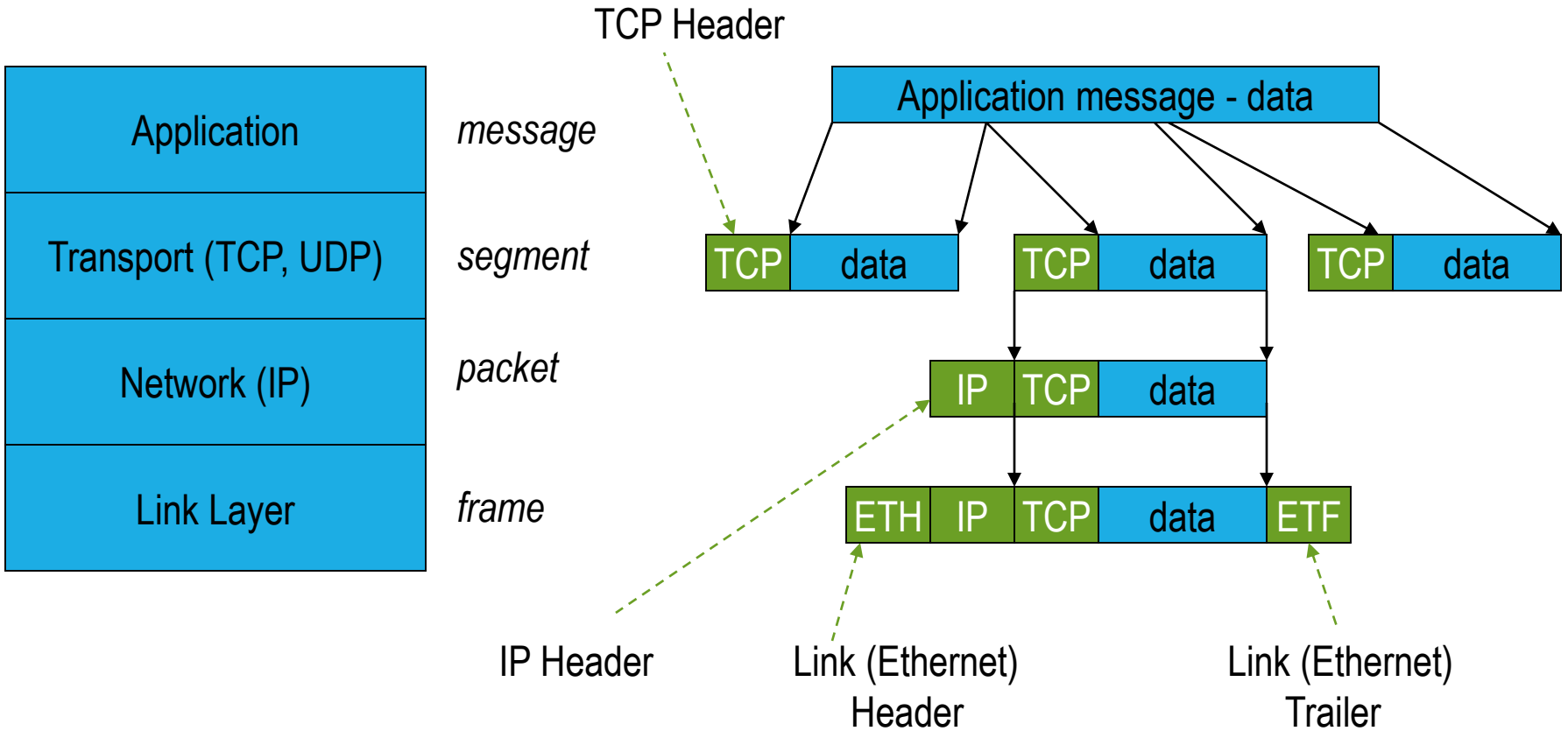
Network Model



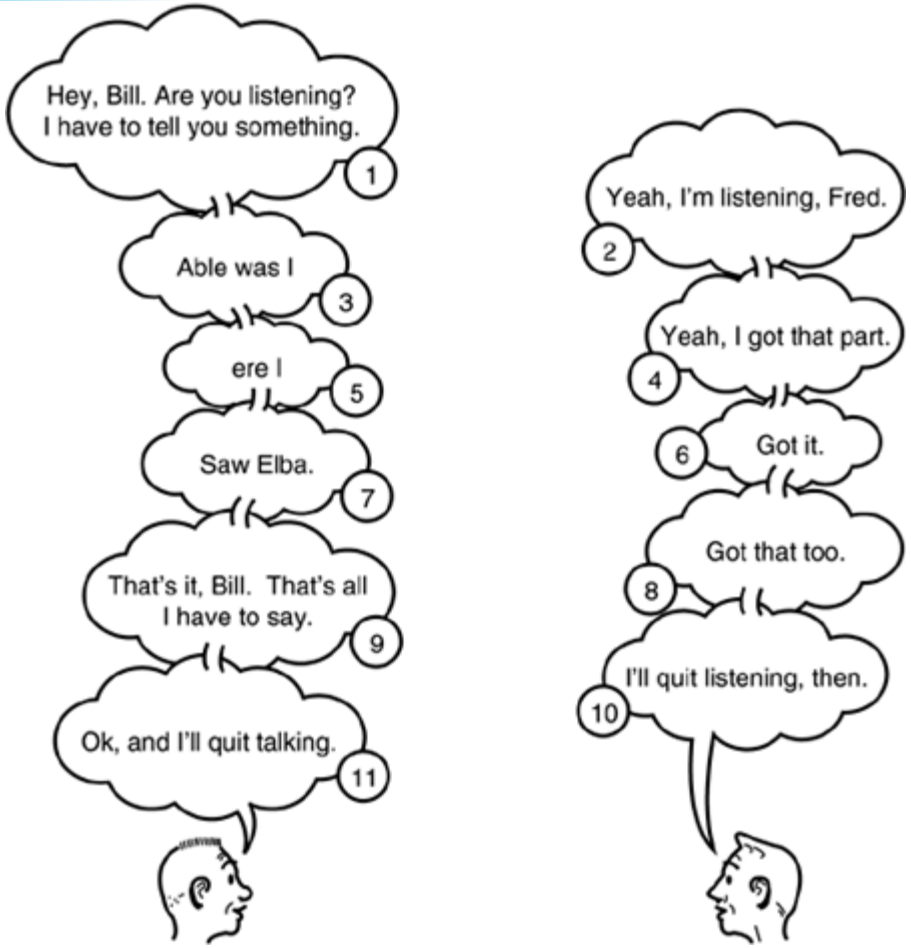
TCP/IP Protocol Stack



Data Formats



Connectionless vs. Connection oriented Protocols:



A connection-oriented protocol

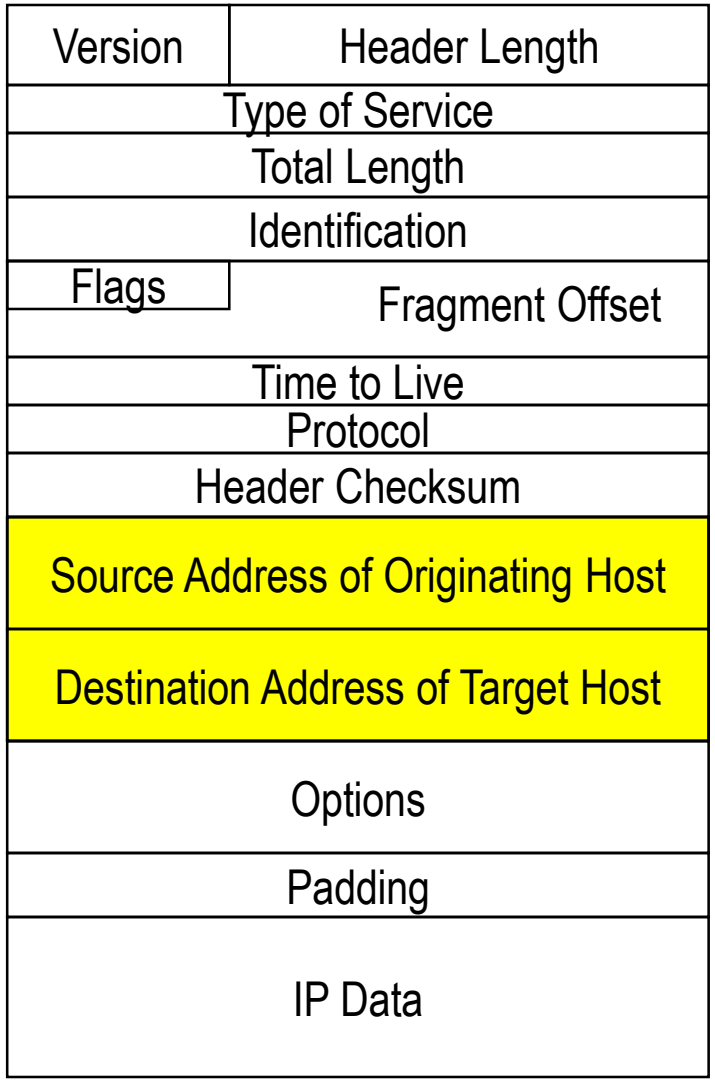
A connectionless protocol

http://www.yaldex.com/tcp_ip/0672325659_ch06lev1sec2.html

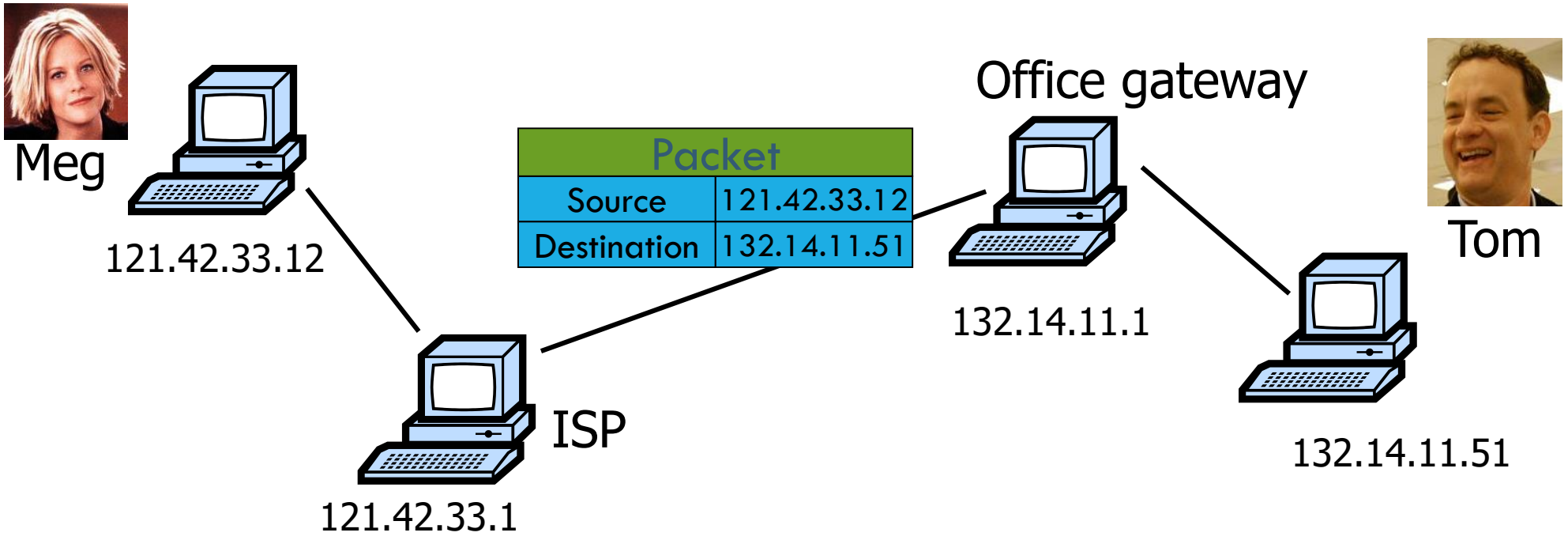
Internet Protocol(IP)

- Connectionless
  - Unreliable
  - Best effort

- Notes:
  - Src. and dest. **ports** not parts of IP hdr



IP Routing/ The Sleepless in Seattle



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

IP Protocol Functions (Summary)

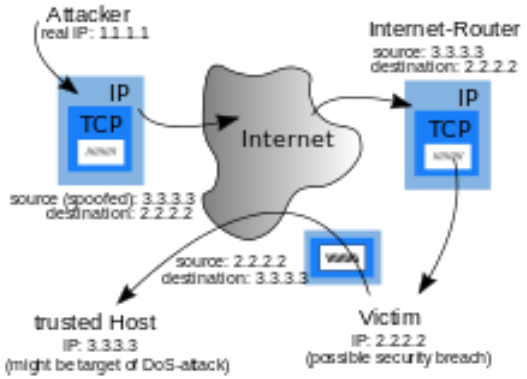
- Routing
  - IP host knows location of router (gateway)
  - IP gateway must know route to other networks
- Fragmentation and reassembly
  - If max-packet-size less than the user-data-size
- Error reporting
  - ICMP packet to source if packet is dropped
- TTL field: decremented after every hop
  - Packet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

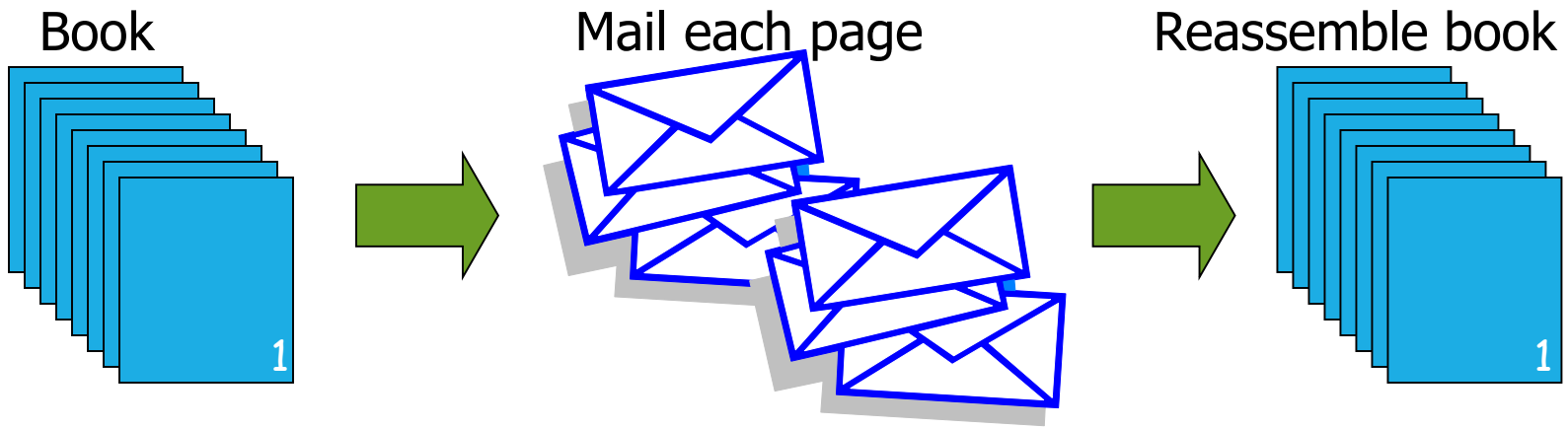
- Implications: (solutions in DDoS lecture[Upper layers: SN])
  - Anonymous DoS attacks;
  - Anonymous infection attacks (e.g. slammer worm)



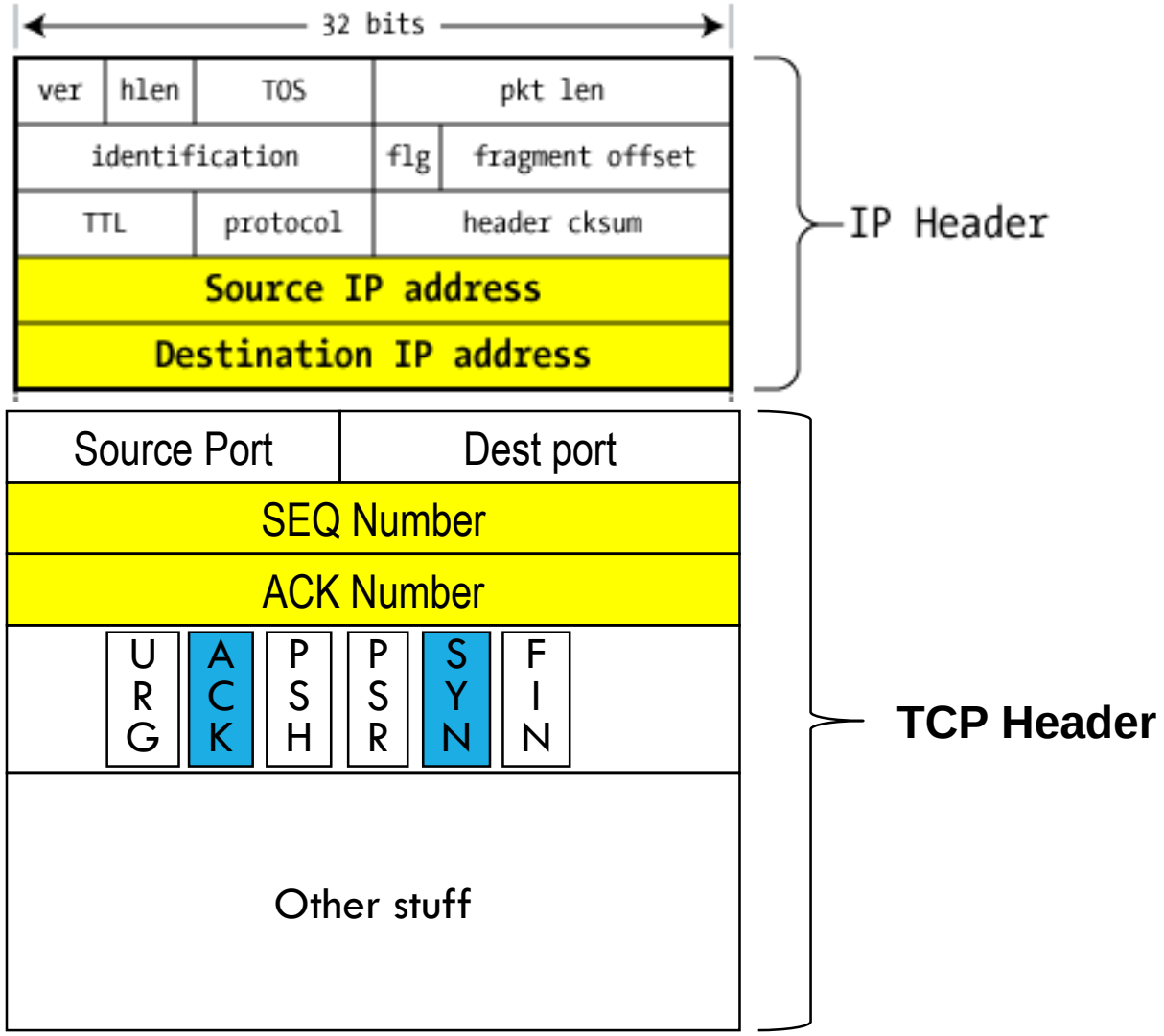
example scenario of IP address spoofing

Transmission Control Protocol(TCP)

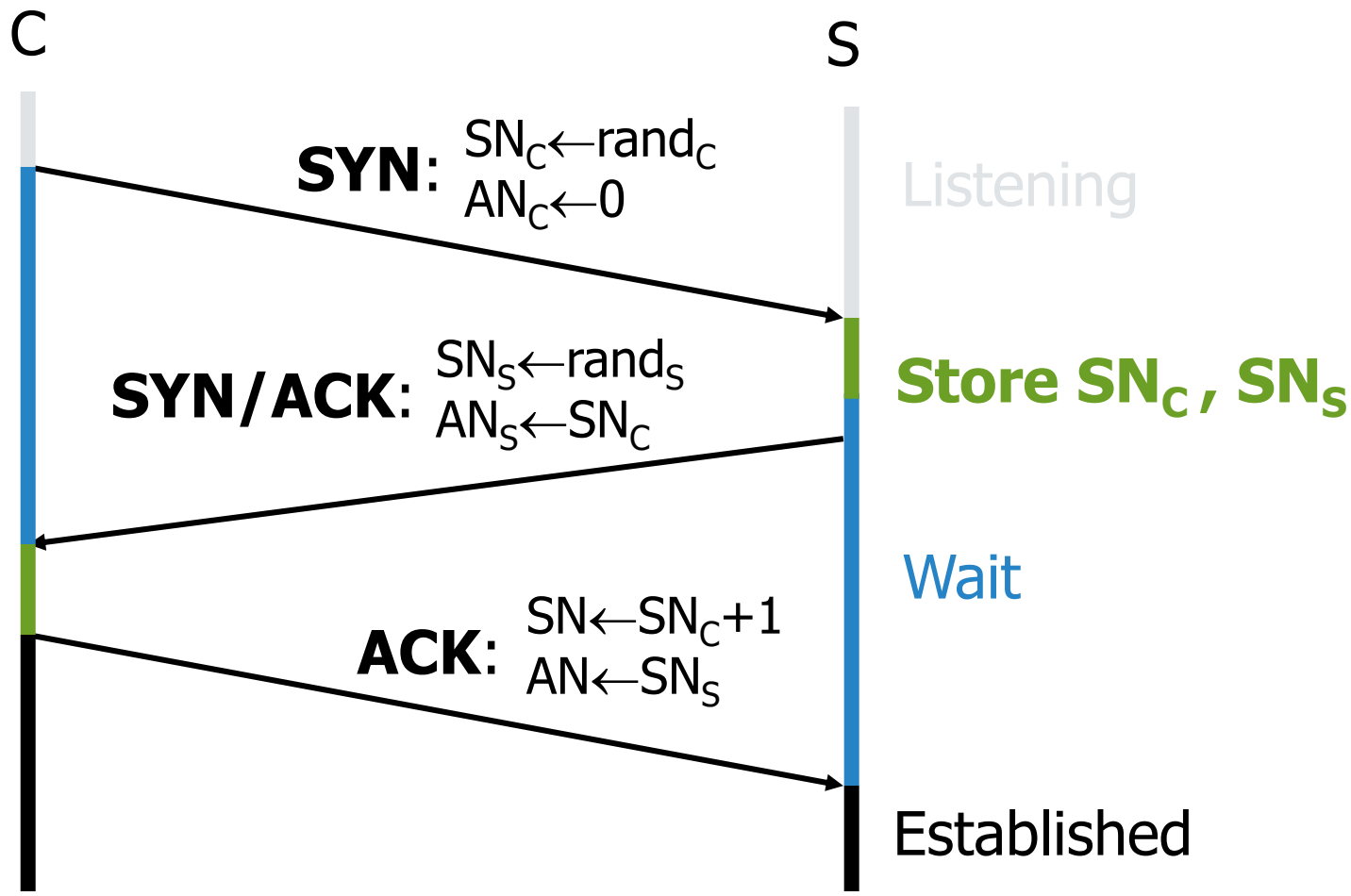
- Connection-oriented, preserves order
  - Sender
    - Break data into packets
    - Attach packet numbers
  - Receiver
    - Acknowledge receipt; lost packets are resent
    - Reassemble packets in correct order



TCP Header


32 bits
ver
hlen
TOS
pkt len
identification
flg
fragment offset
TTL
protocol
header cksum
Source IP address
Destination IP address
IP Header
Source Port
Dest port
SEQ Number
ACK Number
URG
ACK
PSH
PSR
SYN
FIN
Other stuff
TCP Header

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)

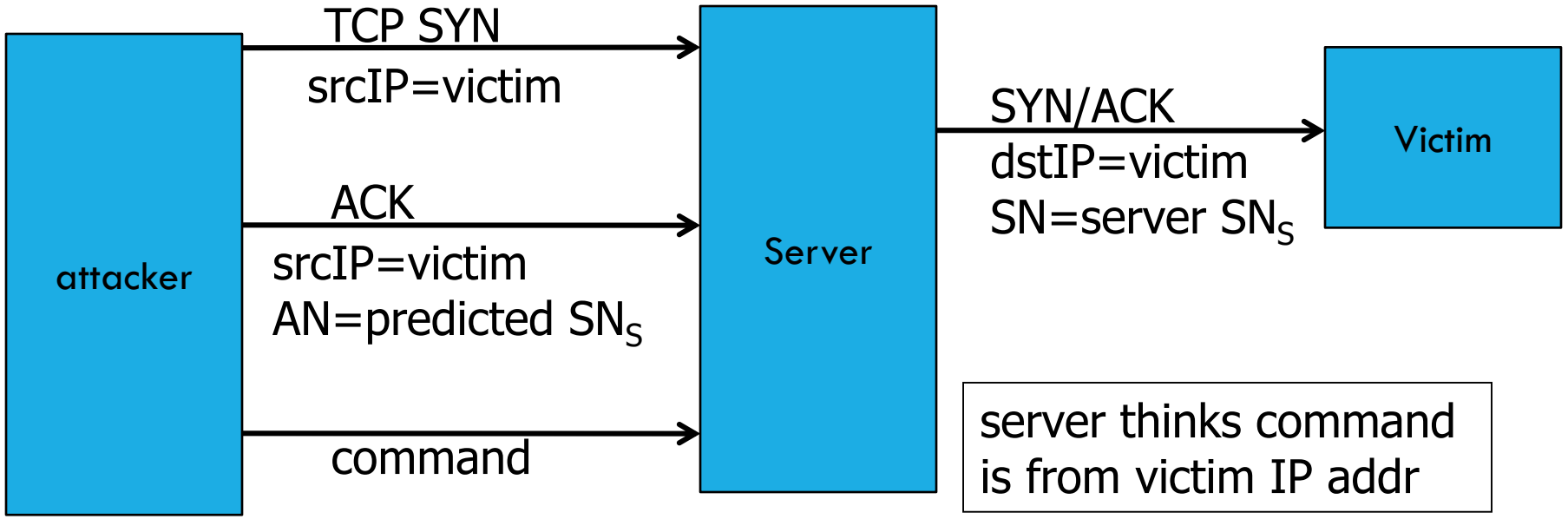
2. TCP state easily obtained by eavesdropping
   - Enables spoofing and session hijacking

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. does not block attack, but makes it harder



Example DoS vulnerability: Reset attack

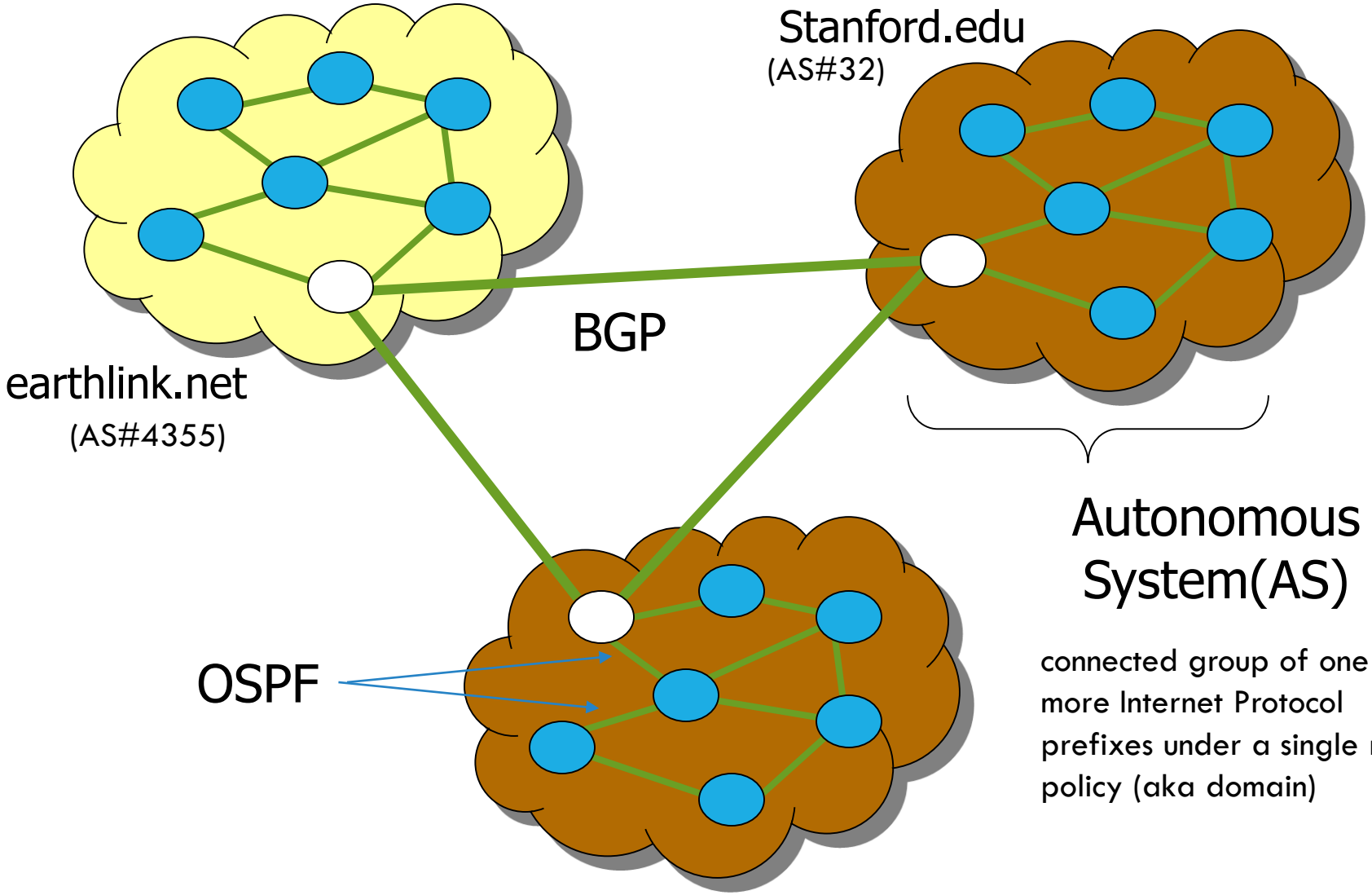
- Attacker sends a Reset packet to an open socket
  - If correct SN_S then connection will close ⇒ DoS
  - Naively, success prob. is $1/2^{32}$ (32-bit seq. #'s).
    - … but, many systems allow for a large window of acceptable seq. #'s. Much higher success probability.
  - Attacker can flood with RST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

Part 2-2

Routing Security: ARP, OSPF, BGP

Mostly Based on and extracted from Dan Boneh Lecures on Computer and Network Security, course material at https://crypto.stanford.edu/cs155/
+ of course with help of internet ☺

Inter-domain Routing



Routing Protocols

- 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)
- OSPF: used for routing within an AS
- ARP (Addr. Resolution Protocol): IP addr. ⟶ Physical addr.
  Security issues: (local network attacks e.g. ARP spoofing)
  - 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)

BGP example

An example of the BGP routes that are exchanged between domains.

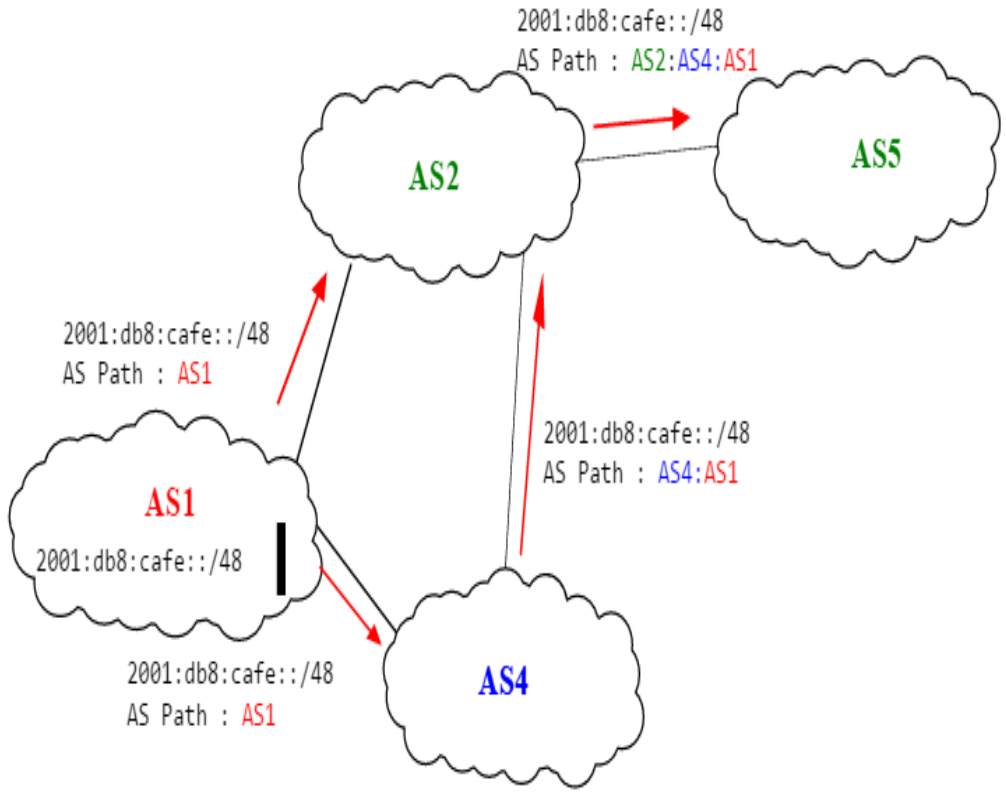
- Prefix *2001:db8:1234/48* is announced by*AS1*.

- *AS1* advertises a BGP route towards this prefix to *AS2*. The AS-Path of this route indicates that *AS1* is the originator of the prefix.
- When *AS4* receives the BGP route from *AS1*, it re-announces it to *AS2* and adds its AS number to the AS-Path.

- *AS2* has learned two routes towards prefix *2001:db8:1234/48*. It compares the two routes and prefers the route learned from *AS4* based on its own ranking algorithm.

- *AS2* advertises to *AS5* a route towards *2001:db8:1234/48* with its AS-Path set to *AS2:AS4:AS1*.

- Thanks to the AS-Path, *AS5* knows that if it sends a packet towards *2001:db8:1234/48* the packet first



http://cnp3book.info.ucl.ac.be/2nd/html/protocols/bgp.html

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

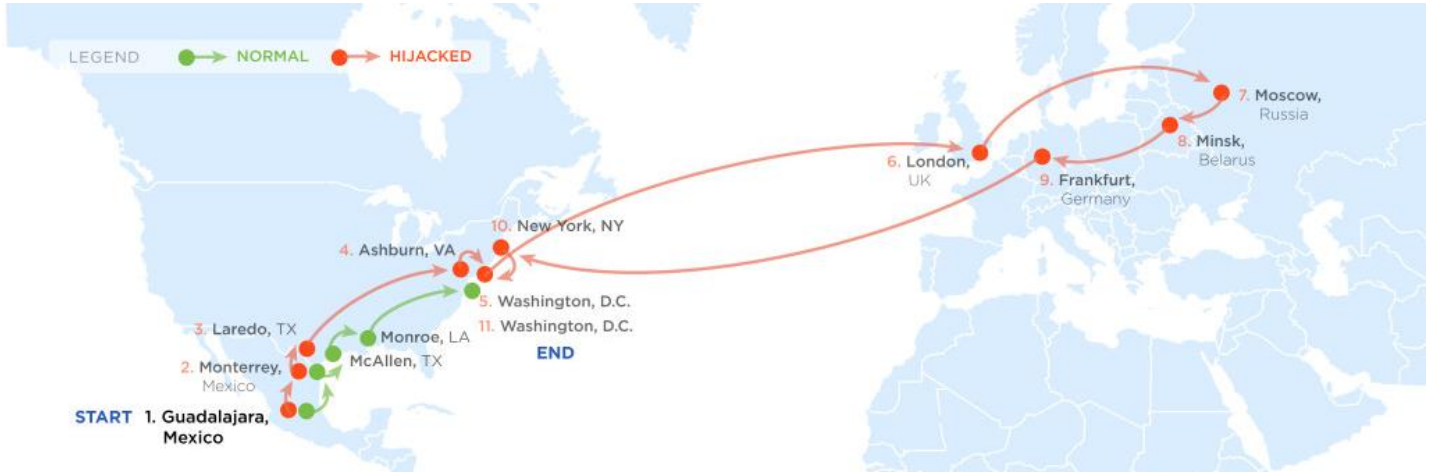
Solutions:

- RPKI: AS obtains a certificate (ROA) from RIR and
  attaches ROA to path advertisements.
  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

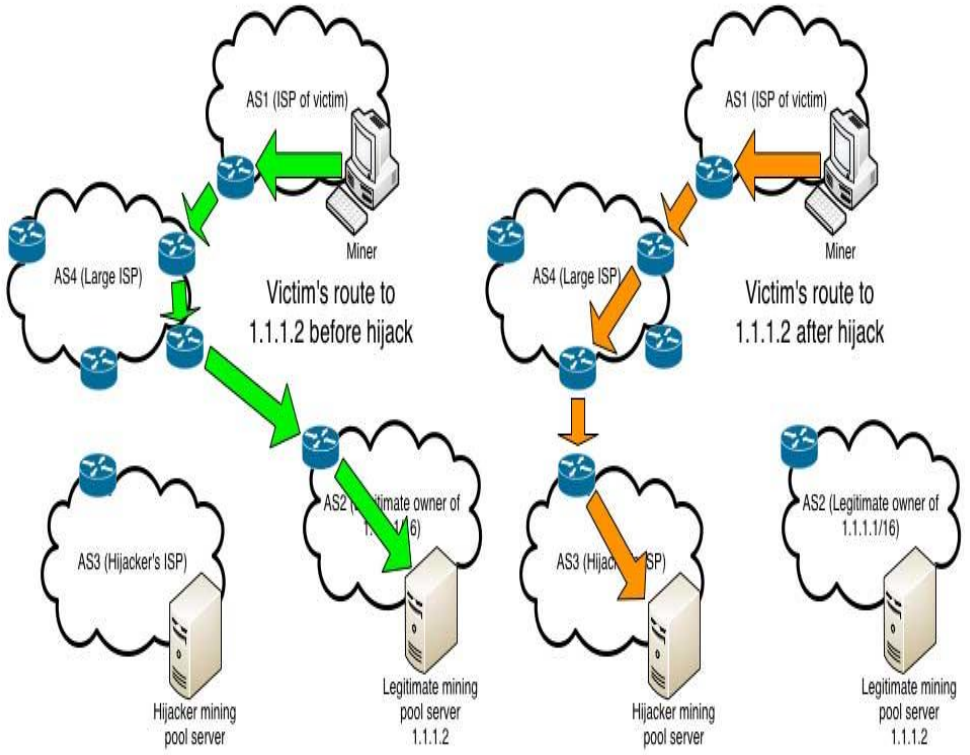
Yes, It is real!

- Network hijacker steals $83,000 in Bitcoin ... and enough Dogecoin for a cup of coffee!
- The Dell researchers eventually traced the bogus BGP broadcasts to a single router at an unnamed Canadian ISP, but no culprit in the attacks has been identified.
- How much a single bitcoin value?

1 Bitcoin equals

448.28 US Dollar

83000	Bitcoin
37207240.00	US Dollar



OSPF: Routing inside an AS

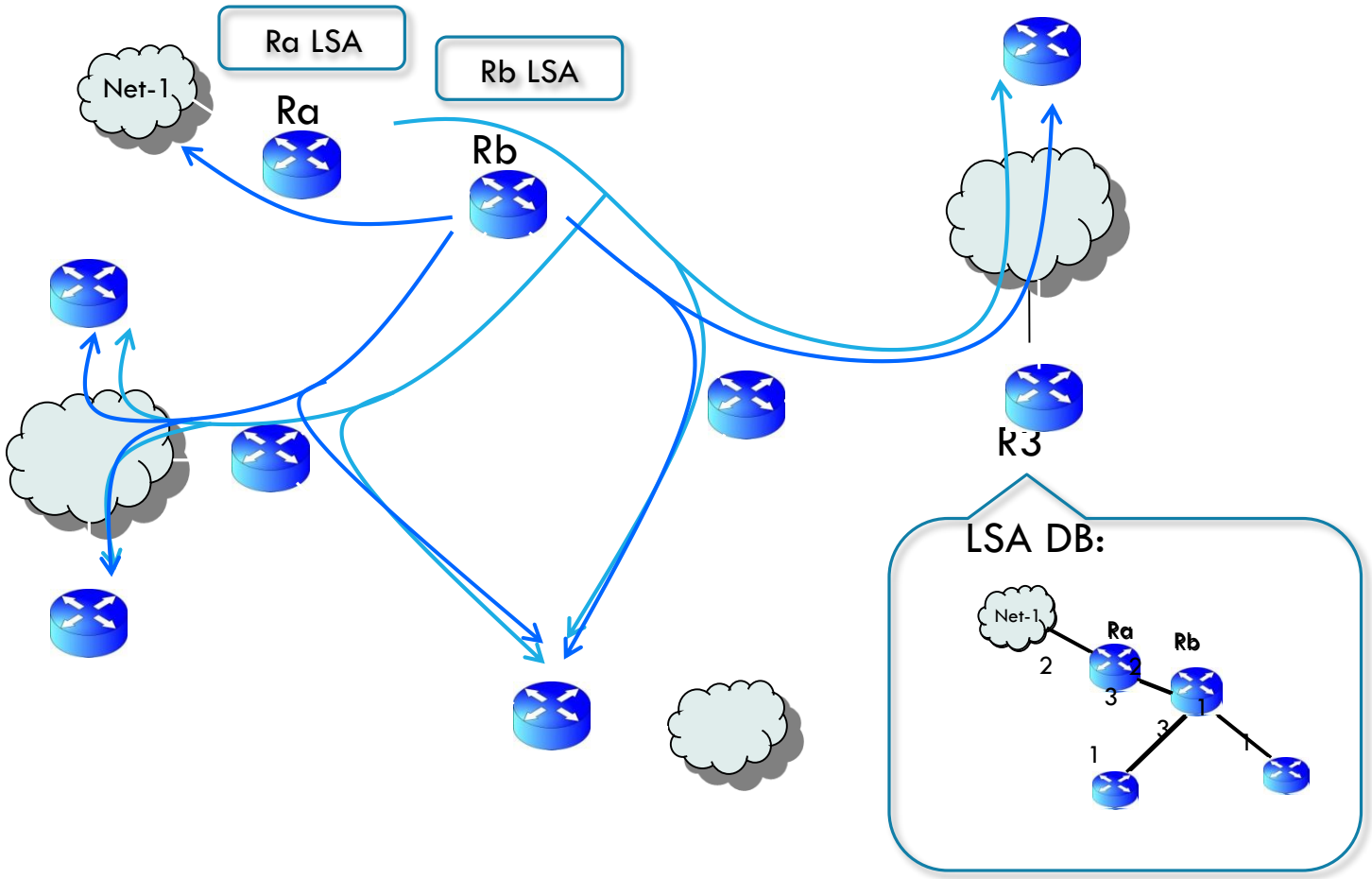
Link State Advertisements (LSA):

- Flooded throughout AS so that all routers in the AS have a complete view of the AS topology
- Transmission: IP datagrams, protocol = 89

Neighbor discovery:

- Routers dynamically discover direct neighbors on attached links --- sets up an "adjacenty"
- Once setup, they exchange their LSA databases

Example: LSA from Ra and Rb



Security features

- OSPF message integrity (unlike BGP)
  - Every link can have its own shared secret
  - Unfortunately, OSPF uses an insecure MAC:

    MAC(k,m) = MD5(data || key || pad || len)

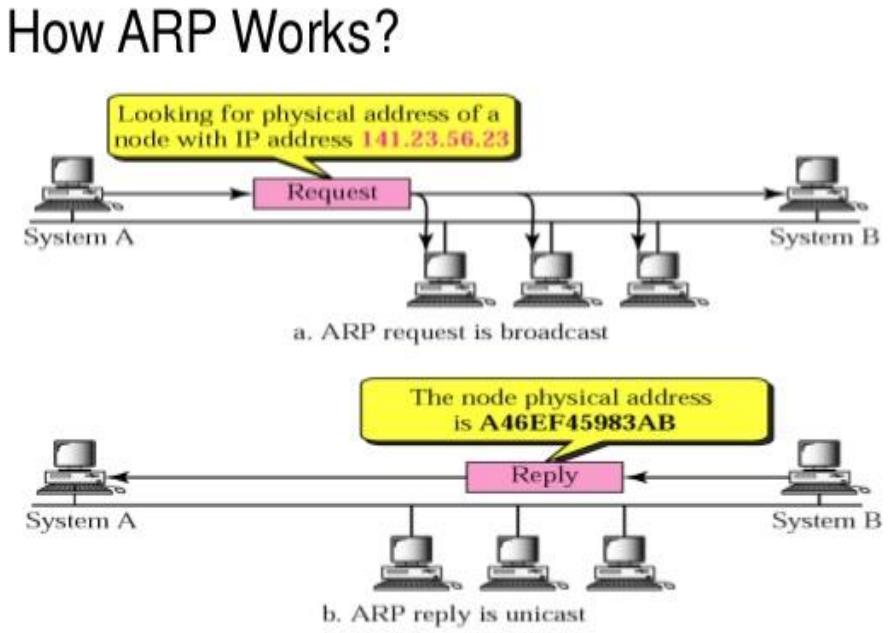
- Every LSA is flooded throughout the AS
  - If a single malicious router, valid LSAs may still reach dest.

- The “fight back” mechanism
  - If a router receives its own LSA with a newer timestamp than the latest it sent, it immediately floods a new LSA

- Links must be advertised by both ends

ARP protocol:

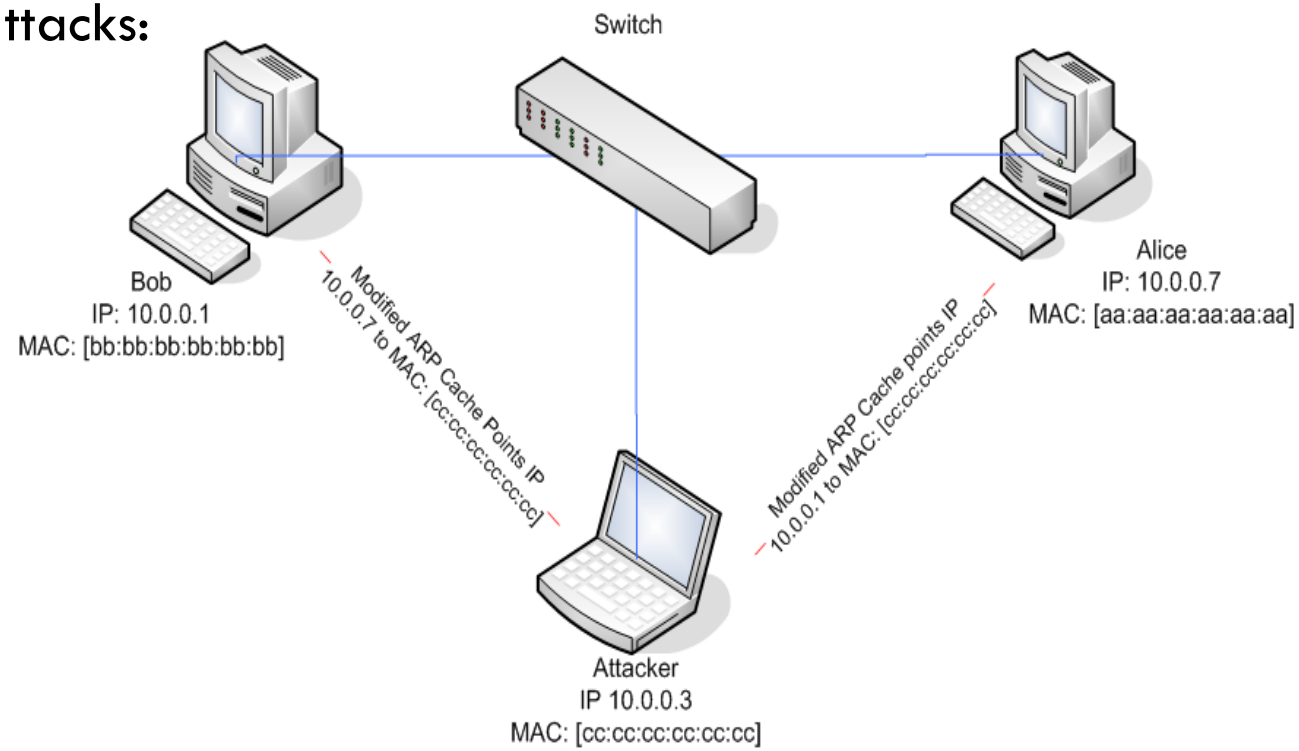
- It maps IP addresses to the hardware address used by a datalink protocol



ARP spoofing

- A technique by which an attacker send (spoofed) ARP messages onto a network

- Usually it is used as an opening to other attacks:
  - DOS
  - MITM
  - Session Hijacking



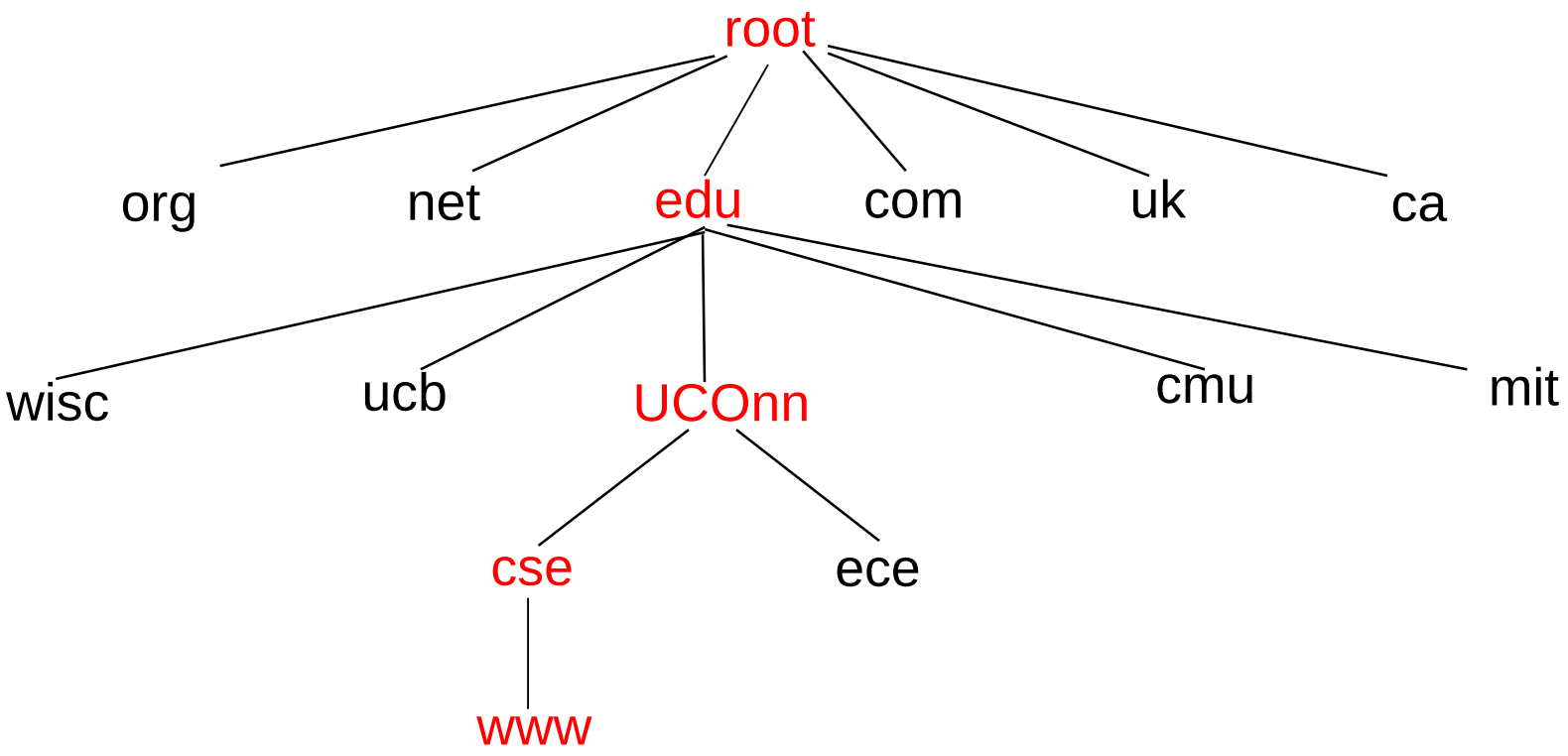
Part 2-3

Domain Name System

Mostly Based on and extracted from Dan Boneh Lecures on Computer and Network Security, course material at https://crypto.stanford.edu/cs155/
+ of course with help of internet ☺

Domain Name System(DNS)

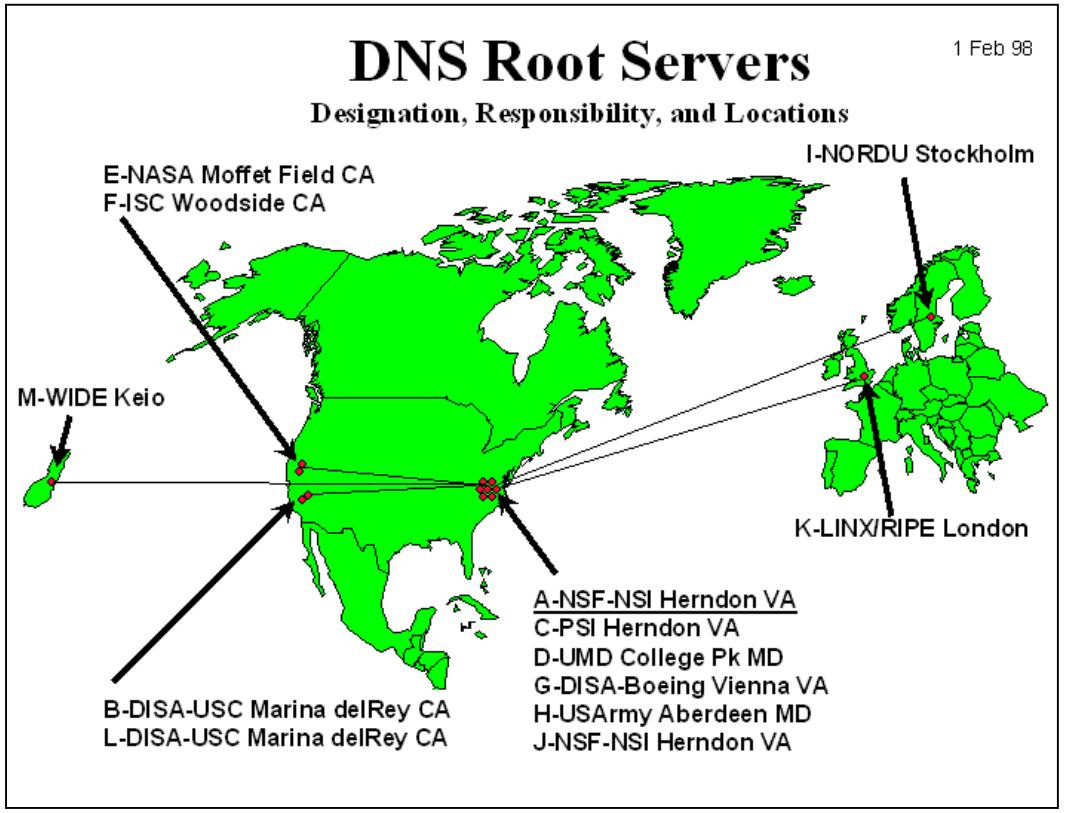
- Hierarchical Name Space



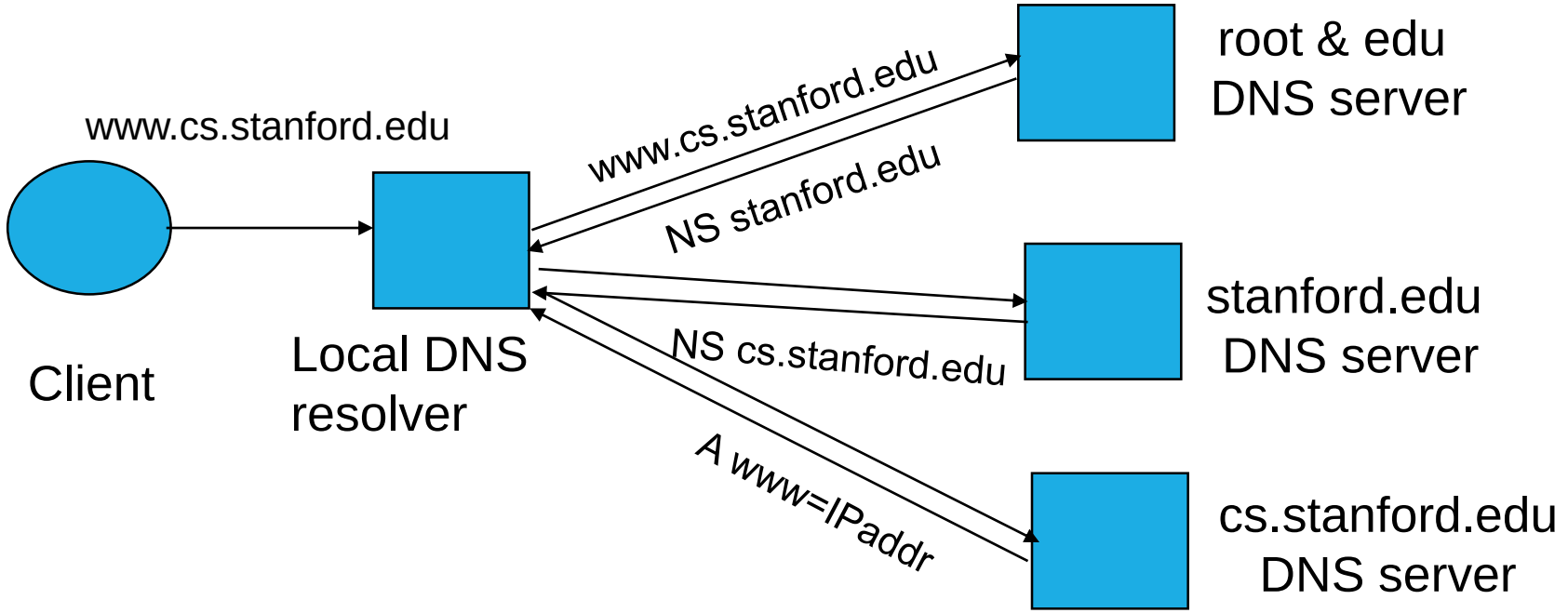
DNS Root Name Servers

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):

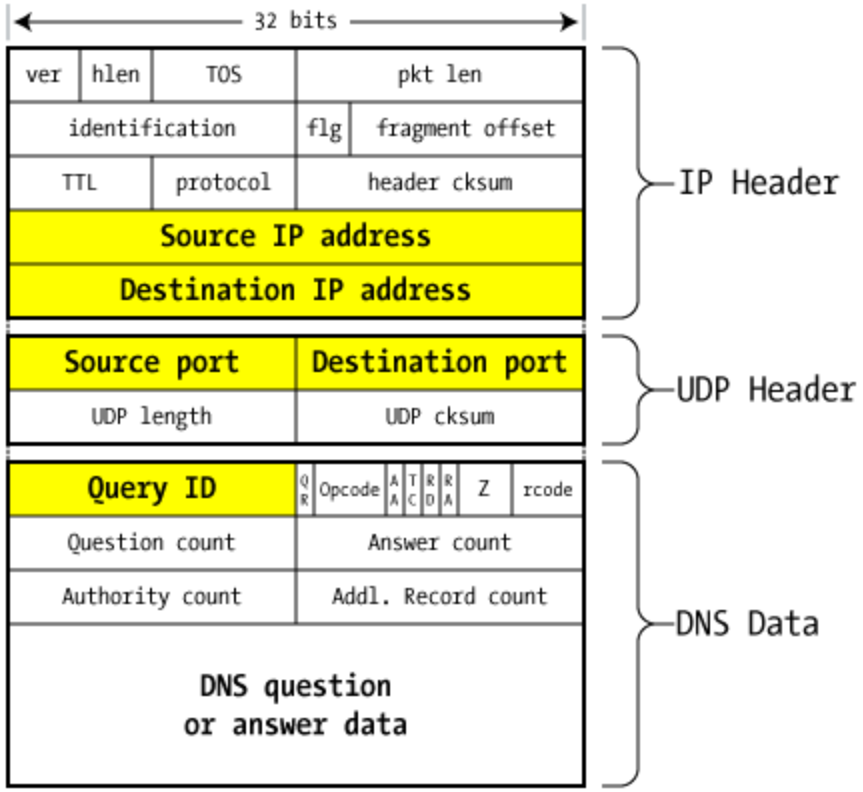
- NS: name server (points to other server)
- A: address record (contains IP address)
- MX: address in charge of handling email
- TXT: generic text (e.g. used to distribute site public keys (DKIM))

Caching

- DNS responses are cached
  - Quick response for repeated translations
  - Useful for finding servers as well as addresses
    - NS records for domains
- 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

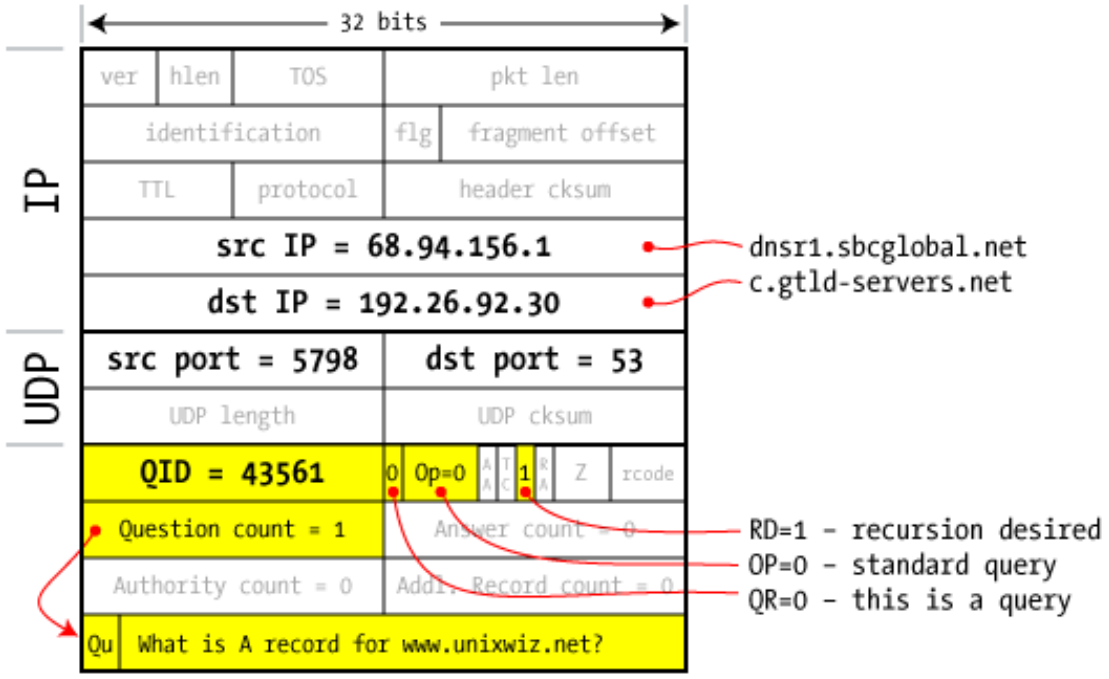
DNS Packet

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



(from Steve Friedl)

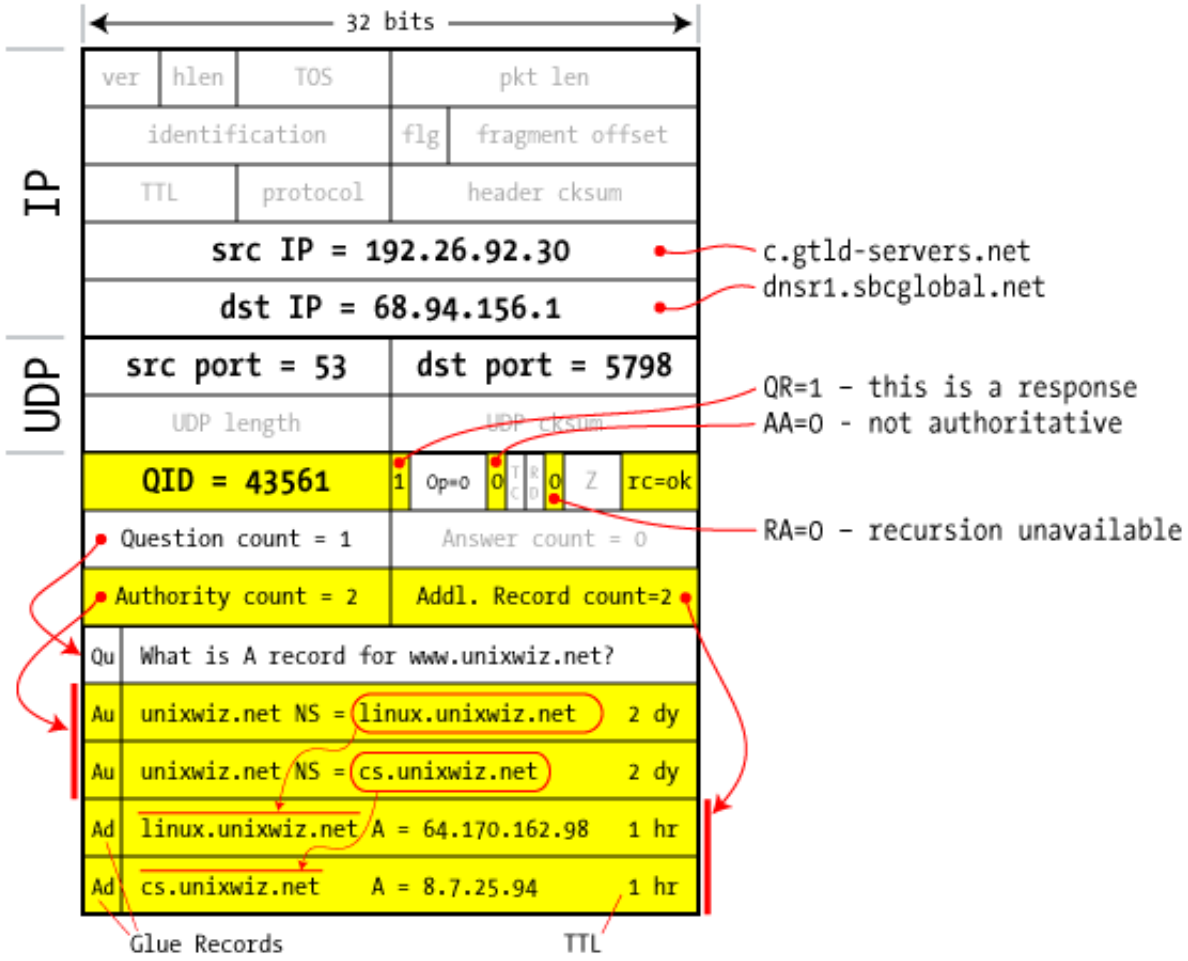
Resolver to NS request



Response to resolver

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

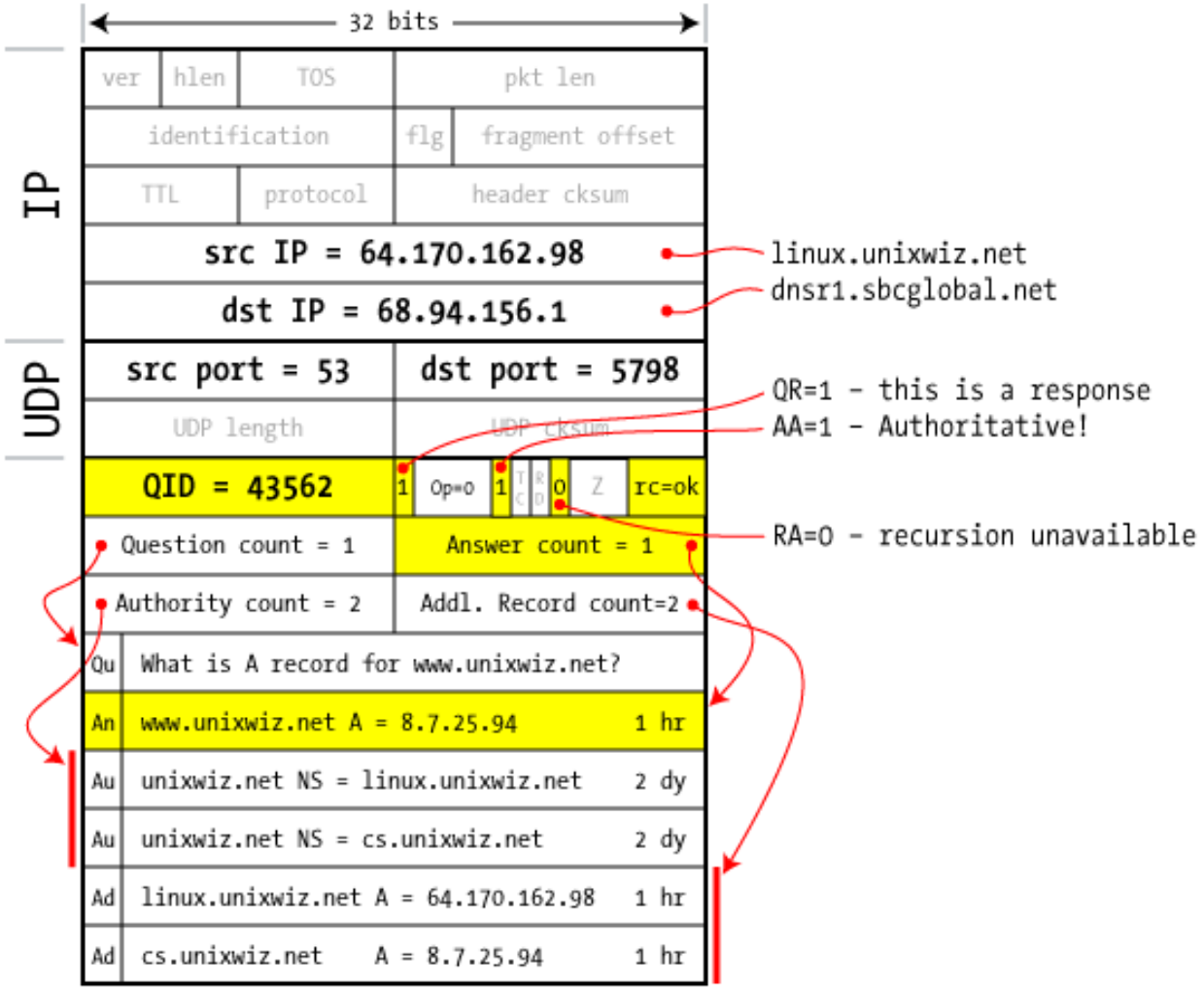
Response ignored if unrecognized QueryID



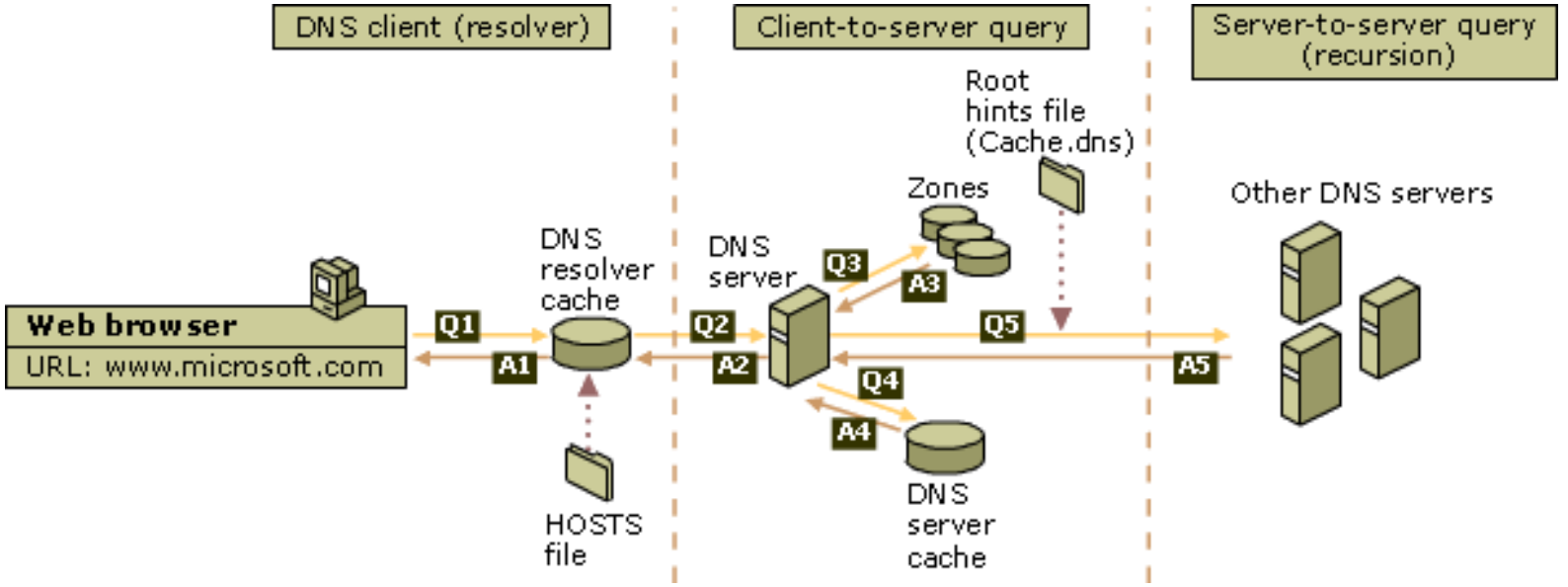
Authoritative response to resolver

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

final answer →



DNS Simple - Visualization



Basic DNS Vulnerabilities

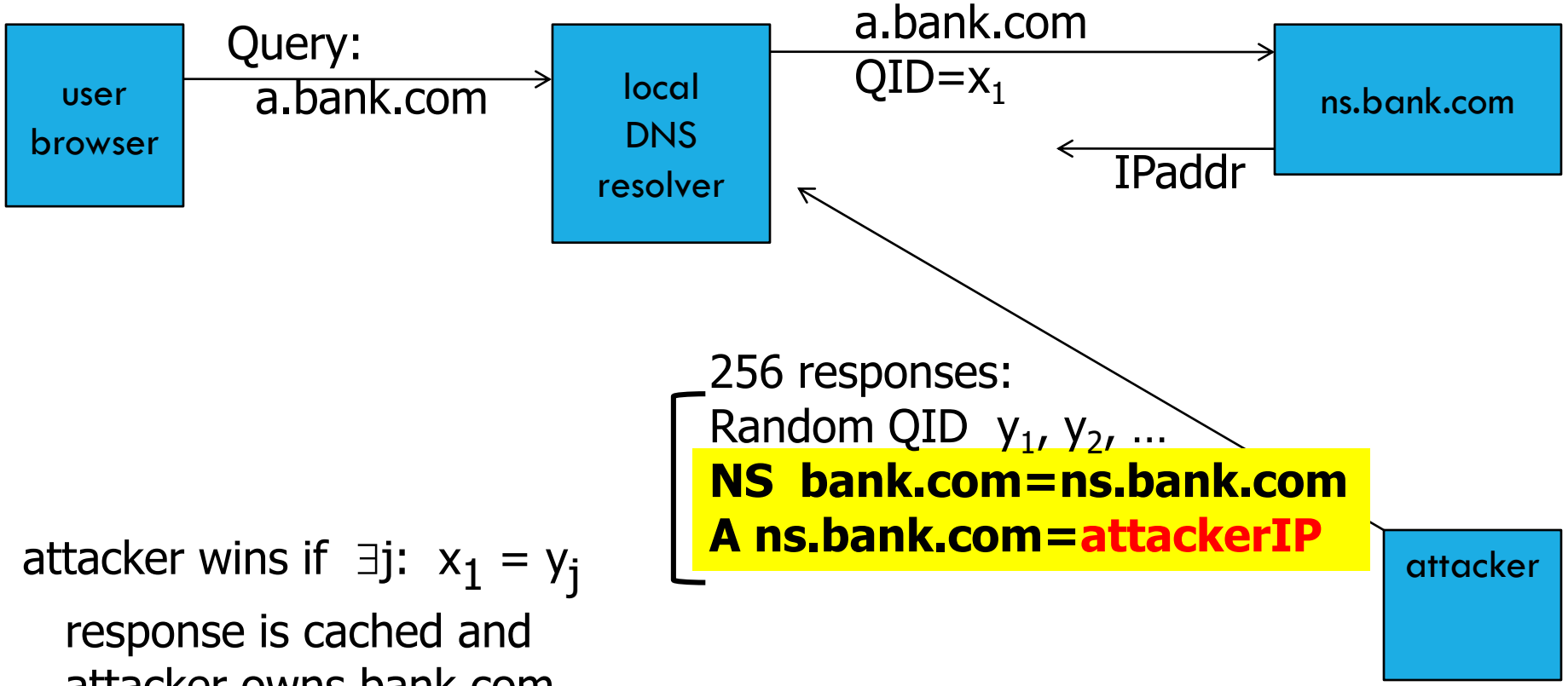
- Users/hosts trust the host-address mapping provided by DNS:
  - Used as basis for many security policies:

    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

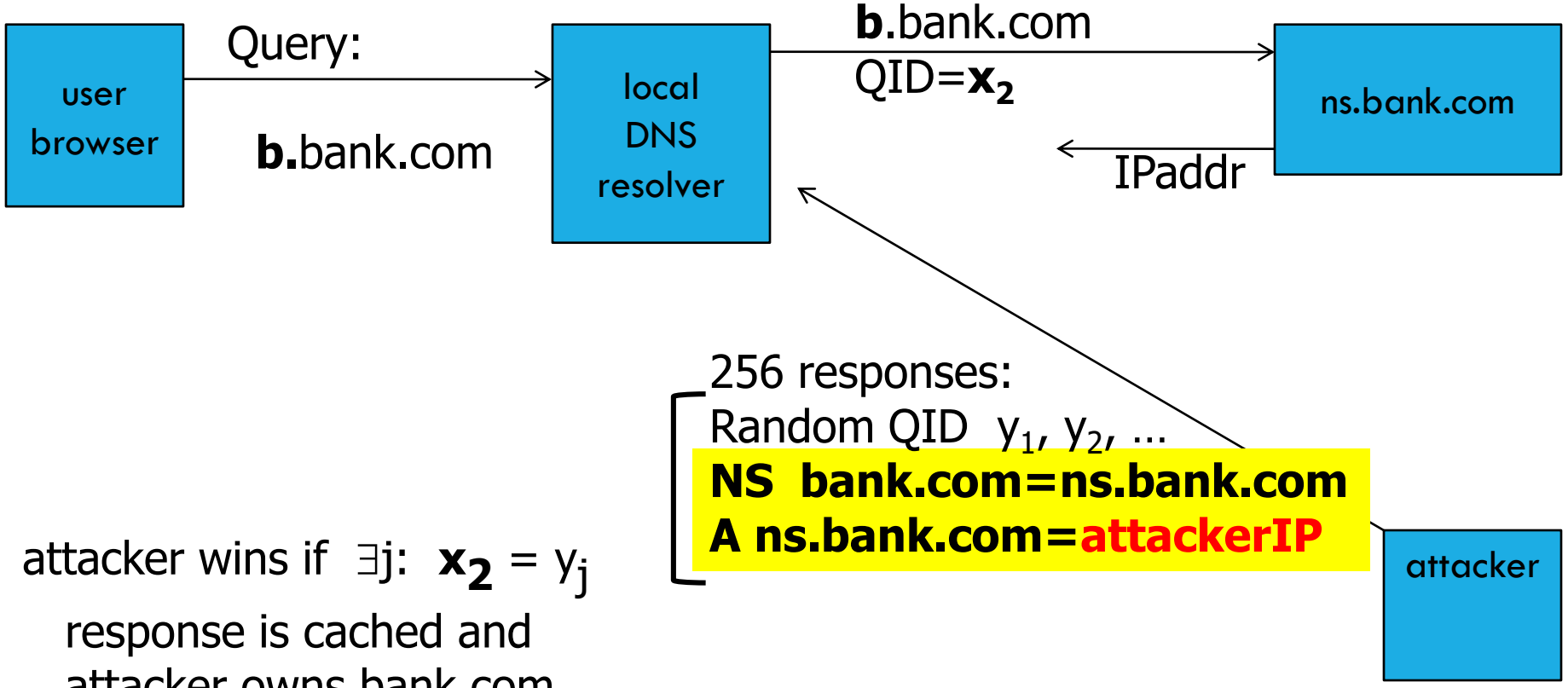
DNS cache poisoning (a la Kaminsky'08)

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



If at first you don't succeed ...

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



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

response is cached and attacker owns bank.com

success after ≈ 256 tries (few minutes)

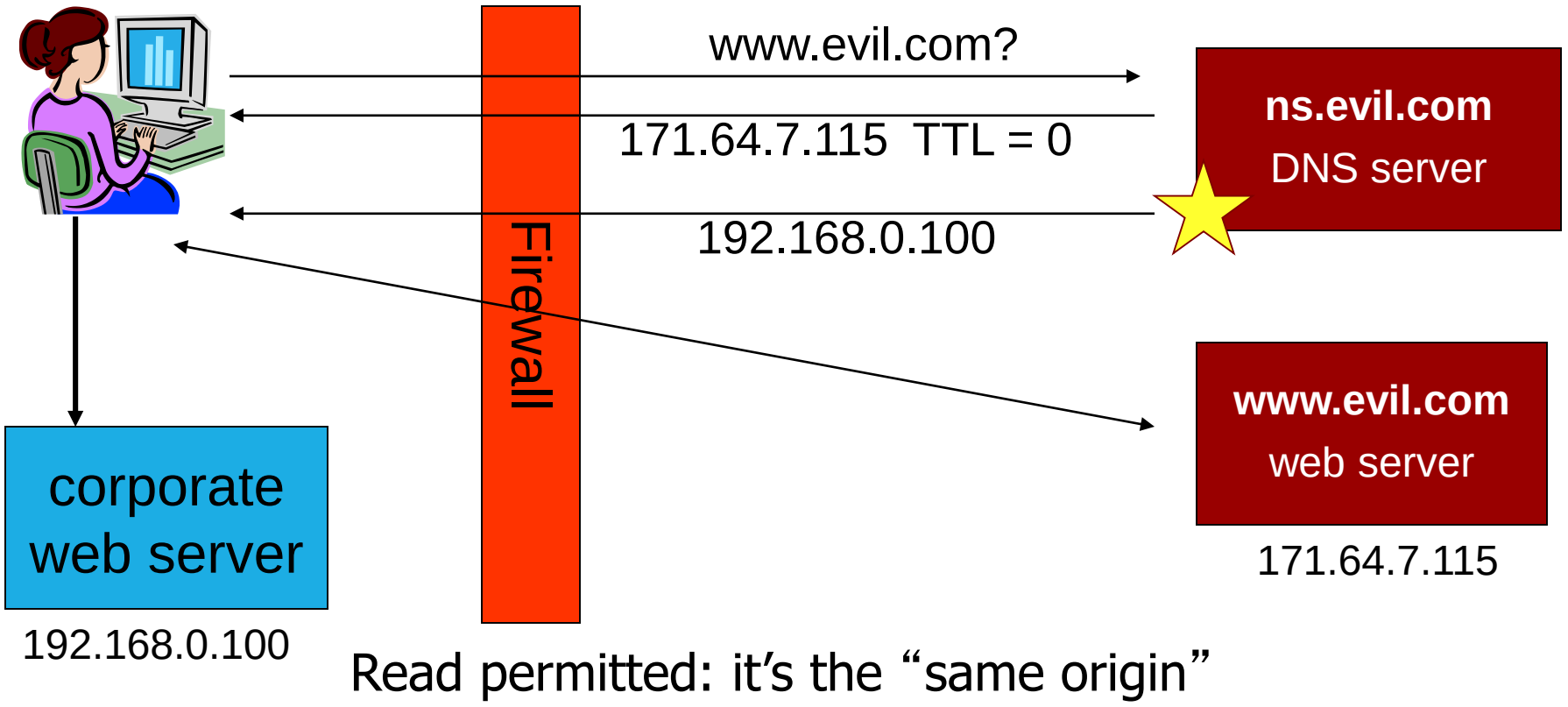
Defenses

- 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

DNS Rebinding Attack

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

DNS-SEC cannot stop this attack



Read permitted: it's the "same origin"

[DWF' 96, R' 01]

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 Slides) :

  IP ⟶ IPsec

  DNS ⟶ DNSsec

  BGP ⟶ SBGP

Until now, we covered:

- Basic network protocols
  - IP, TCP, UDP, BGP, DNS
- Problems with them
  - TCP/IP
    - No SRC authentication: can't tell where packet is from
    - Packet sniffing
    - Connection spoofing, sequence numbers
  - BGP: advertise bad routes or close good ones
  - DNS: cache poisoning, rebinding
    - Web security mechanisms rely on DNS

Part 3-1

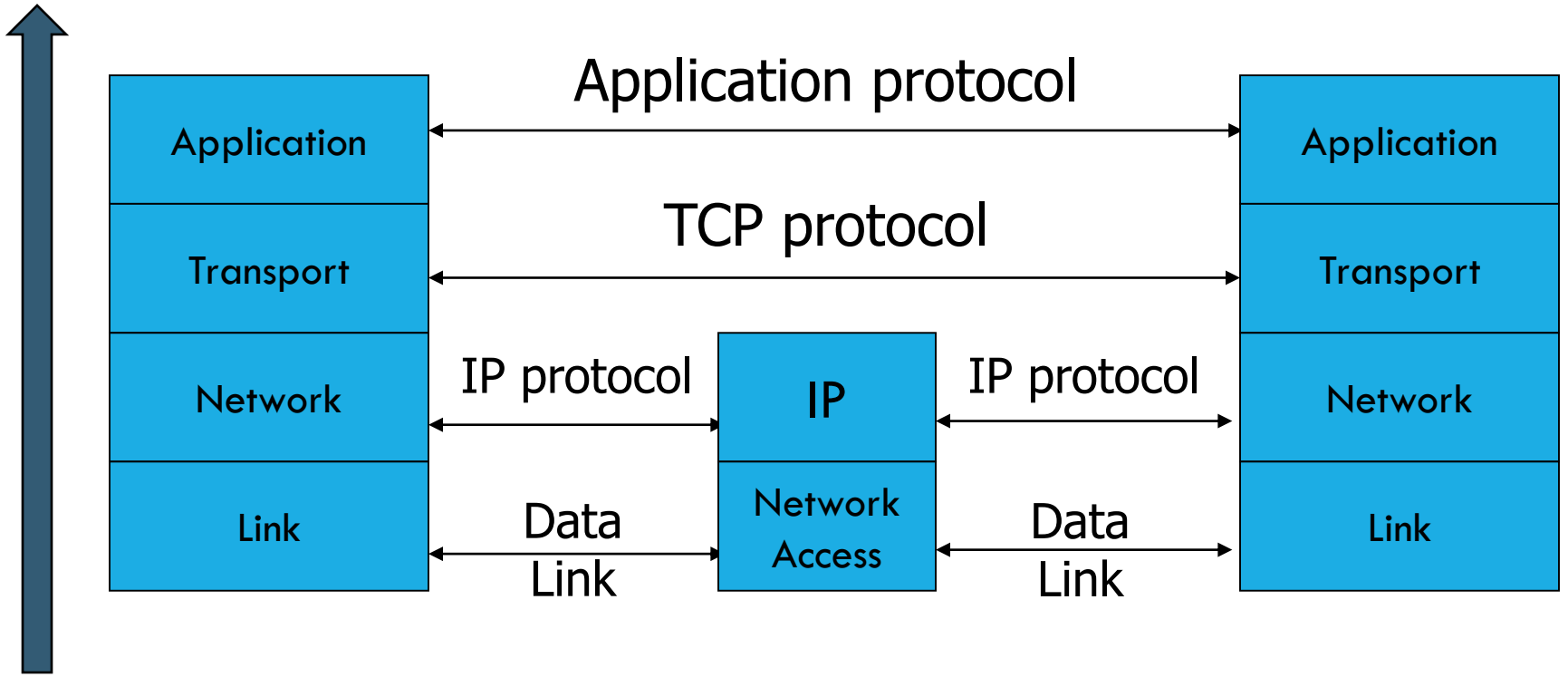
Network Protocol Security

Mostly based on John Mitchell Slides

What is the plan?

- Network protocol security
  - IPSEC
  - BGP instability and S-BGP
  - DNS rebinding and DNSSEC
- Standard network defenses
  - Firewall
    - Packet filter (stateless, stateful), Application layer proxies
  - Intrusion detection
    - Anomaly and misuse detection

Network Protocol Stack



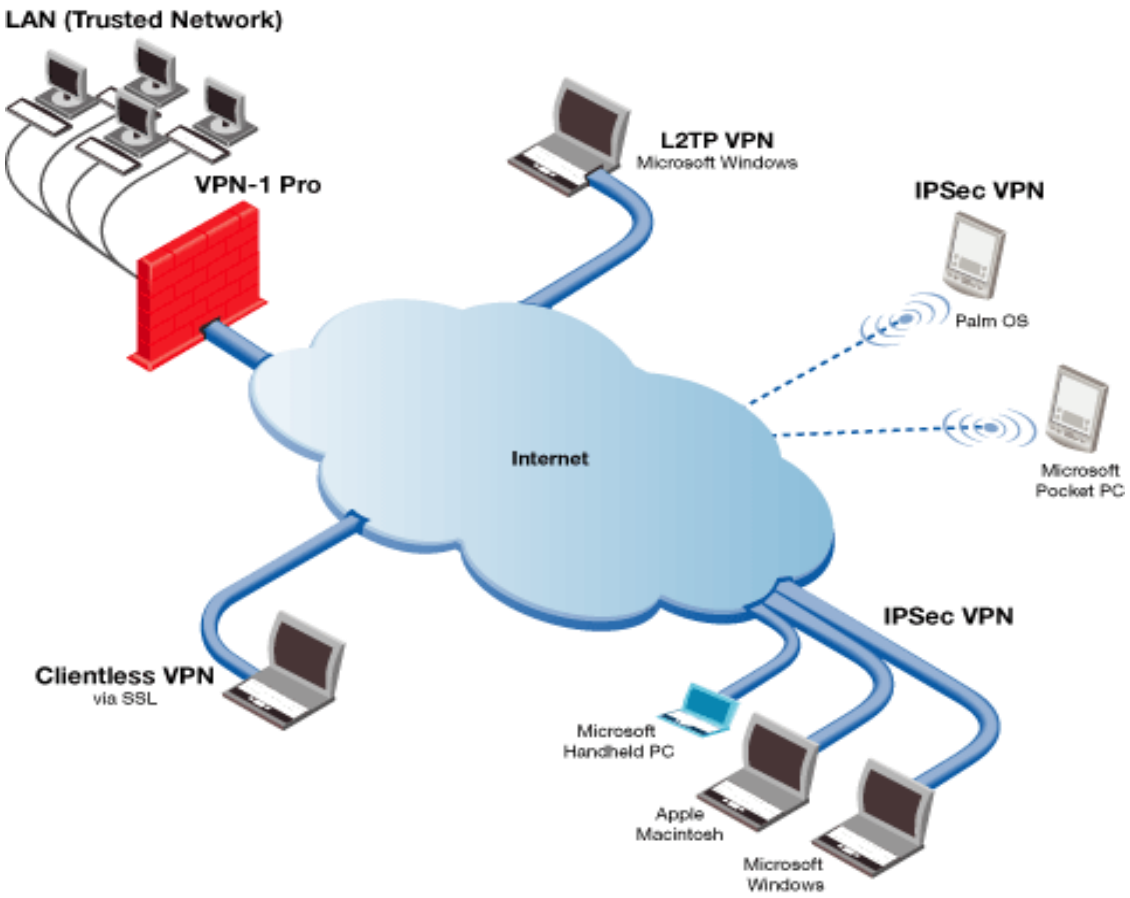
Basic Layer 2-3 Security Problems

- Network packets pass by untrusted hosts
  - Eavesdropping, packet sniffing
  - Especially easy when attacker controls a machine close to victim
- TCP state can be easy to guess
  - Enables spoofing and session hijacking

Virtual Private Network (VPN)

- Three different modes of use:
  - Remote access client connections
  - LAN-to-LAN internetworking
  - Controlled access within an intranet
- Several different protocols
  - PPTP – Point-to-point tunneling protocol
  - L2TP – Layer-2 tunneling protocol
  - IPsec (Layer-3: network layer)

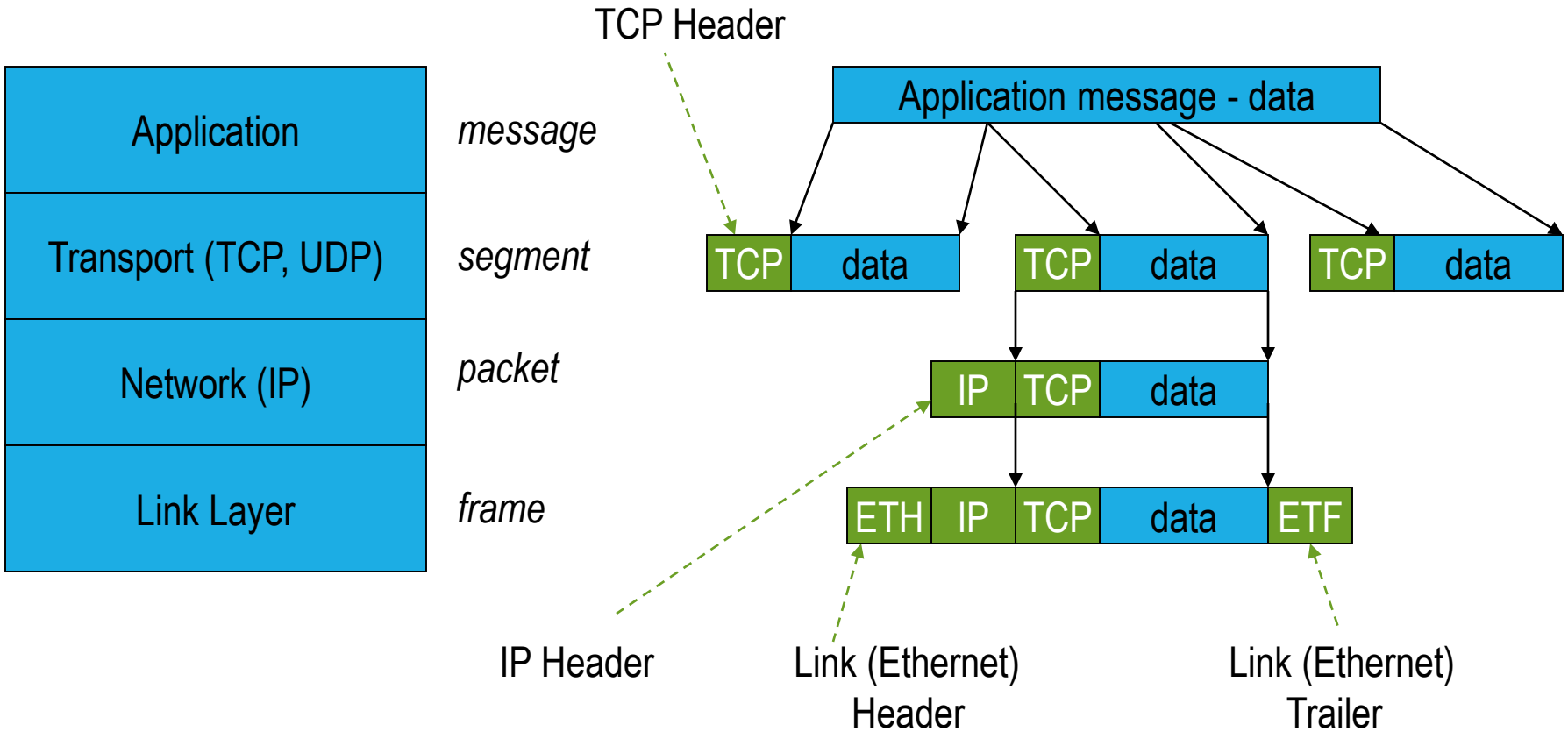
} Data layer



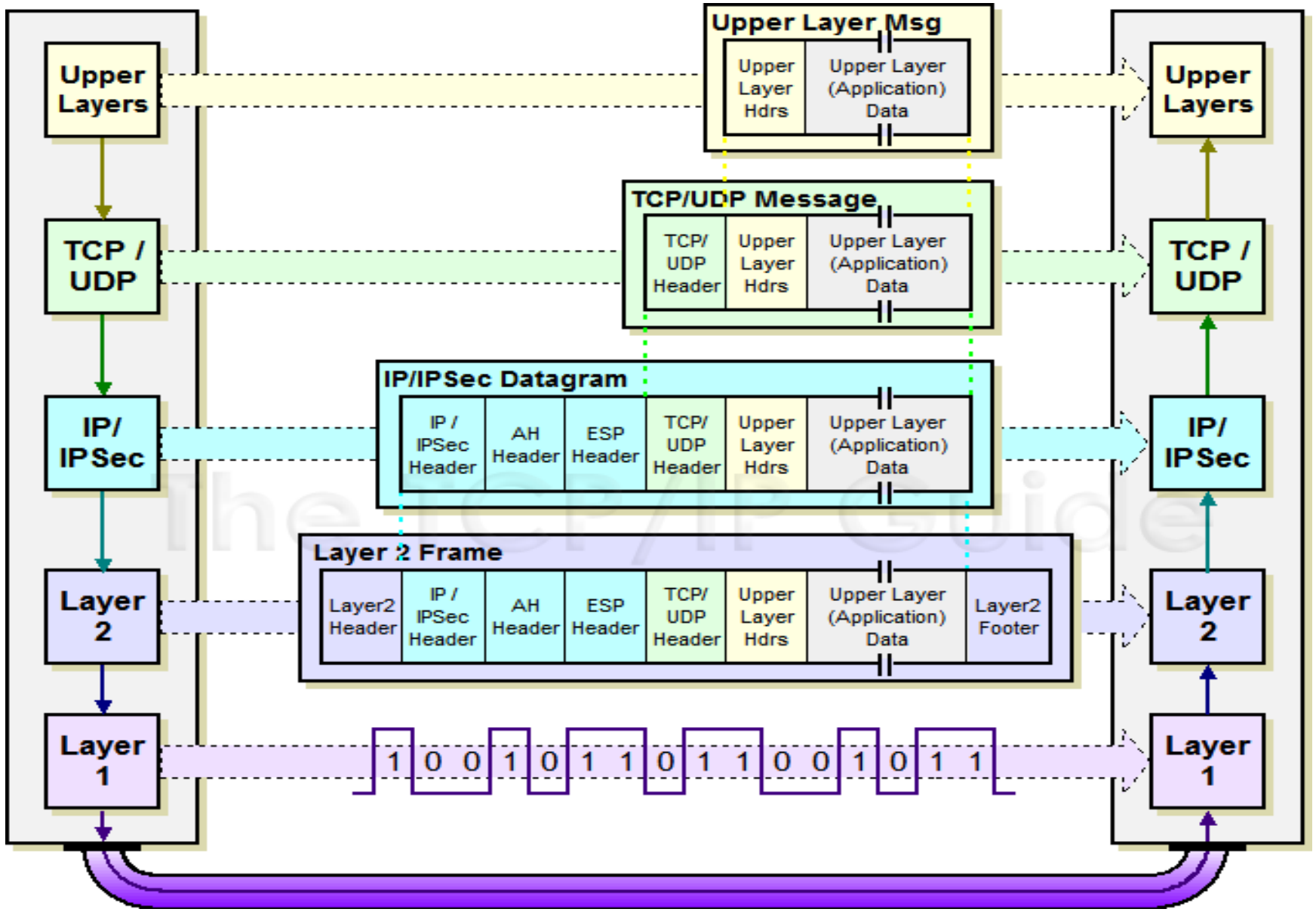
IPSEC

- Security extensions for IPv4 and IPv6
- IP Authentication Header (AH)
  - Authentication and integrity of payload and header
- IP Encapsulating Security Protocol (ESP)
  - Confidentiality of payload
- ESP with optional ICV (integrity check value)
  - Confidentiality, authentication and integrity of payload

Recall packet formats and layers

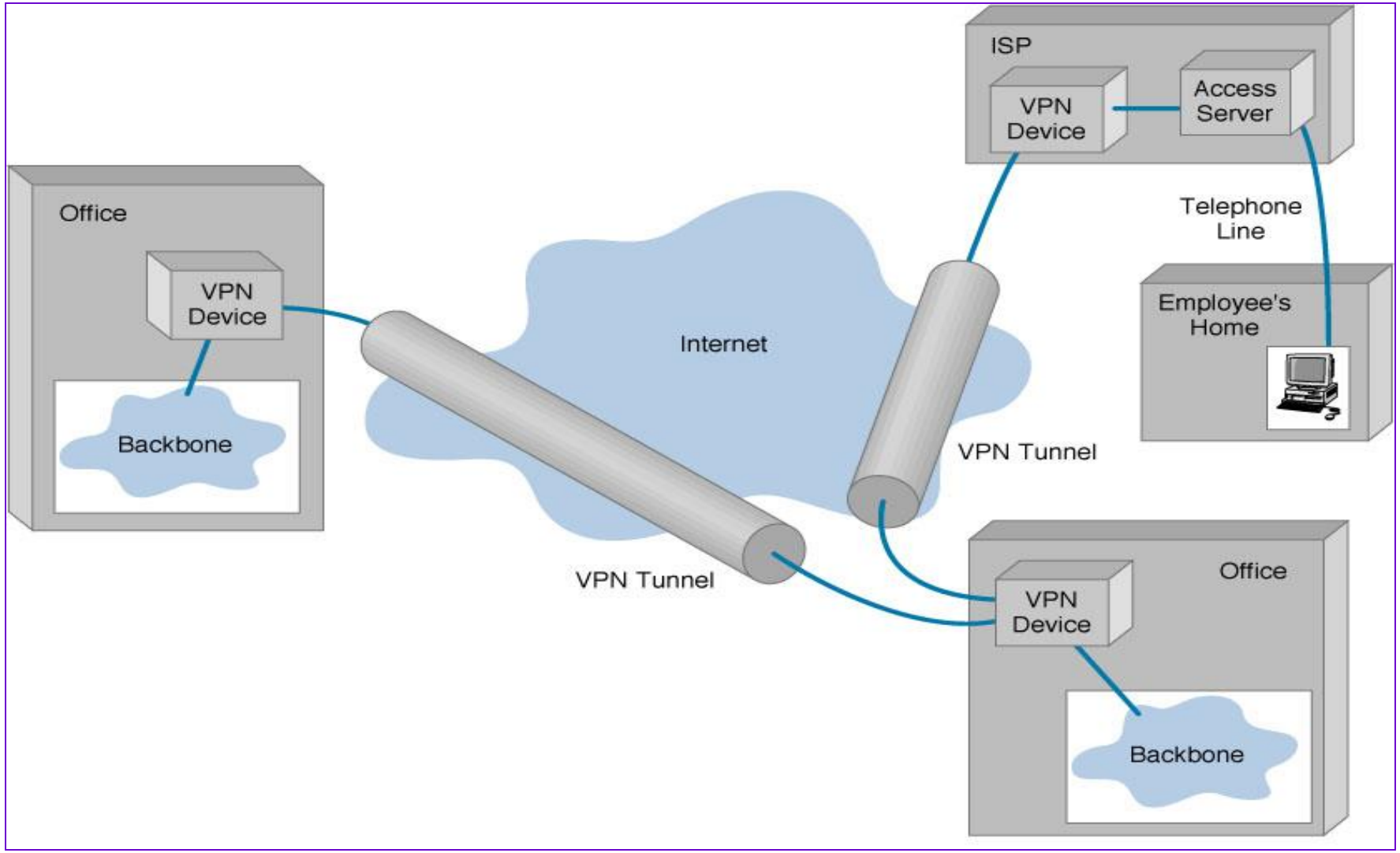


IPSec Transport Mode: IPSEC instead of IP header

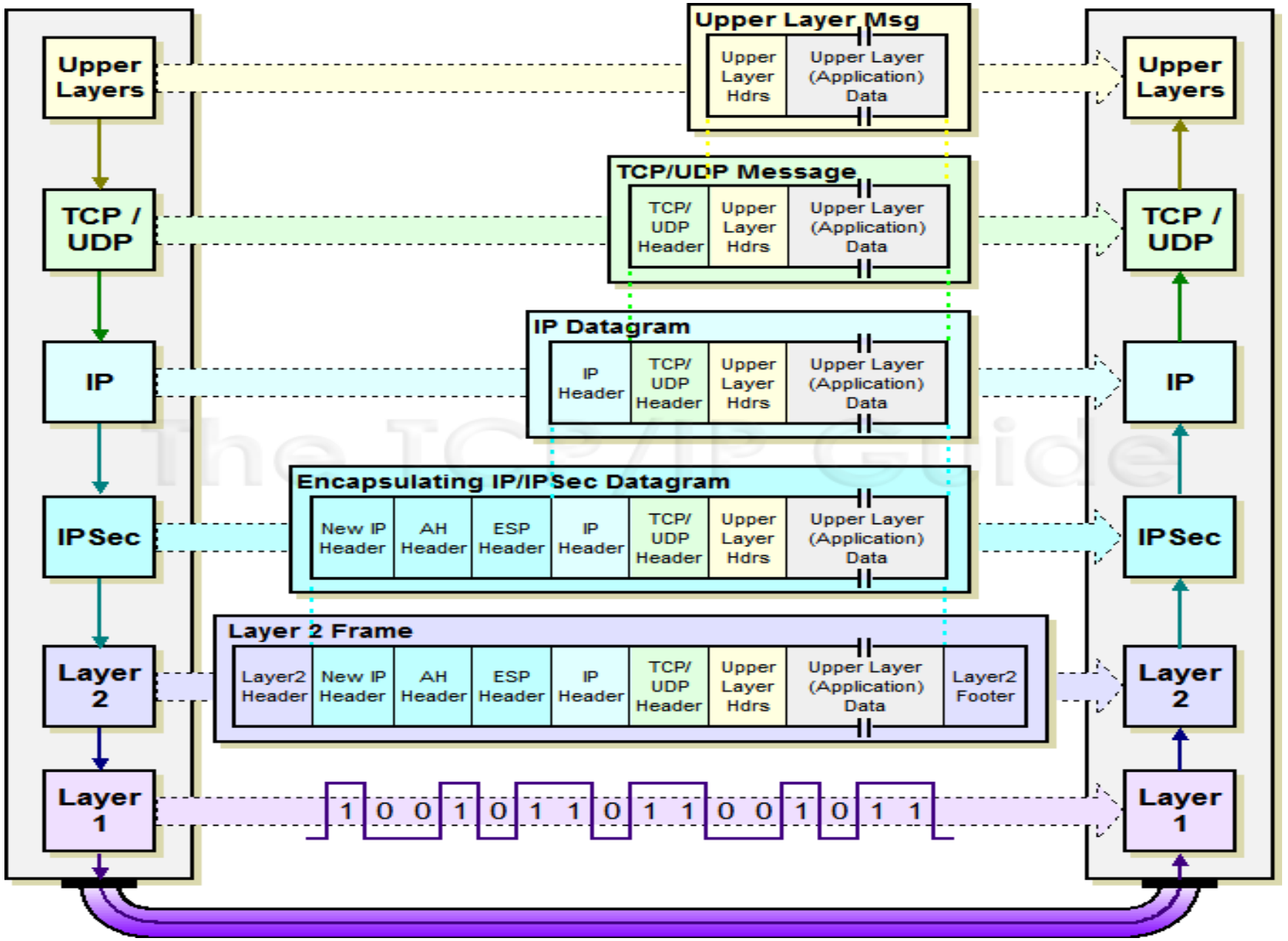


http://www.tcpipguide.com/free/t_IPSecModesTransportandTunnel.htm

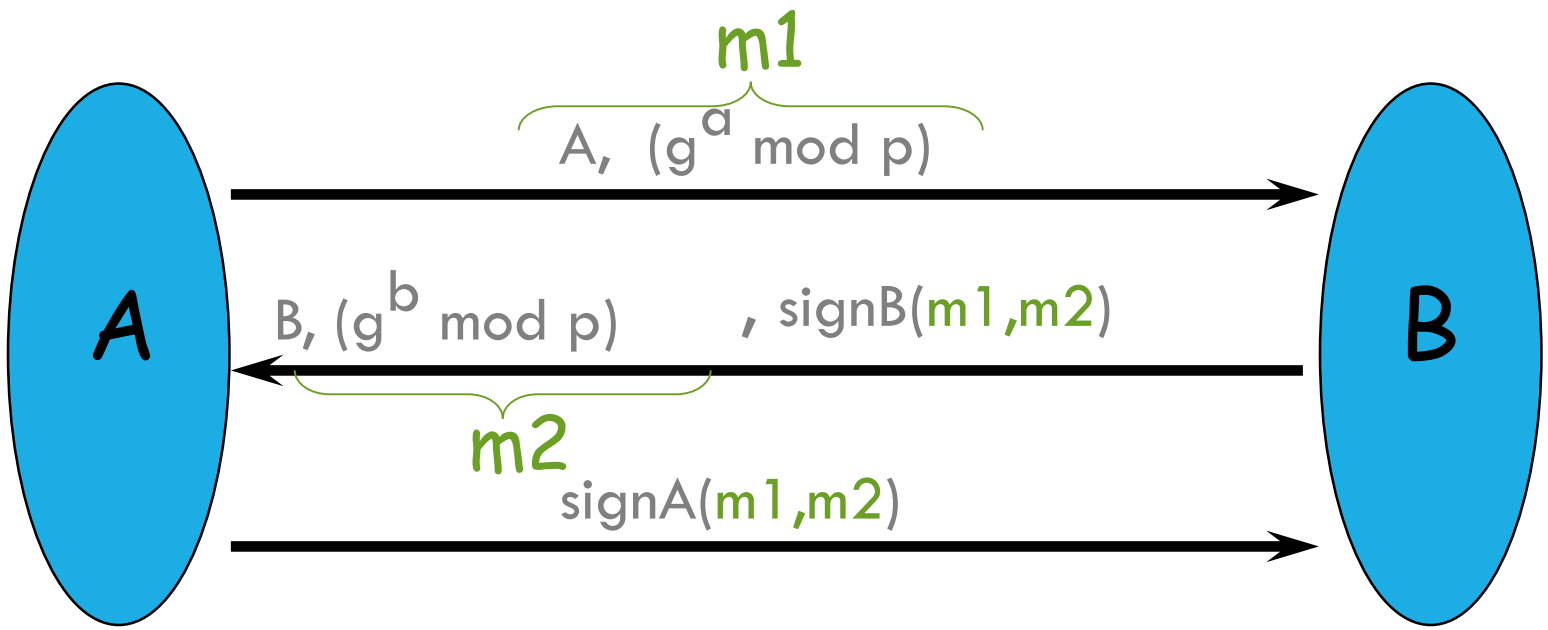
IPSEC Tunnel Mode



IPSec Tunnel Mode: IPSEC header + IP header

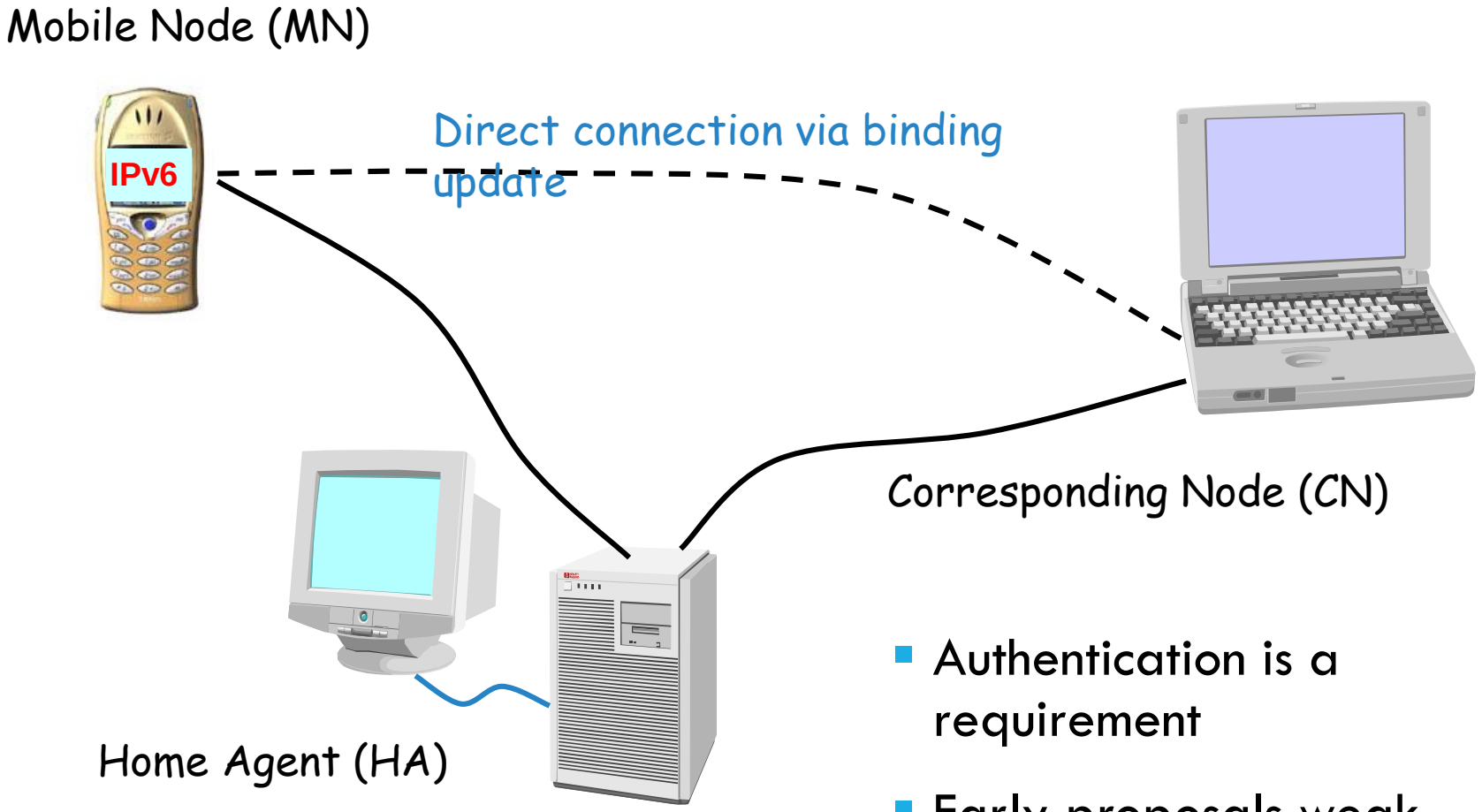


Internet Key Exchange(IKE) subprotocol from IPSEC



Result: A and B share secret $g^{ab} \bmod p$

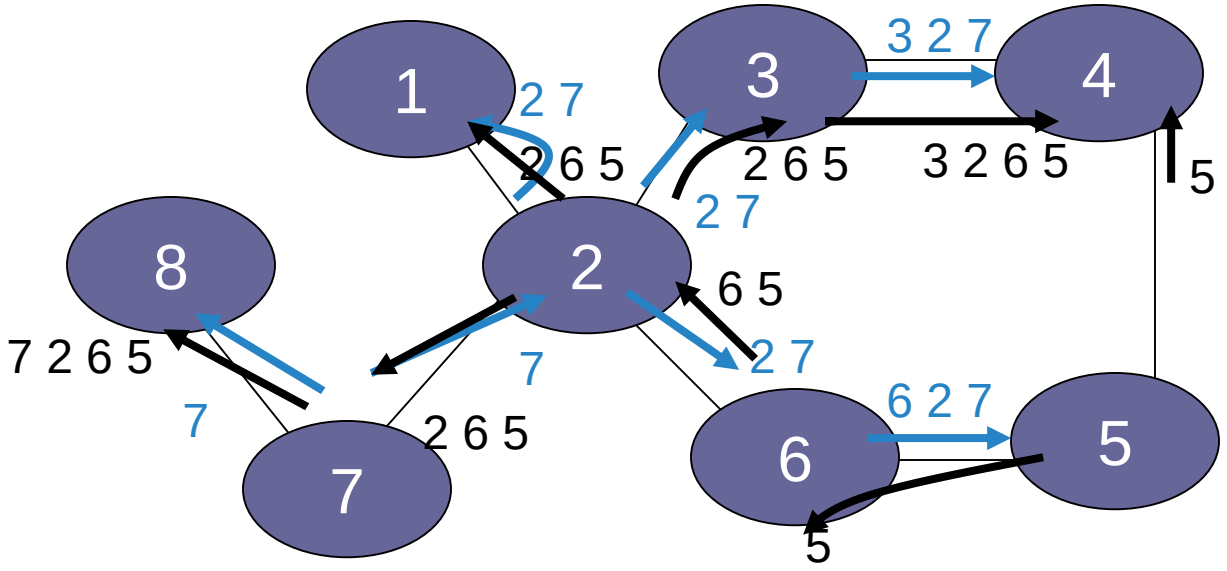
Mobile IPv6 Architecture



- Authentication is a requirement
- Early proposals weak

Infrastructure protocols: BGP

BGP example



- Transit: 2 provides transit for 7
- Algorithm seems to work OK in practice
  - BGP is does not respond well to frequent node outages

Figure: D. Wetherall

BGP Security Issues

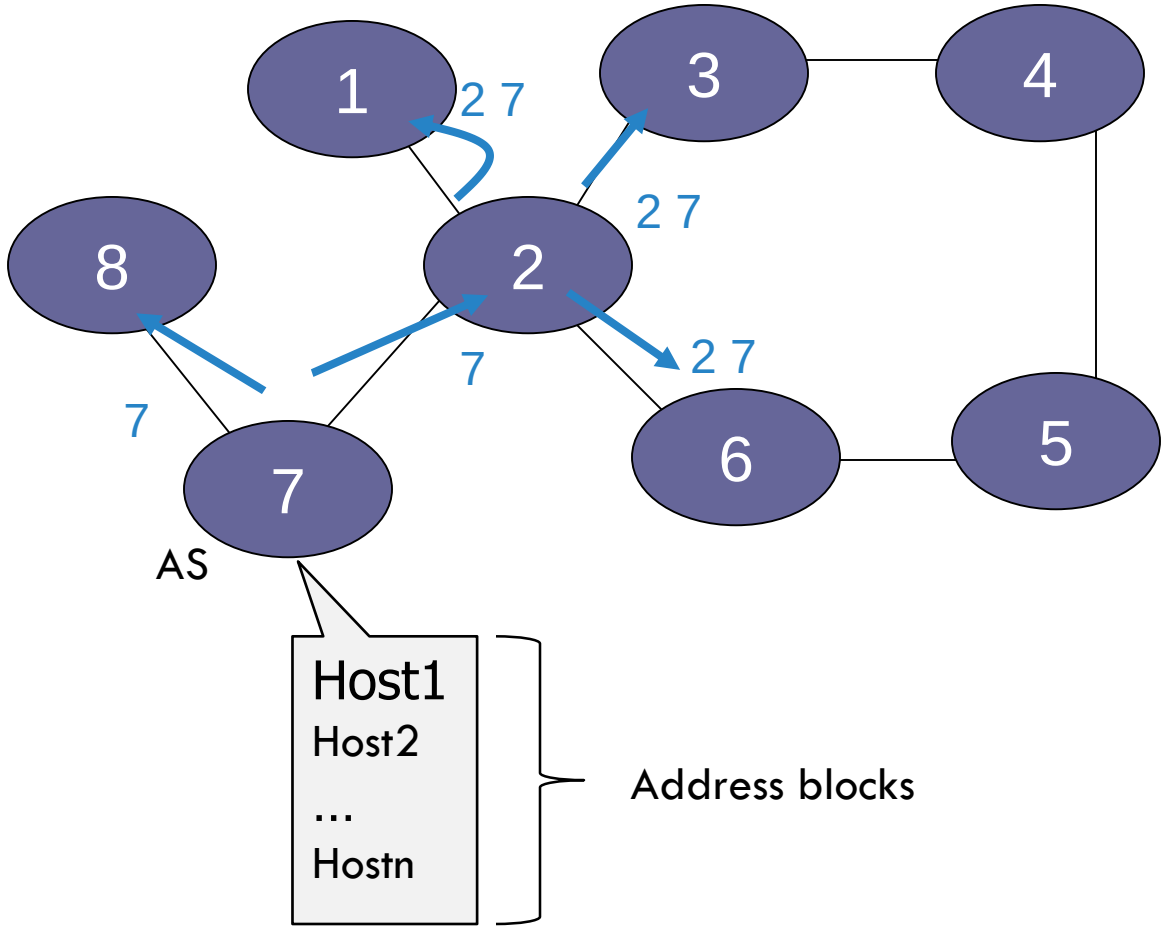
- BGP is used for all inter-ISP routing
- Benign configuration errors affect about 1% of all routing table entries at any time
- Highly vulnerable to human errors, malicious attacks
  - Actual routing policies can be very complicated
- MD5 MAC is rarely used, perhaps due to lack of automated key management, addresses only one class of attacks

S-BGP Design Overview

- IPsec: secure point-to-point router communication
- Public Key Infrastructure: authorization for all S-BGP entities
- Attestations: digitally-signed authorizations
  - Address: authorization to advertise specified address blocks
  - Route: Validation of UPDATEs based on a new path attribute, using PKI certificates and attestations
- Repositories for distribution of certificates, CRLs, and address attestations
- Tools for ISPs to manage address attestations, process certificates & CRLs, etc.

Slide: Steve Kent

BGP example



Address Attestation

- Indicates that the final AS listed in the UPDATE is authorized by the owner of those address blocks to advertise the address blocks in the UPDATE
- Includes identification of:
  - owner's certificate
  - AS to be advertising the address blocks
  - address blocks
  - expiration date
- Digitally signed by owner of the address blocks
- Used to protect BGP from erroneous UPDATEs (authenticated but misbehaving or misconfigured BGP speakers)

Route Attestation

- Indicates that the speaker or its AS authorizes the listener's AS to use the route in the UPDATE
- Includes identification of:
  - AS's or BGP speaker's certificate issued by owner of the AS
  - the address blocks and the list of ASes in the UPDATE
  - the neighbor
  - expiration date
- Digitally signed by owner of the AS (or BGP speaker) distributing the UPDATE, traceable to the IANA …
- Used to protect BGP from erroneous UPDATEs (authenticated but misbehaving or misconfigured BGP speakers)

Validating a Route

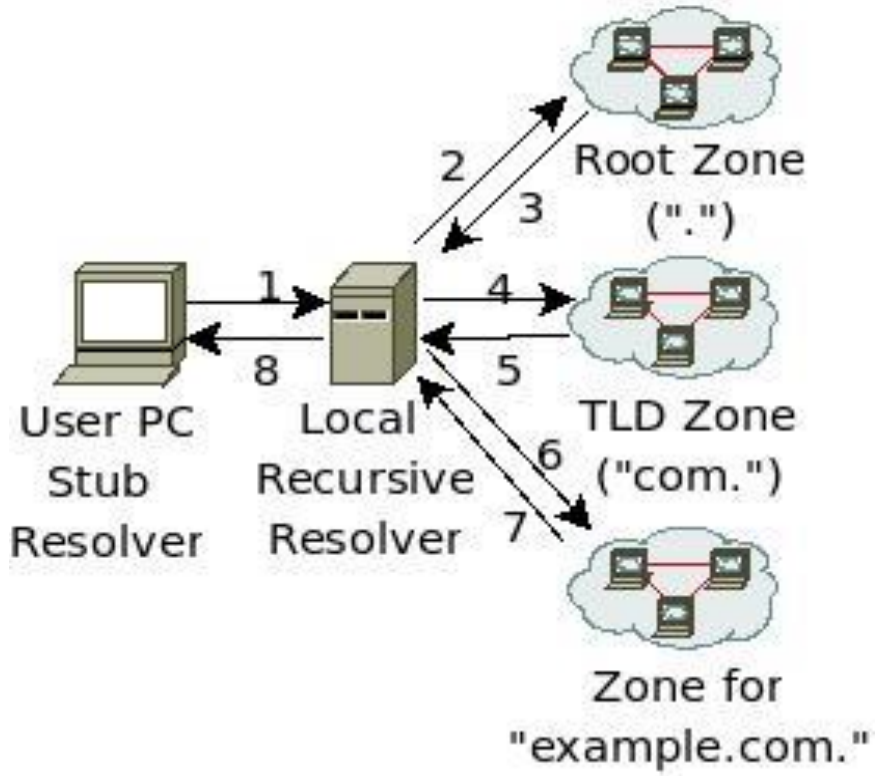
- To validate a route from AS_n, AS_{n+1} needs:
  - address attestation from each organization owning an address block(s) in the NLRI
  - address allocation certificate from each organization owning address blocks in the NLRI
  - route attestation from every AS along the path (AS_1 to AS_n), where the route attestation for AS_k specifies the NLRI and the path up to that point (AS_1 through AS_{k+1})
  - certificate for each AS or router along path (AS_1 to AS_n) to check signatures on the route attestations
  - and, of course, all the relevant CRLs must have been checked

Slide: Kent et al.

Infrastructure protocols: DNS

Recall: DNS Lookup Query: "www.example.com A?"

Reply	Resource Records in Reply
3	"com. NS a.gtld.net" "a.gtld.net A 192.5.6.30"
5	"example.com. NS a.iana.net" "a.iana.net A 192.0.34.43"
7	"www.example.com A 1.2.3.4"
8	"www.example.com A 1.2.3.4"



Local recursive resolver caches these for TTL specified by RR

DNS is Insecure

- Packets sent over UDP, < 512 bytes
- 16-bit TXID, UDP Src port are only “security”
- Resolver accepts packet if above match
- Packet from whom? Was it manipulated?

- Cache poisoning
  - Attacker forges record at resolver
  - Forged record cached, attacks future lookups
  - Kaminsky (BH USA08)
    - Attacks delegations with “birthday problem”

DNSSEC Goal

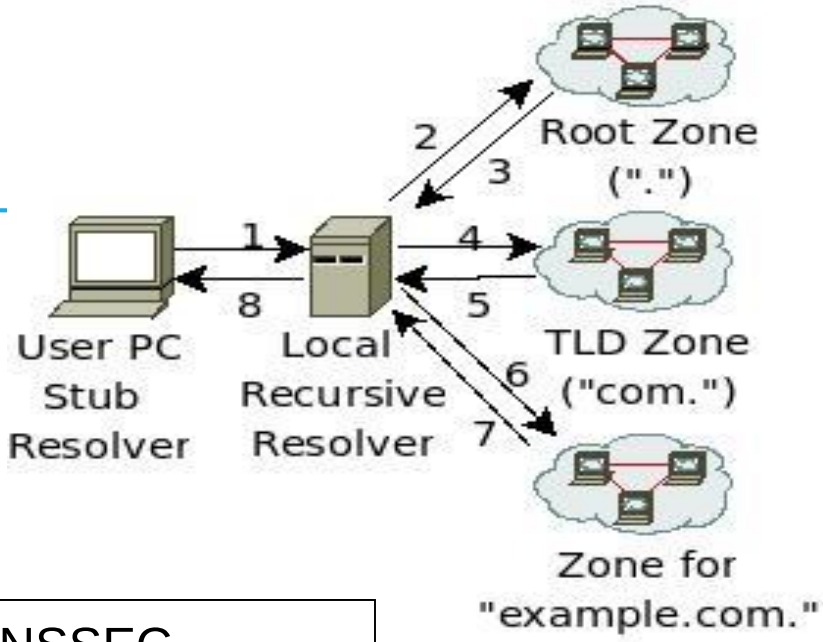
"The Domain Name System (DNS) security extensions provide origin authentication and integrity assurance services for DNS data, including mechanisms for authenticated denial of existence of DNS data."

-RFC 4033

DNSSEC

- Basically no change to packet format
  - Goal is security of DNS data, not channel security
- New Resource Records (RRs)
  - RRSIG : signature of RR by private zone key
  - DNSKEY : public zone key
  - DS : crypto digest of child zone key
  - NSEC / NSEC3 authenticated denial of existence
- Lookup referral chain (unsigned)
- Origin attestation chain (PKI) (signed)
  - Start at pre-configured trust anchors
    - DS/DNSKEY of zone (should include root)
  - DS → DNSKEY → DS forms a link

DNSSEC Lookup



Query: "www.example.com A?"

Reply	RRs in DNS Reply	Added by DNSSEC
3	"com. NS a.gtld.net" "a.gtld.net A 192.5.6.30"	"com. DS" "RRSIG(DS) by ."
5	"example.com. NS a.iana.net" "a.iana.net A 192.0.34.43"	"com. DNSKEY" "RRSIG(DNSKEY) by com." "example.com. DS" "RRSIG(DS) by com."
7	"www.example.com A 1.2.3.4"	"example.com DNSKEY" "RRSIG(DNSKEY) by example.com." "RRSIG(A) by example.com."
8	"www.example.com A 1.2.3.4"	Last Hop?

Authenticated Denial-of-Existence

- Most DNS lookups result in denial-of-existence
- NSEC (Next SECure)
  - Lists all extant RRs associated with an owner name
  - Points to next owner name with extant RR
  - Easy zone enumeration
- NSEC3
  - Hashes owner names
    - Public salt to prevent pre-computed dictionaries
  - NSEC3 chain in hashed order
  - Opt-out bit for TLDs to support incremental adoption
    - For TLD type zones to support incremental adoption
    - Non-DNSSEC children not in NSEC3 chain

Insecure Sub-Namespace

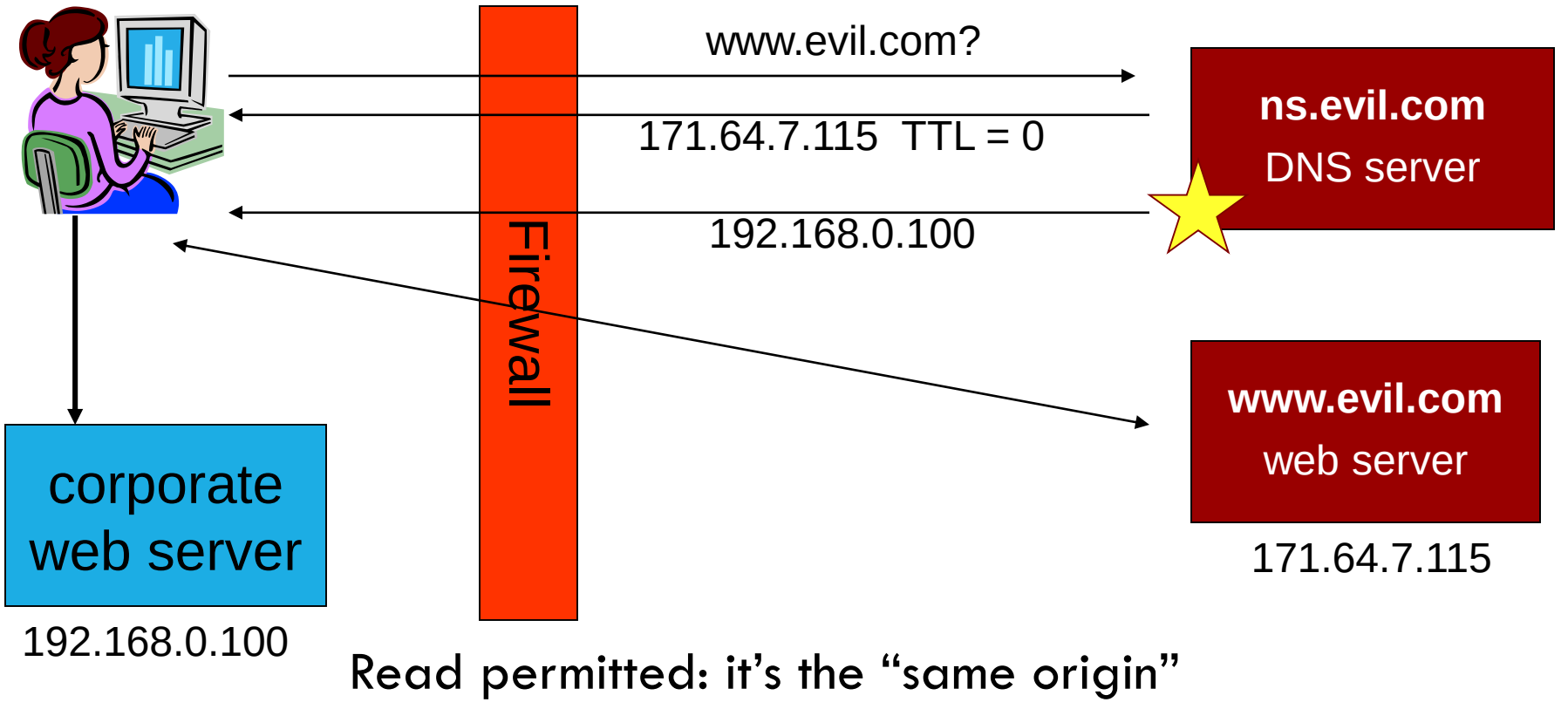
- NSEC3 Opt-out
  - "Does not assert the existence or non-existence of the insecure delegations that it may cover" (RFC 5155)
  - Only thing asserting this is insecure glue records
- Property: Possible to insert bogus pre-pended name into otherwise secure zone. (RFC 5155)
- Insecure delegation from secure zone
  - Spoofs possible for resultant lookup results
- Acceptable for TLD, bad for enterprises

[DWF'96, R'01]

DNS Rebinding Attack

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

DNSSEC 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

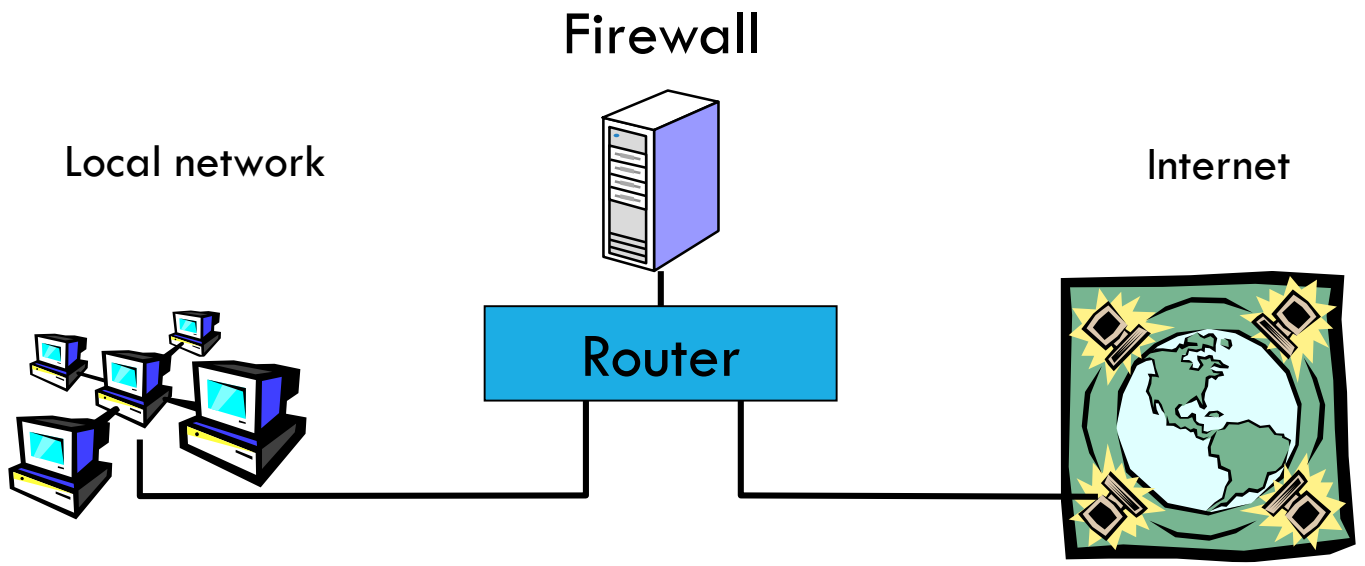
Part 3-2

Standard network defenses

Mostly based on John Mitchell Slides

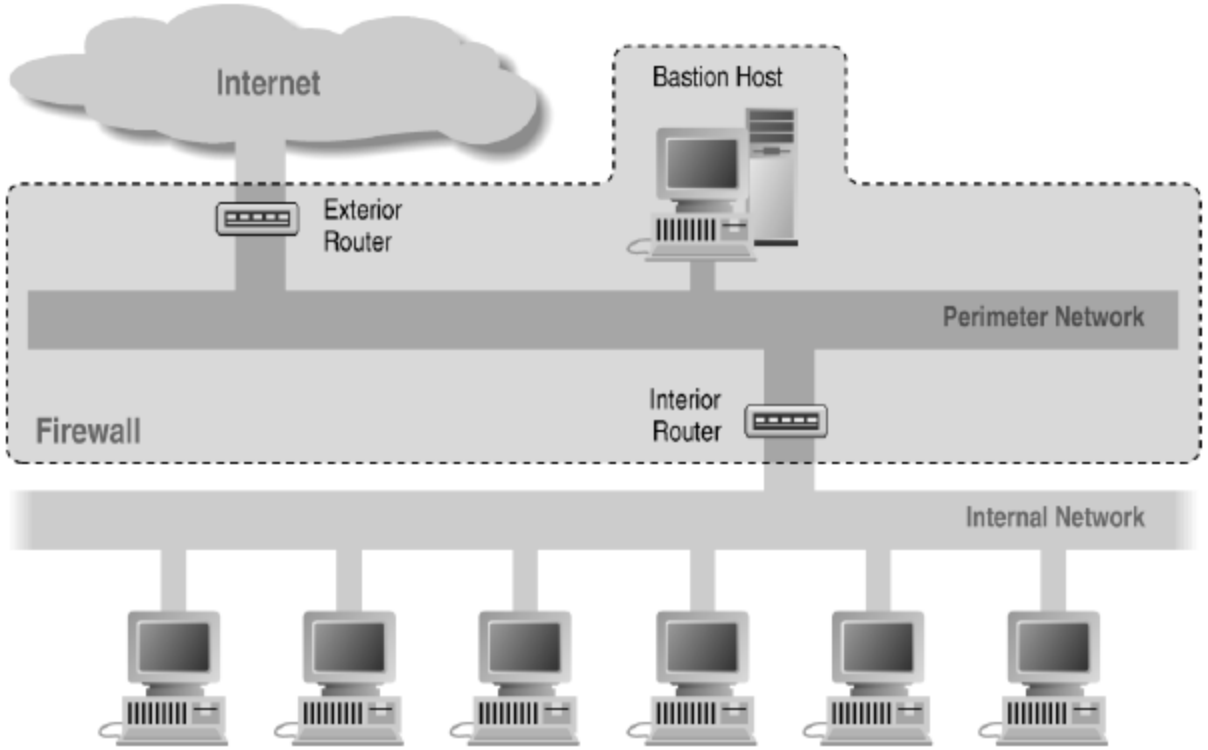
Basic Firewall Concept

- Separate local area net from internet

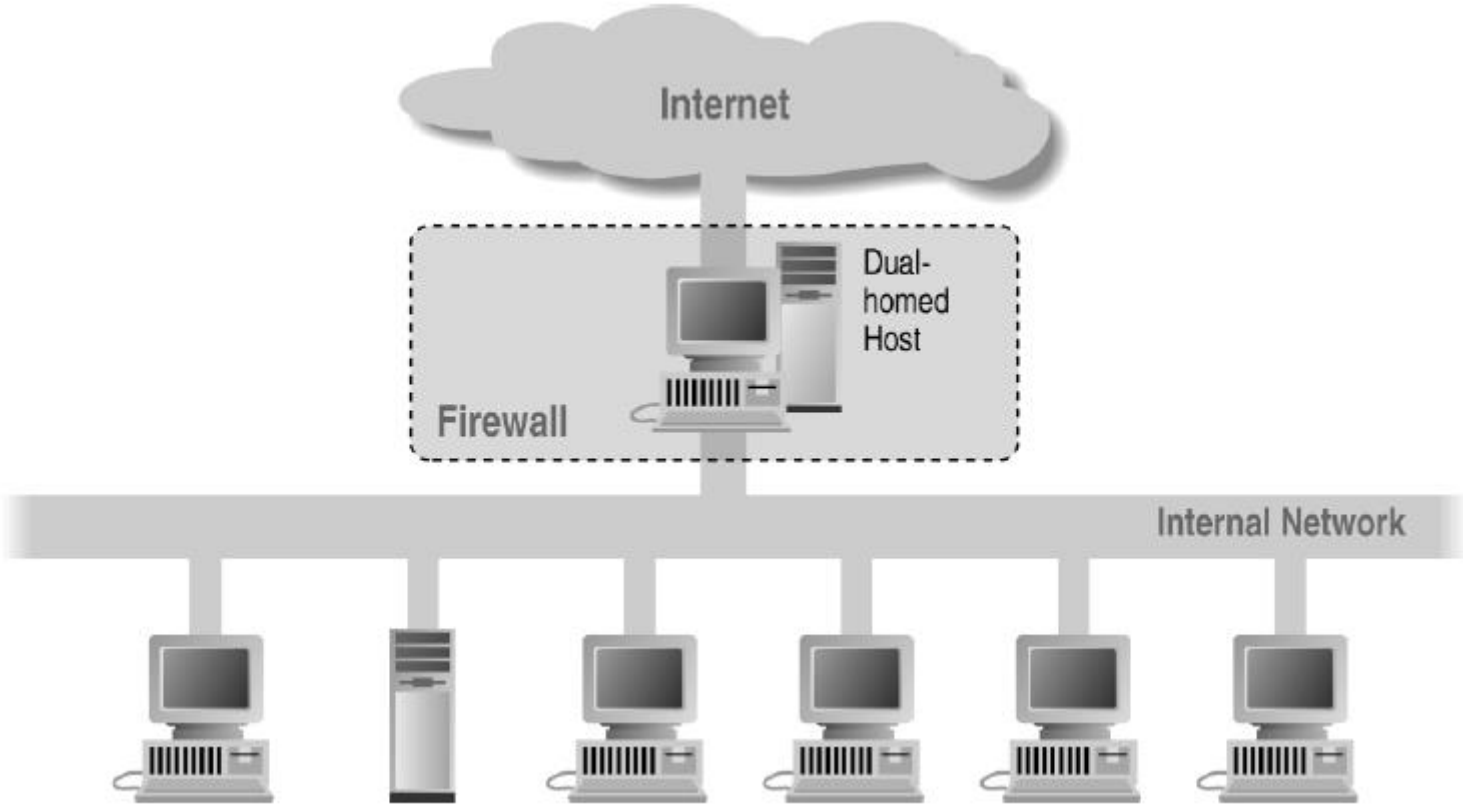


All packets between LAN and internet routed through firewall

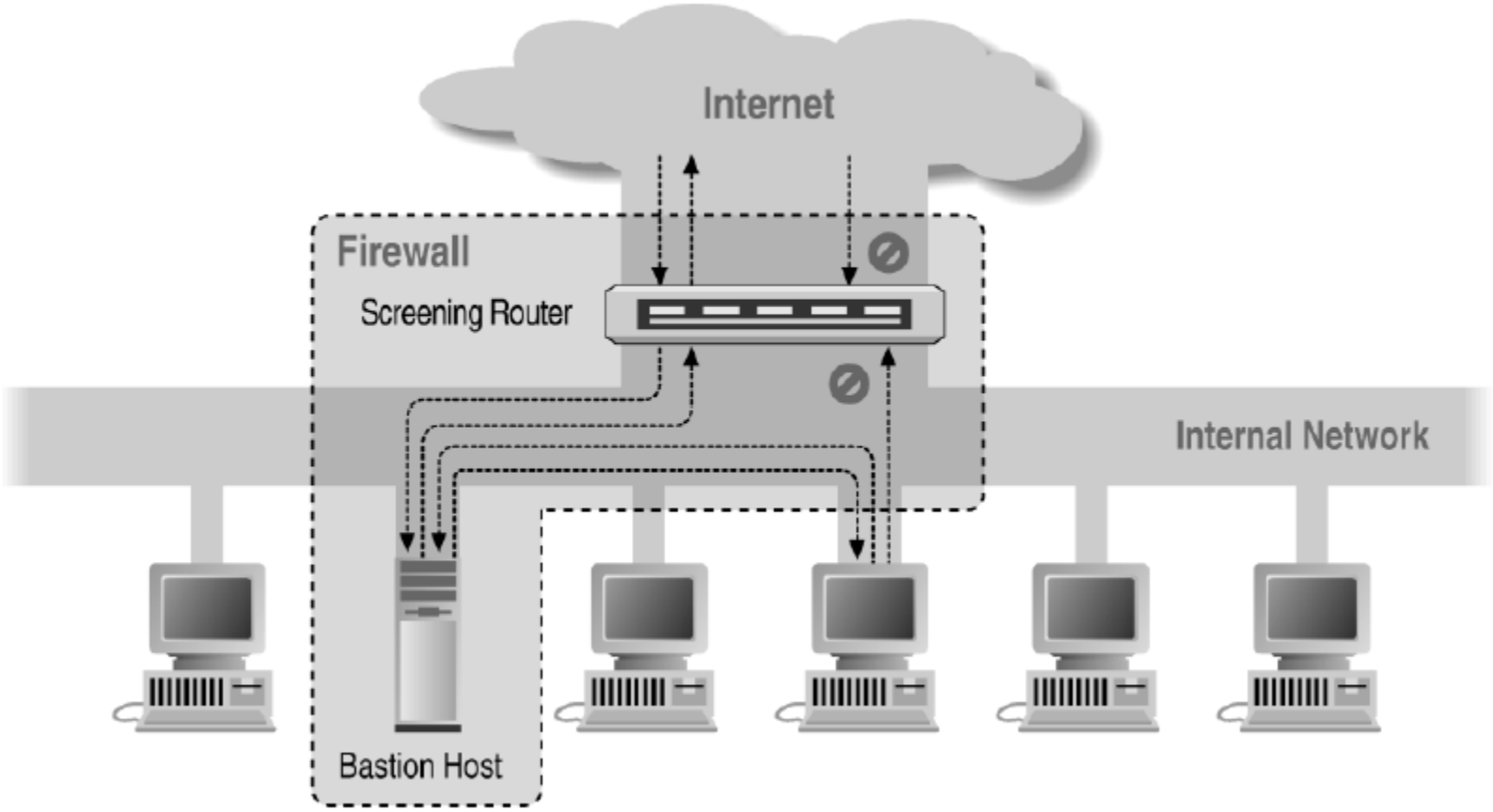
Screened Subnet Using Two Routers



Alternate 1: Dual-Homed Host



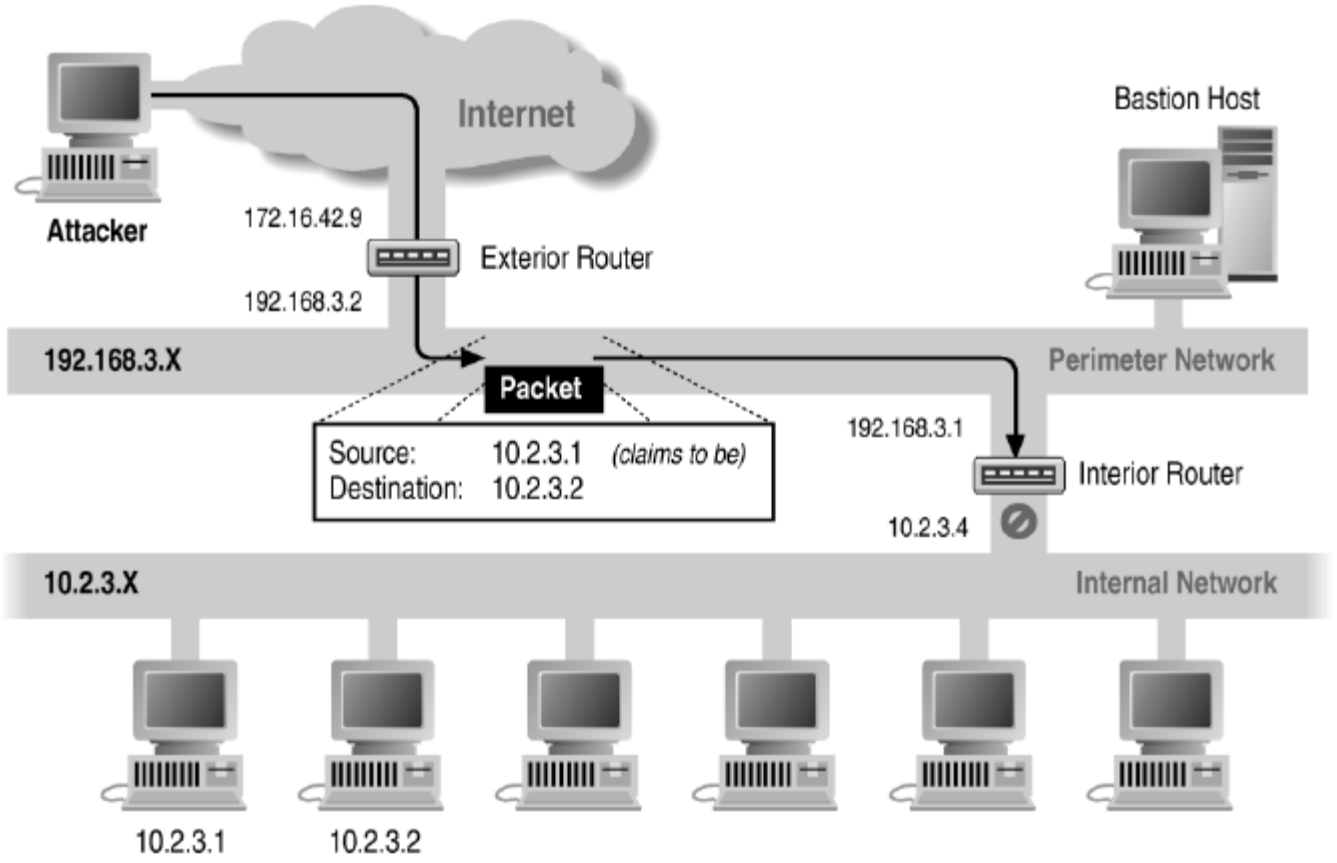
Alternate 2: Screened Host



Basic Packet Filtering

- Uses transport-layer information only
  - IP Source Address, Destination Address
  - Protocol (TCP, UDP, ICMP, etc)
  - TCP or UDP source & destination ports
  - TCP Flags (SYN, ACK, FIN, RST, PSH, etc)
  - ICMP message type
- Examples
  - DNS uses port 53
    - Block incoming port 53 packets except known trusted servers
- Issues
  - Stateful filtering
  - Encapsulation: address translation, other complications
  - Fragmentation

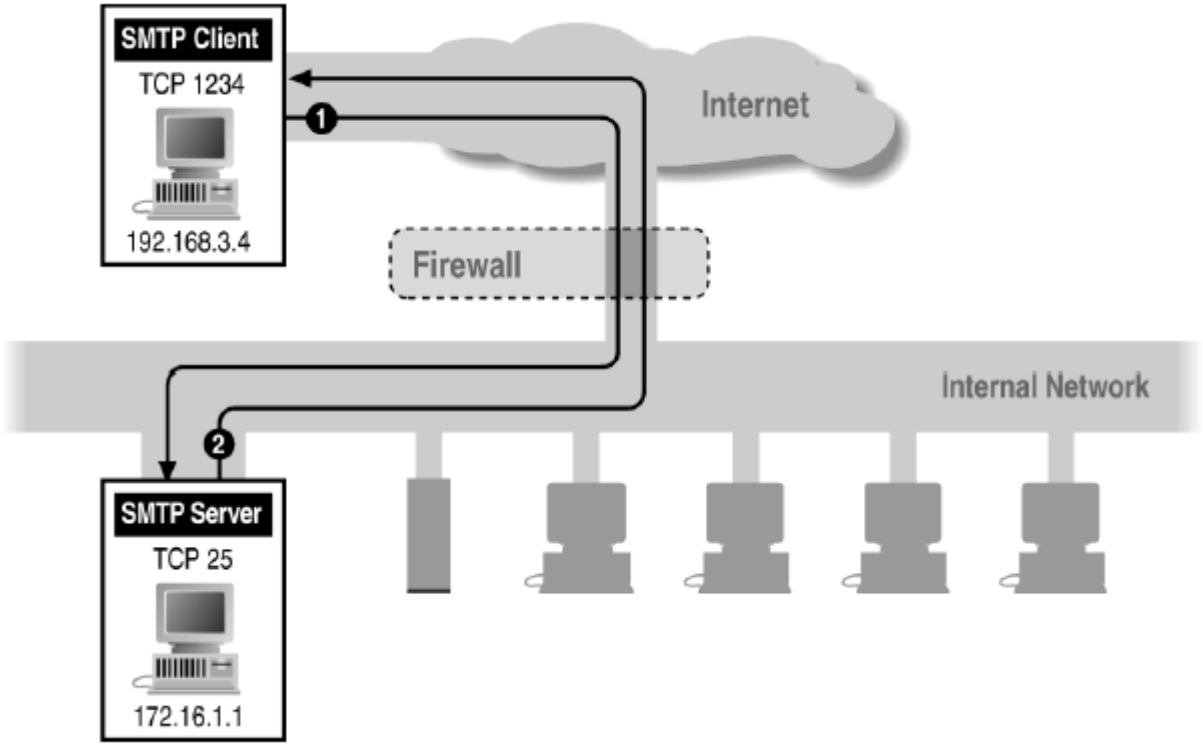
Source/Destination Address Forgery



More about networking: port numbering

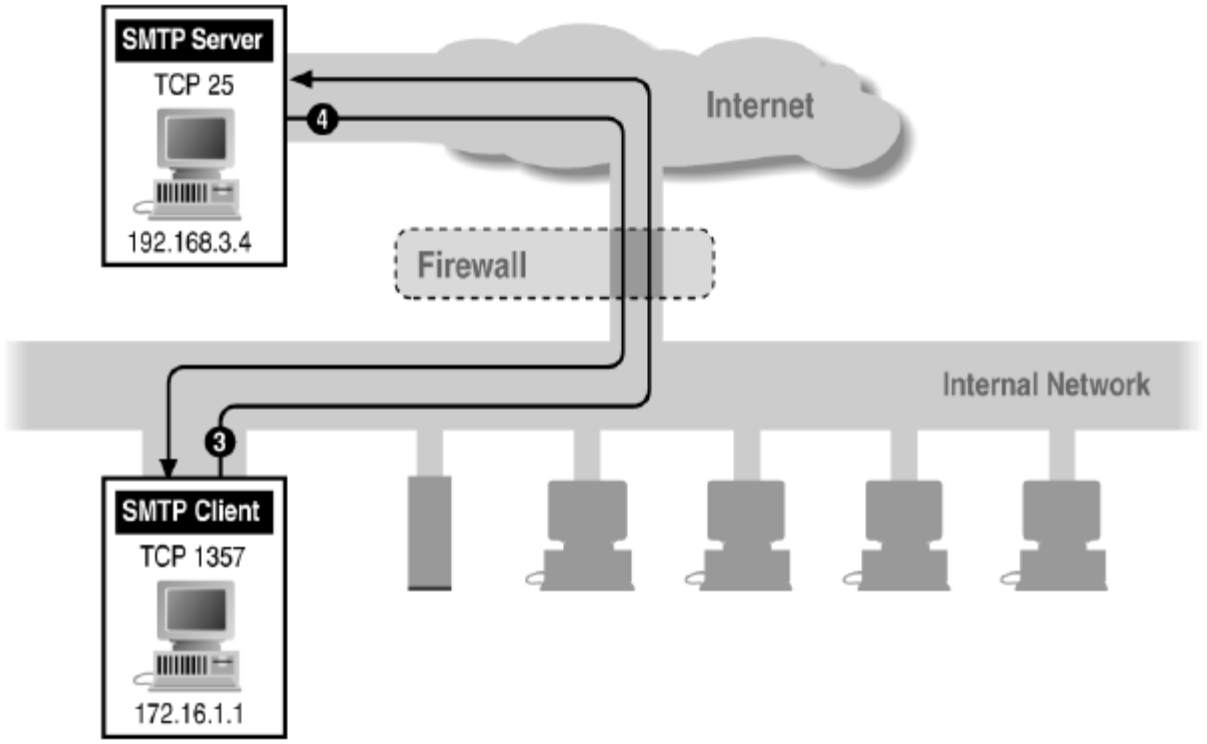
- TCP connection
  - Server port uses number less than 1024
  - Client port uses number between 1024 and 16383
- Permanent assignment
  - Ports <1024 assigned permanently
    - 20,21 for FTP 23 for Telnet
    - 25 for server SMTP 80 for HTTP
- Variable use
  - Ports >1024 must be available for client to make connection
  - Limitation for stateless packet filtering
    - If client wants port 2048, firewall must allow incoming traffic
  - Better: stateful filtering knows outgoing requests
    - Only allow incoming traffic on high port to a machine that has initiated an outgoing request on low port

Filtering Example: Inbound SMTP



Can block external request to internal server based on port number

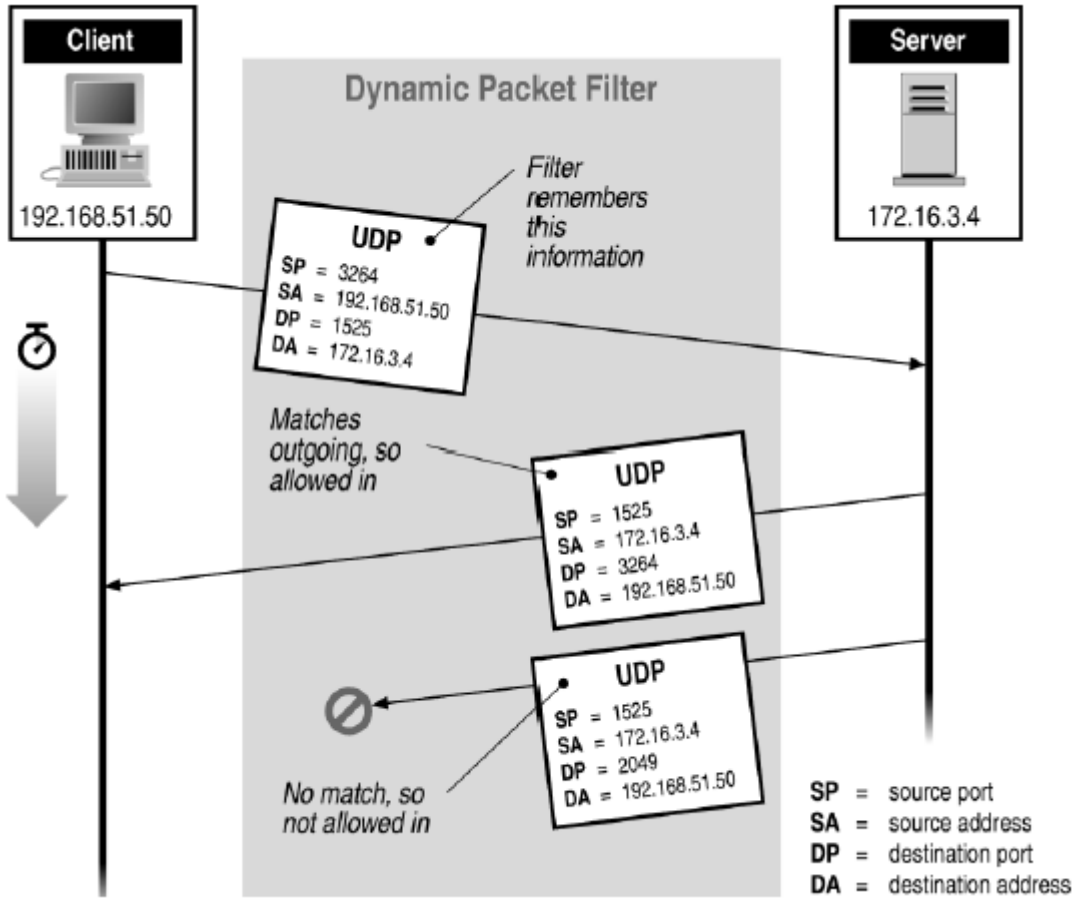
Filtering Example: Outbound SMTP



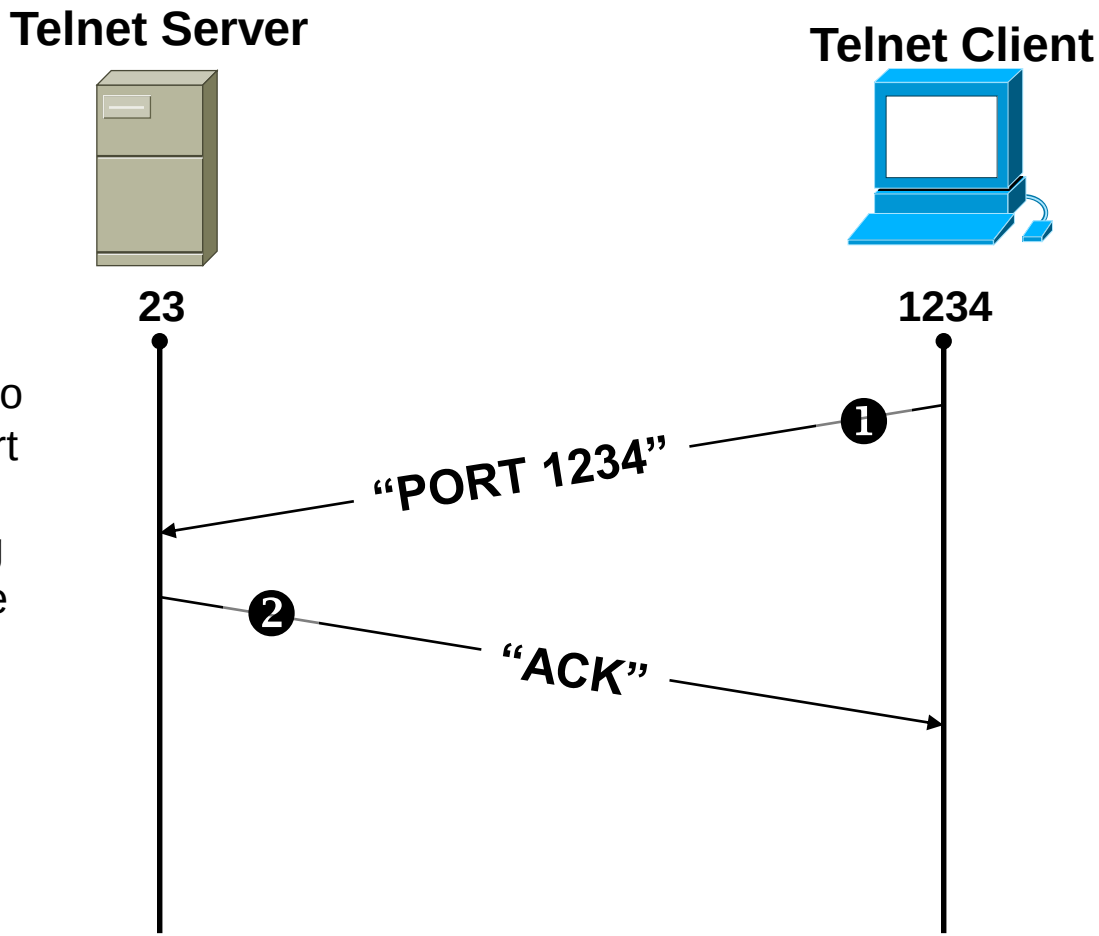
Known low port out, arbitrary high port in

If firewall blocks incoming port 1357 traffic then connection fails

Stateful or Dynamic Packet Filtering



Telnet



❶ Client opens channel to server; tells server its port number. The ACK bit is not set while establishing the connection but will be set on the remaining packets

❷ Server acknowledges

Stateful filtering can use this pattern to identify legitimate sessions

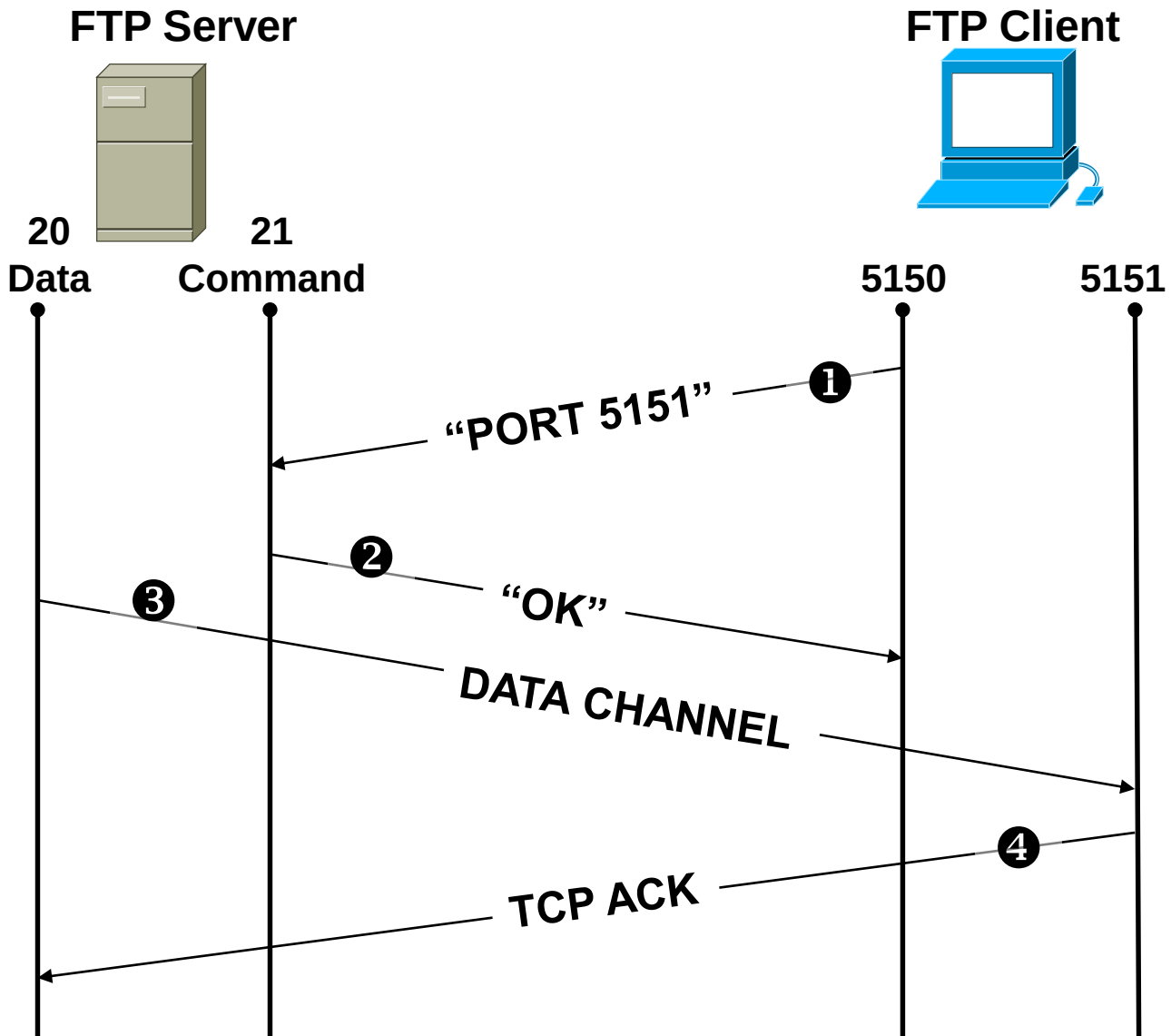
FTP

❶ Client opens command channel to server; tells server second port number

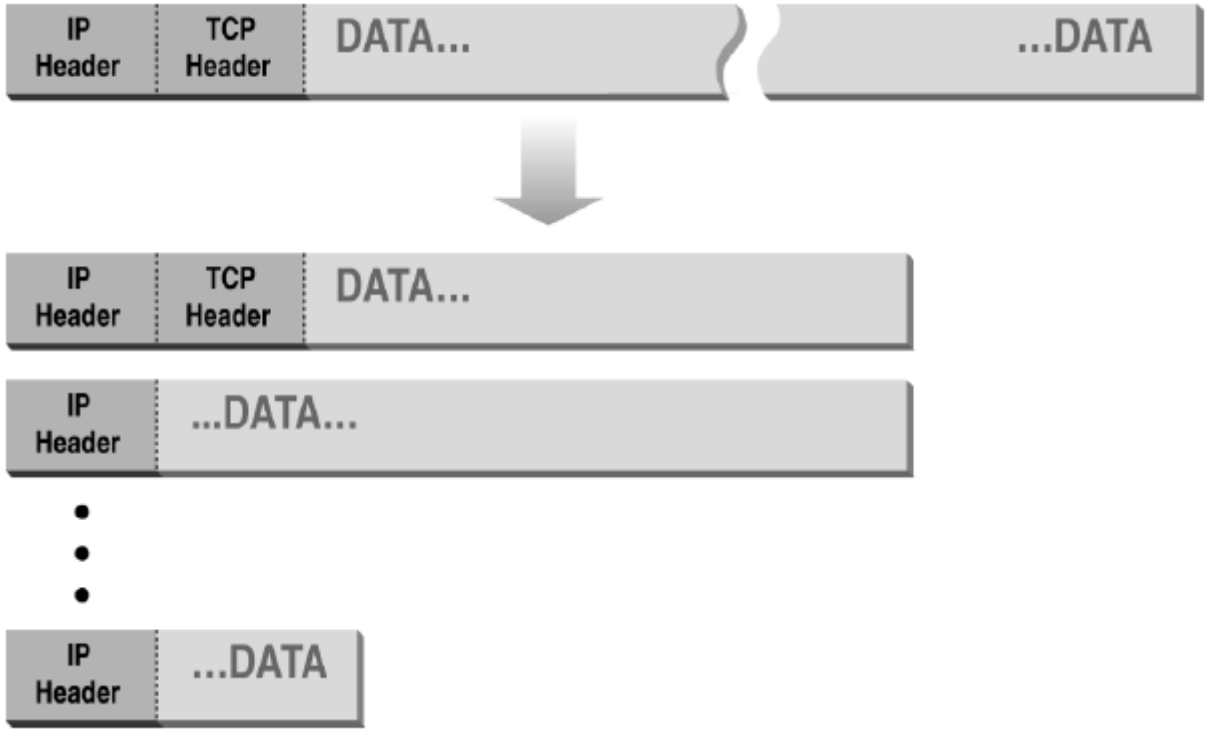
❷ Server acknowledges

❸ Server opens data channel to client's second port

❹ Client acknowledges

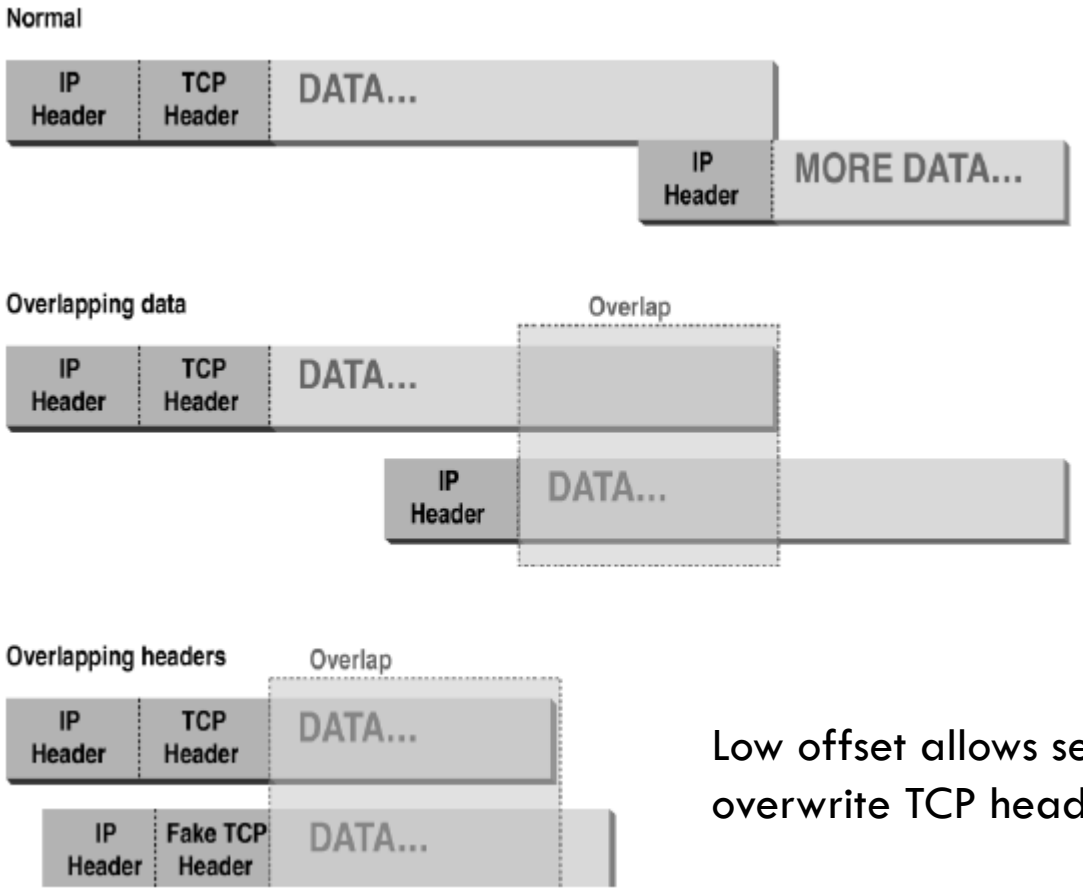


Normal IP Fragmentation



Flags and offset inside IP header indicate packet fragmentation

Abnormal Fragmentation

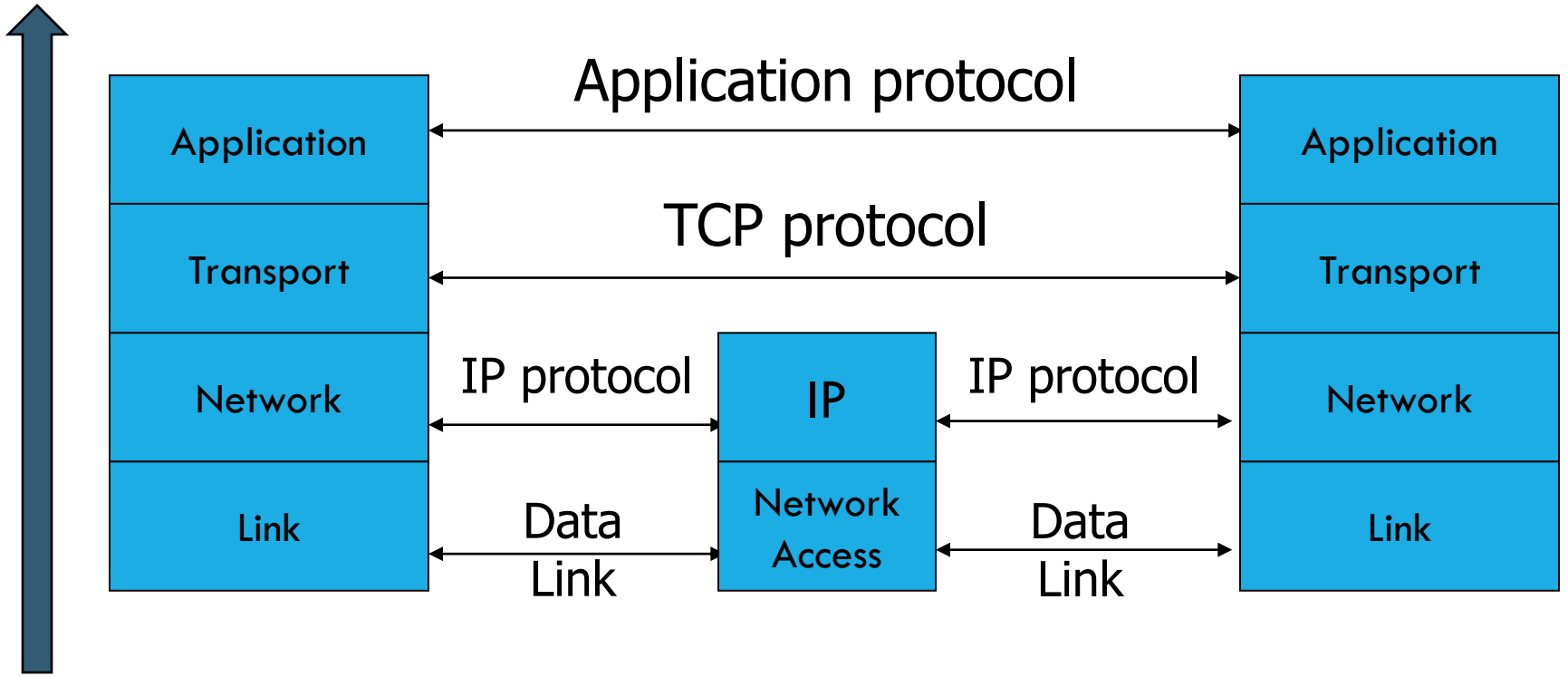


Low offset allows second packet to overwrite TCP header at receiving host

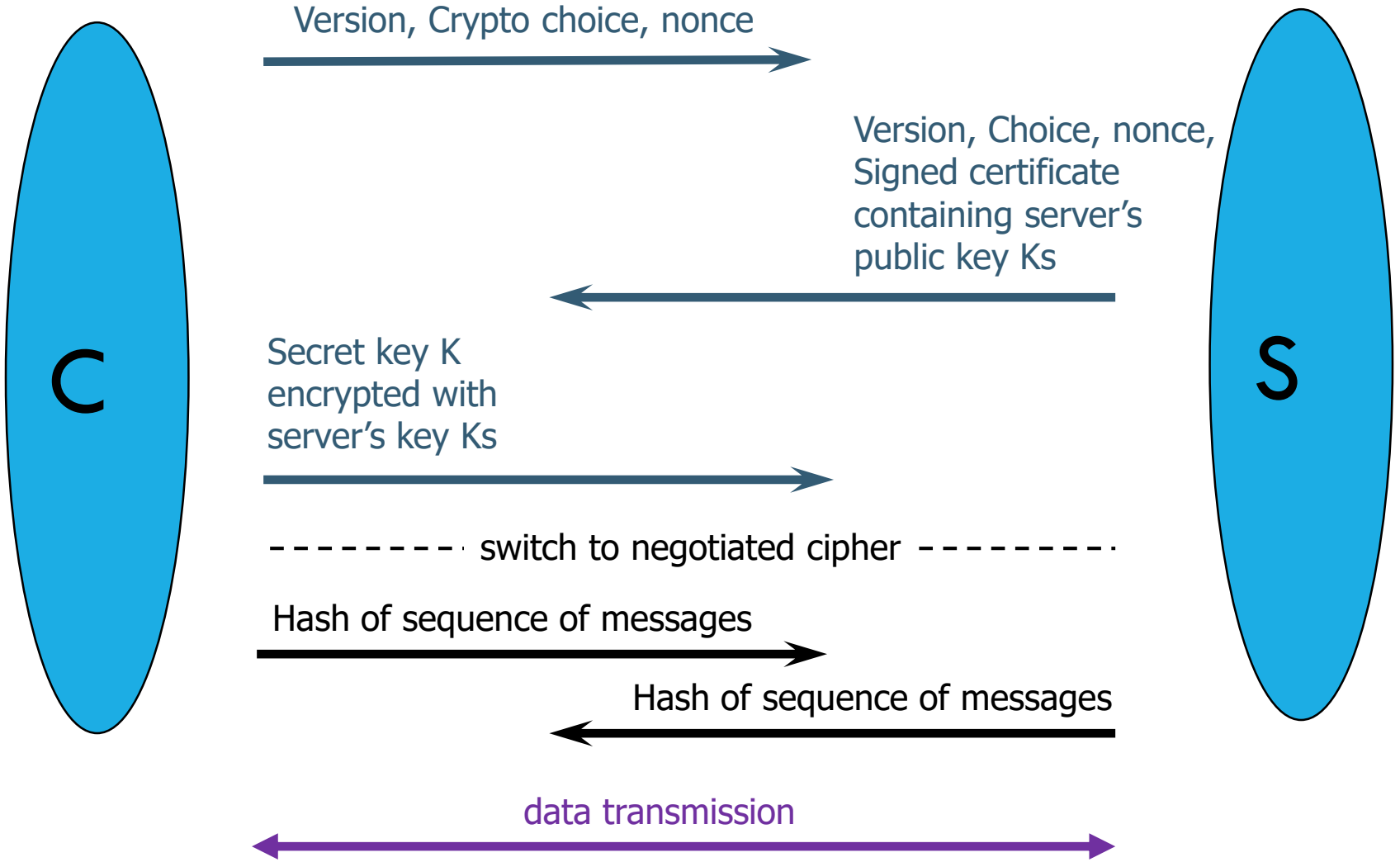
Packet Fragmentation Attack

- Firewall configuration
  - TCP port 23 is blocked but SMTP port 25 is allowed
- First packet
  - Fragmentation Offset = 0.
  - DF bit = 0 : "May Fragment"
  - MF bit = 1 : "More Fragments"
  - Destination Port = 25. TCP port 25 is allowed, so firewall allows packet
- Second packet
  - Fragmentation Offset = 1: second packet overwrites all but first 8 bits of the first packet
  - DF bit = 0 : "May Fragment"
  - MF bit = 0 : "Last Fragment."
  - Destination Port = 23. Normally be blocked, but sneaks by!
- What happens
  - Firewall ignores second packet “TCP header” because it is fragment of first
  - At host, packet reassembled and received at port 23

TCP Protocol Stack



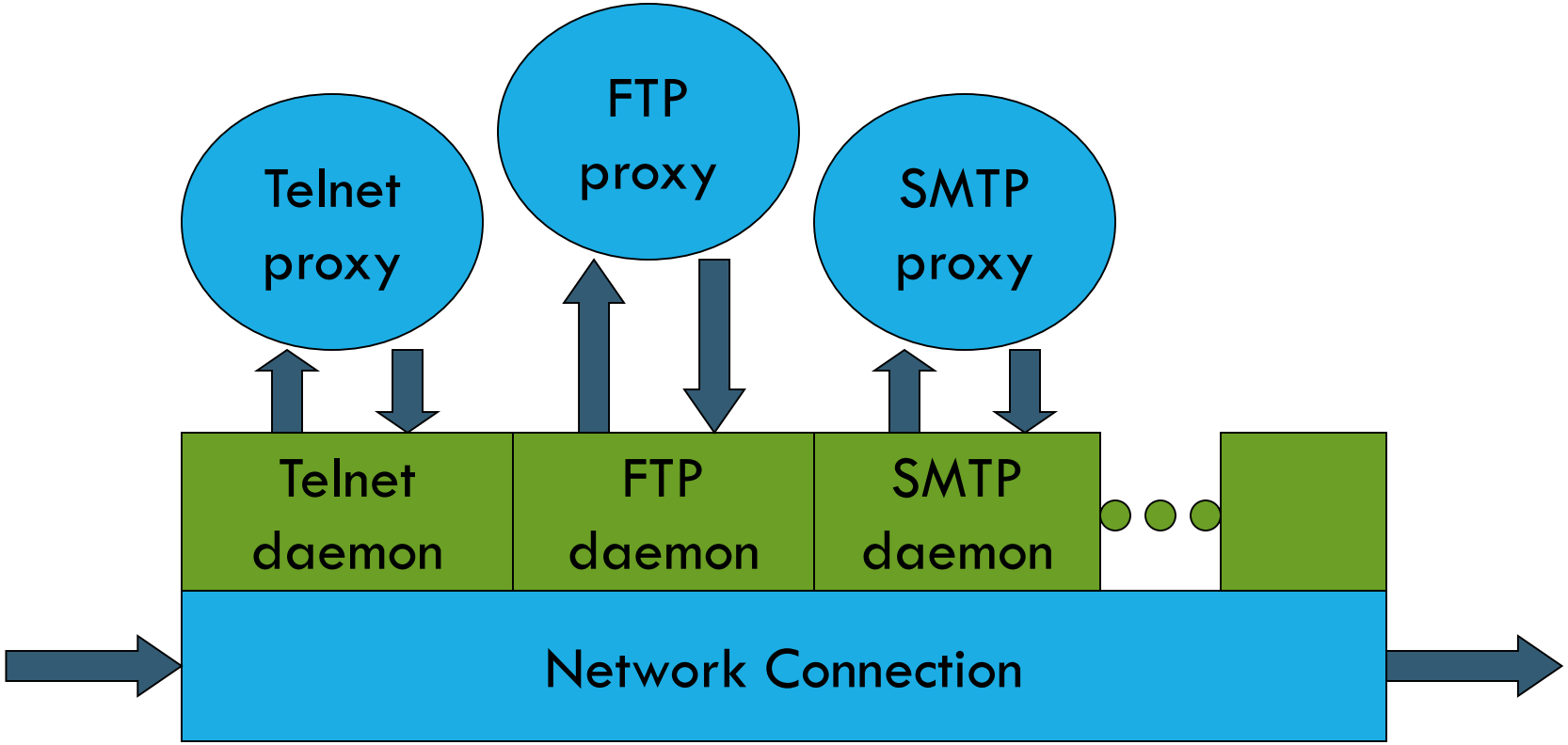
Remember SSL/TLS



Proxying Firewall

- Application-level proxies
  - Tailored to http, ftp, smtp, etc.
  - Some protocols easier to proxy than others
- Policy embedded in proxy programs
  - Proxies filter incoming, outgoing packets
  - Reconstruct application-layer messages
  - Can filter specific application-layer commands, etc.
    - Example: only allow specific ftp commands
    - Other examples: ?
- Several network locations – see next slides

Firewall with application proxies



Daemon spawns proxy when communication detected …

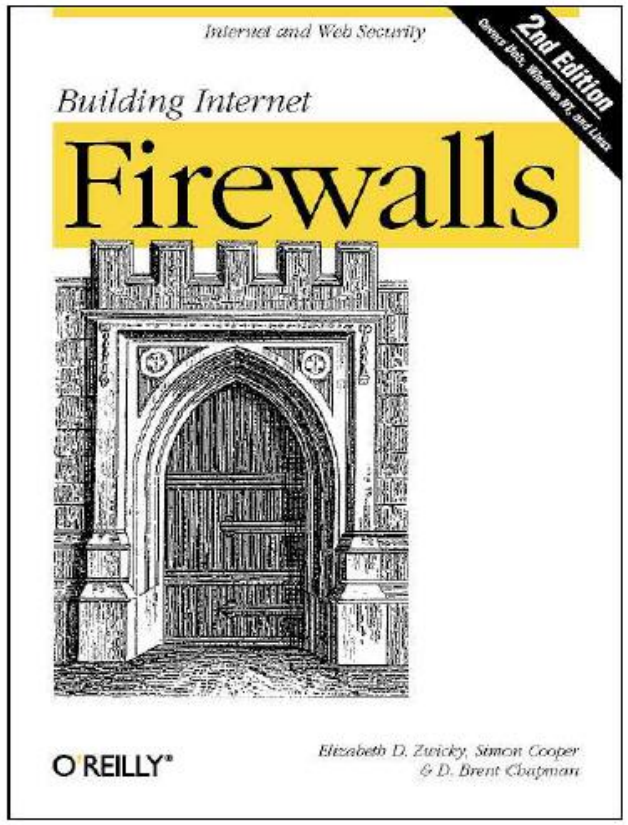
Application-level proxies

- Enforce policy for specific protocols
  - E.g., Virus scanning for SMTP
    - Need to understand MIME, encoding, Zip archives
  - Flexible approach, but may introduce network delays
- "Batch" protocols are natural to proxy
  - SMTP (E-Mail)
  - NNTP (Net news)
  - DNS (Domain Name System)
  - NTP (Network Time Protocol
- Must protect host running protocol stack
  - Disable all non-required services; keep it simple
  - Install/modify services you want
  - Run security audit to establish baseline
  - Be prepared for the system to be compromised

Web traffic scanning

- Intercept and proxy web traffic
  - Can be host-based
  - Usually at enterprise gateway
- Block known bad sites
- Block pages with known attacks
- Scan attachments
  - Usually traditional virus scanning methods

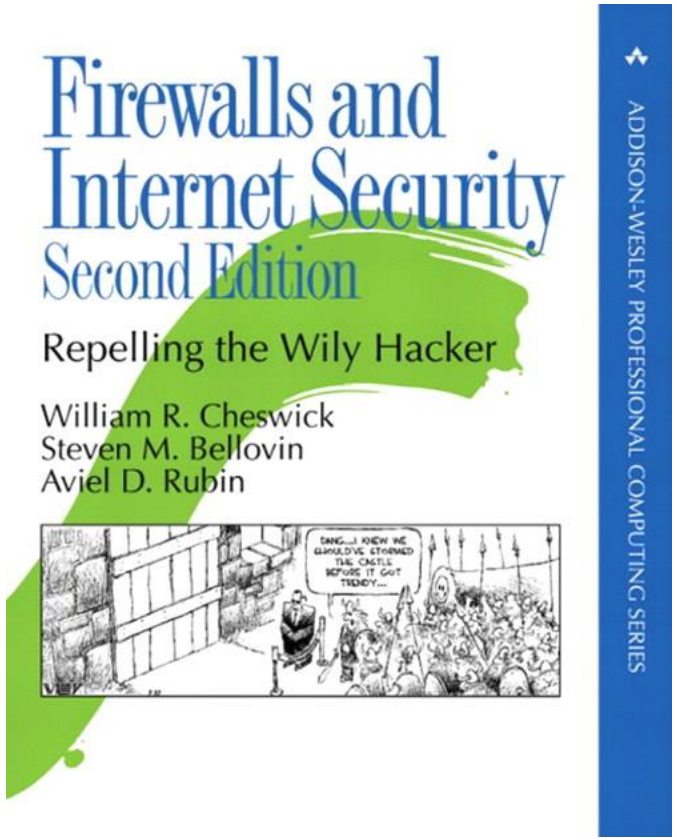
Firewall references



Elizabeth D. Zwicky

Simon Cooper

D. Brent Chapman



William R Cheswick

Steven M Bellovin

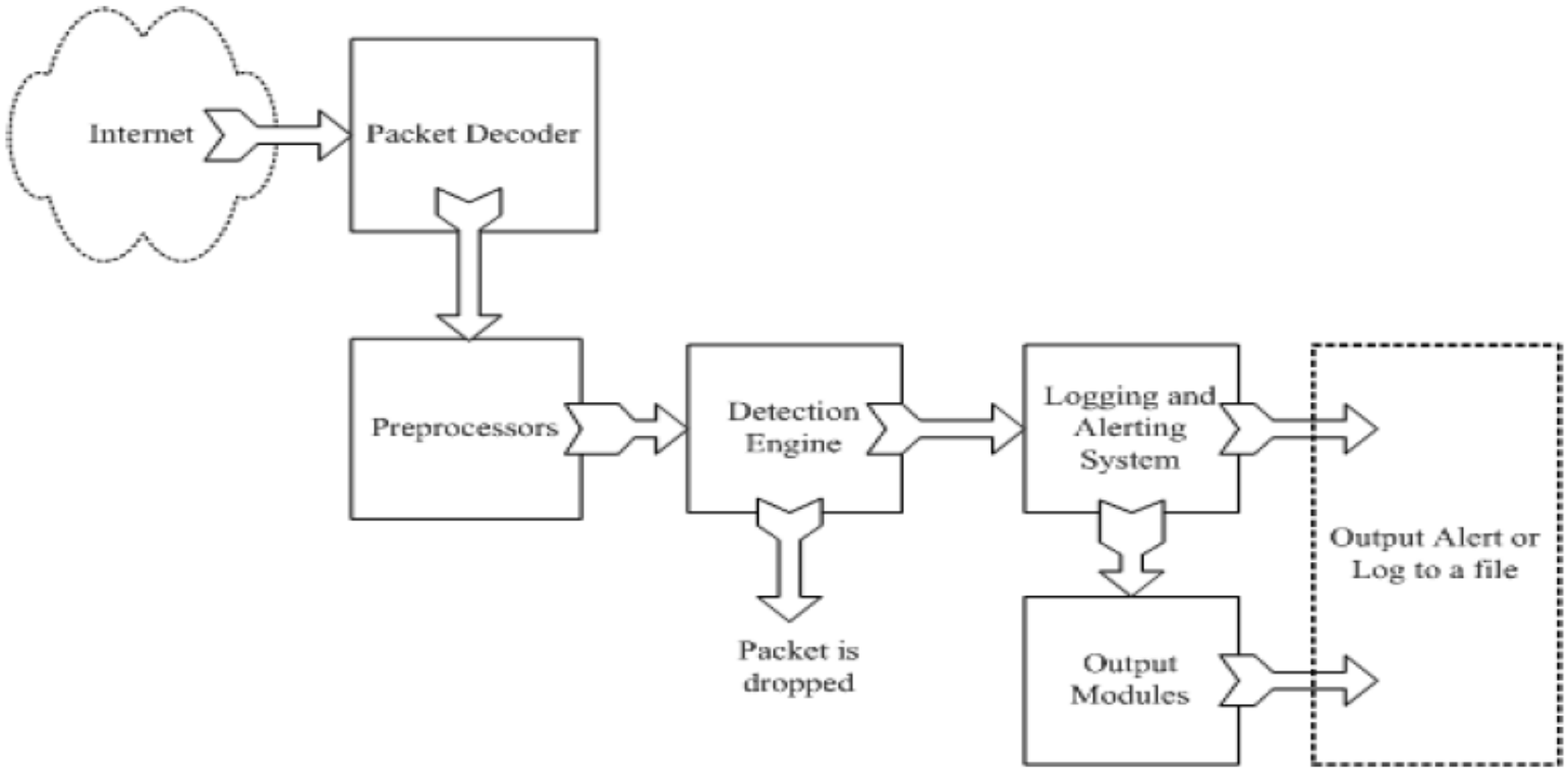
Aviel D Rubin

Intrusion detection

- Many intrusion detection systems
  - Close to 100 systems with current web pages
  - Network-based, host-based, or combination
- Two basic models
  - Misuse detection model
    - Maintain data on known attacks
    - Look for activity with corresponding signatures
  - Anomaly detection model
    - Try to figure out what is “normal”
    - Report anomalous behavior
- Fundamental problem: too many false alarms

Example: Snort

http://www.snort.org/

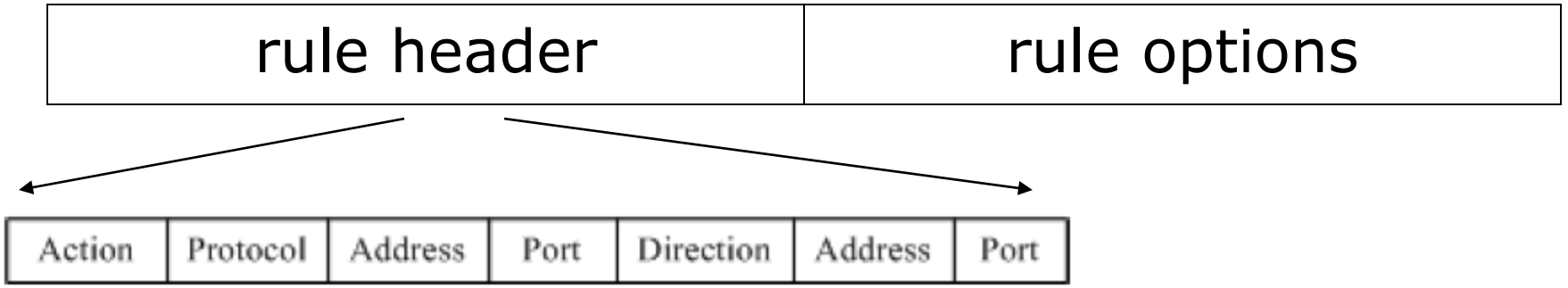


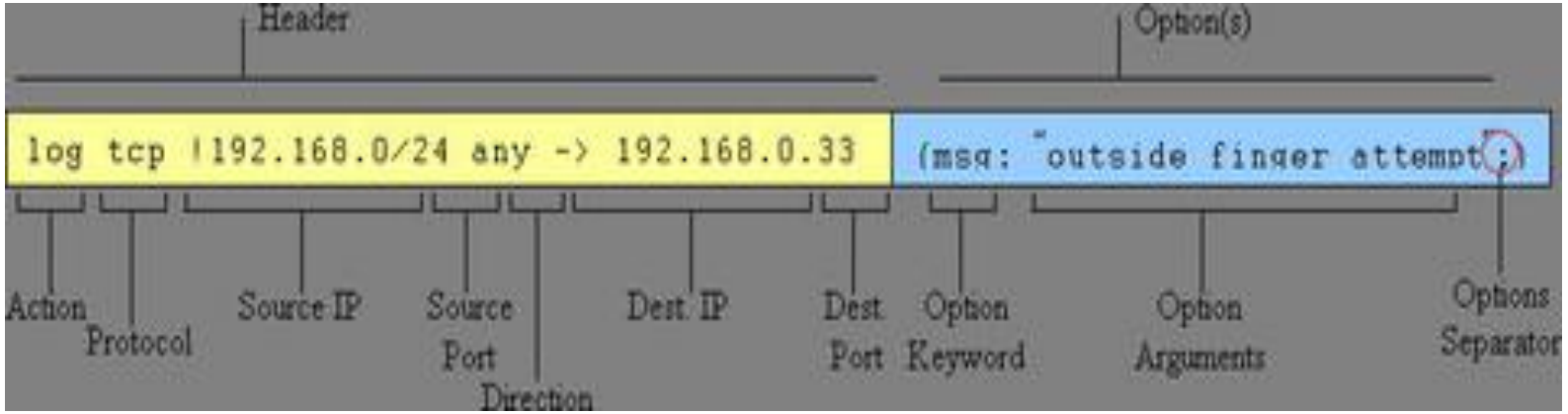
From: Rafeeq Ur Rehman, *Intrusion Detection Systems with Snort: Advanced IDS Techniques with Snort, Apache, MySQL, PHP, and ACID.*

Snort components

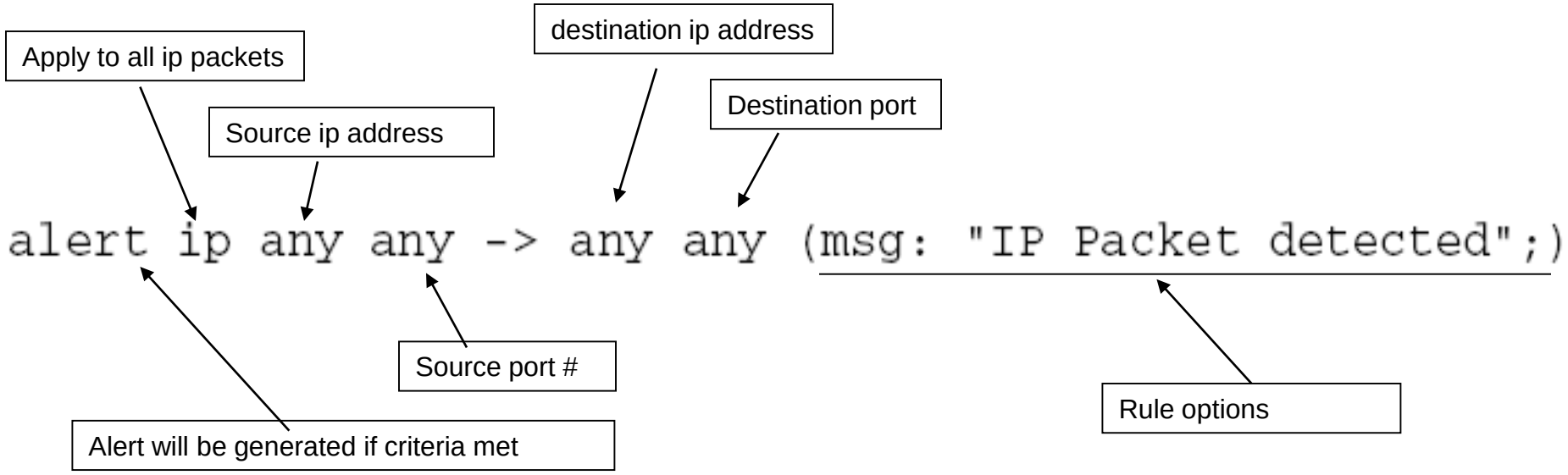
- Packet Decoder
  - input from Ethernet, SLIP, PPP…
- Preprocessor:
  - detect anomalies in packet headers
  - packet defragmentation
  - decode HTTP URI
  - reassemble TCP streams
- Detection Engine: applies rules to packets
- Logging and Alerting System
- Output Modules: alerts, log, other output

Snort detection rules





Additional examples




```
alert tcp $TELNET_SERVERS 23 -> $EXTERNAL_NET any (msg:"TELNET
  Attempted SU from wrong group"; flow:
from_server,established; content:"to su root"; nocase;
  classtype:attempted-admin; sid:715; rev:6;)
```

Snort challenges

- Misuse detection – avoid known intrusions
  - Database size continues to grow
    - Snort version 2.3.2 had 2,600 rules
  - Snort spends 80% of time doing string match

- Anomaly detection – identify new attacks
  - Probability of detection is low

Difficulties in anomaly detection

- Lack of training data
  - Lots of “normal” network, system call data
  - Little data containing realistic attacks, anomalies
- Data drift
  - Statistical methods detect changes in behavior
  - Attacker can attack gradually and incrementally
- Main characteristics not well understood
  - By many measures, attack may be within bounds of “normal” range of activities
- False identifications are very costly
  - Sys Admin spend many hours examining evidence

Summary

- Network protocol security
  - IPSEC
  - BGP instability and S-BGP
  - DNSSEC, DNS rebinding
- Standard network perimeter defenses
  - Firewall
    - Packet filter (stateless, stateful), Application layer proxies
  - Traffic shaping
  - Intrusion detection
    - Anomaly and misuse detection

Part 4-1

Unwanted Traffic: Denial of Service Attacks

Mostly based on Dan Boneh Slides

What is network DoS?

- Goal: take out a large site with little computing work
- How: **Amplification**
  - Small number of packets $\Rightarrow$ big effect
- Two types of amplification attacks:
  - DoS bug:
    - Design flaw allowing one machine to disrupt a service
  - DoS flood:
    - Command bot-net to generate flood of requests

DoS can happen at any layer

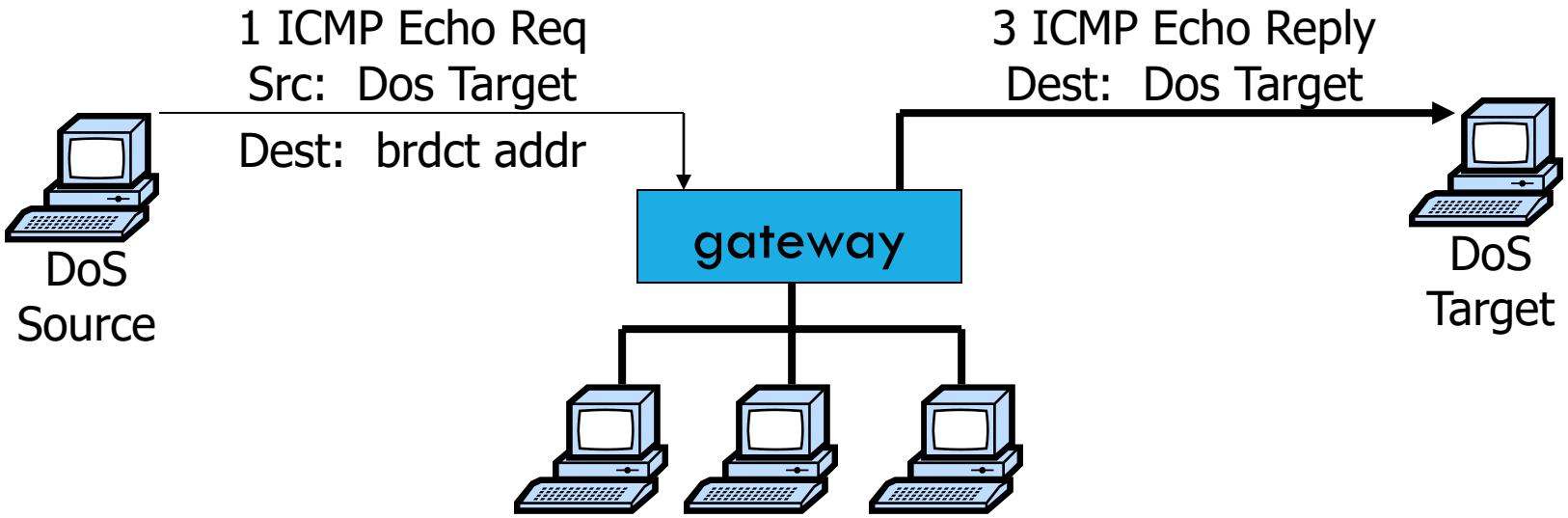
- This lecture:
  - Sample Dos at different layers (by order):
    - Link
    - TCP/UDP
    - Application
  - Generic DoS solutions
  - Network DoS solutions
- Sad truth:
  - Current Internet not designed to handle DDoS attacks

Warm up: 802.11b DoS bugs

- Radio jamming attacks: trivial, not our focus.
- Protocol DoS bugs: [Bellardo, Savage, '03]
  - NAV (Network Allocation Vector):
    - 15-bit field. Max value: 32767
    - Any node can reserve channel for NAV seconds
    - No one else should transmit during NAV period
    - … but not followed by most 802.11b cards

  - De-authentication bug:
    - Any node can send deauth packet to AP
    - Deauth packet unauthenticated
    - … attacker can repeatedly deauth anyone

Smurf amplification DoS attack

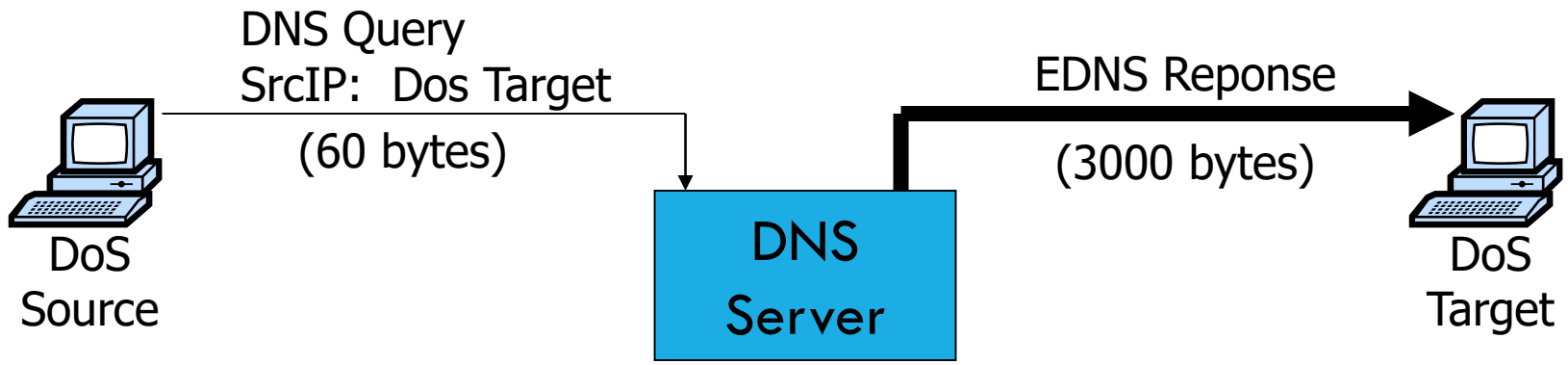


- Send ping request to broadcast addr (ICMP Echo Req)
- Lots of responses:
  - Every host on target network generates a ping reply (ICMP Echo Reply) to victim

Prevention: reject external packets to broadcast address

Modern day example

DNS Amplification attack: (×50 amplification)



2006: 0.58M open resolvers on Internet (Kaminsky-Shiffman)

2014: 28M open resolvers (openresolverproject.org)

⇒ 3/2013: DDoS attack generating 309 Gbps for 28 mins.

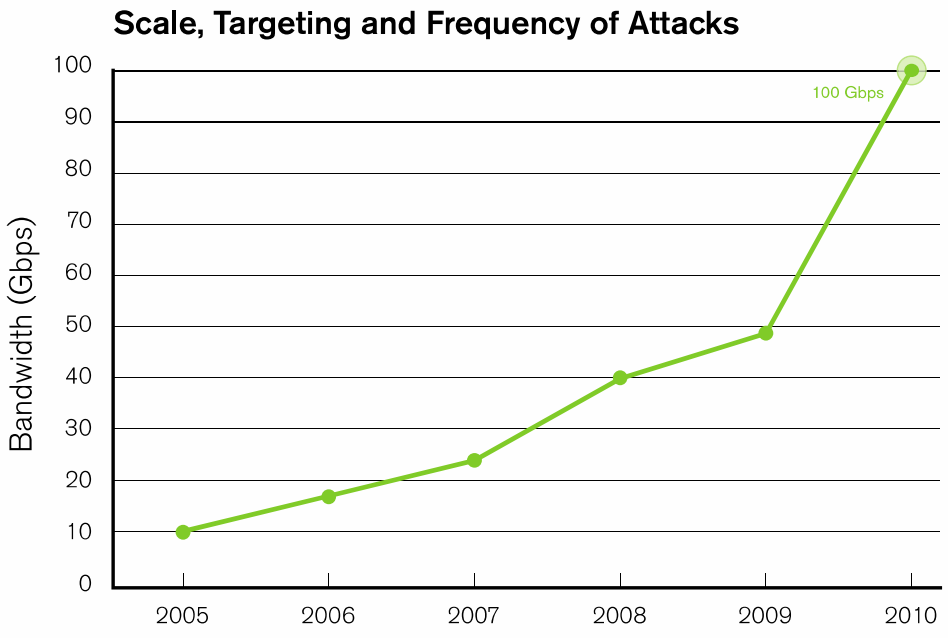
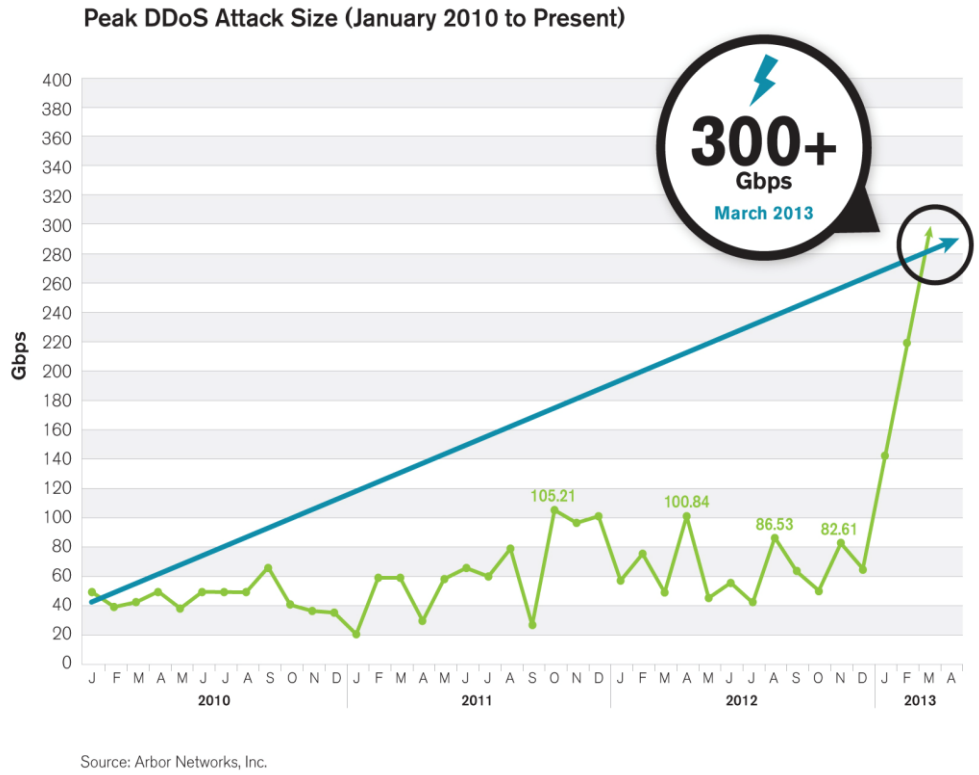


Figure 13
Source: Arbor Networks, Inc.

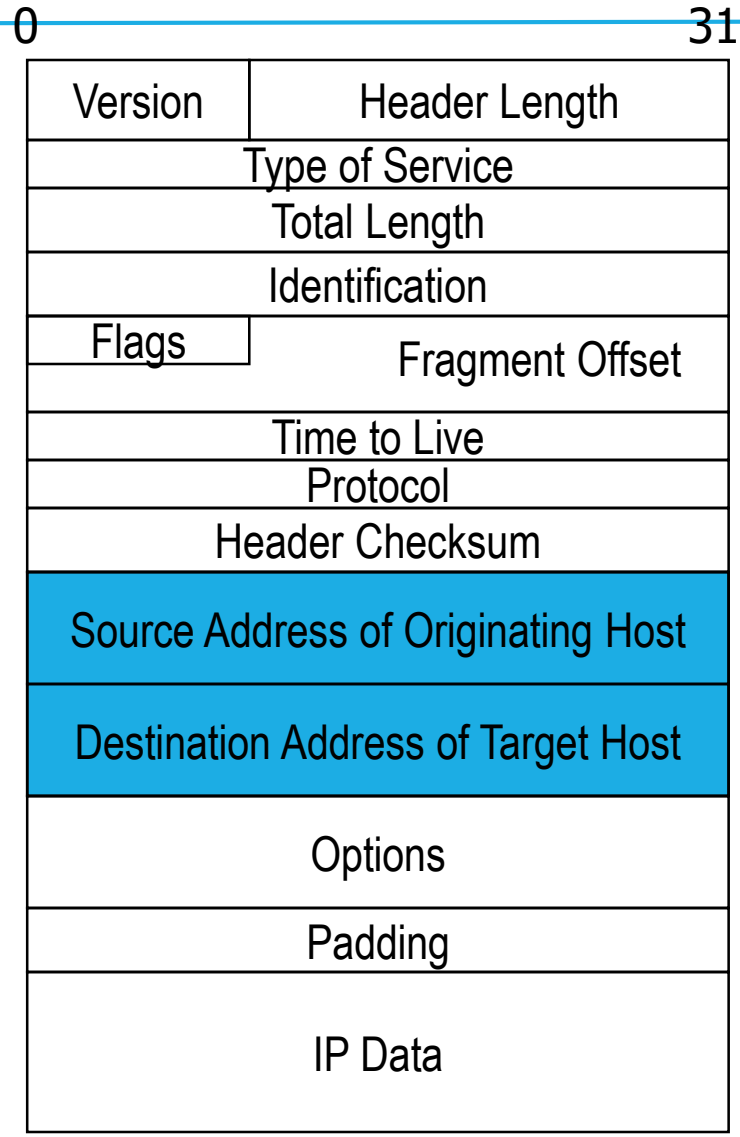


Source: Arbor Networks, Inc.

Feb. 2014: 400 Gbps via NTP amplification (4500 NTP servers)

Review: IP Header format

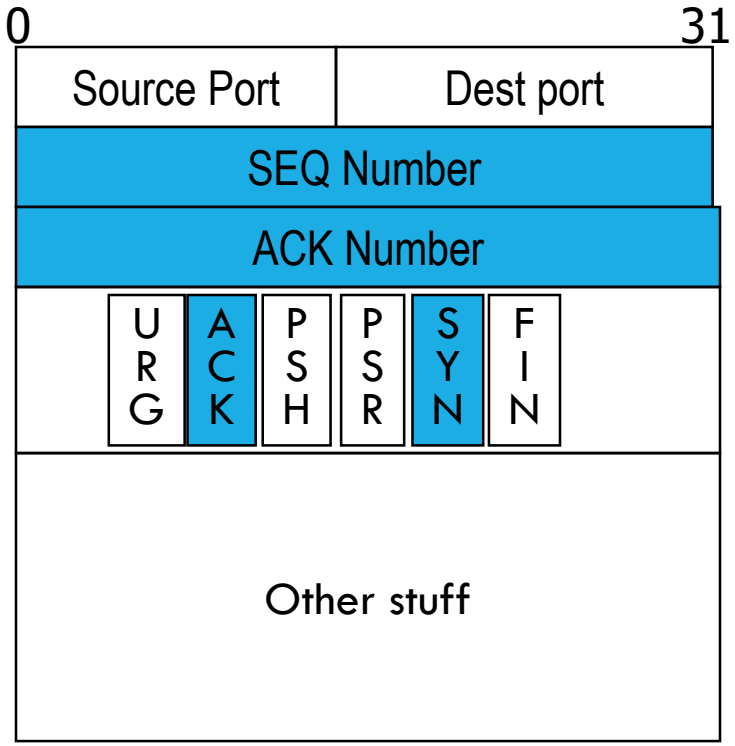
- Connectionless
  - Unreliable
  - Best effort



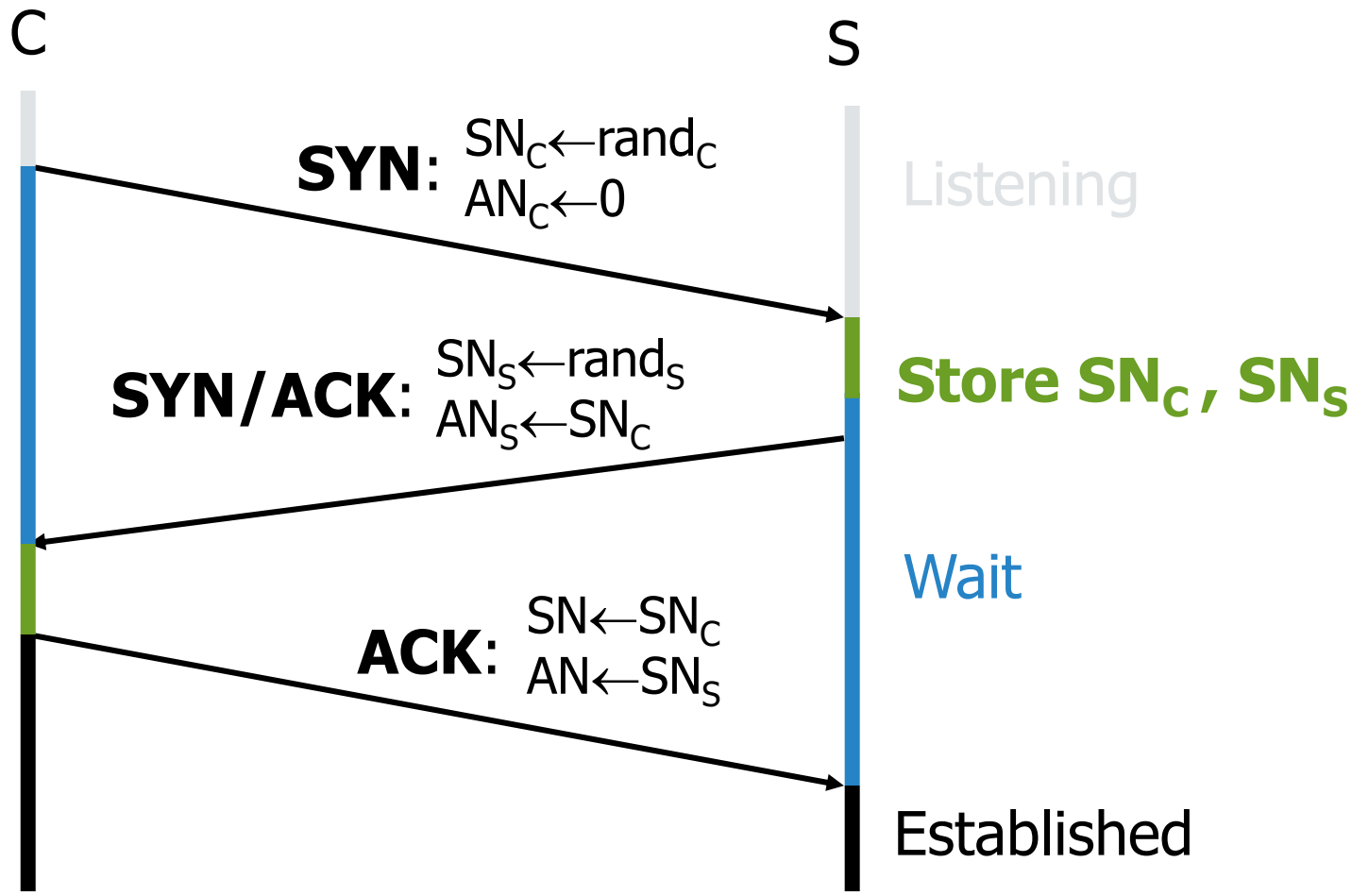
Review: TCP Header format

TCP:

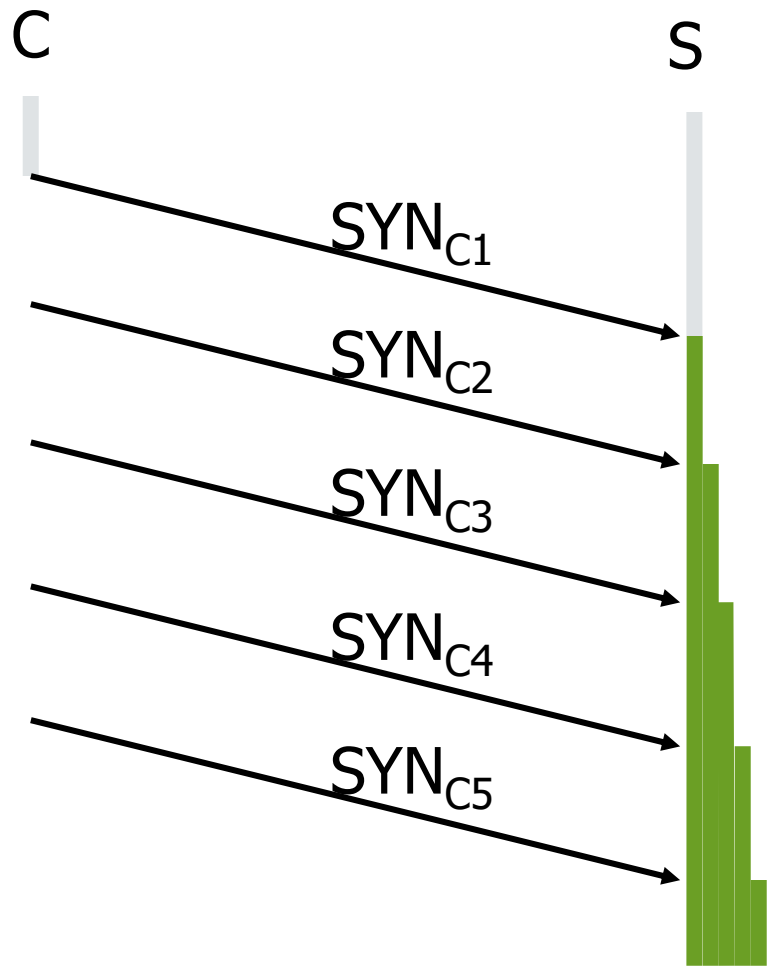
- Session based
- Congestion control
- In order delivery



Review: TCP Handshake



TCP SYN Flood I: low rate (DoS bug)



Single machine:

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

SYN Floods (phrack 48, no 13, 1996)

OS	Backlog queue size
Linux 1.2.x	10
FreeBSD 2.1.5	128
WinNT 4.0	6

Backlog timeout: 3 minutes

⇒ Attacker need only send 128 SYN packets every 3 minutes.

⇒ Low rate SYN flood

A classic SYN flood example

- MS Blaster worm (2003)
  - Infected machines at noon on Aug 16th:
    - SYN flood on port 80 to **windowsupdate.com**
    - 50 SYN packets every second.
      - each packet is 40 bytes.
    - Spoofed source IP: a.b.X.Y where X,Y random.
- MS solution:
  - new name: windowsupdate.microsoft.com
  - Win update file delivered by Akamai

Low rate SYN flood defenses

- Non-solution:
  - Increase backlog queue size or decrease timeout

- Correct solution (when under attack) :
  - **Syncookies**: remove state from server
  - Small performance overhead

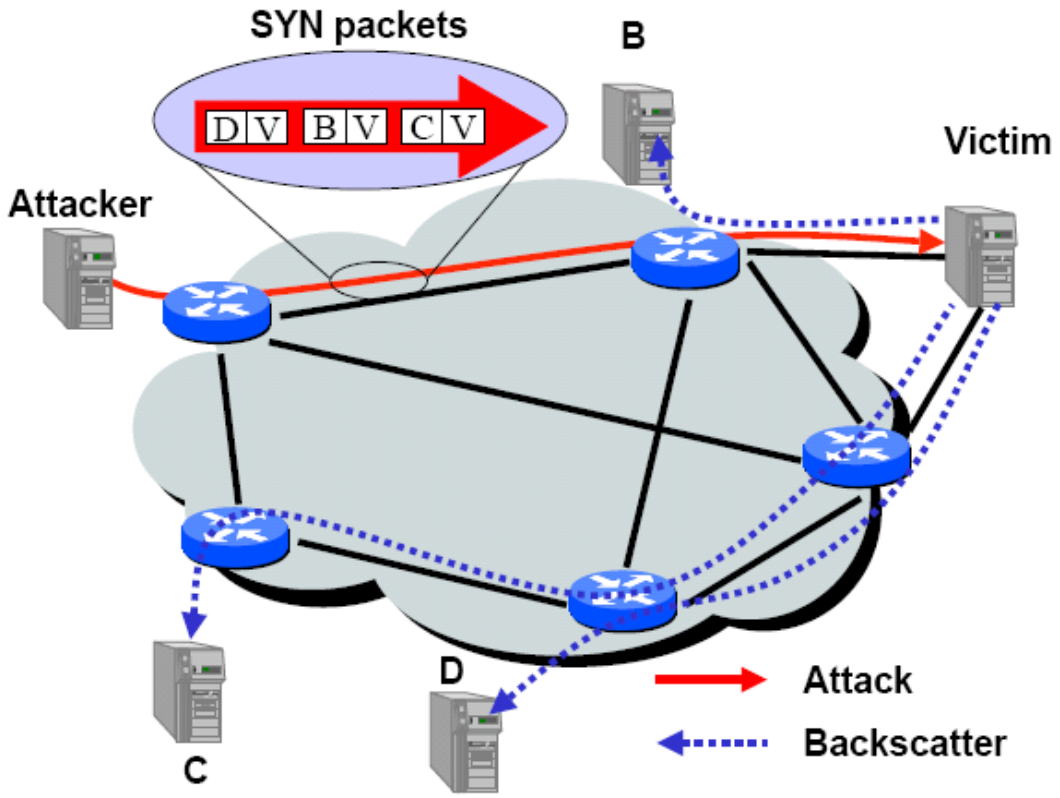
Syncookies [Bernstein, Schenk]

- Idea: use secret key and data in packet to gen. server SN
- Server responds to Client with SYN-ACK cookie:
  - T = 5-bit counter incremented every 64 secs.
  - $L = MAC_{key}$ (SAddr, SPort, DAddr, DPort, SN_C, T) [24 bits]
    - key: picked at random during boot
  - SN_S = (T . mss . L) (|L| = 24 bits)
  - **Server does not save state** (other TCP options are lost)
- Honest client responds with ACK ($AN=SN_S$, $SN=SN_C+1$)
  - Server allocates space for socket only if valid SN_S

SYN floods: backscatter

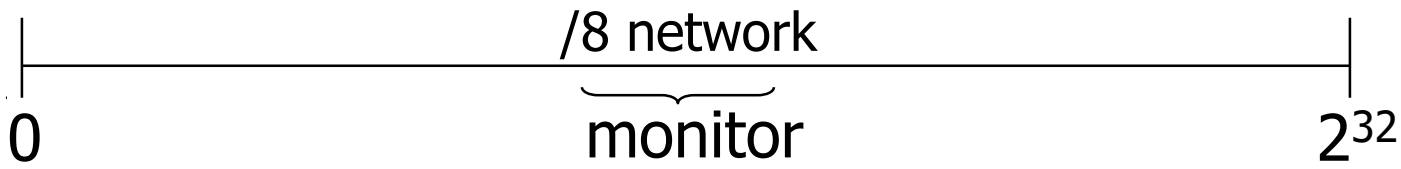
[MVS'01]

- SYN with forged source IP ⇒ SYN/ACK to random host



Backscatter measurement [MVS'01]

- Listen to unused IP addresss space (darknet)



- Lonely SYN/ACK packet likely to be result of SYN attack

- 2001: **400** SYN attacks/week
- 2013: **773** SYN attacks/24 hours (arbor networks ATLAS)
  - Larger experiments: (monitor many ISP darknets)
    - Arbor networks

Estonia attack (ATLAS '07)

- Attack types detected:
  - 115 ICMP floods, 4 TCP SYN floods
- Bandwidth:
  - 12 attacks: **70-95 Mbps for over 10 hours**
- All attack traffic was coming from outside Estonia
  - Estonia's solution:
    - Estonian ISPs blocked all foreign traffic until attacks stopped

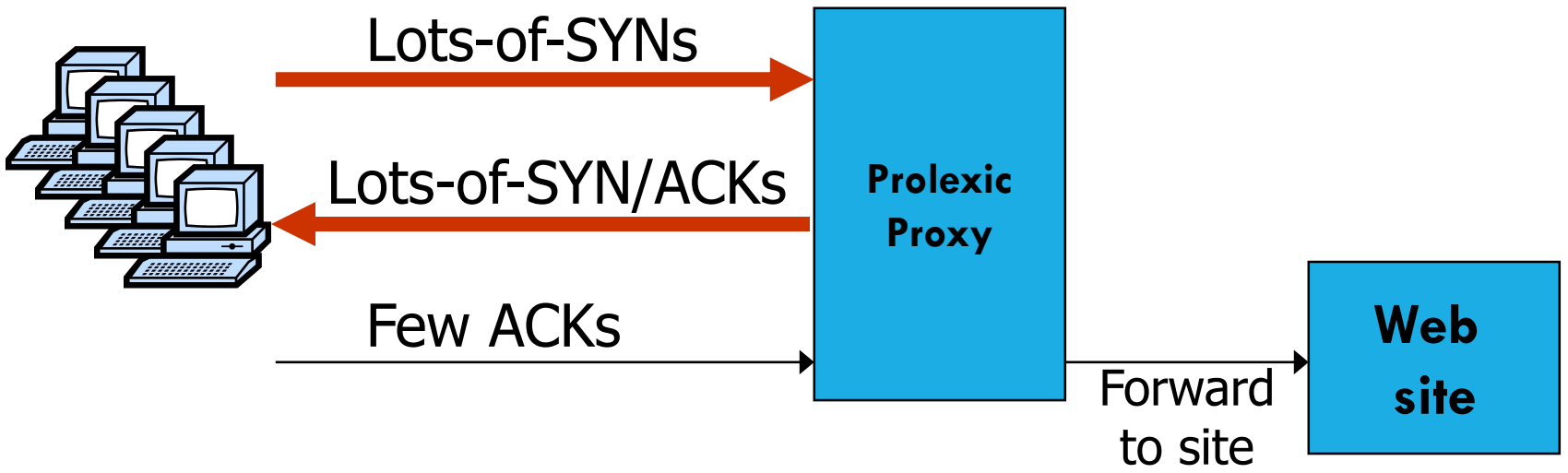
    => DoS attack had little impact inside Estonia

SYN Floods II: Massive flood
(e.g BetCris.com '03)

- Command bot army to flood specific target: (DDoS)
  - **20,000** bots can generate **2Gb/sec** of SYNs (2003)
  - At web site:
    - Saturates network uplink or network router
    - Random source IP ⇒
      attack SYNs look the same as real SYNs
  - What to do ???

Prolexic / CloudFlare

- Idea: only forward established TCP connections to site



Other junk packets

Attack Packet	**Victim Response**	**Rate: attk/day [ATLAS 2013]**
TCP SYN to open port	TCP SYN/ACK	773
TCP SYN to closed port	TCP RST	
TCP ACK or TCP DATA	TCP RST	
TCP RST	No response	
TCP NULL	TCP RST	
ICMP ECHO Request	ICMP ECHO Response	**50**
UDP to closed port	ICMP Port unreachable	387

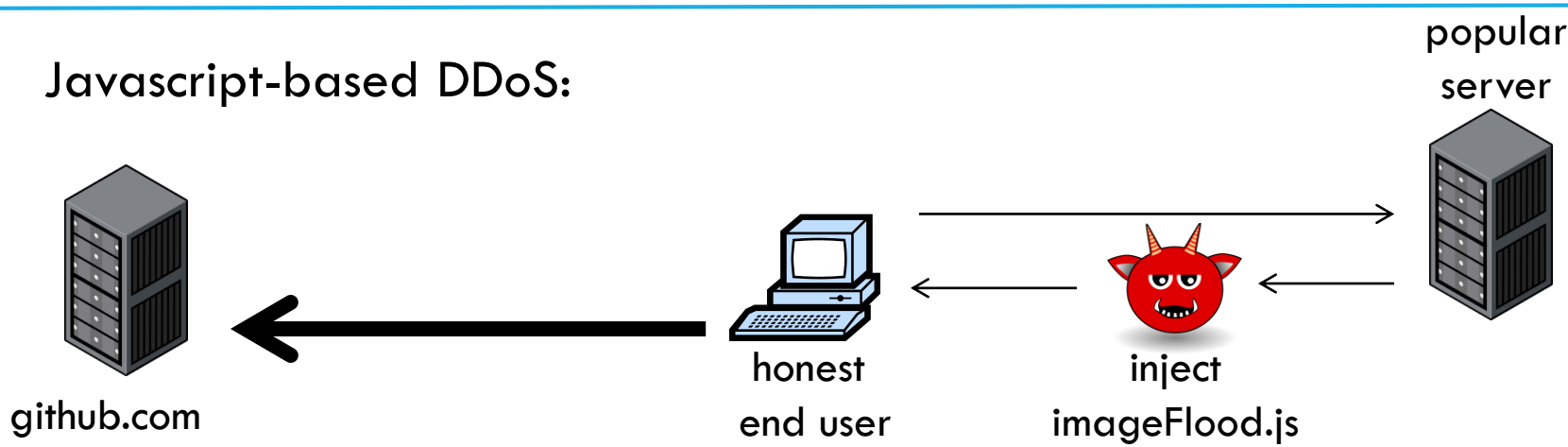
Proxy must keep floods of these away from web site

Stronger attacks: TCP con flood

- Command bot army to:
  - Complete TCP connection to web site
  - Send short HTTP HEAD request
  - Repeat
- Will bypass SYN flood protection proxy
- … but:
  - Attacker can no longer use random source IPs.
    - Reveals location of bot zombies
  - Proxy can now block or rate-limit bots.

A real-world example: GitHub (3/2015)

Javascript-based DDoS:



imageFlood.js

```
function imgflood() {
  var TARGET = 'victim-website.com/index.php?'
  var rand = Math.floor(Math.random() * 1000)
  var pic = new Image()
  pic.src = 'http://'+TARGET+rand+'=val'
}
setInterval(imgflood, 10)
```

Would HTTPS
prevent this DDoS?

DoS via route hijacking

- YouTube is 208.65.152.0**/22** (includes 2^{10} IP addr)

  youtube.com is 208.65.153.238, …

- Feb. 2008:
  - Pakistan telecom advertised a BGP path for

    208.65.153.0**/24** (includes 2^{8} IP addr)
  - Routing decisions use most specific prefix
  - The entire Internet now thinks

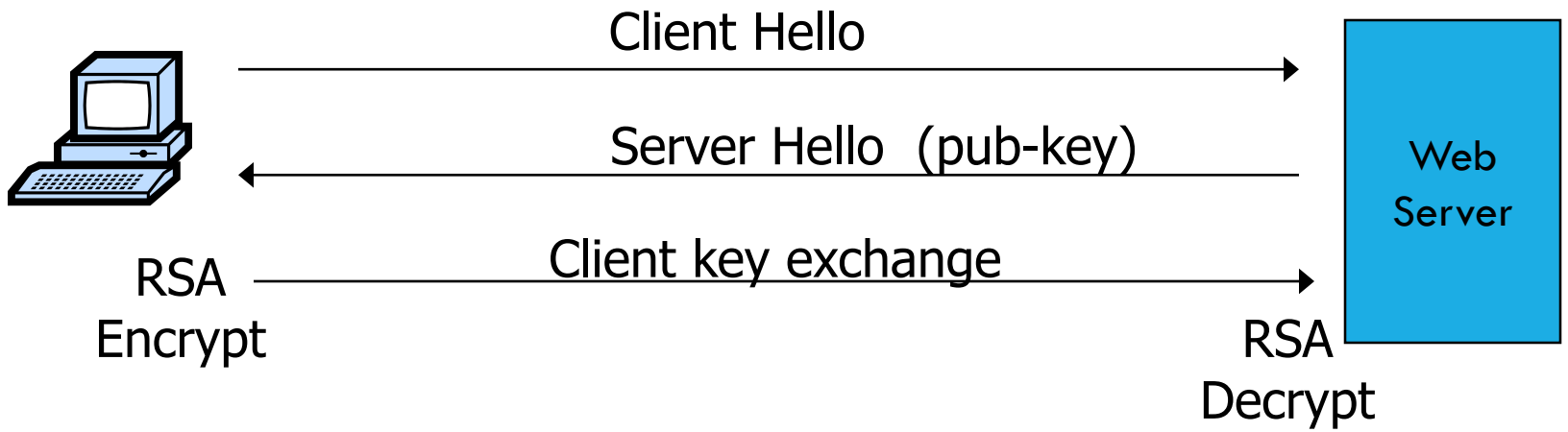
    208.65.153.238 is in Pakistan

- Outage resolved within two hours

  … but demonstrates huge DoS vuln. with no solution!

DoS at higher layers

- SSL/TLS handshake [SD'03]



  - RSA-encrypt speed ≈ 10× RSA-decrypt speed

  ⇒ Single machine can bring down ten web servers

- Similar problem with application DoS:
  - Send HTTP request for some large PDF file

  ⇒ Easy work for client, hard work for server.

Part 4-2

Dos Mitigation

Mostly based on Dan Boneh Slides

DoS Mitigation

- 1- Client Puzzles
- 2- CAPTCHAs
- 3- Source Identification:
  - Goal: identify packet source

1. Client puzzles

- Idea: slow down attacker

- Moderately hard problem:
  - Given challenge C find X such that

$$\mathbf{LSB}_n\left(\mathbf{SHA\text{-}1}(\mathbf{C} \,\|\, \mathbf{X})\right) = \mathbf{0}^n$$

  - Assumption: takes expected 2^n time to solve
  - For n=16 takes about .3sec on 1GhZ machine
  - Main point: checking puzzle solution is easy.

- During DoS attack:
  - Everyone must submit puzzle solution with requests
  - When no attack: do not require puzzle solution

Examples

- TCP connection floods (RSA '99)
  - Example challenge: C = TCP server-seq-num
  - First data packet must contain puzzle solution
    - Otherwise TCP connection is closed
- SSL handshake DoS: (SD'03)
  - Challenge C based on TLS session ID
  - Server: check puzzle solution before RSA decrypt.
- Same for application layer DoS and payment DoS.

Benefits and limitations

- Hardness of challenge: n
  - Decided based on DoS attack volume.

- Limitations:
  - Requires changes to both clients and servers
  - Hurts low power legitimate clients during attack:
    - Clients on cell phones and tablets cannot connect

Memory-bound functions

- CPU power ratio:
  - high end server / low end cell phone = 8000

  $\Rightarrow$ Impossible to scale to hard puzzles

- Interesting observation:
  - Main memory access time ratio:
    - high end server / low end cell phone = 2
- Better puzzles:
  - Solution requires many main memory accesses
    - Dwork-Goldberg-Naor, Crypto '03
    - Abadi-Burrows-Manasse-Wobber, ACM ToIT '05

2. CAPTCHAs

- Idea: verify that connection is from a human

- Applies to application layer DDoS [Killbots '05]
  - During attack: generate CAPTCHAs and process request only if valid solution
  - Present one CAPTCHA per source IP address.

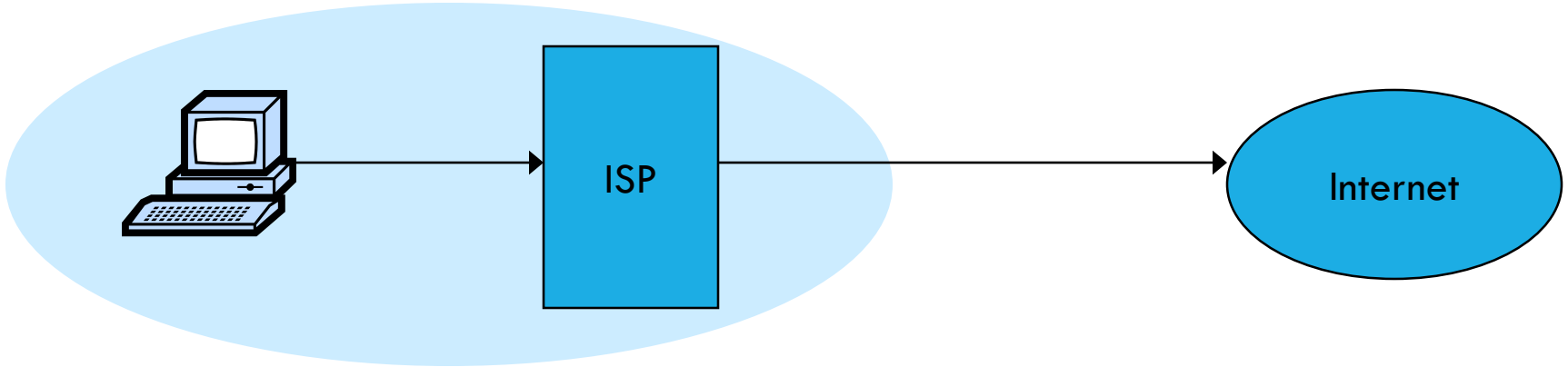
3. Source Identification

- Goal: Identify packet source

- Ultimate Goal: Block attack at the source

3.1. Ingress filtering (RFC 2827, 3704)

- Big problem: DDoS with spoofed source IPs



- Ingress filtering policy: ISP only forwards packets with legitimate source IP (see also SAVE protocol)

Implementation problems

ALL ISPs must do this. Requires global trust.

- If 10% of ISPs do not implement $\Rightarrow$ no defense
- No incentive for deployment

2014:

- 25% of Auto. Systems are fully spoofable (spoofer.cmand.org)
- 13% of announced IP address space is spoofable

Recall: 309 Gbps attack used only 3 networks (3/2013)

3.2. Traceback [Savage et al. ’00]

- Goal:
  - Given set of attack packets
  - Determine path to source
- How: change routers to record info in packets
- Assumptions:
  - Most routers remain uncompromised
  - Attacker sends many packets
  - Route from attacker to victim remains relatively stable

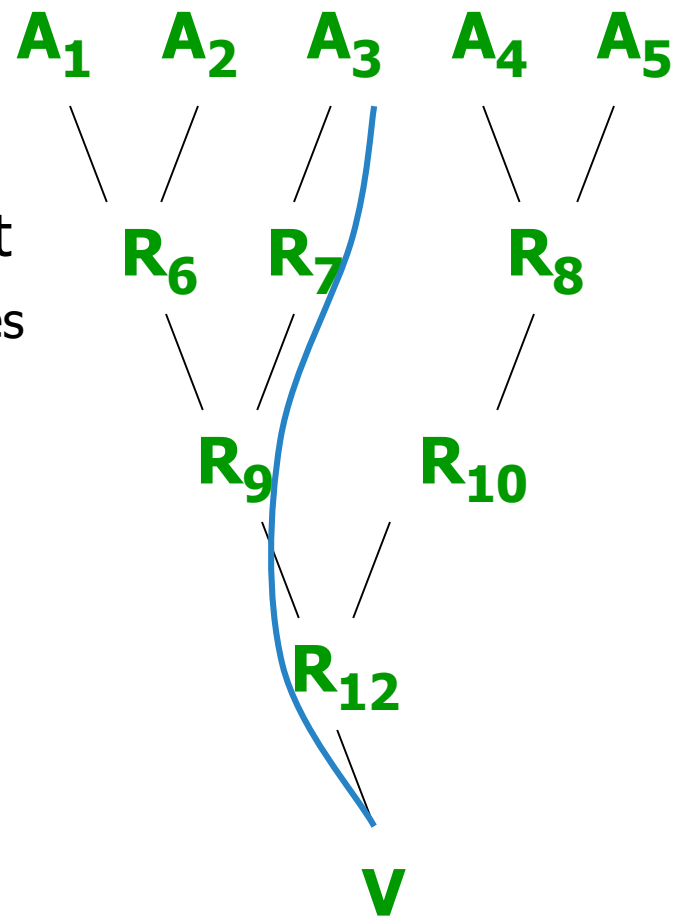
Simple method

- Write path into network packet
  - Each router adds its own IP address to packet
  - Victim reads path from packet

- Problem:
  - Requires space in packet
    - Path can be long
    - No extra fields in current IP format
      - Changes to packet format too much to expect

Better idea

- DDoS involves many packets on same path
- Store one link in each packet
  - Each router probabilistically stores own address
  - Fixed space regardless of path length



Edge Sampling

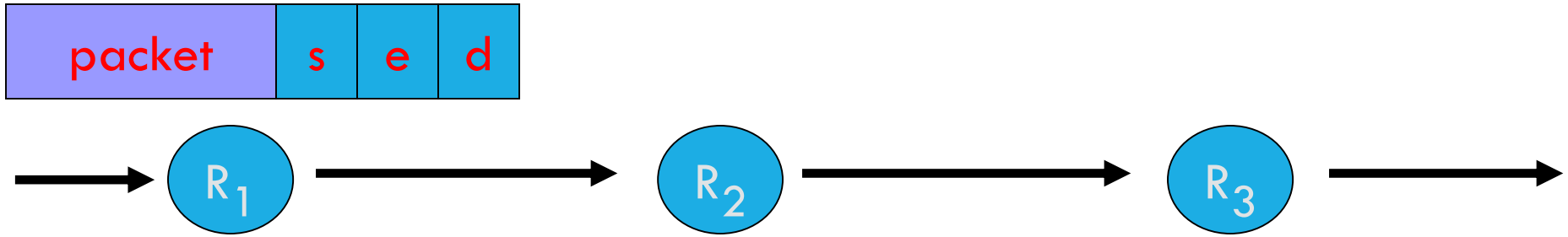
- Data fields written to packet:
  - Edge: *start* and *end* IP addresses
  - Distance: number of hops since edge stored

- Marking procedure for router R

```
if coin turns up heads (with probability p) then
        write R into start address
        write 0 into distance field
else
        if distance == 0 write R into end field
        increment distance field
```

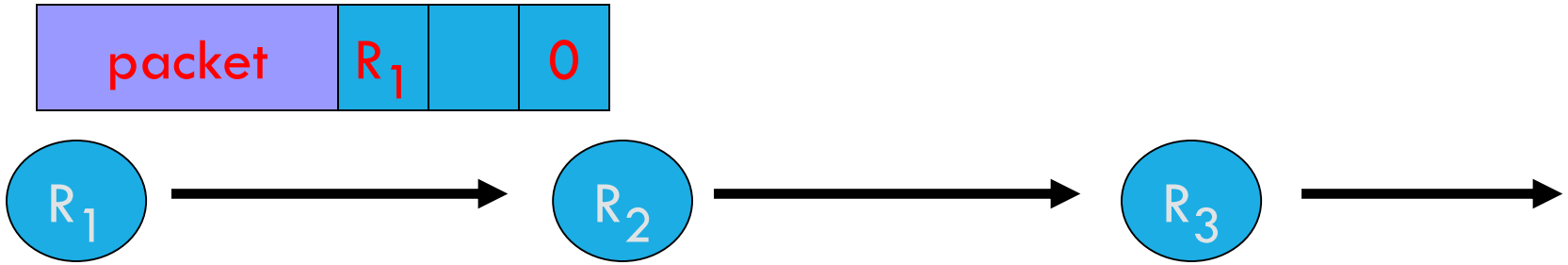
Edge Sampling: picture

- Packet received
  - R_1 receives packet from source or another router
  - Packet contains space for start, end, distance



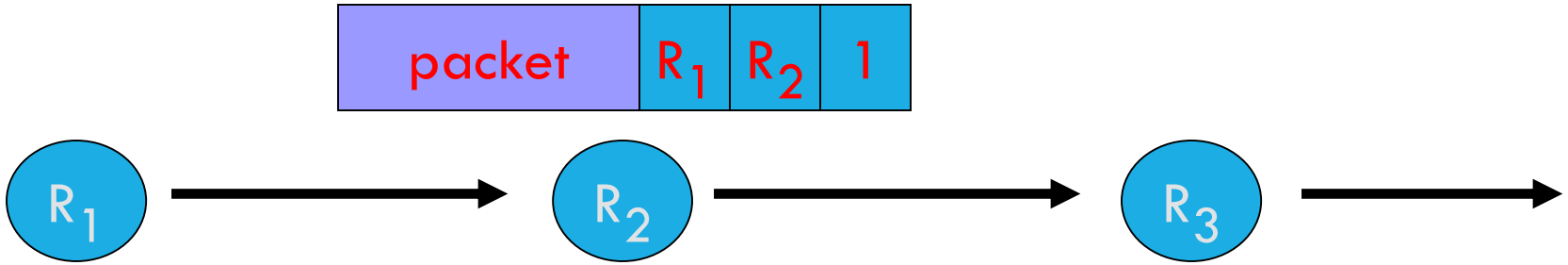
Edge Sampling: picture

- Begin writing edge
  - R_1 chooses to write start of edge
  - Sets distance to 0



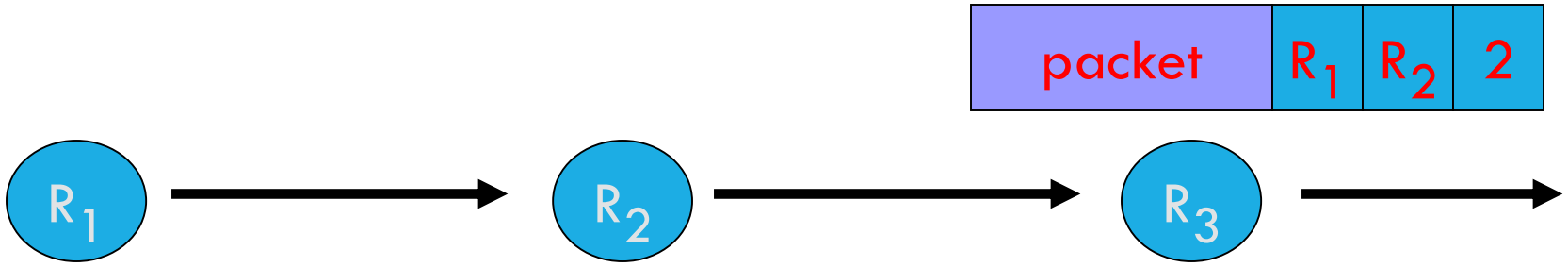
Edge Sampling

- Finish writing edge
  - R_2 chooses not to overwrite edge
  - Distance is 0
    - Write end of edge, increment distance to 1



Edge Sampling

- Increment distance
  - R_3 chooses not to overwrite edge
  - Distance >0
    - Increment distance to 2



Path reconstruction

- Extract information from attack packets
- Build graph rooted at victim
  - Each (start,end,distance) tuple provides an edge
- # packets needed to reconstruct path

  $$E(X) < \frac{\ln(d)}{p(1-p)^{d-1}}$$

  where p is marking probability, d is length of path

More traceback proposals

- Advanced and Authenticated Marking Schemes for IP Traceback
  - Song, Perrig. IEEE Infocomm ’01
  - Reduces noisy data and time to reconstruct paths
- An algebraic approach to IP traceback
  - Stubblefield, Dean, Franklin. NDSS ’02
- Hash-Based IP Traceback
  - Snoeren, Partridge, Sanchez, Jones, Tchakountio, Kent, Strayer. SIGCOMM ‘01

Problem: Reflector attacks [Paxson '01]

- **Reflector**:
  - A network component that responds to packets
  - Response sent to victim (spoofed source IP)

- Examples:
  - DNS Resolvers: UDP 53 with victim.com source
    - At victim: DNS response
  - Web servers: TCP SYN 80 with victim.com source
    - At victim: TCP SYN ACK packet
  - Gnutella servers

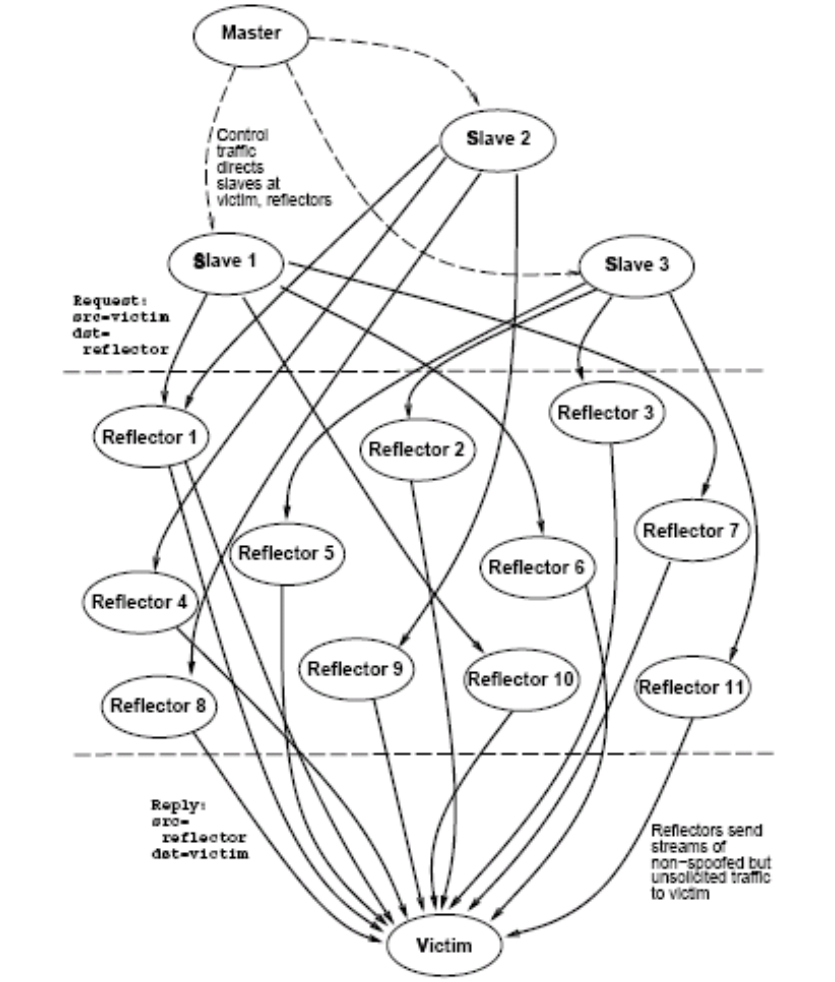
DoS Attack

Single Master

Many bots to generate flood

Zillions of reflectors to hide bots

- Kills traceback and pushback methods



Part 4-3

Dos Defense mechanisms:

- Capability based defense
- Pushback Traffic + Overlay Filtering

Mostly based on Dan Boneh Slides

Capability based defense

- Anderson, Roscoe, Wetherall.
  - Preventing internet denial-of-service with capabilities. SIGCOMM '04.
- Yaar, Perrig, and Song.
  - Siff: A stateless internet flow filter to mitigate DDoS flooding attacks. IEEE S&P '04.
- Yang, Wetherall, Anderson.
  - A DoS-limiting network architecture. SIGCOMM '05

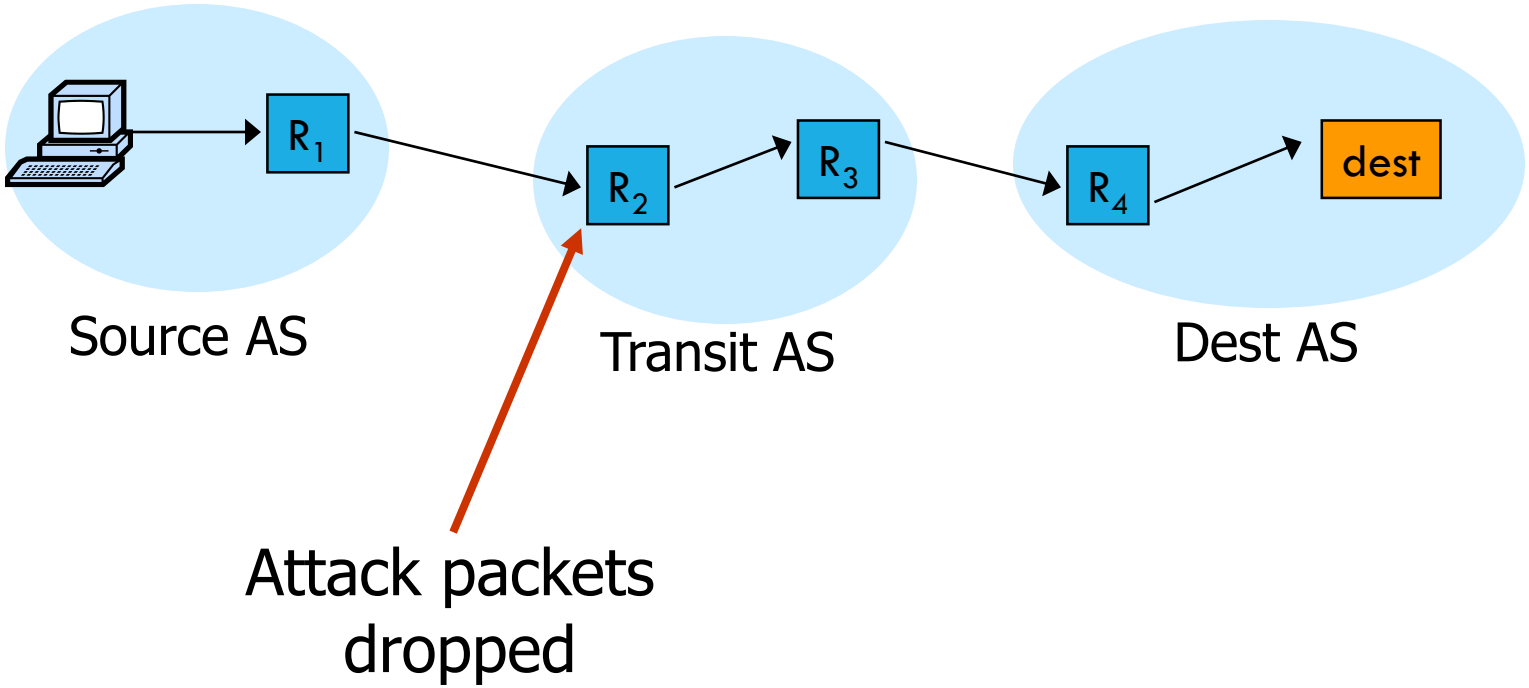
Capability based defense

- Basic idea:
  - Receivers can specify what packets they want

- How:
  - Sender requests capability in SYN packet
    - Path identifier used to limit # reqs from one source
  - Receiver responds with capability
  - Sender includes capability in all future packets
  - Main point: Routers only forward:
    - Request packets, and
    - Packets with valid capability

Capability based defense

- Capabilities can be revoked if source is attacking
  - Blocks attack packets close to source



Pushback filtering

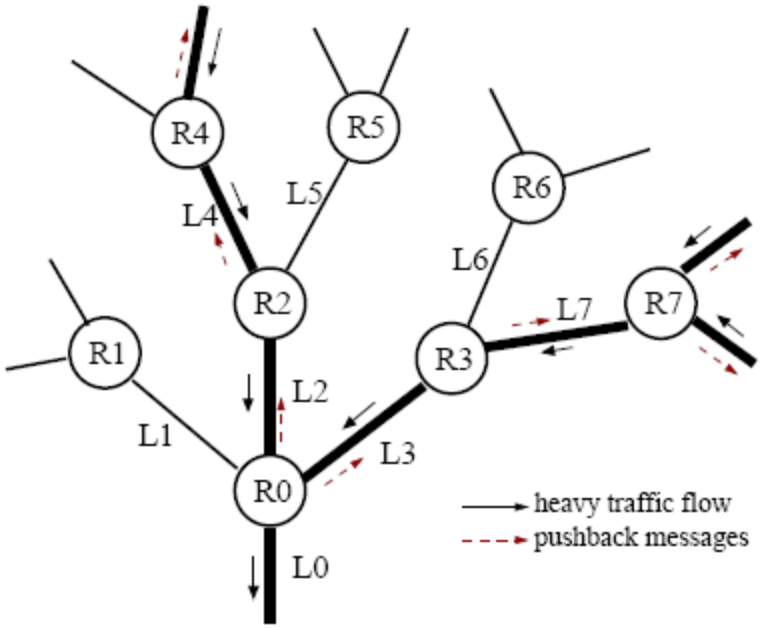
- Mahajan, Bellovin, Floyd, Ioannidis, Paxson, Shenker. Controlling High Bandwidth Aggregates in the Network. *Computer Communications Review* '02.

- Ioannidis, Bellovin.
  Implementing Pushback: Router-Based Defense Against DoS Attacks.
  NDSS '02

- Argyraki, Cheriton.
  Active Internet Traffic Filtering: Real-Time Response to Denial-of-Service Attacks. USENIX '05.

Pushback Traffic Filtering

- Assumption: DoS attack from few sources



- Iteratively block attacking network segments.

Overlay filtering

- Keromytis, Misra, Rubenstein.
  SOS: Secure Overlay Services. SIGCOMM '02.

- D. Andersen. Mayday.
  Distributed Filtering for Internet Services.
  Usenix USITS '03.

- Lakshminarayanan, Adkins, Perrig, Stoica.
  Taming IP Packet Flooding Attacks. HotNets '03.

What we learn…

- Denial of Service attacks are real.
  Must be considered at design time.

- Sad truth:
  - Internet is ill-equipped to handle DDoS attacks
  - Commercial solutions: CloudFlare, Prolexic

- Many good proposals for core redesign.

Questions?

THE END