https://brown-csci1660.github.io

CS1660: Intro to Computer Systems Security
Spring 2026

Lecture 20: Network Security II

Instructor: **Nikos Triandopoulos**

April 16, 2026

BROWN

20.1 Networking & Security

Computer networks and security…

Remote Communication

- Networks enable distant interactions

Data Exchange Infrastructure

- Network devices allow the creation of an efficient digital domain

Cyber Attack Vectors

- Networks are common targets needing solid defenses.

There is a dual nature of networks as both enablers and potential risks

Networks

Types

Virtual Circuit

- Legacy phone network
- Single route through sequence of hardware devices established when two nodes start communication
- Data sent along route
- Route maintained until communication ends

Packet Routing

- Internet
- Data split into packets
- Packets transported independently through network
- Each packet handled on a best-effort basis
- Packets may follow different routes

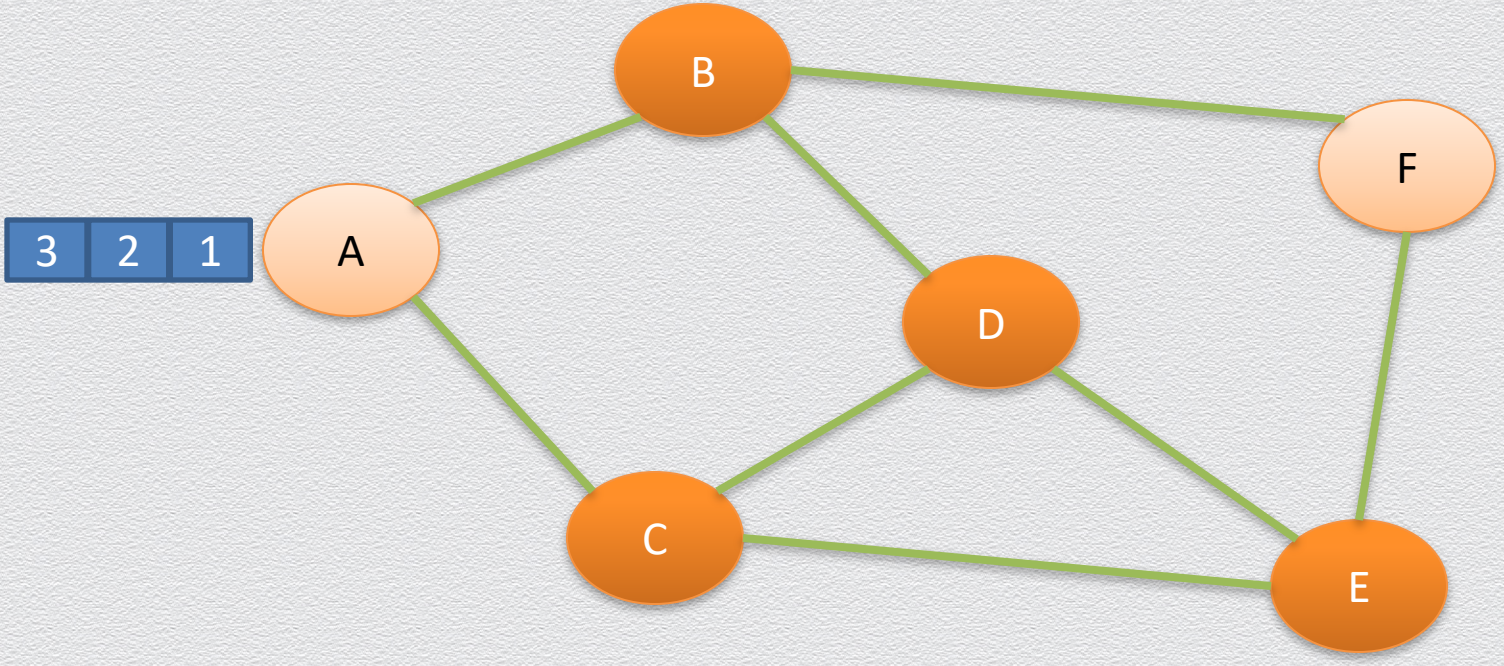
Virtual circuit

- Analog rotary phones

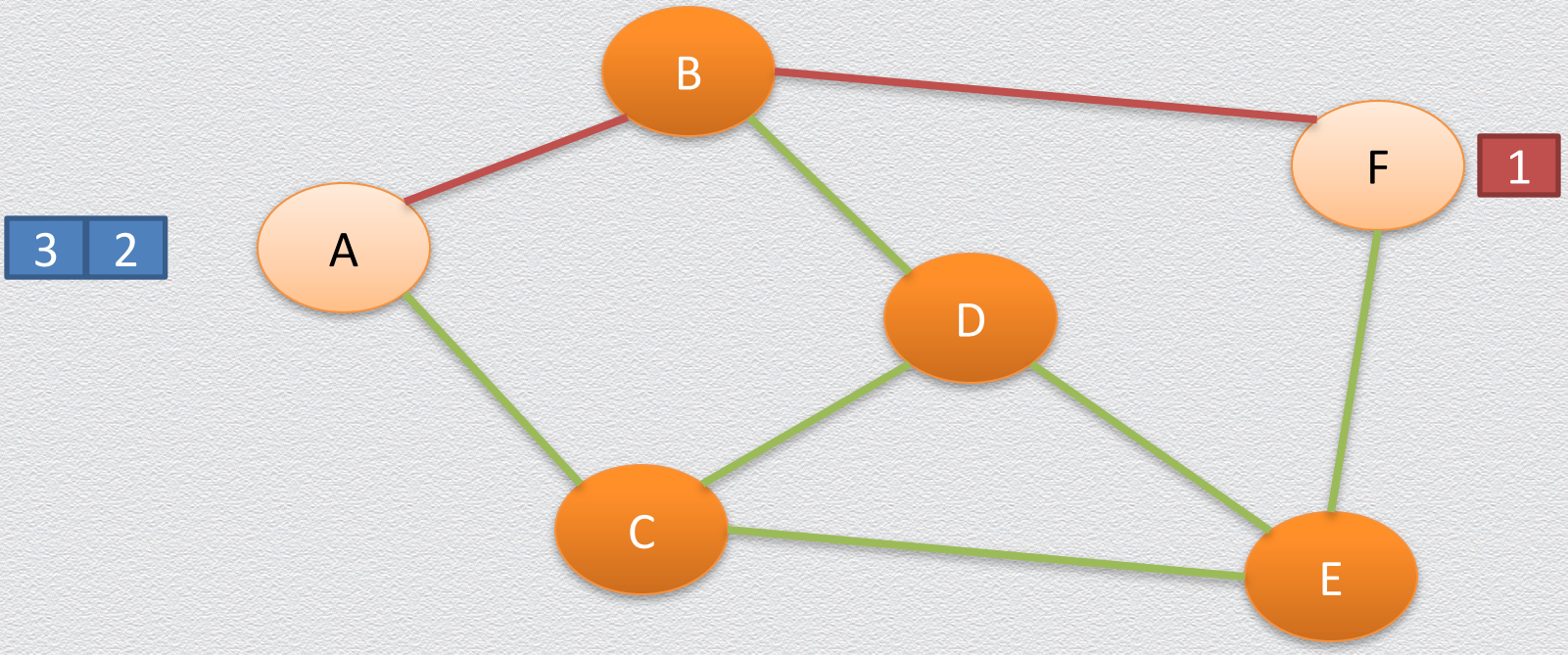
Packet routing (aka switching)

- Data split into packets
- Each packet is
  - Transported independently through network
  - Handled on a best-effort basis by each device
- Packets may
  - Follow different routes between the same endpoints
  - Be dropped by an intermediate device and never delivered

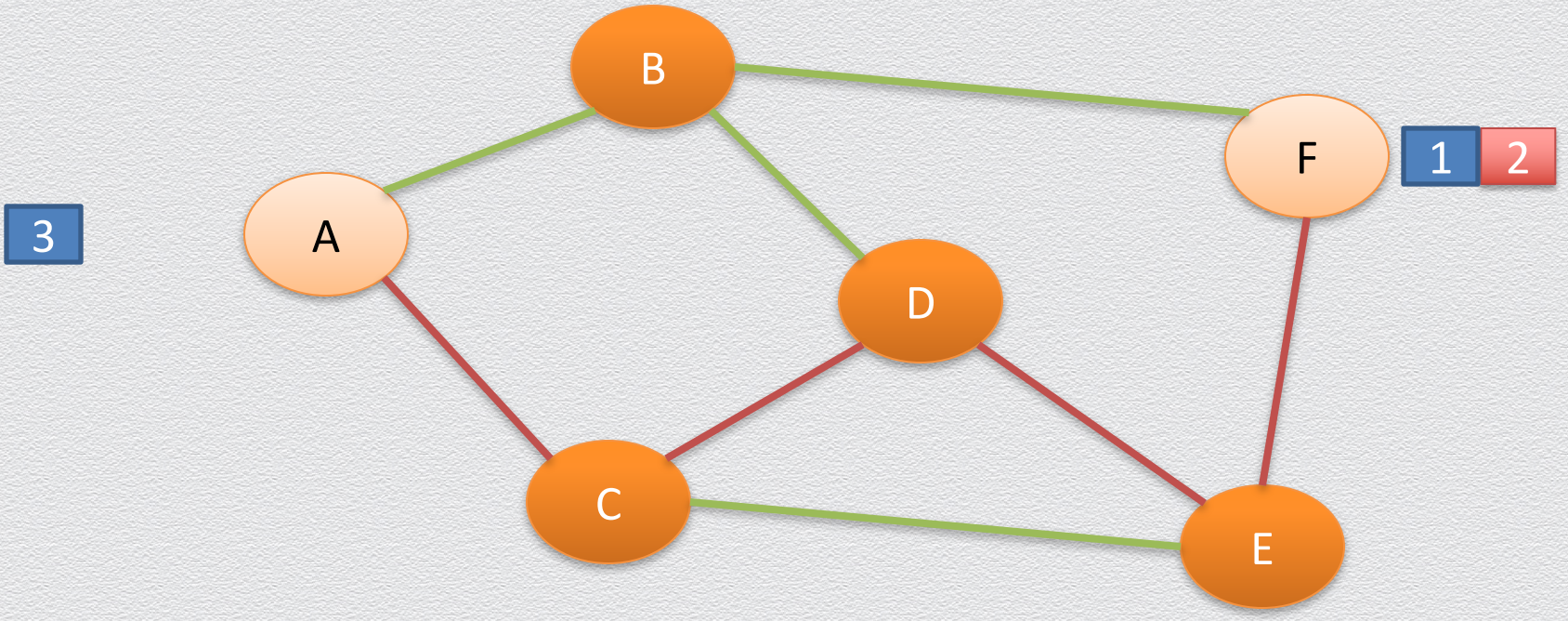
Packet routing: Example



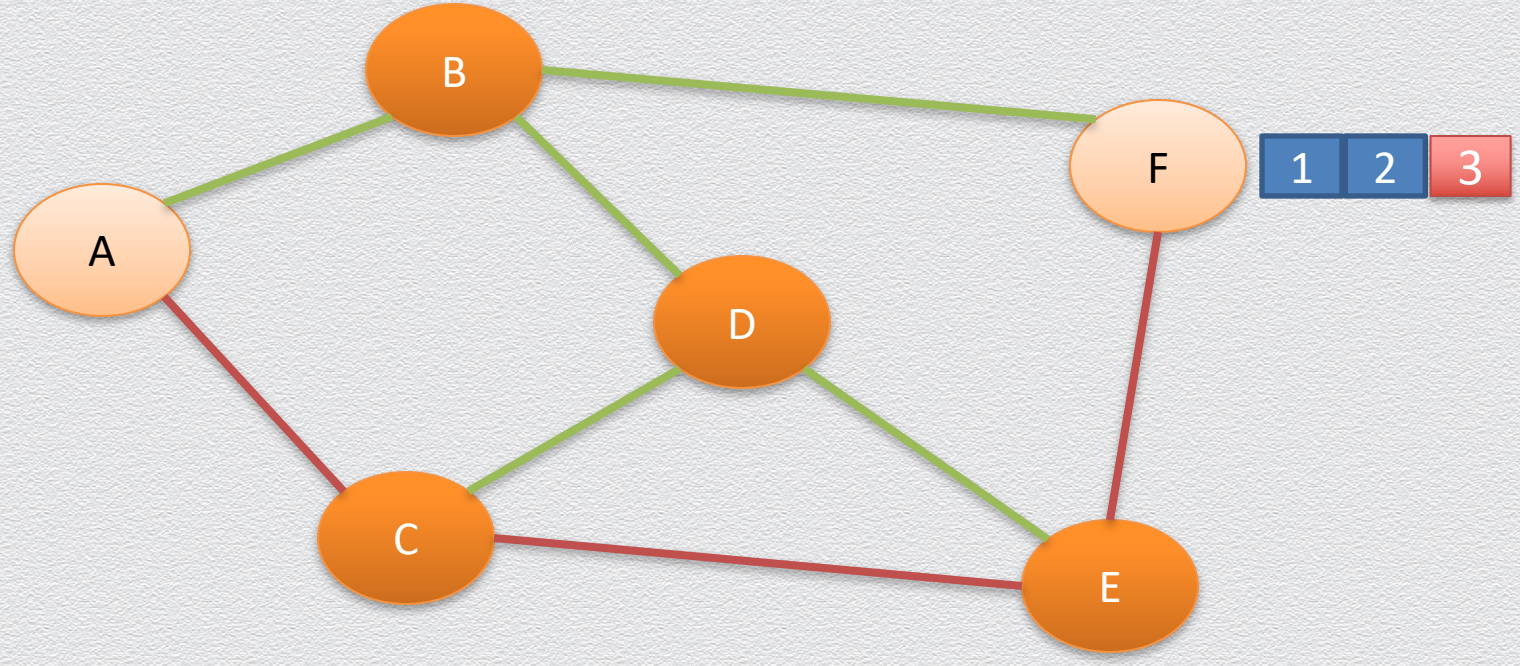
Packet routing: Example



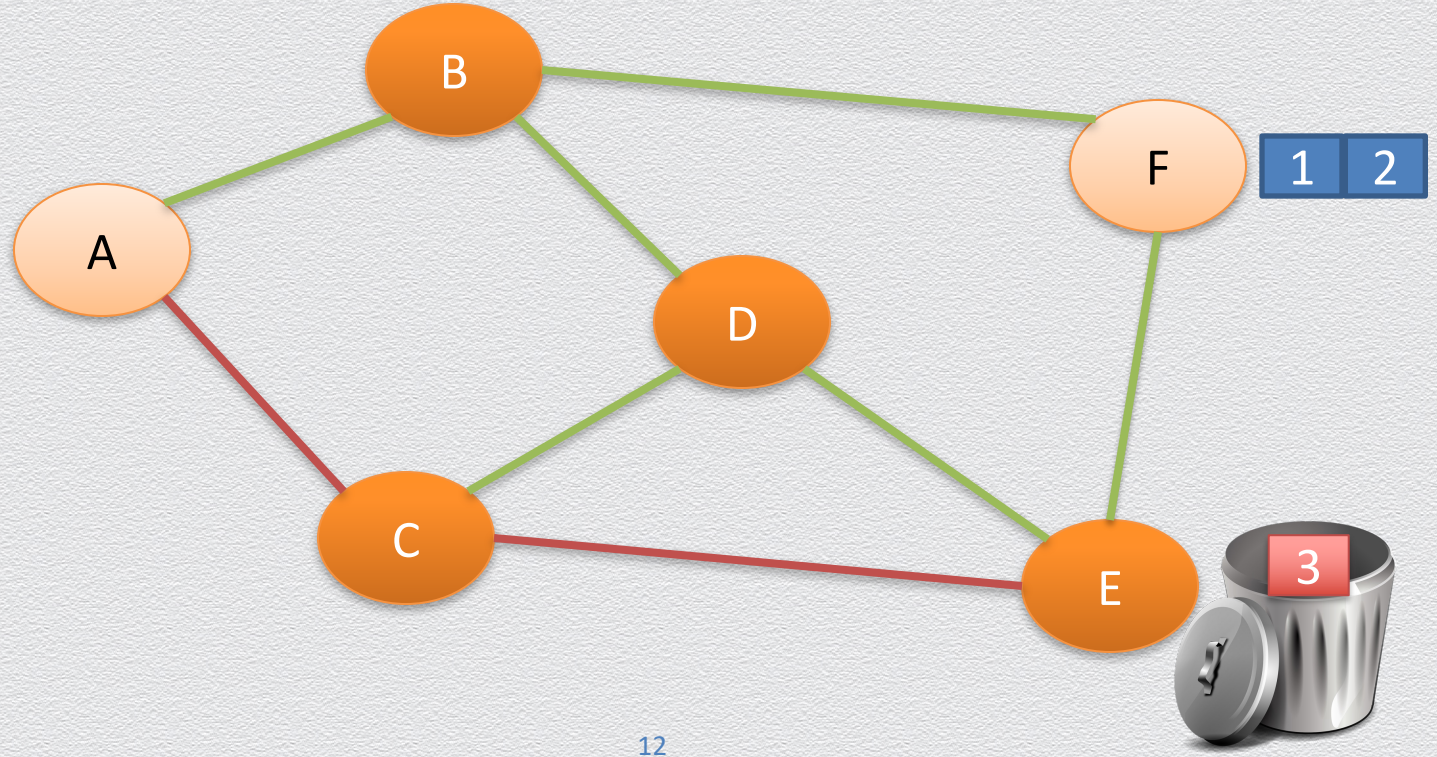
Packet routing: Example



Packet routing: Example



Packet routing: Problem



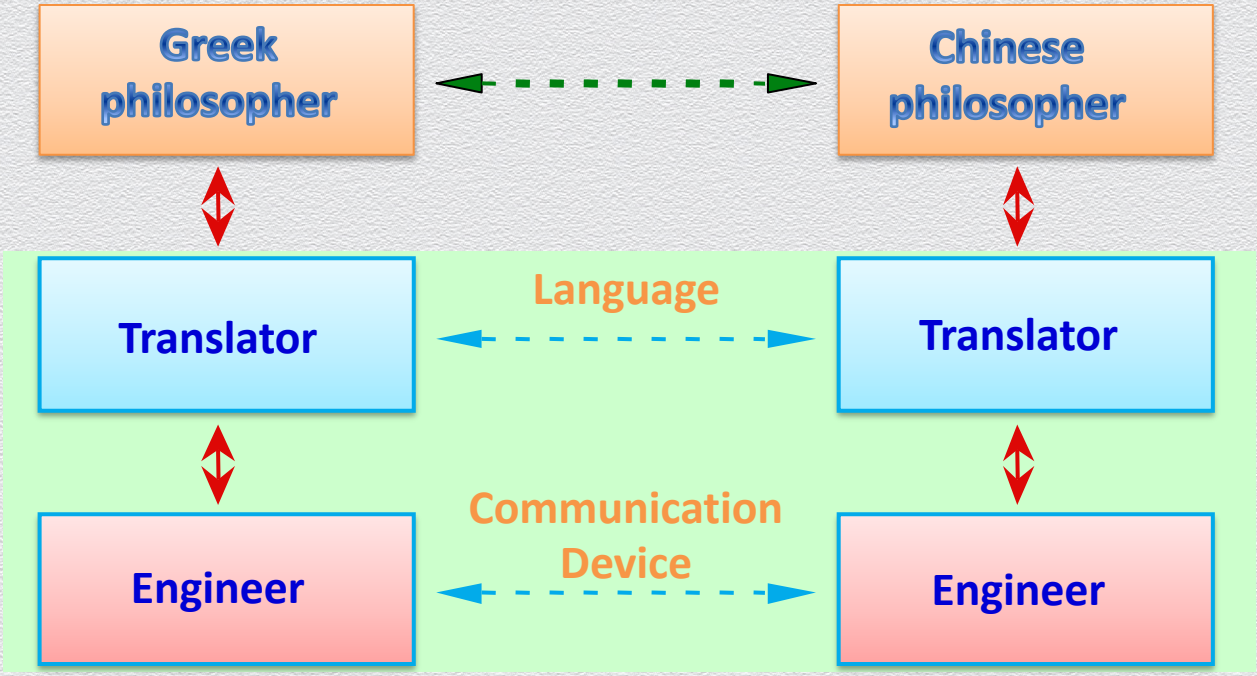
Communication

Characterized by the following fundamental principles

- Packet routing/switching
- Stack of layers (virtual layers)
- Encapsulation

20.2 Protocol layers and encapsulation

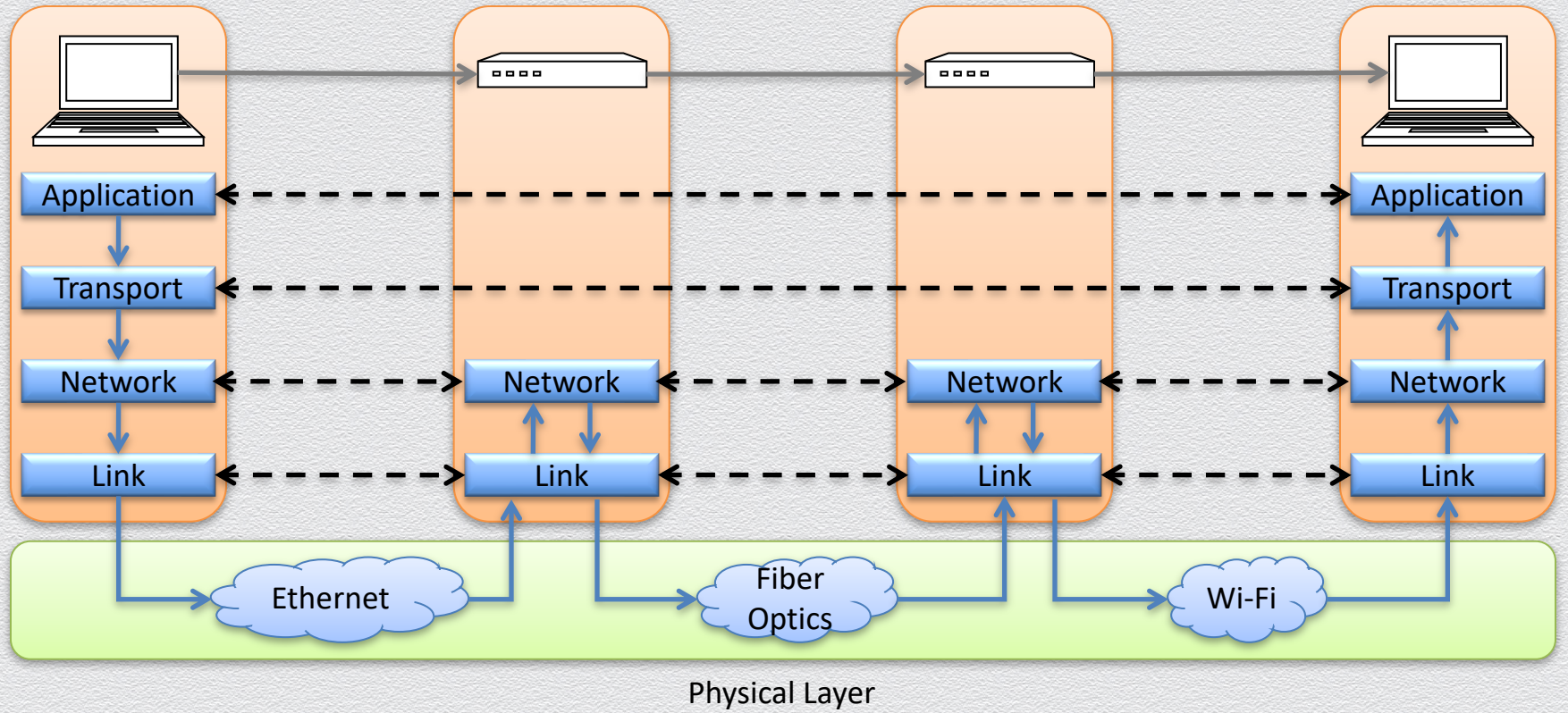
Two philosophers example



Stack of layers

- Network communication models use a **stack of layers**
  - Higher layers use services of lower layers
  - Physical channel at the bottommost layer
- A network device implements several layers
- A communication channel between two devices is established for each layer
  - Actual channel at the bottom layer
  - Virtual channel at higher layers

Internet Layers: How your computer talks to a website



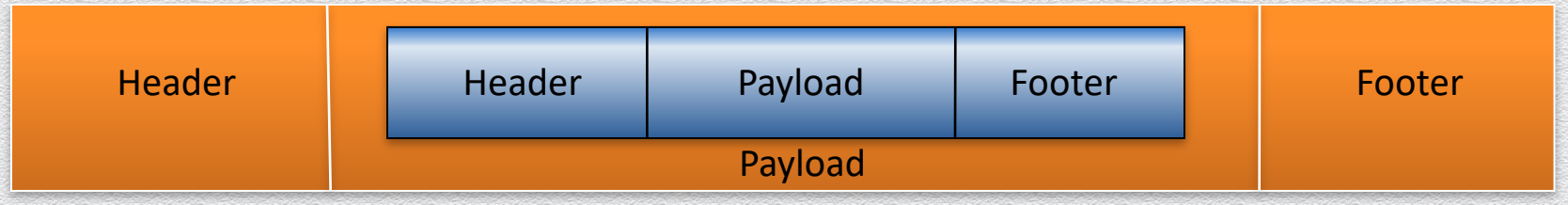
Encapsulation

Packets

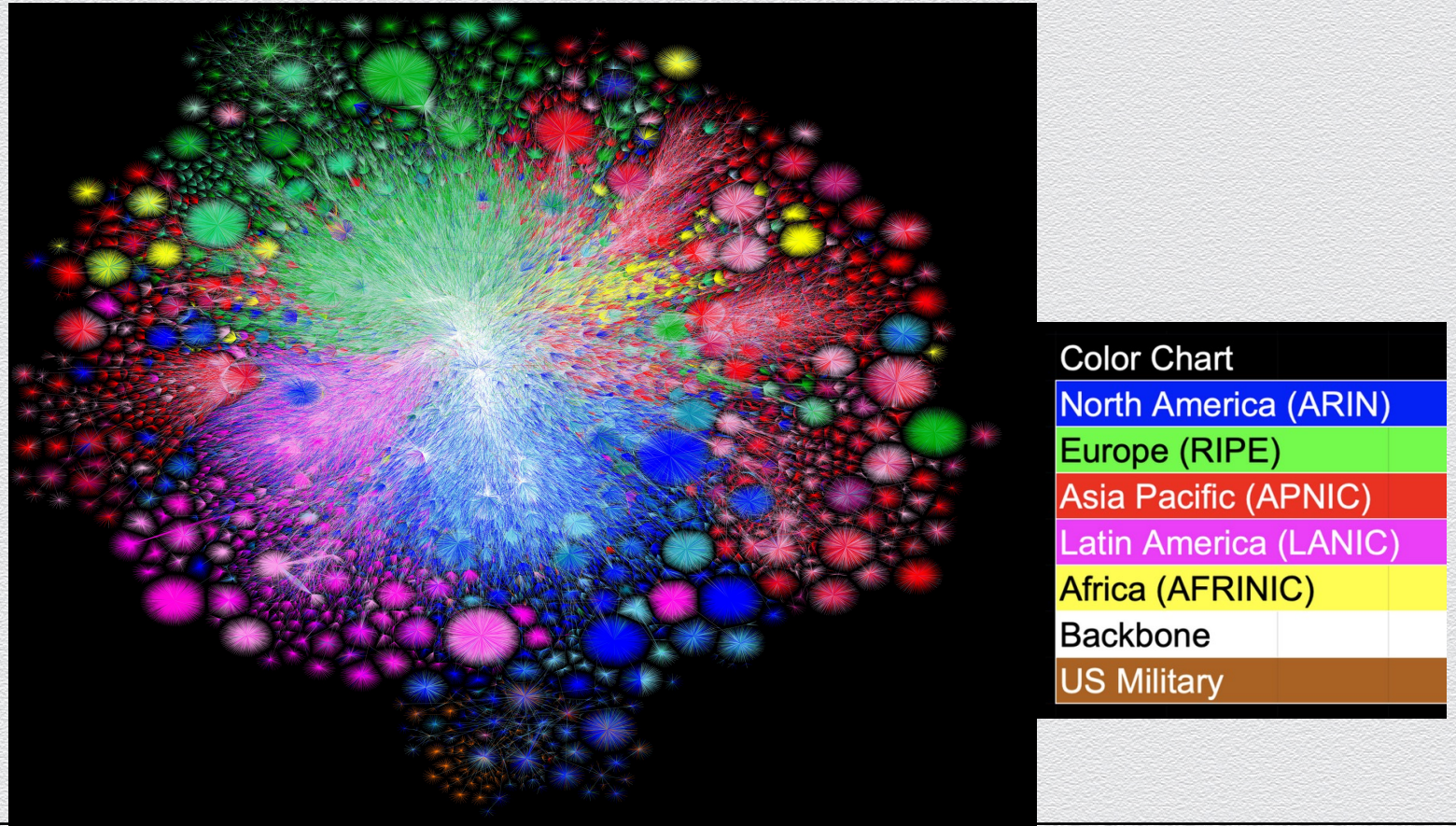
- A packet typically consists of
  - Control information
    - header and footer
  - Data
    - Payload

Protocols

- A protocol P uses the services of another protocol Q through **encapsulation**
  - A packet p of protocol P is encapsulated into a packet q of protocol Q
  - The payload of q is p
  - The control information of q is derived from that of p



Map of the Internet (2021 via BGP, OPTE project)



How do we make sense of this?

Network abstractions model how we build protocols and applications

- How data gets encapsulated
- What services are provided at each layer
- What they rely on from other layers

The OSI model

Networks are complex

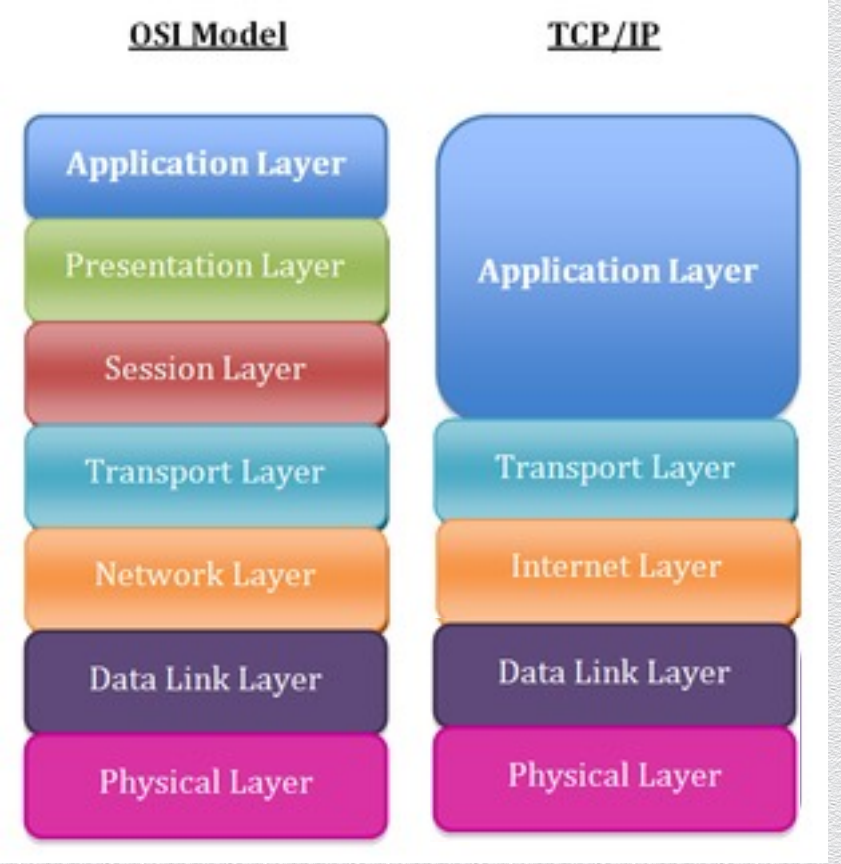
- Abstractions help us deal with them and build extensible, scalable systems

Some problems

- Different media: Wifi, Ethernet, Cellular, Bluetooth, ..
- No single managing entity: many ISPs, organizations, countries with own goals/policies
- Support for different types of applications using networks in different ways

Network layers

- The OSI (Open System Interconnect) Reference Model is a network model consisting of seven layers
- Created in 1983, OSI is promoted by the International Standard Organization (ISO)



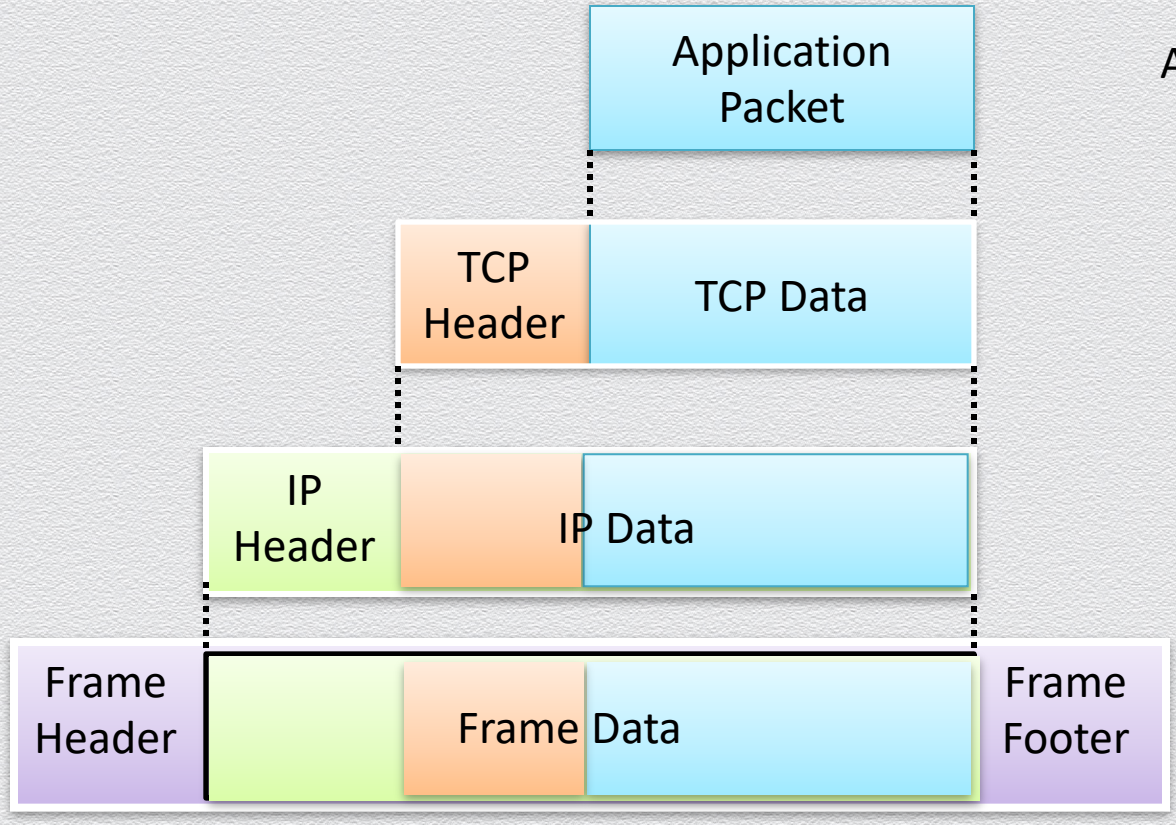
Layers: the classical picture

- Application – what users see, e.g., web page via HTTP
- Presentation – crypto, conversion between representations
- Session – can tie together multiple streams (e.g., audio & video)
- Transport – abstractions for getting data between applications
- Network – consider packets moving across entire network
- Link layer – consider frames moving between individual links
- Physical – moving bits across a link

A high-level picture

7. Application	Provides applications to users (eg. HTTP, SSH, …) Application-defined messages
4. Transport	Abstracts methods use to send data Examples: TCP, UDP Defines: port numbers;
3. Network	Provides way to get a packet to any other node on the Internet Protocols: IP (IPv4, IPv6) Defines: IP address (eg. 1.2.3.4)
2. Link	Protocols for sending data on individual links Examples: Wifi, Ethernet, Bluetooth, ... Defines: MAC address *(more on this later)*
1. Physical	Service: move bits to other node across link (Electrical engineering problem)

Internet packet encapsulation



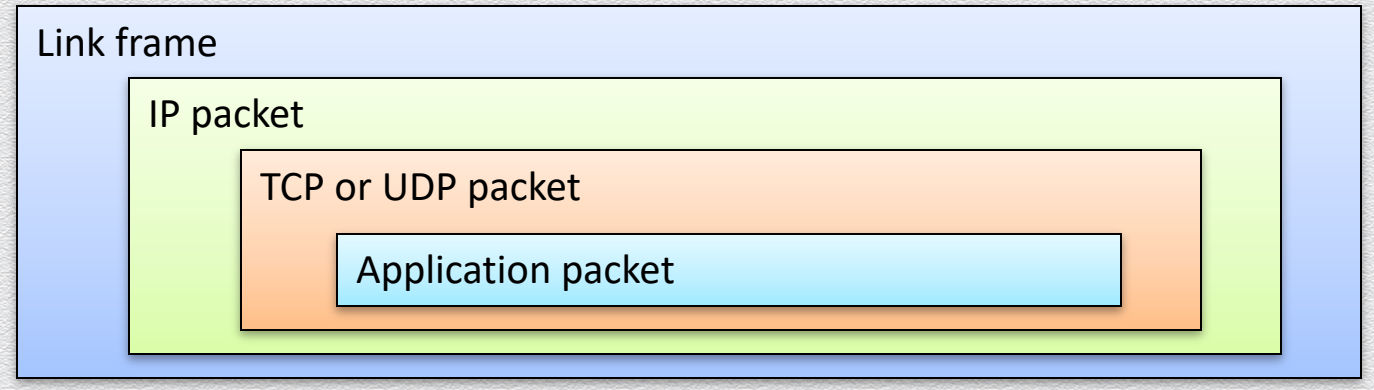
Application Layer

Transport Layer

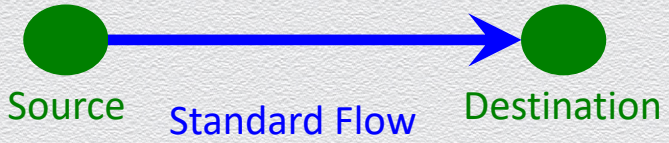
Network Layer

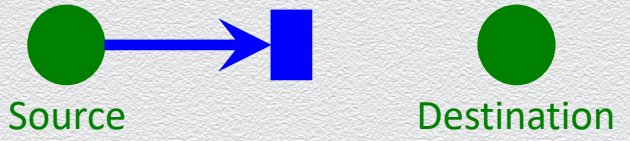
Link Layer

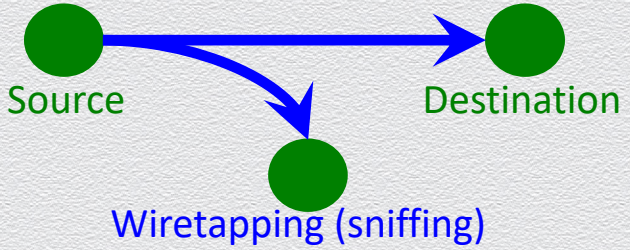
Internet packet encapsulation (cont.)

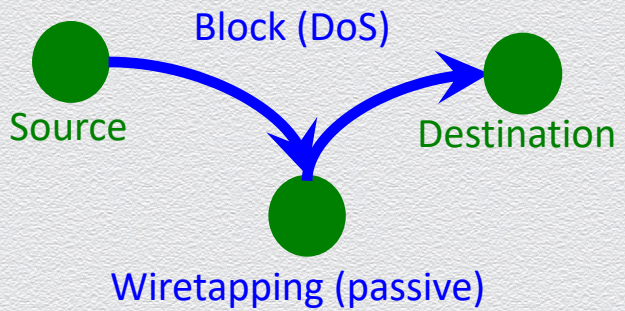


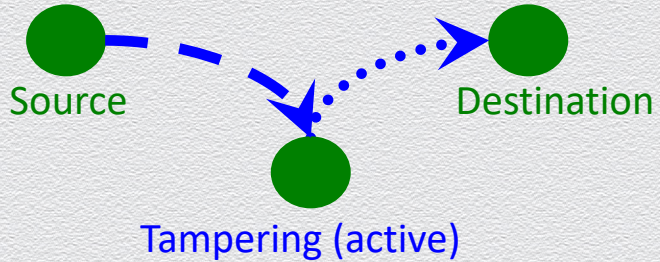
Network attacks

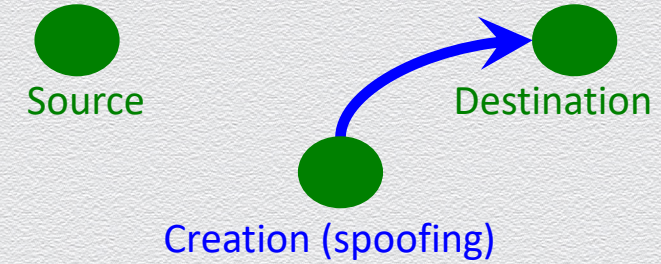












A framework of possible threats: The STRIDE model

Threats (STRIDE)	Description	Violation
Spoofing	Creating content by impersonating another account	Authentication
Tampering	Alteration of data or system	**Integrity**
Repudiation	Do not recognize actions performed	Non-repudiation (Certification)
Information Revelation	Exposing information to an unauthorized party	**Confidentiality**
Denial of Service	Inability to use services	**Availability**
Elevation of privileges	Possibility to carry out privileges without authorization	Authorization

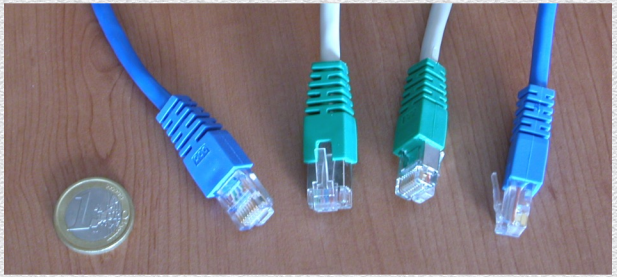
20.3 Physical & link layers

Network interfaces

Connects a computer or other device to a network

- Ethernet card, RJ-45 plug and cables
- WiFi adapter
- Bluetooth
- Cellular
- ...

A device may have multiple network interfaces



MAC addresses

- All interfaces have a MAC address
  - 48-bit number in hex (eg. 00-1A-92-D4-BF-86)
- Used to identify devices on a local network (eg. single house or building)
- First three bytes: assigned to manufacturers
  - E.g., 00-1A-A1 Cisco, 00-1B-11 D-Link , 00-0a-95 Apple
- Next three bytes: assigned per device, by manufacturer
  - Pre-programmed at factory, but can be changed by OS

20.4 Network layer

Internet Protocol (IP) goals

Addressing

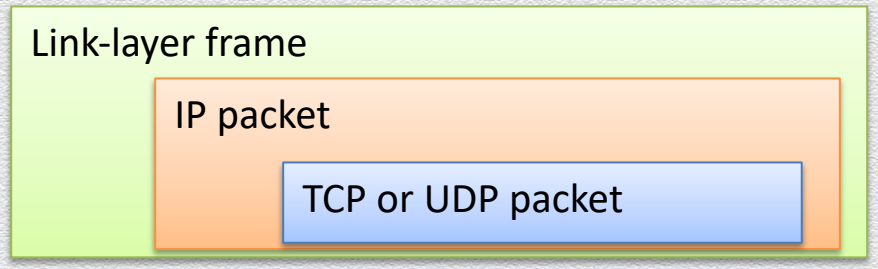
- Provide a unique identifier to every host on the Internet

Routing

- Unified abstraction to route between any two hosts, regardless of the type of networks involved (Ethernet, Wifi, Cellular, ...)

The Internet

- A network of networks



IP addressing

IP Version 4

- Each address is a 32-bit number

128.148.16.7

10000000 10010100 00010000 00000111

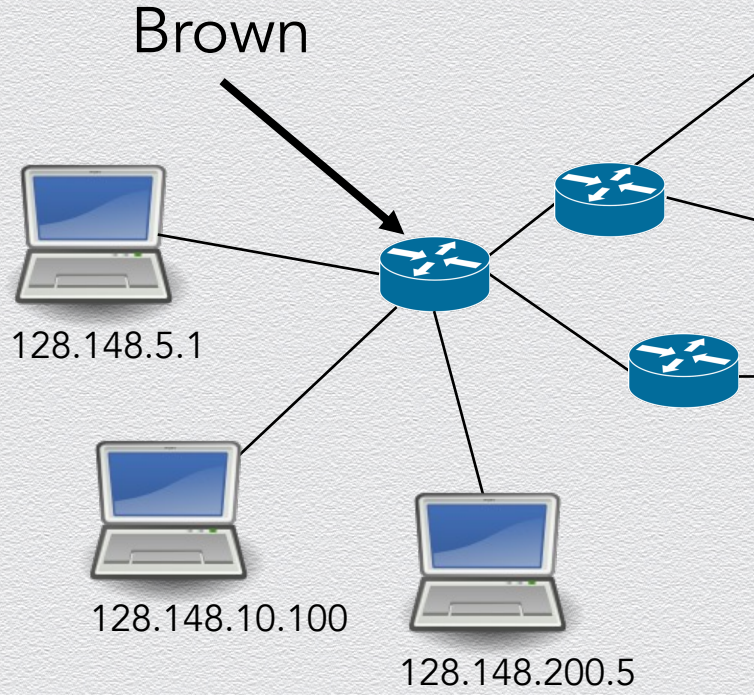
<u>Notation</u>

- Write each byte (“octet”) as a decimal number 0-255
- Called “dotted decimal” or ”dotted quad” notation

32 bits =>
2^{32} possible addresses

IP addressing (cont.)

A network can designate IP addresses
for its own hosts within its address range

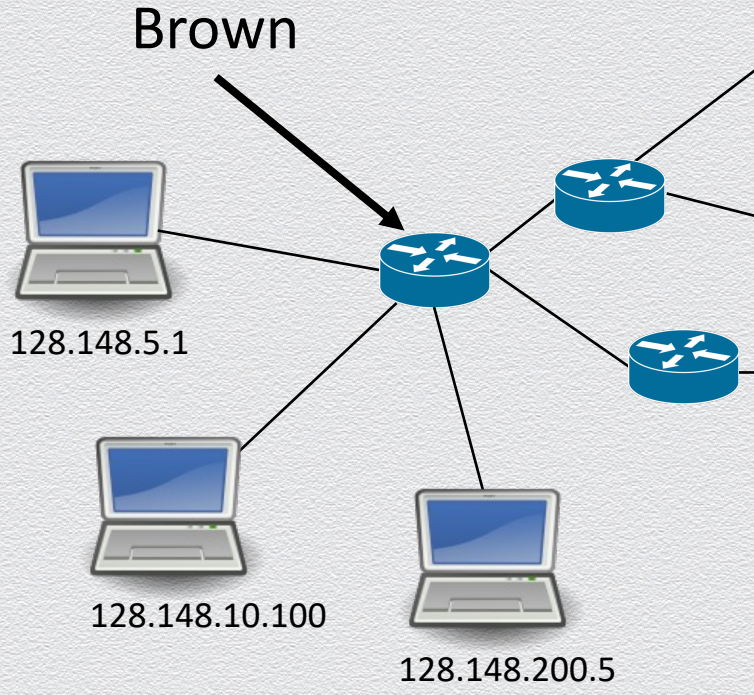


IP addressing (cont.)

*ICANN (Internet Corporation for Assigned Names and Numbers)

An IP address identifies…

- Who a host is: A unique number
- Where it is on the Internet
- Networks are allocated ranges of IPs by global authority (ICANN*)
  - Further subdivided by regions, ISPs, …
  - US-biased, especially in early internet
- Some IPs have special uses (eg. 127.0.0.1)
- E.g., Brown owns 128.148.xxx.xxx, 138.16.xxx.xxx



Viewing network configuration

MAC address

IPv4 address

```
deemer@ceres ~ % ip addr
2: enp7s0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu default ...
   link/ether c8:f7:50:55:9e:29 brd ff:ff:ff:ff:ff:ff
   inet 172.17.48.25/24 scope global enp0s31f6
      valid_lft forever preferred_lft forever
   inet6 fe80::caf7:50ff:fe55:9e29/64 scope link
      valid_lft forever preferred_lft forever
```

MAC OS: ifconfig
Windows: ipconfig

Gateway IP address

```
deemer@ceres ~ % ip route
127.0.0.0/8 via 127.0.0.1 dev lo
172.17.48.0/24 dev enp7s0  proto kernel
default via 172.17.48.1 dev eth0 src 172.17.44.22
```

MAC OS: route get <destination>
Windows: route print

Brown's IP space

Brown separates the network connecting dorms offices and academic buildings

- Class B network 138.16.0.0/16 (64K addresses)
- Class B network 128.148.0.0/16 (64K addresses)

CS department

- Several class C (/24) networks, each with 254 addresses
- Tstaff supported machines: 128.148.31.0/24, 128.148.33.0/24, 128.148.38.0/24
- Unsupported machines:128.148.36.0/24

Public information available: e.g. bgp.he.net

A simple Internet protocol

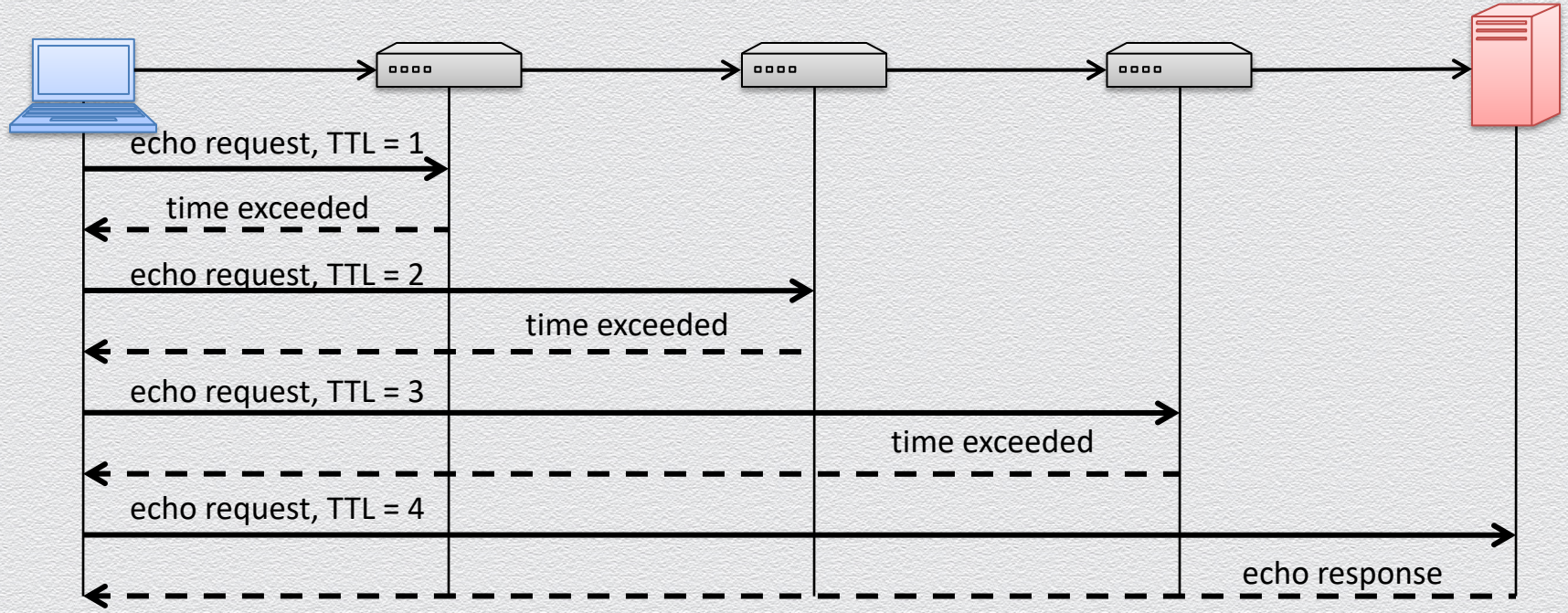
Brown separates the network connecting dorms offices and academic buildings

- Class B network 138.16.0.0/16 (64K addresses)
- Internet Control Message Protocol (ICMP)
- Used for network testing and debugging
- Network-layer protocol: simple messages about IP forwarding/routing
- Tools based on ICMP
  - Ping: send a message to an IP, get a response back
  - Traceroute: sends series ICMP packets with increasing TTL value to discover routes

Time to Live (TTL)

- When TTL reaches 0, router may send back an error
  - "ICMP TTL exceeded" message
- If it does, we can identify a path used by a packet!
- Traceroute takes advantage of this

Traceroute


echo request, TTL = 1
time exceeded
echo request, TTL = 2
time exceeded
echo request, TTL = 3
time exceeded
echo request, TTL = 4
echo response

Traceroute: Example

```
[deemer@Warsprite ~]$ traceroute -q 1 google.com
traceroute to google.com (142.251.40.174), 30 hops max, 60 byte packets
 1  router1-nac.linode.com (207.99.1.13)  0.621 ms
 2  if-0-1-0-0-0.gw1.cjj1.us.linode.com (173.255.239.26)  0.499 ms
 3  72.14.222.136 (72.14.222.136)  0.949 ms
 4  72.14.222.136 (72.14.222.136)  0.919 ms
 5  108.170.248.65 (108.170.248.65)  1.842 ms
 6  lga25s81-in-f14.1e100.net (142.251.40.174)  1.812 ms
```

Traceroute: Example (cont.)

```
[deemer@Warsprite ~]$ traceroute -q 1 amazon.co.uk
traceroute to amazon.co.uk (178.236.7.220), 30 hops max, 60 byte packets
 1  router2-nac.linode.com (207.99.1.14)  0.577 ms
 2  if-11-1-0-1-0.gw2.cjj1.us.linode.com (173.255.239.16)  0.461 ms
 3  ix-et-2-0-2-0.tcore3.njy-newark.as6453.net (66.198.70.104)  1.025 ms
 4  be3294.ccr41.jfk02.atlas.cogentco.com (154.54.47.217)  2.938 ms
 5  be2317.ccr41.lon13.atlas.cogentco.com (154.54.30.186)  69.725 ms
 6  be2350.rcr21.b023101-0.lon13.atlas.cogentco.com (130.117.51.138)  69.947 ms
 7  a100-row.demarc.cogentco.com (149.11.173.122)  71.639 ms
 8  150.222.15.28 (150.222.15.28)  78.217 ms
 9  150.222.15.21 (150.222.15.21)  84.383 ms
10  *
11  150.222.15.4 (150.222.15.4)  74.529 ms
                                        . . .
30  178.236.14.162 (178.236.14.162)  83.659 ms
```

Practicing Ping and Traceroute

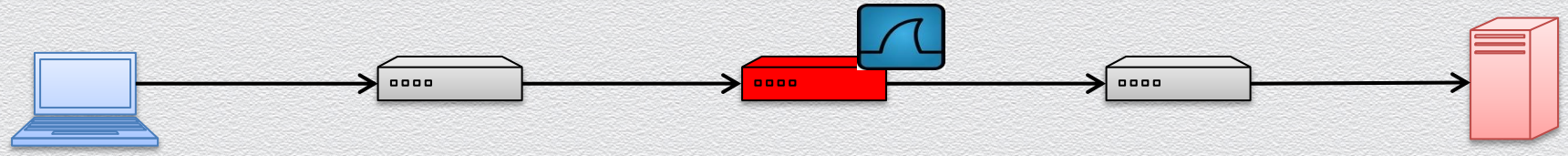
- Linux/Unix/Macos
  - ifconfig
  - ping www.brown.edu
  - traceroute www.brown.edu
- Windows
  - ipconfig
  - tracert www.brown.edu

Practice with Wireshark

- Checking a connection
  - Ping 127.0.0.1 (localhost)
  - Ping <your-ip-address> (ifconfig)
  - Ping www.brown.edu
- Traceroute www.brown.edu
- Let’s see in Wireshark
- Let’s see in geotraceroute.com

Sniffing: Not just for hosts?

- Any network device that sees packets could be an eavesdropper
- This is why we encrypt traffic in transit!



IP addressing (cont.)

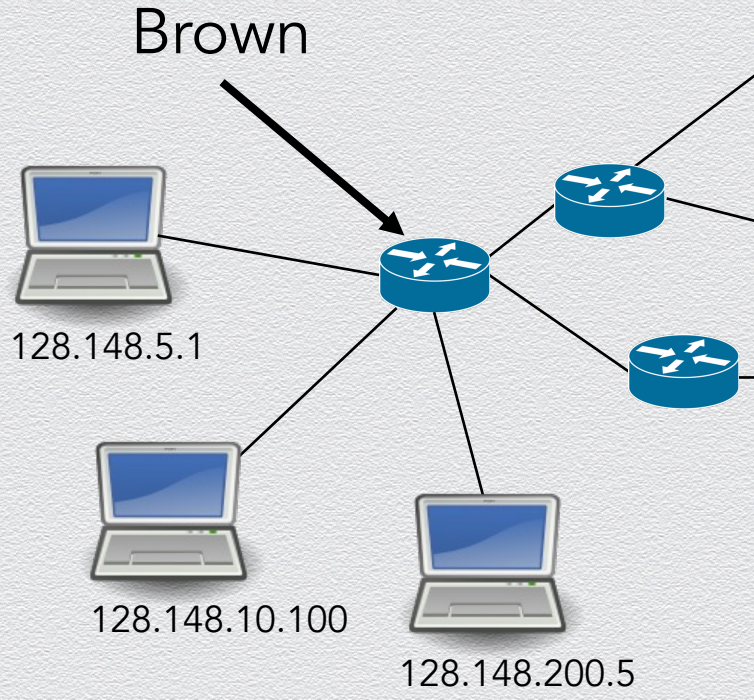
A network can designate IP addresses
for its own hosts within its address range

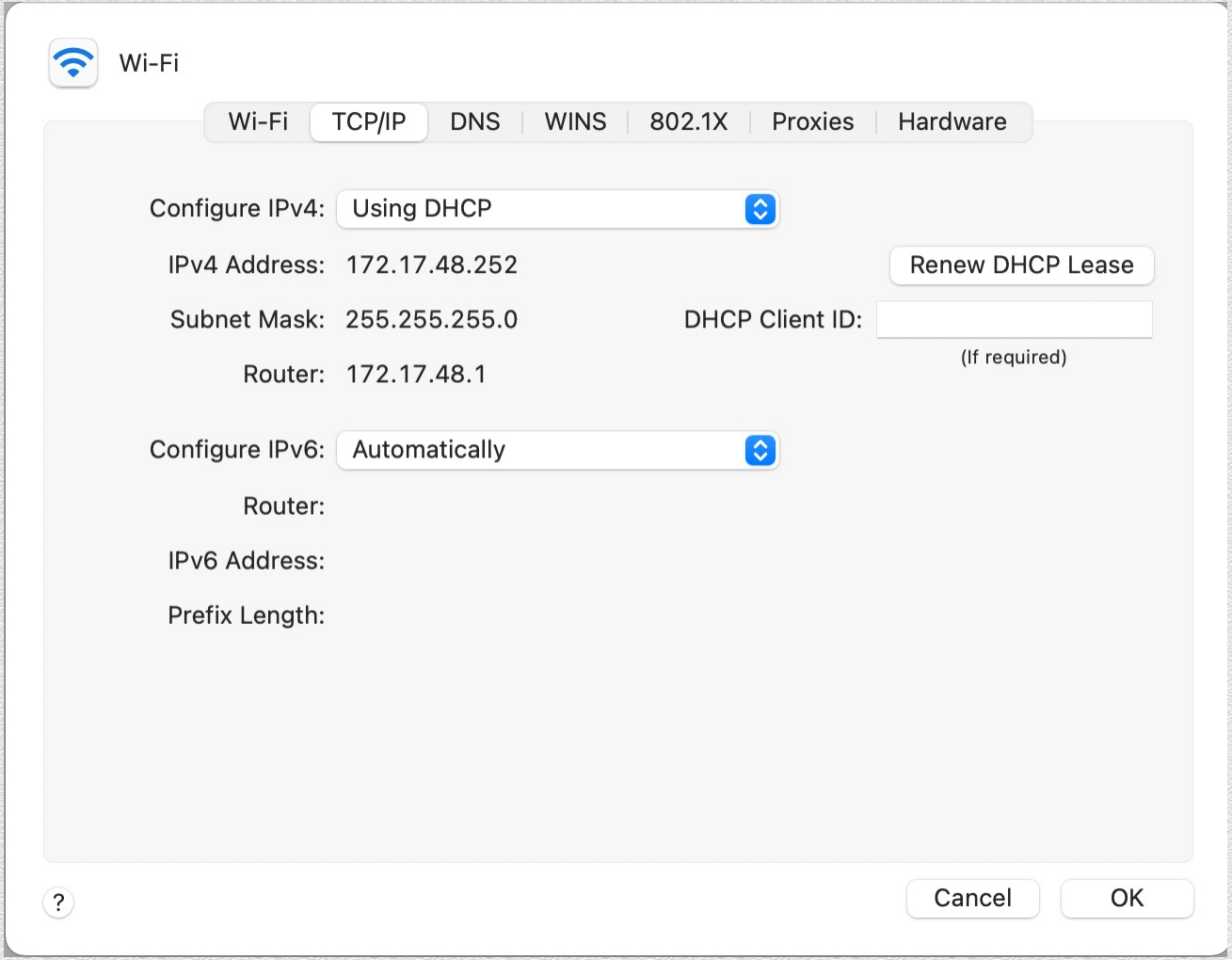
How? Every address has two parts

- Network part
  - Identifies the network (eg. "Brown") to the Internet
- Host part
  - Identifies individual hosts within Brown

Why?

- Routers need to check which network an address belongs to…




Wi-Fi
Wi-Fi
TCP/IP
DNS
WINS
802.1X
Proxies
Hardware
Configure IPv4:
Using DHCP
IPv4 Address:
172.17.48.252
Renew DHCP Lease
Subnet Mask:
255.255.255.0
DHCP Client ID:
(If required)
Router:
172.17.48.1
Configure IPv6:
Automatically
Router:
IPv6 Address:
Prefix Length:
?
Cancel
OK

Components of an IP

172.17.48.252

IPv4 Address: 172.17.48.252
Subnet Mask: 255.255.255.0
Router: 172.17.48.1

Addr: 172.17.48.252 **10101100 00010001 00110000 11111100**

Mask: 255.255.255.0 **11111111 11111111 11111111 00000000**

Key point: networks can be of different sizes!

The "subnet mask" defines what part of is the network part

Common prefix sizes

Prefix	IPs	Number of hosts	Note
1.2.3.0/24	1.2.3.*	2^8 = 256	Common for local networks (LANs) Old term: "Class C"
1.2.0.0/16	1.2.*.*	2^16 = 65536	Old term: "Class B" Large (or older) organizations
1.0.0.0/8	1.*.*.*	2^24 = ~16M	Old term: "Class A"
1.2.3.100/30	1.2.3.1-1.2.3.3	4	A smaller prefix

Special/private IP ranges

Prefix	Note
127.0.0.0/8	Localhost (for networks on same system), usually 127.0.0.1
192.168.0.0/16	Private: often used for home networks
10.0.0.0/8	Private: often used for larger organizations (eg. Brown)
172.16.0.0/12	Private: larger space for organizations, systems (eg. Docker)

- Used for LANs, private networks not publicly routable on the global internet RFC 1918

IP Address space and ICANN

Hosts on the internet must have unique IP addresses

Internet Corporation for Assigned Names and Numbers

- International nonprofit organization
- Incorporated in the US
- Allocates IP address space
- Manages top-level domains

Historical bias in favor of US corporations and nonprofit organizations

003/8 May 94 General Electric

009/8 Aug 92 IBM

012/8 Jun 95 AT&T Bell Labs

013/8 Sep 91 Xerox Corporation

015/8 Jul 94 Hewlett-Packard

017/8 Jul 92 Apple Computer

018/8 Jan 94 MIT

019/8 May 95 Ford Motor

040/8 Jun 94 Eli Lily

043/8 Jan 91 Japan Inet

044/8 Jul 92 Amateur Radio Digital

047/8 Jan 91 Bell-Northern Res.

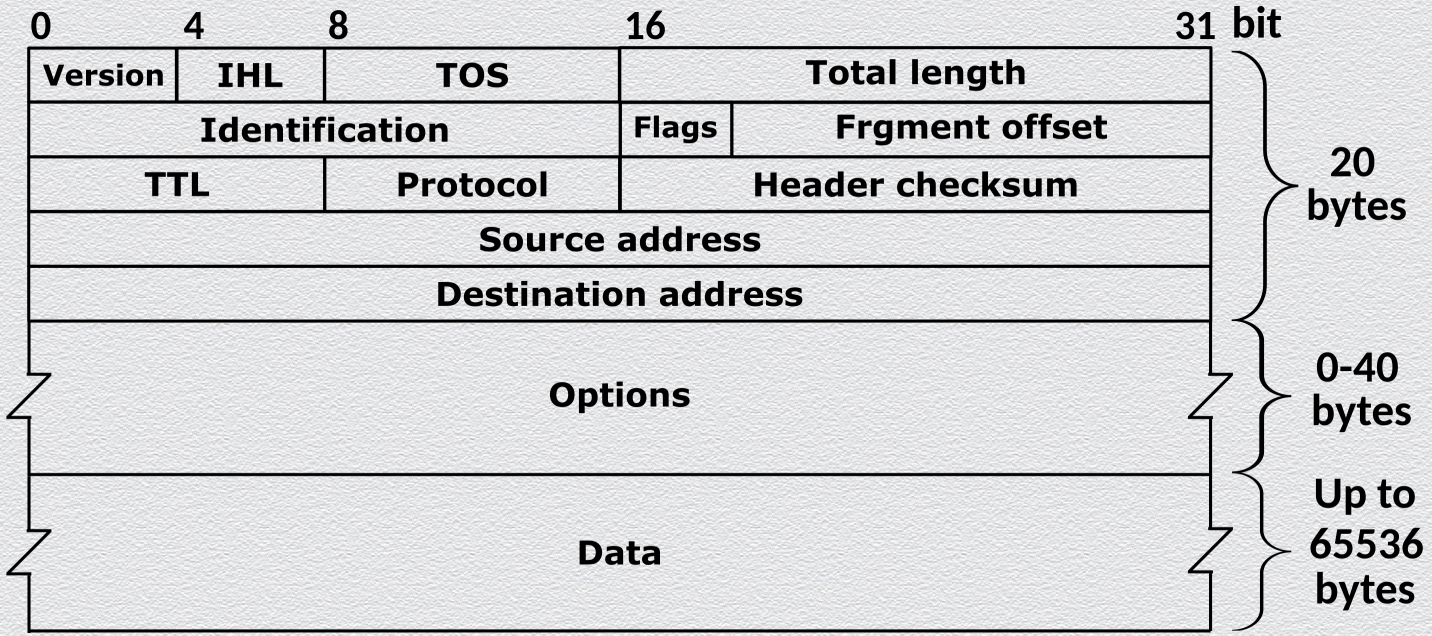
048/8 May 95 Prudential Securities

054/8 Mar 92 Merck

055/8 Apr 95 Boeing

056/8 Jun 94 U.S. Postal Service

The IPv4 Header

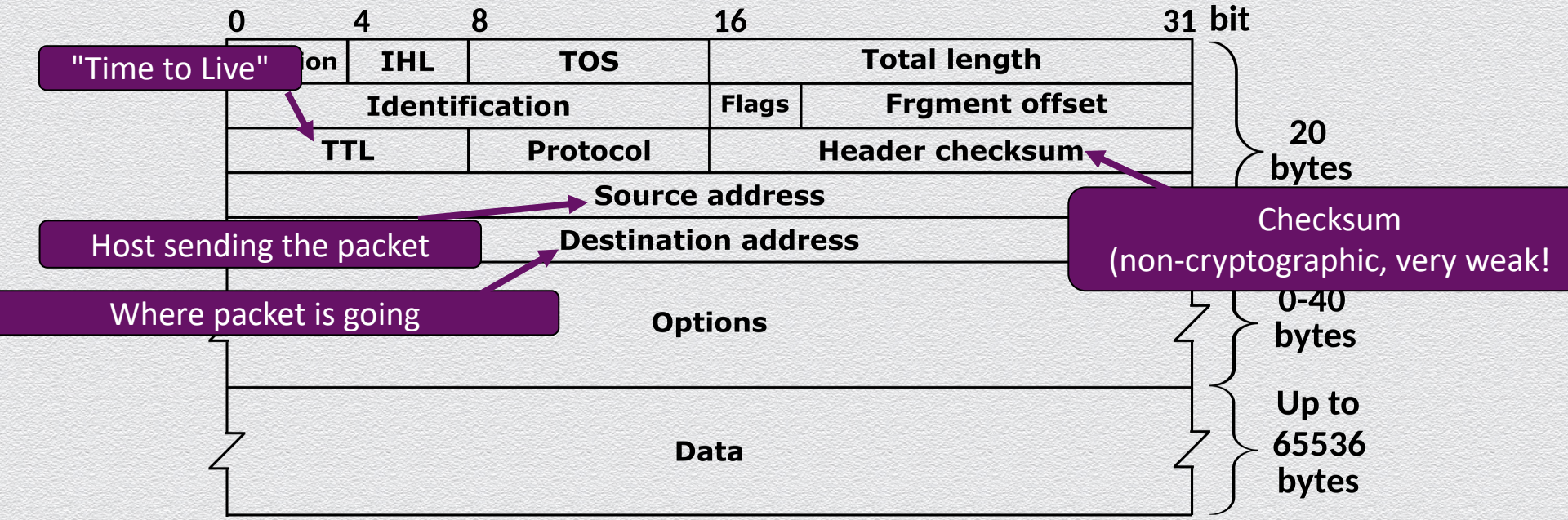


RFC 791

IP routing

- A router connects two or more networks
  - Maintains tables to forward packets to the appropriate network
  - Forwarding decisions based solely on the destination address
  - Hosts (regular systems) can be routers too!
- Routing table
  - Maps ranges of addresses to LANs or other gateway routers

The IPv4 Header

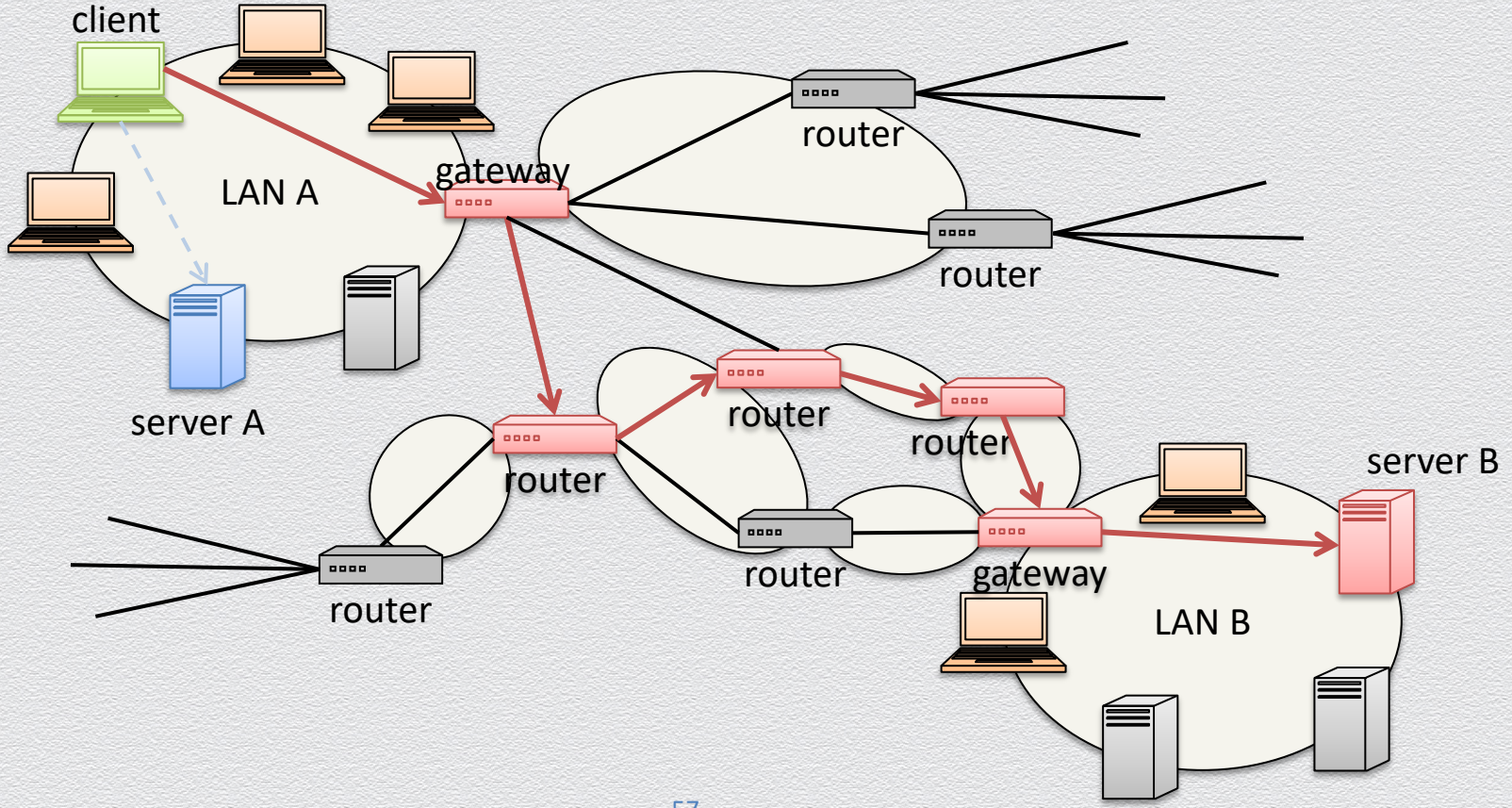


Example routing table

- "Default": where to send packets when they go to a network you don't know about
- Also known as "next hop"

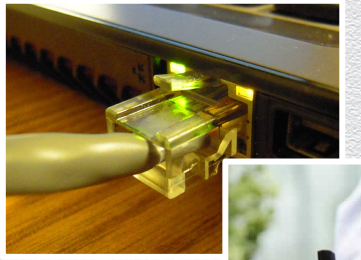
```
deemer@ceres ~ % ip route
127.0.0.0/8 via 127.0.0.1 dev lo
172.17.48.0/24 dev enp7s0  proto kernel
default via 172.17.48.1 dev eth0 src 172.17.44.22
```

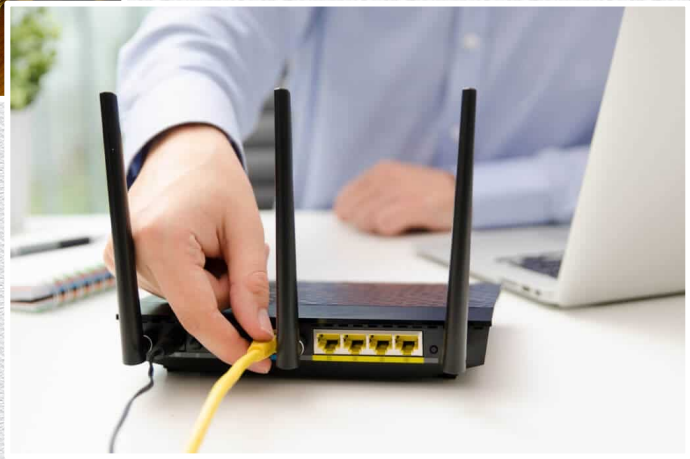
Routing Examples



20.5 Local network attacks

Where we are…

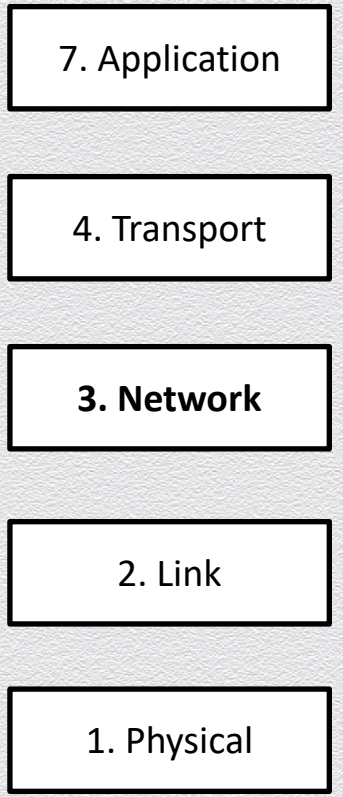




Local Area Network (LAN): ”small” network: within a building, house, floor of office, etc.

Security concerns before we even start talking to the wider Internet!

A high-level picture



Provides applications to users (eg. HTTP, SSH, …)
Application-defined messages

Abstracts methods use to send data
Examples: TCP, UDP
Defines: port numbers;

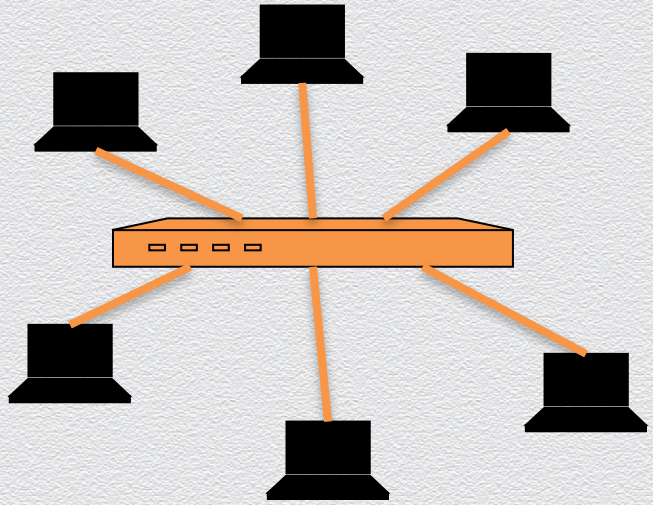
Provides way to get a packet to any other node on the Internet
Protocols: IP (IPv4, IPv6)
Defines: IP address (eg. 1.2.3.4)

Protocols for sending data on individual links
Examples: Wifi, Ethernet, Bluetooth, ...
Defines: MAC address *(more on this later)*

Service: move bits to other node across link
(Electrical engineering problem)

Switching

- A switch connects devices on a local area network (LAN)
- Has multiple interfaces, or ports
- Operates on link-layer frames
- As devices connect, learns MAC addresses of some or all the devices on the network



Recap: MAC Addresses

- All interfaces have a MAC address
  - 48-bit number in hex (eg. 00-1A-92-D4-BF-86)
- Used to identify devices on a *local* network (eg. single house or building)
- First three bytes: assigned to manufacturers
  - E.g., 00-1A-A1 Cisco, 00-1B-11 D-Link , 00-0a-95 Apple
- Next three bytes: assigned per device, by manufacturer
  - Pre-programmed at factory, but can be changed by OS

MAC Address Authentication/Filtering

- Link-layer security which effectively allows network to grant and deny access to specific devices
- Administrator configures lists of allowed and blocked MAC addresses, which may change over time
- When is necessary a mac address authentication?
  - E.g. Systems without a user interface (a keyboard, a touch screen, etc.)
  - https://guestwifi.net.brown.edu/guest/mac_create.php

IP and MAC Addresses

- Devices on a local area network have
  - IP addresses (network layer)
  - MAC addresses (data link layer)
- IP addresses are used for high level protocols
- MAC addresses are used for low level protocol
- Network administrator configures IP address and subnet on each machine
- How to translate IP Addresses into MAC addresses?

Address Resolution Protocol (ARP)

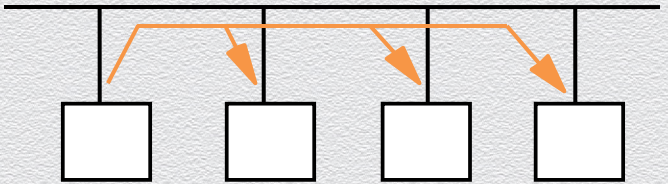
- Connects the network layer to the data link layer
- Maps IP addresses to MAC addresses
- Based on broadcast messages and local caching
- Does not support confidentiality, integrity, or authentication
- Defined as a part of RFC 826

ARP messages

- ARP broadcasts in a frame a requests of type

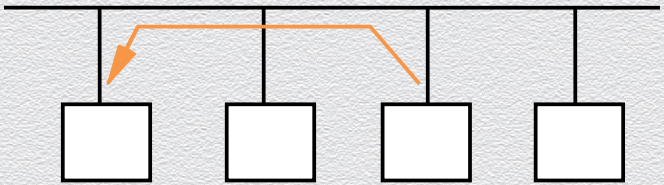
  who has <IP addressC >

  tell <IP addressA >



- Machine with <IP addressC> responds to requesting machine message

  <IP addressC > is at <MAC address>



- Requesting machine caches response

ARP Cache

- The Linux, Windows and OSX command arp - a displays the ARP table

Internet Address	Physical Address	Type
128.148.31.1	00-00-0c-07-ac-00	dynamic
128.148.31.15	00-0c-76-b2-d7-1d	dynamic
128.148.31.71	00-0c-76-b2-d0-d2	dynamic
128.148.31.75	00-0c-76-b2-d7-1d	dynamic
128.148.31.102	00-22-0c-a3-e4-00	dynamic

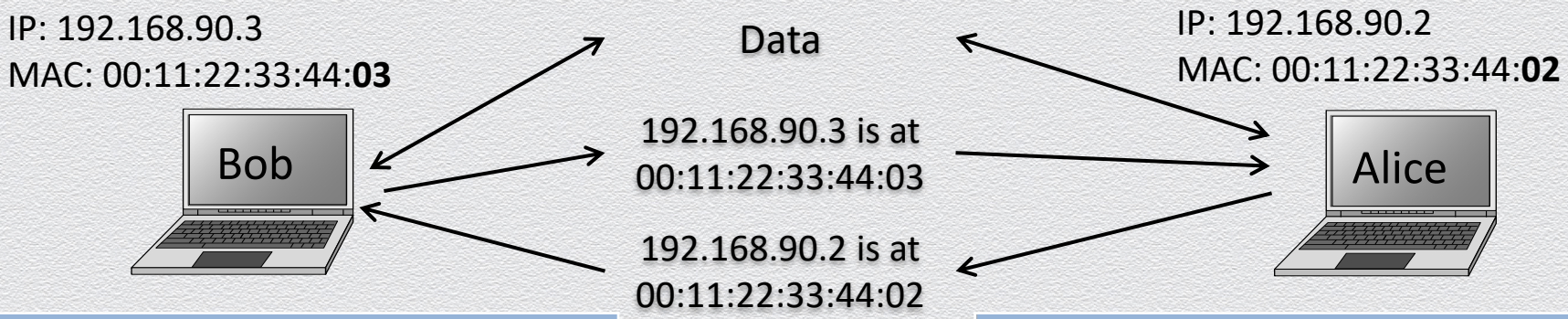
- Command arp –a –d flushes the ARP cache
  (with administrative privileges)
- ARP cache entries are stored for a configurable amount of time

ARP Security: Spoofing

- The ARP table is updated whenever an ARP response is received
- Requests are not tracked
- ARP announcements are not authenticated
- Machines trust each other
- A rogue machine can spoof other machines

ARP normal operation

- Normal operation
  - Alice communicates with Bob



ARP Cache	
192.168.90.2	00:11:22:33:44:**02**

ARP Cache	
192.168.90.3	00:11:22:33:44:**03**

Question (1)

After a great experience at CS1660 TA hours, Bob decides to message Alice about how much he appreciates the CS1660 staff. Eve would like to trick Bob into sending this network traffic to her (instead of Alice). Assuming Eve has access to everyone's MAC and IP, what ARP response could Eve send to Bob to accomplish this?

A. <Eve's IP> is at <Eve's MAC>

B. <Eve's IP> is at <Alice's MAC>

C. <Alice's IP> is at <Eve's MAC>

D. <Alice's IP> is at <Alice's MAC>

Answer (1)

After a great experience at CS1660 TA hours, Bob decides to message Alice about how much he appreciates the CS1660 staff. Eve would like to trick Bob into sending this network traffic to her (instead of Alice). Assuming Eve has access to everyone's MAC and IP, what ARP response could Eve send to Bob to accomplish this?

A. <Eve's IP> is at <Eve's MAC>

B. <Eve's IP> is at <Alice's MAC>

C. **<Alice's IP> is at <Eve's MAC>**

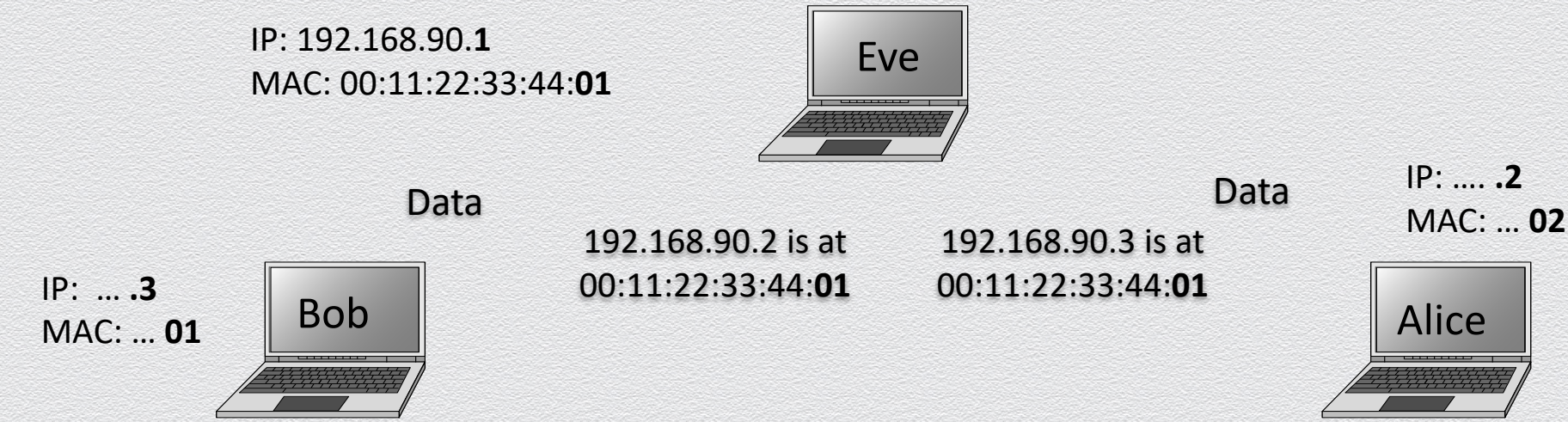
D. <Alice's IP> is at <Alice's MAC>

ARP Poisoning & ARP Spoofing

- Almost all ARP implementations are stateless
- An ARP cache updates every time that it receives an ARP reply
  - … even if it did not send any ARP request!
- Can “poison” ARP cache with gratuitous ARP replies
- Using static entries solves the problem but it is cumbersome to manage!

ARP Poisoning Attack

- Man-in-the-middle attack
  - ARP cache poisoning leads to eavesdropping



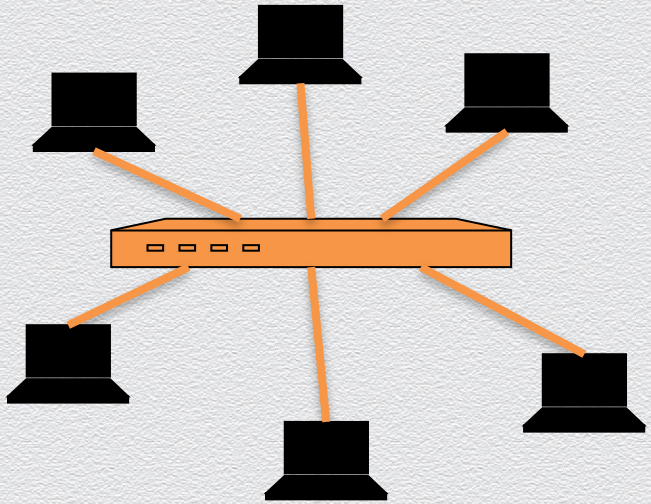
Poisoned ARP Cache	
192.168.90.2	00:11:22:33:44:01

Poisoned ARP Cache	
192.168.90.3	00:11:22:33:44:01

Background: switch operation

- As switch sees packets, it *learns* which MAC address is on which port
- MAC table: map of MAC address => Port
- When packet arrives
  - If destination MAC address is in table => send to that port
  - Otherwise, broadcast to all ports

Problems?

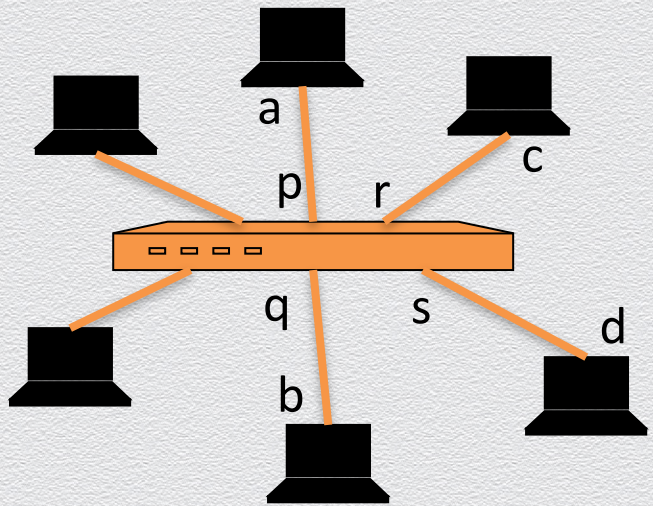


MAC learning: Example

- Table initially empty
- Frame (a, b) broadcast;
  - entry (a, p) added to table
- Frame (c, a) forwarded on p
  - entry (c, r) added to table
- Frame (a, c) forwarded on r
  - table unchanged
- Frame (a, d) broadcast
  - table unchanged

a	p

a	p
c	r

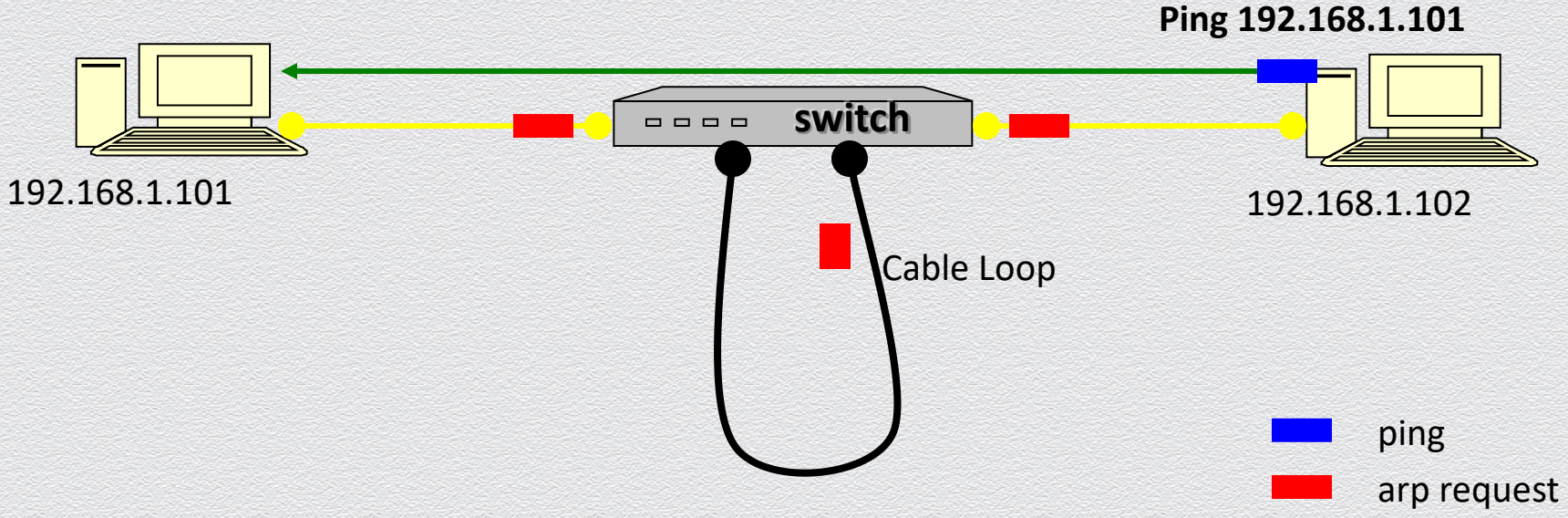


Attack on a learning switch

Idea

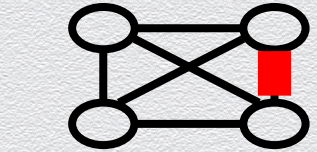
- Flood the switch with many packets from different source MAC addresses
- If MAC table is full, switch just broadcasts all packets to all ports

Network DOS using ARP

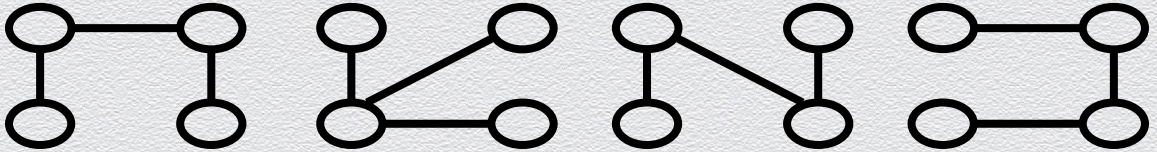


How can it be solved?

Spanning Tree Protocol (ISO 802.1D)



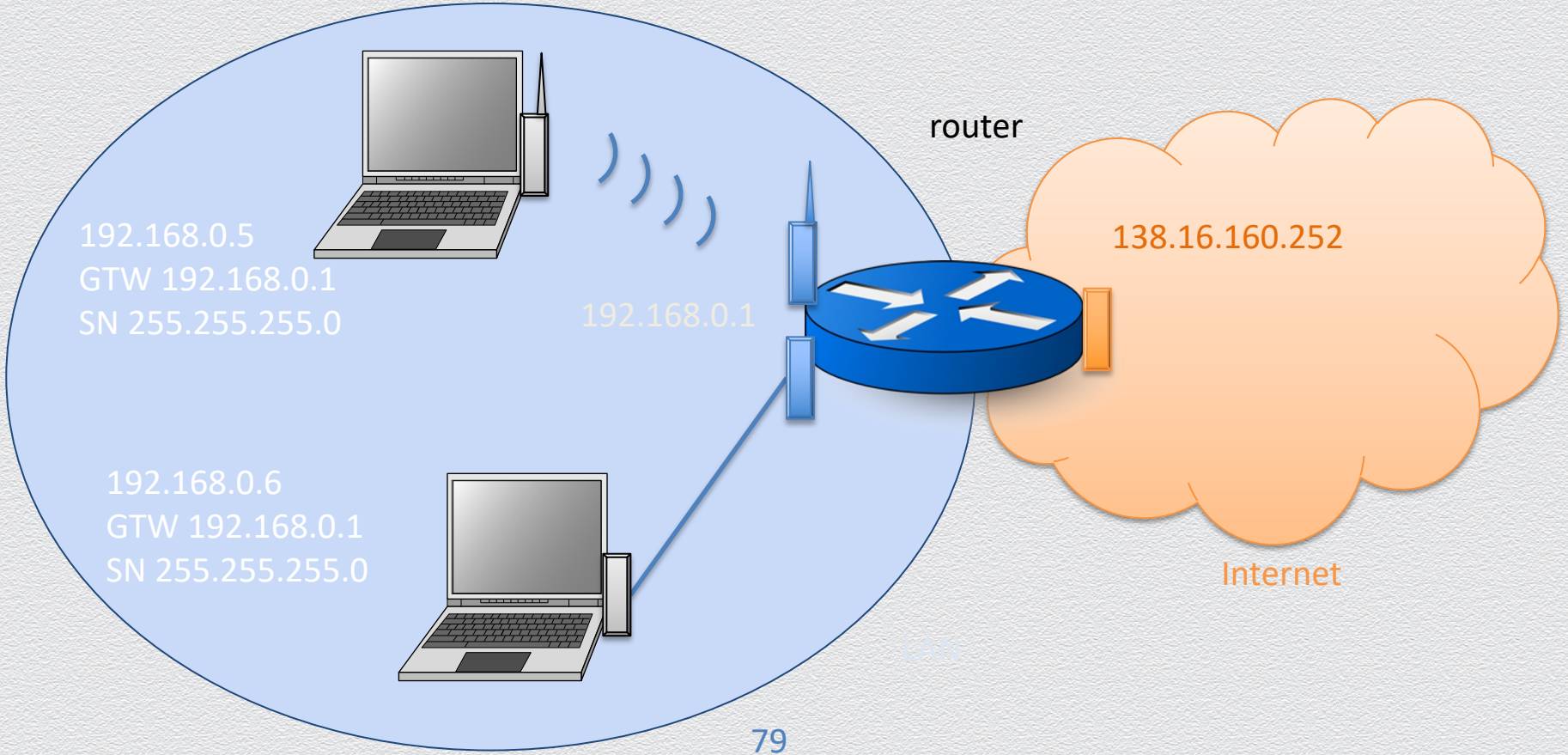
A Meshed Network



Four spanning trees of the Meshed Network

- Suppose you have a Meshed Network with bidirectional links that make loops/cycles…
- A spanning tree of the Meshed Network is the same network and no loops/cycles

From the LAN to the Internet



ARP, IP, TCP, UDP

How do you get an IP address?

Obtaining Host IP Addresses - DHCP

- Networks are free to assign addresses within block to hosts
- Tedious and error-prone: e.g., laptop going from CIT to library to coffee shop
- Idea: client asks network for IP on connection

=> But how? How to send packets with no IP address?

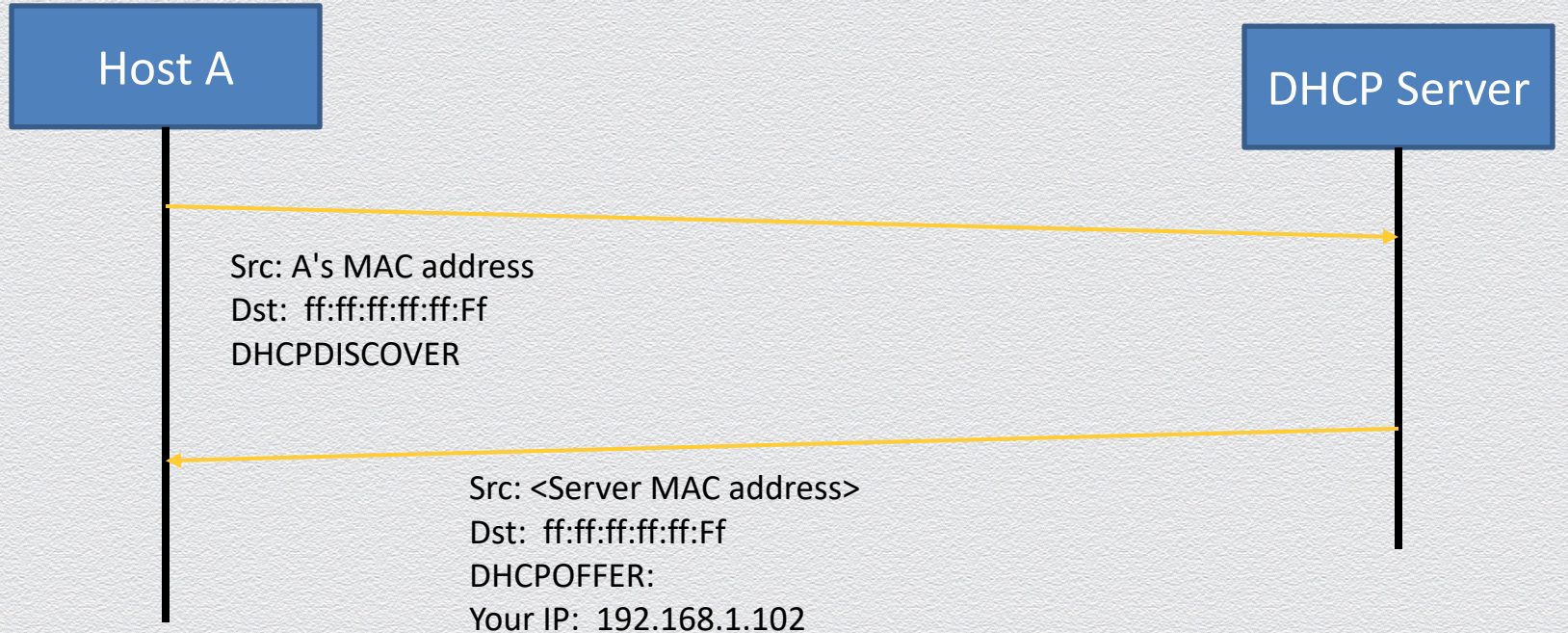
Broadcast traffic

Special MAC address: ff:ff:ff:ff:ff:ff

- Forwarded to all hosts on network!
- Used for link-layer protocols, particularly for finding IP addresses (DHCP, ARP)

Each IP subnet also has a broadcast address, usually last IP
(eg. 192.168.1.255)

Start of DHCP



Note: full protocol has more steps than this

DHCP server knows about "pool" of valid IPs + other settings for new hosts

Problems with DHCP?

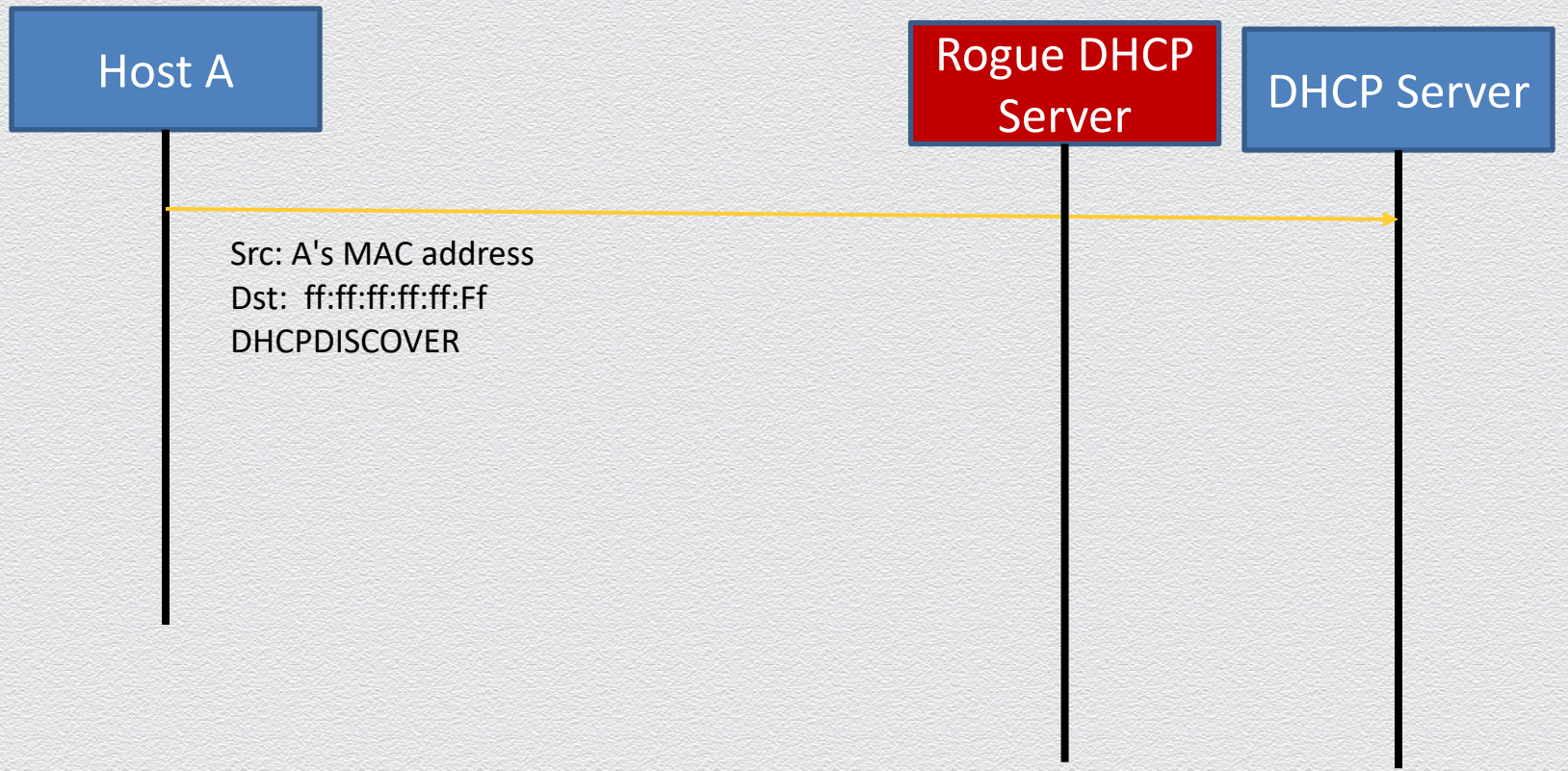
What happens if a random host decides to be a DHCP server?

Race condition!

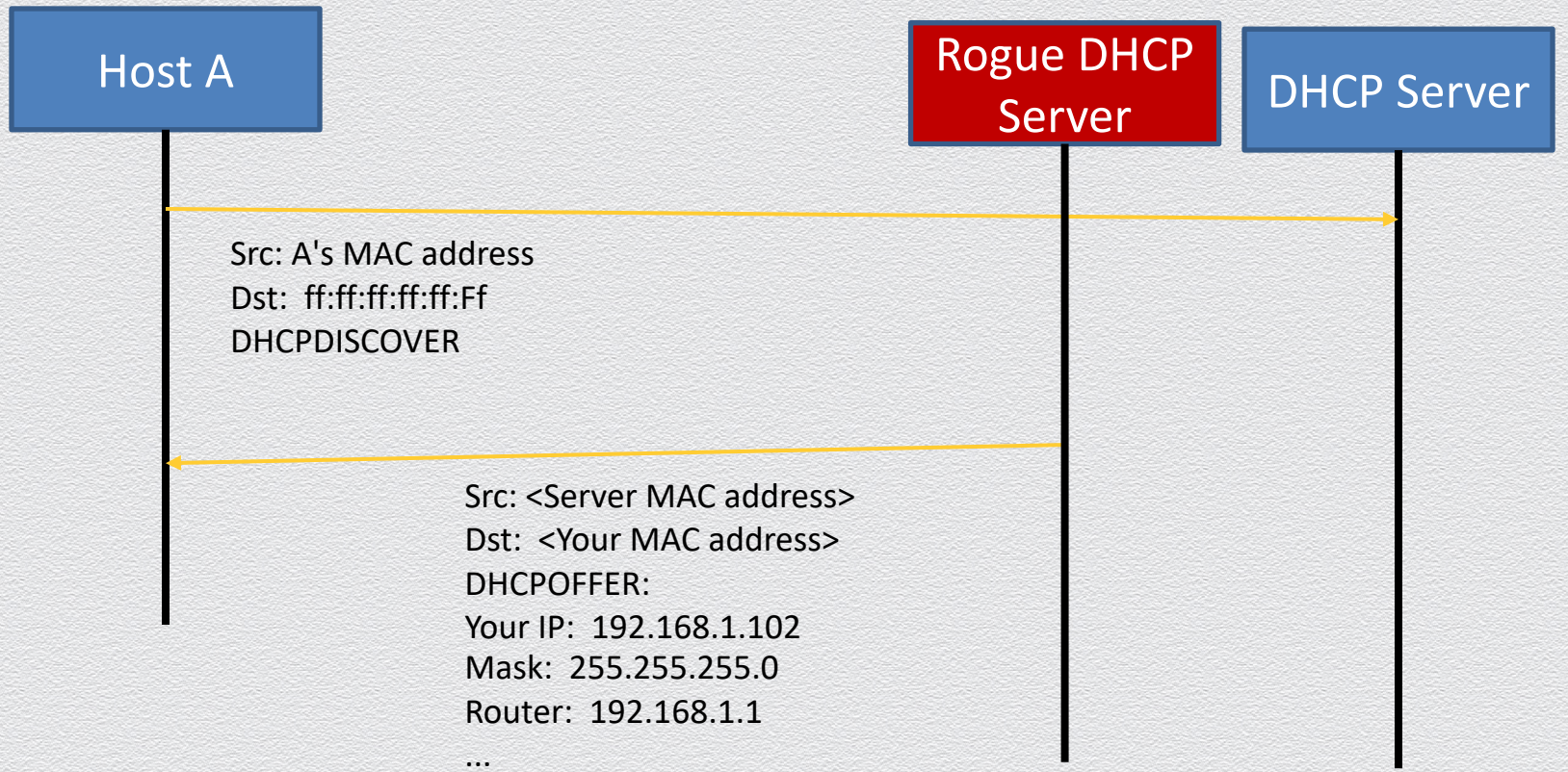
If an attacker can make an offer more quickly than the server, can assign a host's IP settings

Would be detected by the real DHCP server, though (why?)

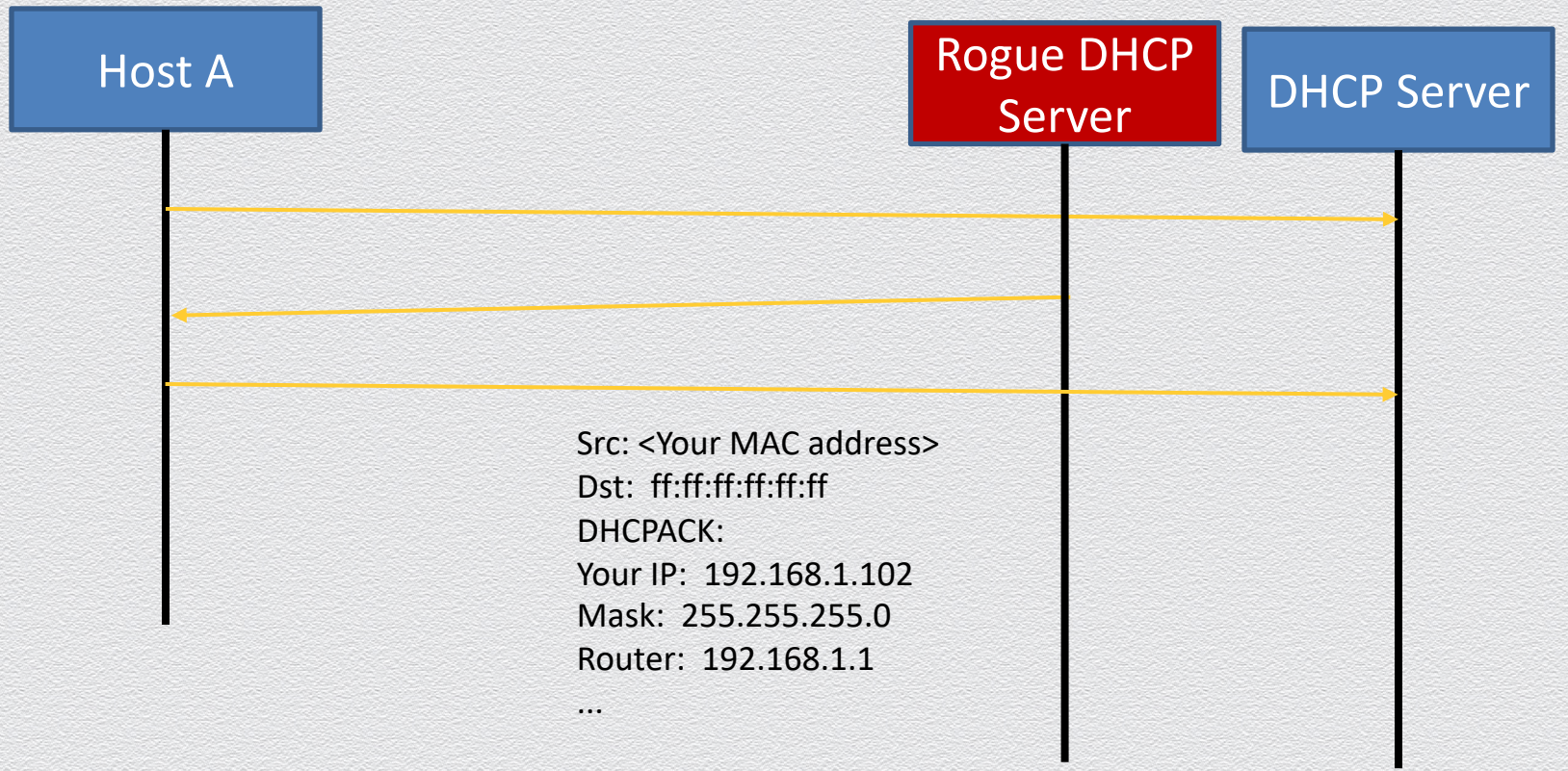
DHCP Spoofing



DHCP Spoofing



DHCP Spoofing



How to defend?

Initial DHCP messages are broadcast, so real server will see the rogue server's response

Can detect the attack!

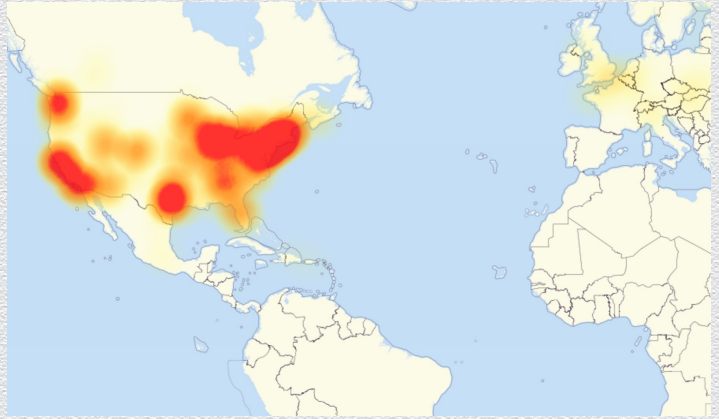
Why use broadcast? Allows multiple, redundant DHCP servers without extra coordination

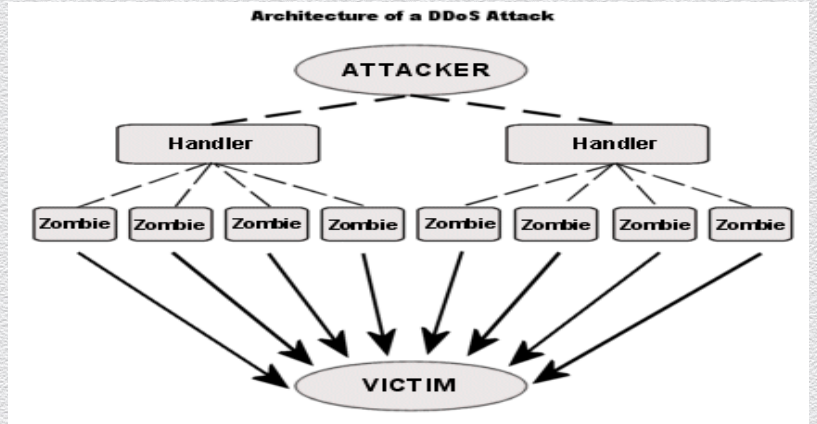
20.6 Dyn DDoS attack

It’s unfair! – I had no class but couldn’t watch my Netflix series!

On October 21, 2016, a large-scale cyber was launched

- it affected globally the entire Internet but particularly hit U.S. east coast
- during most of the day, no one could access a long list of major Internet platforms and services, e.g., Netflix, CNN, Airbnb, PayPal, Zillow, …
- this was a **Distributed Denial-of-Service (DDoS)** attack





DoS: A threat (mainly) against availability

Which main security property does a Denial-of-Service (DoS) attack attempt to defeat?

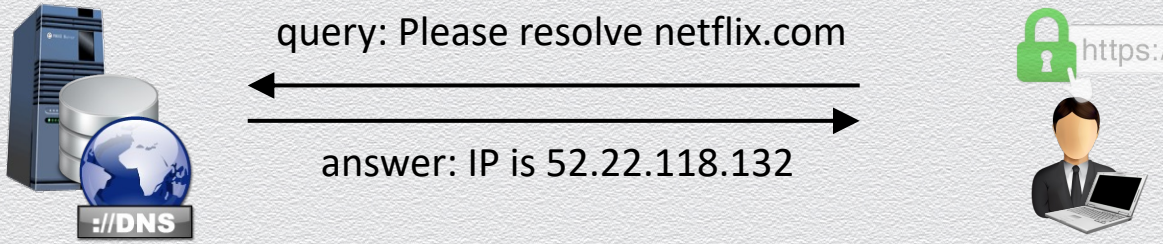
- availability; a user is denied access to authorized services or data
  - availability is concerned with preserving authorized access to assets
  - a DoS attack aims against this property; its name itself implies its main goal
- integrity & confidentiality; services or data are modified or accessed by an unauthorized user
  - elements of a DoS attack may include breaching the integrity or confidentiality of a system
  - but the end goal is disruption of a service or data flow; not the manipulation, fabrication or interception of data and services

10.1.1 DNS

The Domain Name Service (DNS) protocol

Resolving domain names to IP addresses

- when you type a URL in your Web browser, its IP address must be found
  - e.g., domain name “netflix.com” has IP address “52.22.118.132”
  - larger websites have multiple IP responses for redundancy to distributing load
- at the heart of Internet addressing is a protocol called DNS
  - a database translating Internet names to addresses



DNS name resolution is a critical asset – a target itself!

What main security properties must be preserved in such an important service?

- all properties in CIA triad are relevant!
- resolving domain names to IP addresses is a service that
  - must critically be available during all times – availability
    - or else your browser does not know how to connect to Netflix…
  - must critically be trustworthy – integrity
    - or else connections to malicious sites may occur (e.g., DNS-spoofing attacks)
  - must also protect database entries that are not queried – confidentiality
    - or else an attacker may find out about the structure of a target organization (e.g., zone-enumeration attacks)

Recursive name resolution: hierarchical search

Search is performed recursively and hierarchically across different type of DNS resolvers

- application-level (e.g., Web browser), OS-level (e.g., stub resolver): locally managed
- recursive DNS servers: query other resolvers and cache recent results

DNS entries:
<netflix.com, 52.22.118.132>

primary

subset of cached queried entries
(or information of other resolvers)

secondary

netflix.com

52.22.118.132
(or “non-existent”)

locally cached IP addresses
(at Web browser and OS)

Recursive name resolution: hierarchical search

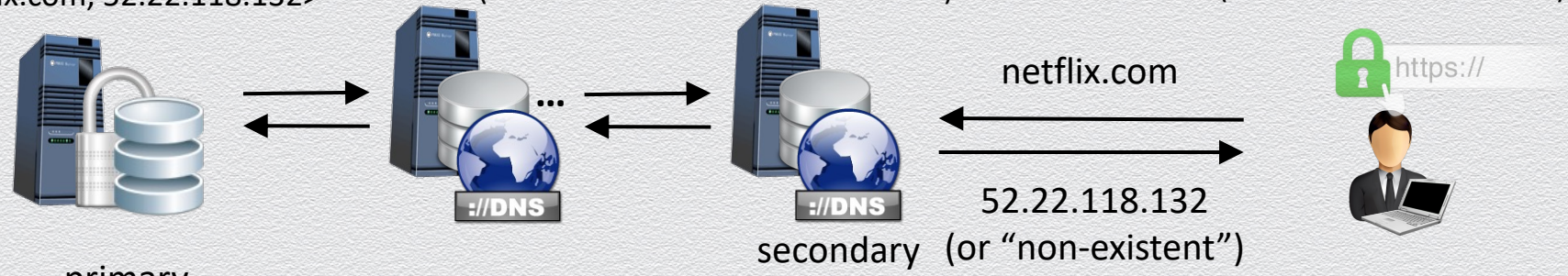
Search is performed recursively and hierarchically across different type of DNS resolvers

- application-level (e.g., Web browser), OS-level (e.g., stub resolver): locally managed
- recursive DNS servers: query other resolvers and cache recent results
- root name servers: refer to appropriate TLD (top-level domain) server
- TLD servers: control TLD zones such as .com, .org, .net, etc.

DNS entries:
<netflix.com, 52.22.118.132>

subset of cached queried entries
(or information of other resolvers)

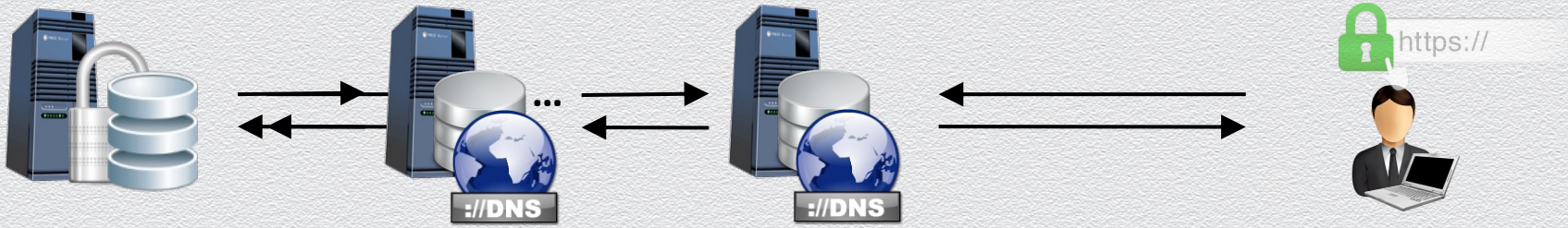
locally cached IP addresses
(at Web browser and OS)



Recursive name resolution: flexibility

Infrastructure allows for different configurations

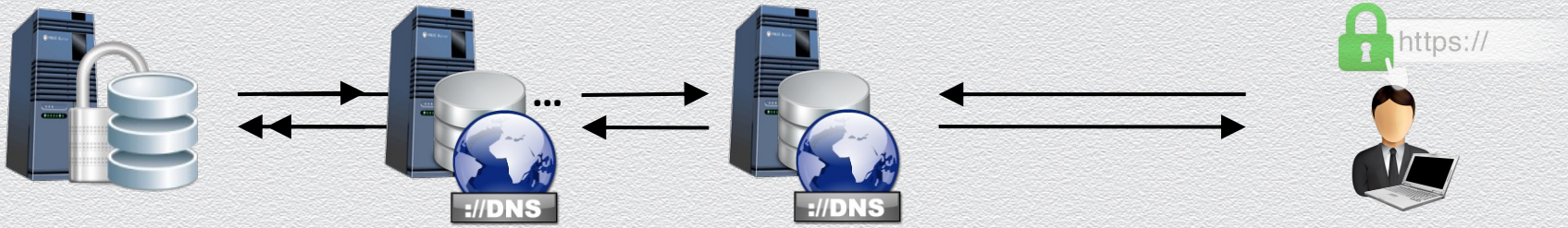
- authoritative-only servers: answer queries on zones they are responsible for
  - fast resolution, no forwarding, no cache
- caching / forwarding DNS servers: answer queries on any public domain name
  - recursive search / request forwarding, caching for speed, first-hop resolvers
- master / slaves DNS servers: authoritative servers replicating DNS data of their domains
- public / private DNS servers: control access to protected resources within an organization



Recursive name resolution: benefits

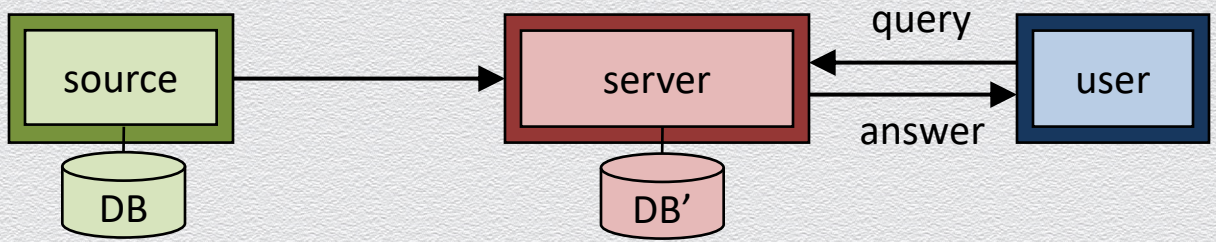
Why DNS uses non-authoritative name servers (that is, recursive resolution)?

- for more scalability & locality
  - high query loads can saturate the response capacity of primary servers
  - secondary do not have to store large volumes of DNS entries
  - cached recently queried domain names speed up searches due to locality of queries
- for added security / locality / scalability alone – not quite
  - e.g., non-authoritative name servers are untrusted and thus possibly compromised



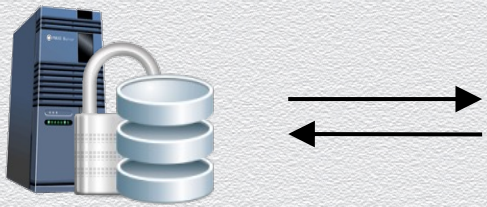
20.6.2 DNS integrity: Protocols DNSSEC & NSEC

DNS as a (distributed) database-as-a-service



DNS entries:
<netflix.com, 52.22.118.132>

subset of cached queried entries
(or information of other resolvers)



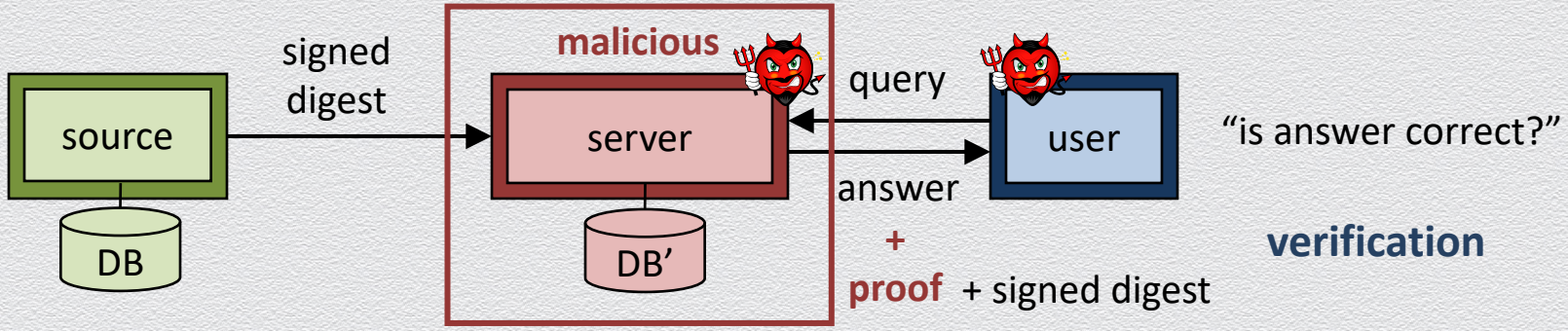
"primary"
name server

"secondary"
name server

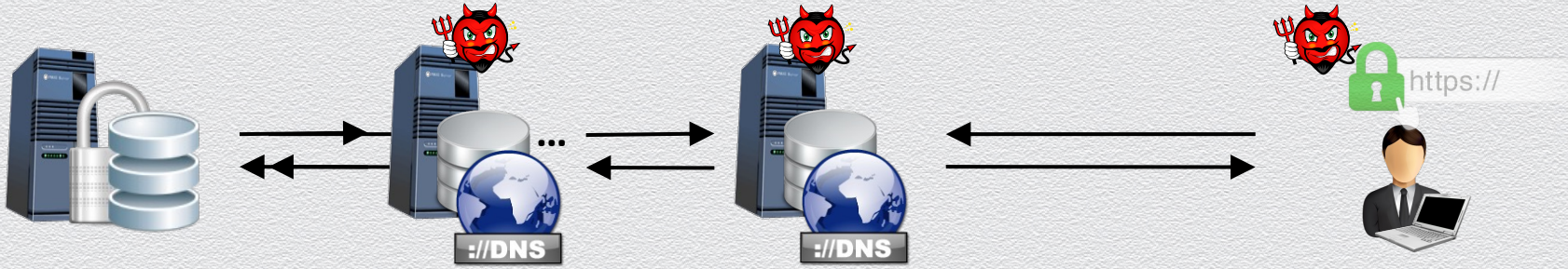
please resolve netflix.com

IP is 52.22.118.132
(or "aWa2j3netflix.com
is a non-existent domain")

A critical asset prone to attacks

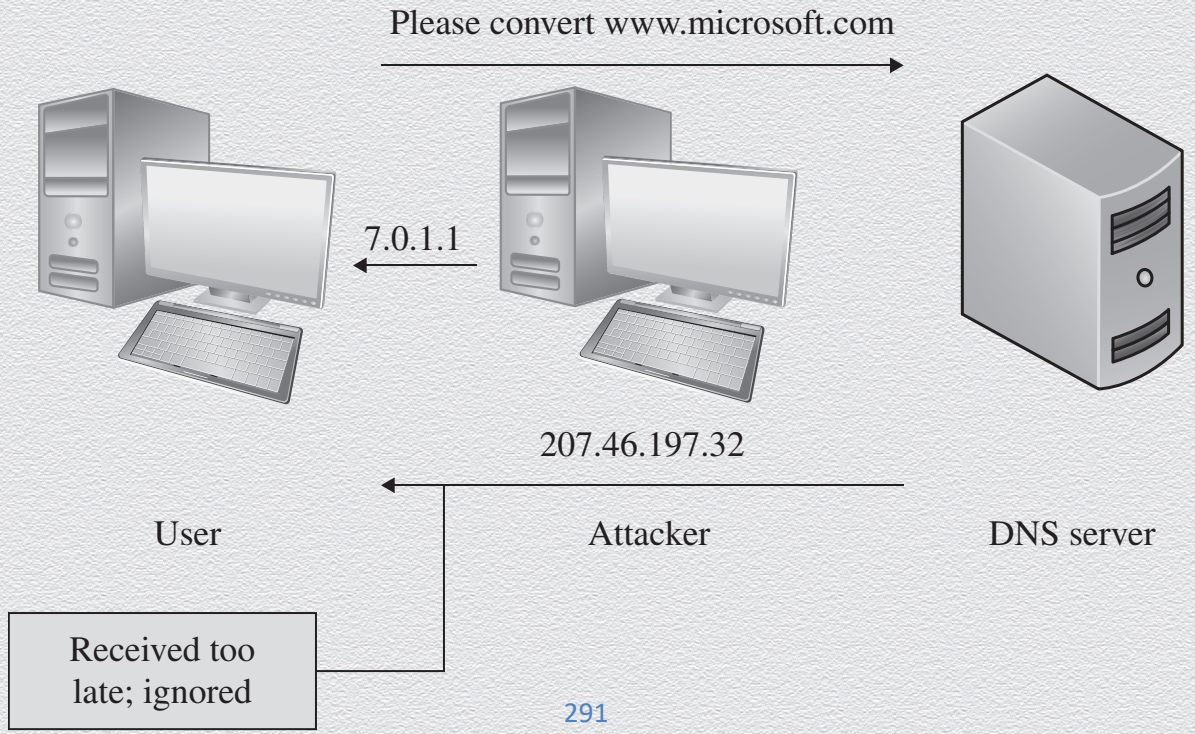


integrity **availability / confidentiality**



DNS spoofing (or cache poisoning)

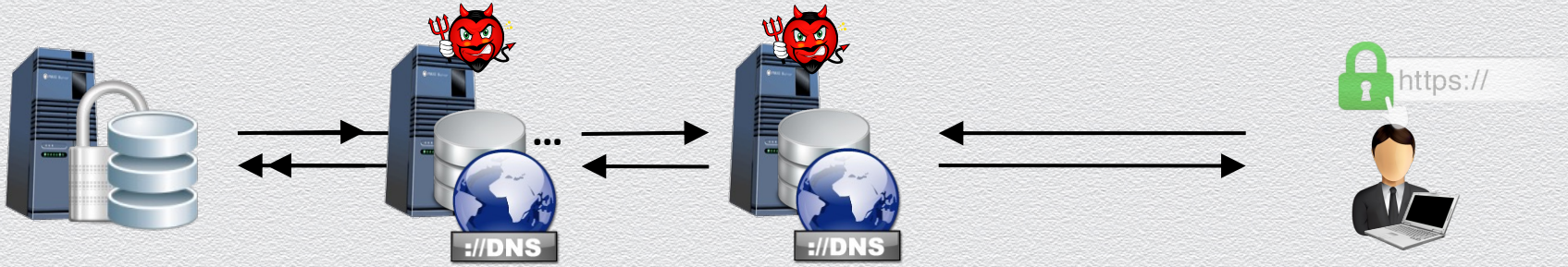
The attacker acts as the DNS server in order to redirect the user to malicious sites



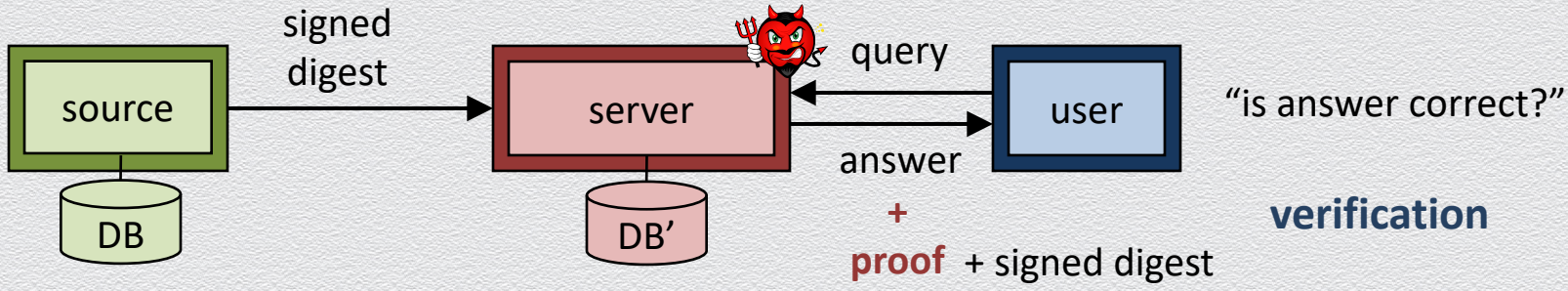
DNSSEC & NSEC

Security extension of DNS protocol to protect integrity of DNS data

- correct resolution, origin authentication, authenticated denial of existence
- specifications made by Internet Engineering Task Force (IETF) via RFCs
  - an RFC (request for comments) is a suggested solution under peer review
- challenges: backward-compatible, simplicity, confidentiality, who signs
  - NSEC (next secure record): extension that provides proofs of denial of existence



DNSSEC & NSEC: core idea



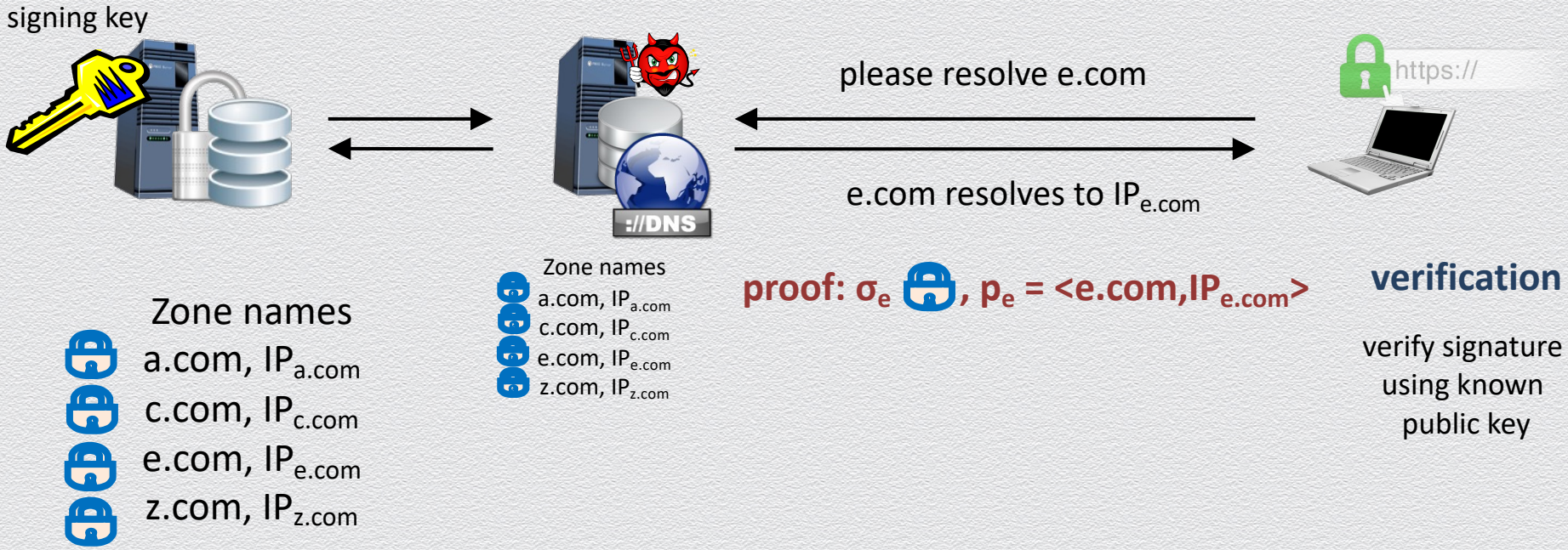
DNSSEC protocol: each DNS entry is pre-signed by primary name server

NSEC protocol:

- domain names are lexicographically ordered and then each pair of neighboring existing domain names is pre-signed by the primary name server
- non-existing names, e.g., aWa2j3netflix.com are proved by providing this pair "containing" missed query name, e.g., <awa.com, awb.com>

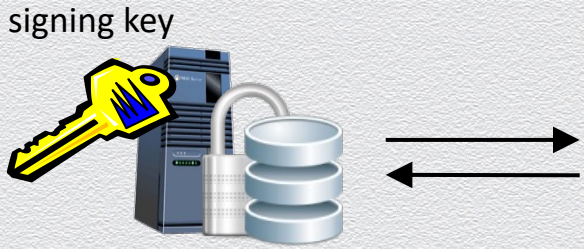
DNSSEC: example

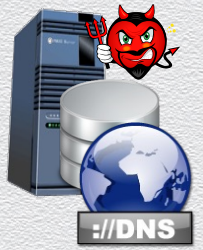
Each entry <domain name, IP address> in the database is individually signed by a primary DNS server and uploaded to secondary DNS servers in signed form



NSEC: example

Additionally, pairs of consecutive (in alphabetical order) domain names are individually signed by a primary DNS server and uploaded to secondary DNS servers in signed form





please resolve b.com

domain name b.com doesn't exist

proof: σ_1, p_1 = <a.com, c.com>

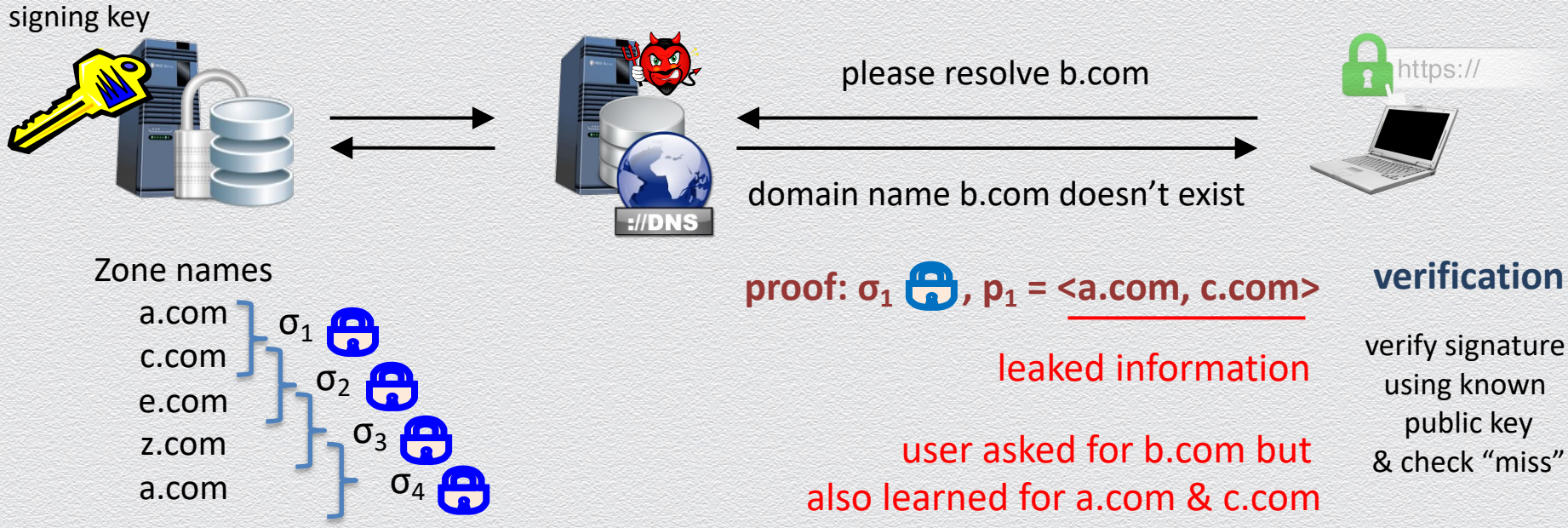
verification

verify signature using known public key & check "miss"

20.6.3 NSEC vulnerability: Protocols NSEC3 & NSEC5

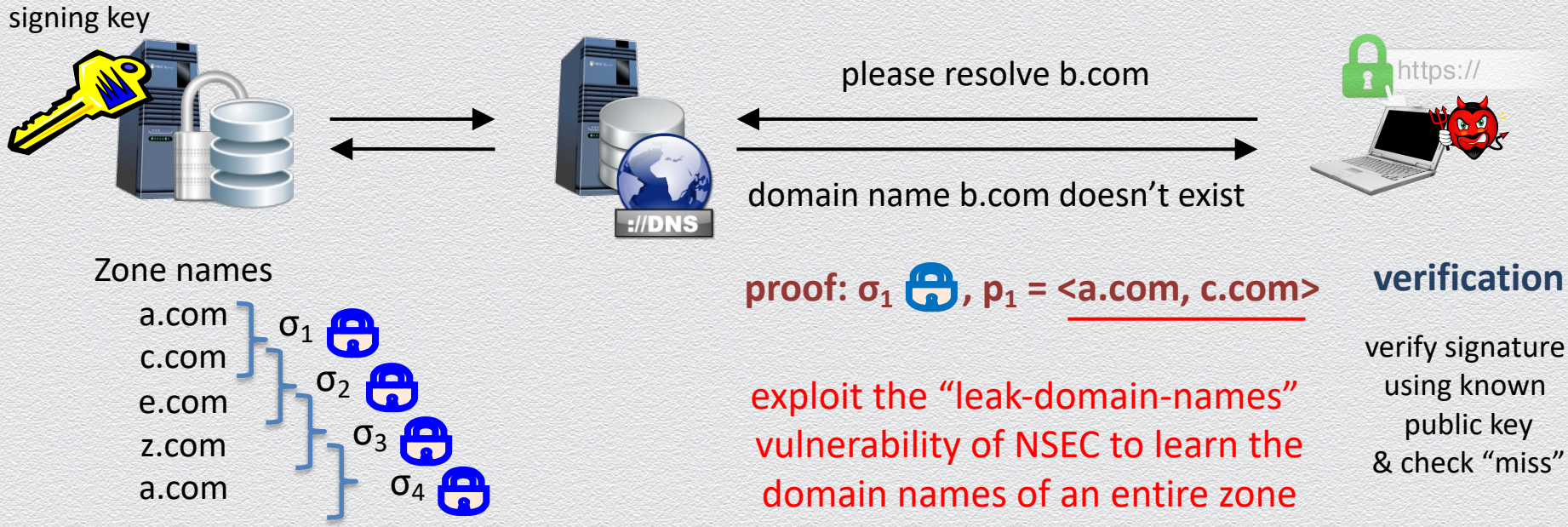
The problem

Proofs of non-existing names leak information about other unknown domain names



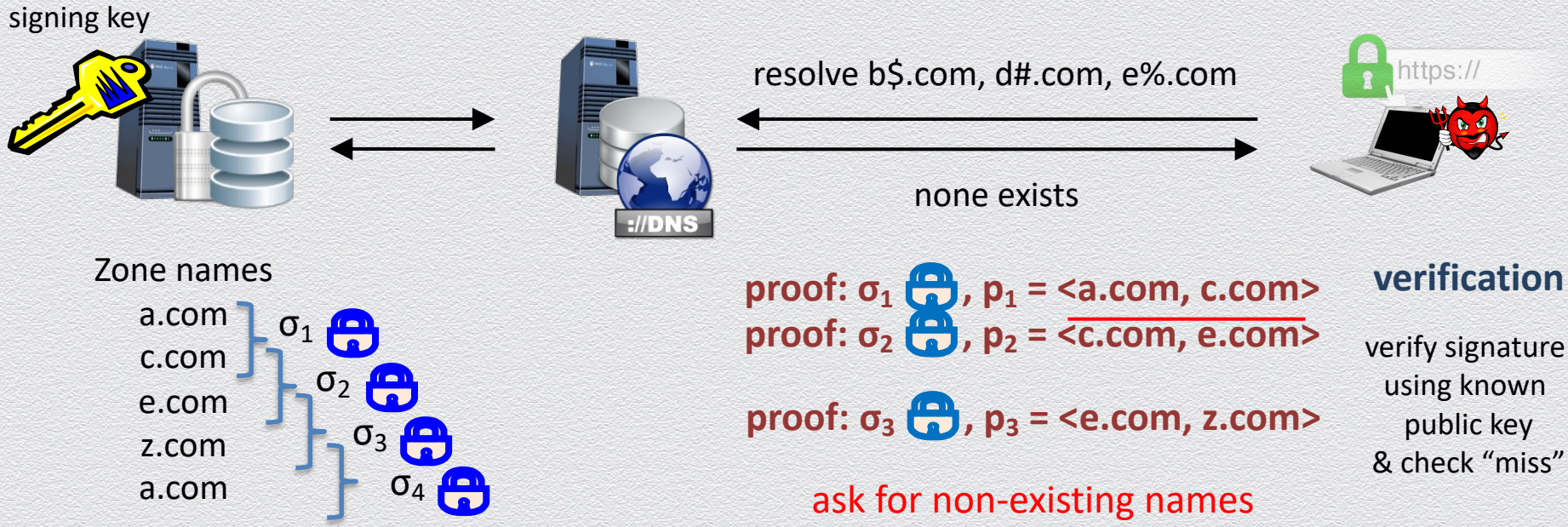
Zone enumeration attack: Main idea

An attacker can simply act as a “querier” to learn target organization’s network structure!



Zone enumeration attack: Example

An attacker can simply act as a “querier” to learn target organization’s network structure!

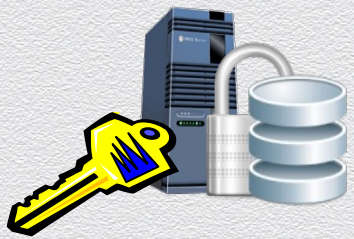


Zone enumeration attack: Result

An attacker can simply act as a “querier” to learn target organization’s network structure!

This attack may expose private device names (e.g., IoT devices which can be toehold for other attacks) or reveal other private data that many registries may have legal obligations to protect

NSEC3: NSEC in the hash domain



please resolve b.com

b.com is a non-existent domain

proof: σ_3 , p_3 = <dde45,zrit5>

Zone names

a.com
c.com
e.com
z.com

hash h →

a1bb5
23ced
zrit5
dde45

sort →

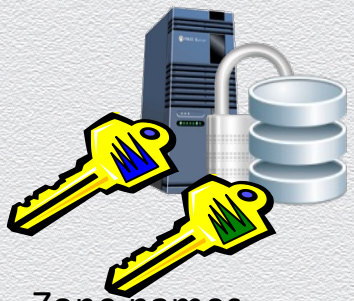
23ced
a1bb5
dde45
zrit5
23ced

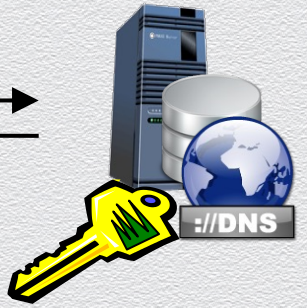
σ_1
σ_2
σ_3
σ_4

asked for b.com but
learned h(e.com) & h(z.com)

h(b.com) = ntwo4
e.g., h is SHA-256

NSEC5: A secure solution





please resolve b.com

b.com is a non-existent domain

proof: σ_3 , p_3 = <dde45,zrit5>

RSA-signature of f(b.com)

Zone names

a.com
c.com
e.com
z.com

hash h’ →

a1bb5
23ced
zrit5
dde45

sort →

23ced
a1bb5
dde45
zrit5
23ced

σ_1
σ_2
σ_3
σ_4

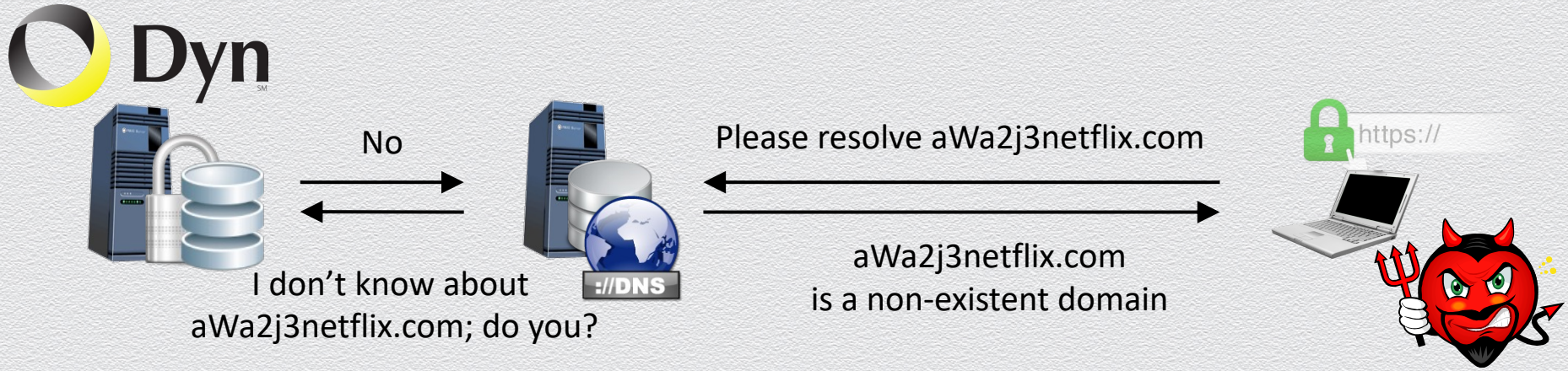
asked for b.com but
learned h’(e.com) & h’(z.com)

h’(b.com) = ntwo4
h: as in NSEC3
f: “message transformation” hash

h’(x) = h(RSA-Sign( , f(x)))

20.6.4 The Dyn DDoS attack

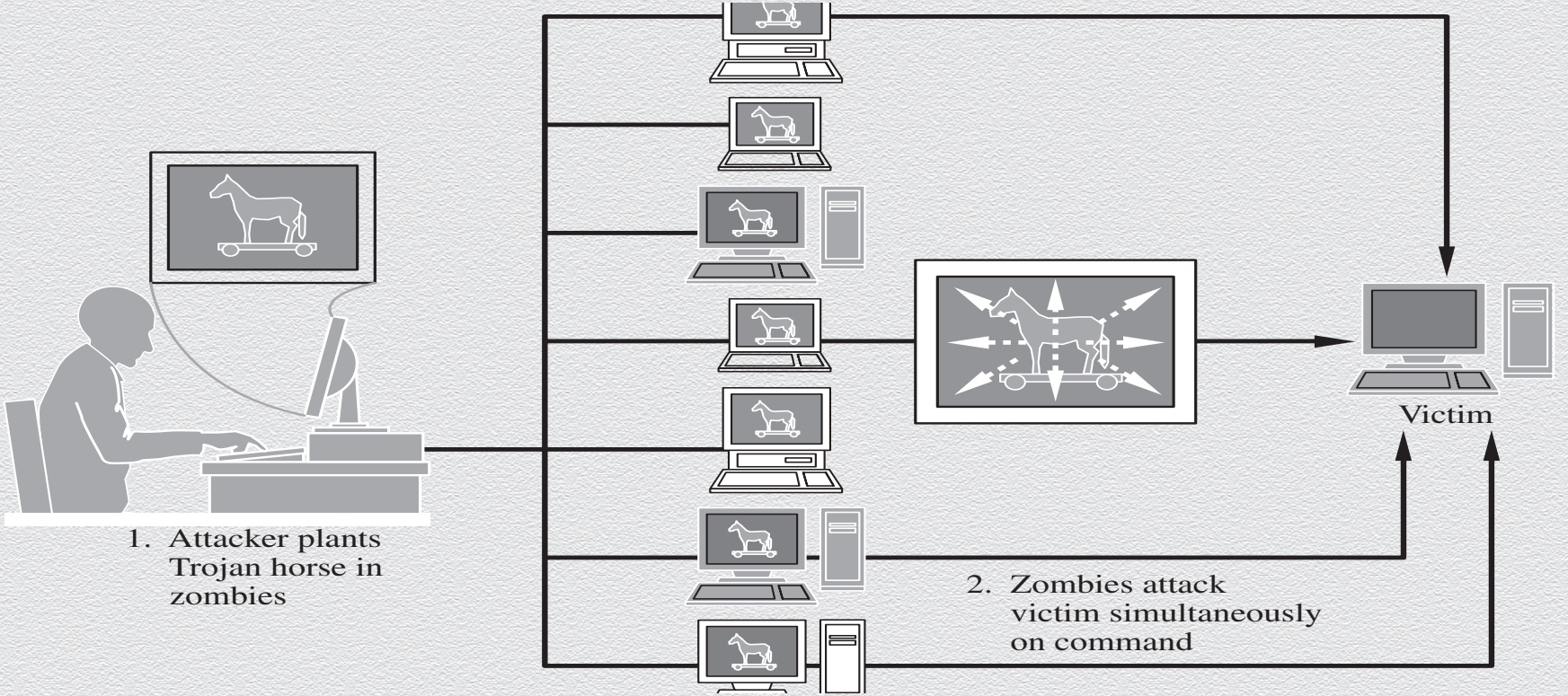
Core idea of attack: Saturate Dyn's primary servers



Attack:

- from a compromised machine ask for domain names that do not exist
- query is forwarded to fewer primary Dyn servers, i.e., defeating benefits of distribution
- ask **A LOT** of such queries to bring down the Dyn DNS service!

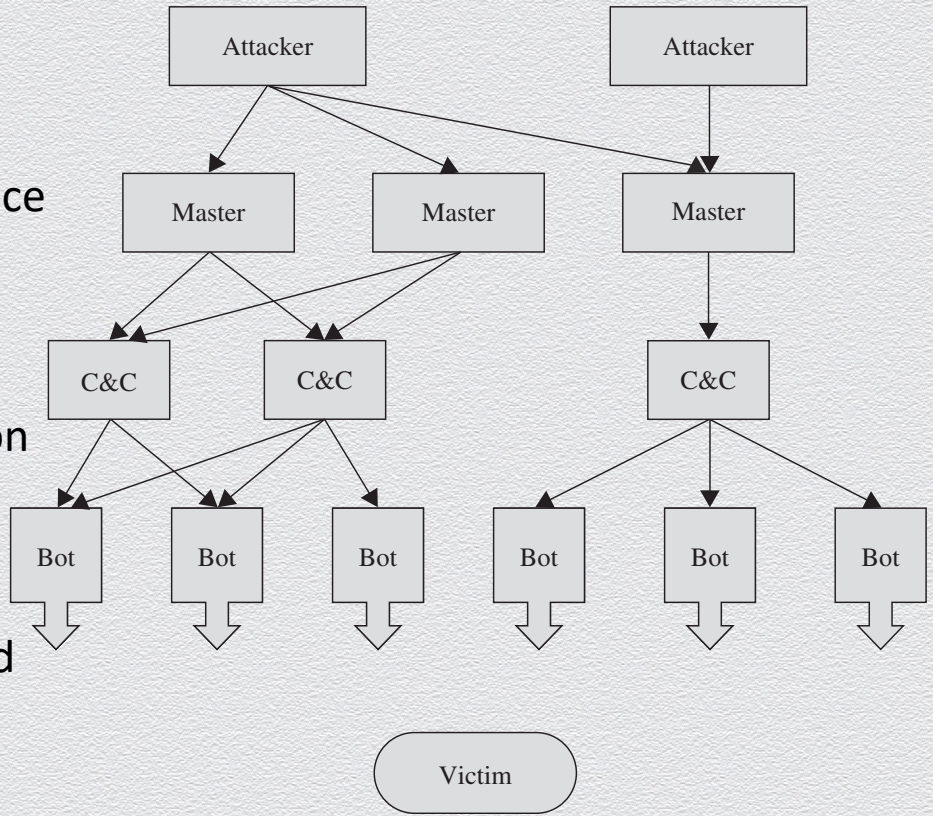
Distributed Denial of Service (DDoS)



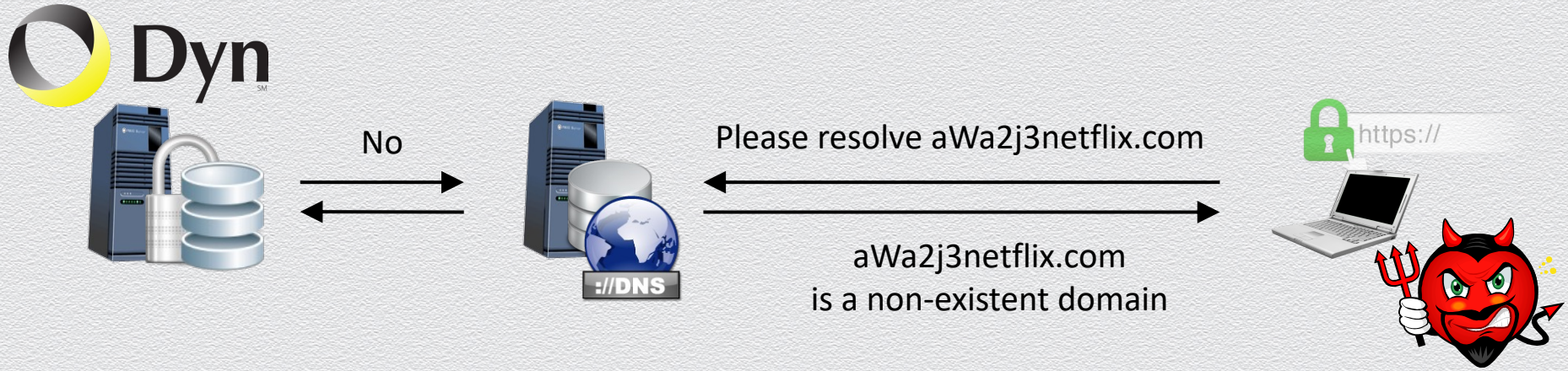
Botnets

Networks of machines running malicious code under remote control

- massive: scale to million of bots
  - comprise main tool for DDoS attacks
- stealth: remain undetected & difficult to trace
  - do little harm to the host machines
    - users won't likely remove malware
  - multiple-level attacker Vs. bots separation
- resilient: have redundant components
  - even if one master or C&C node is taken down, connectivity is maintained

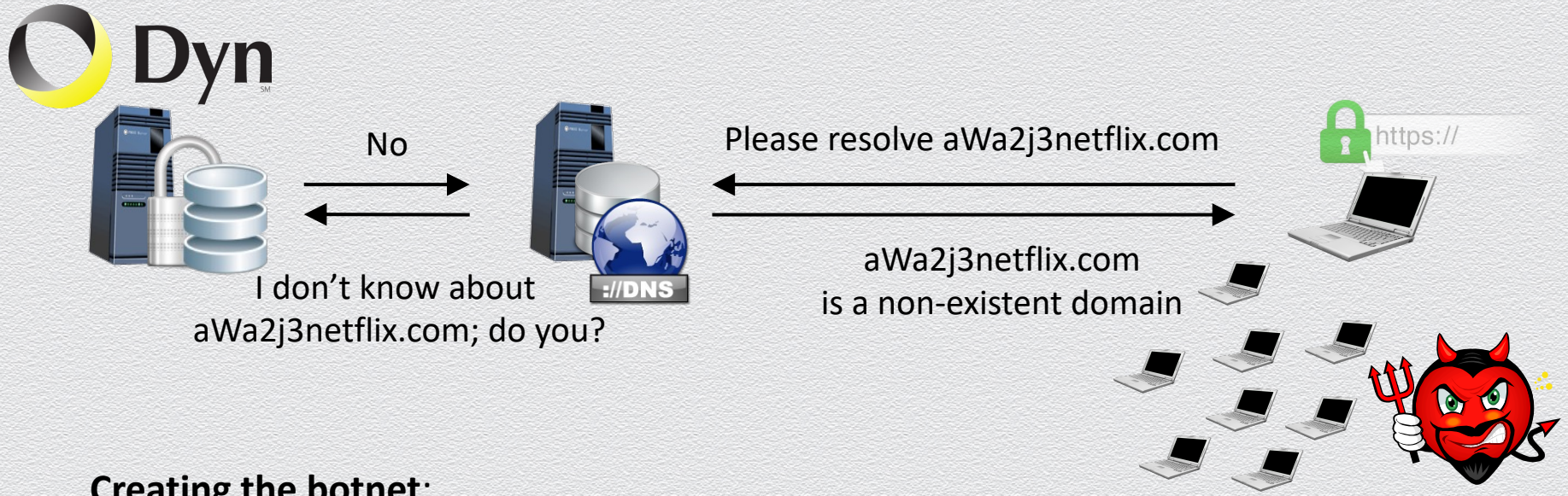


Why botnets are often behind DoS attacks?



- to avoid effective countermeasures and increase "attack" traffic
  - if the high-volume "attack" traffic comes from few devices, they can be filtered out by blocking their connections to the Dyn servers
  - by employing a large botnet of millions of devices the attacker inflicts a larger, more devastating "attack" traffic against the victim Dyn servers

Recruiting an army: Internet of Things (IoT)

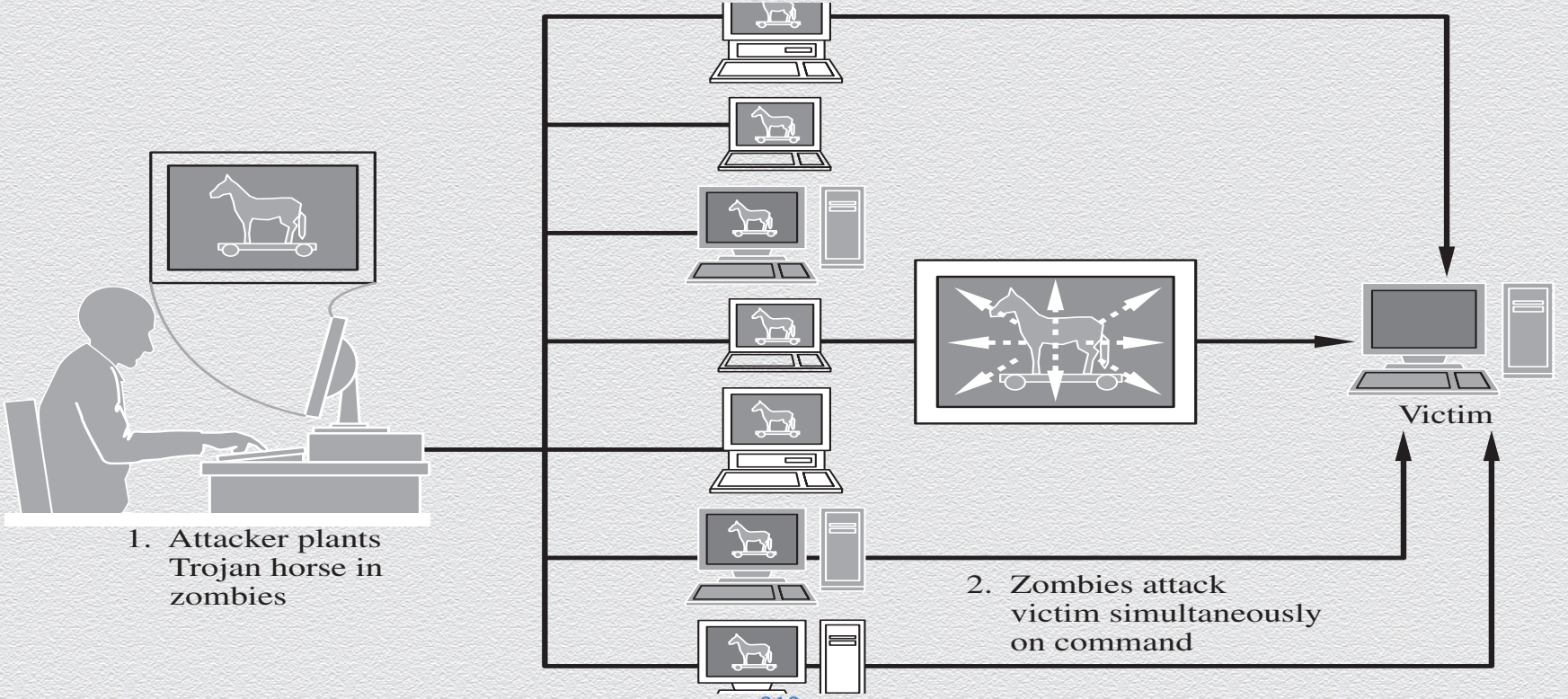


Creating the botnet:

- compromise easy targets: IoT "thin" devices, e.g., printers, cameras, home routers, …
- how? find a vulnerability on these devices...
  - all such devices used an OS with a static, hard-wired, thus known, admin password...!

20.7 DDoS attacks

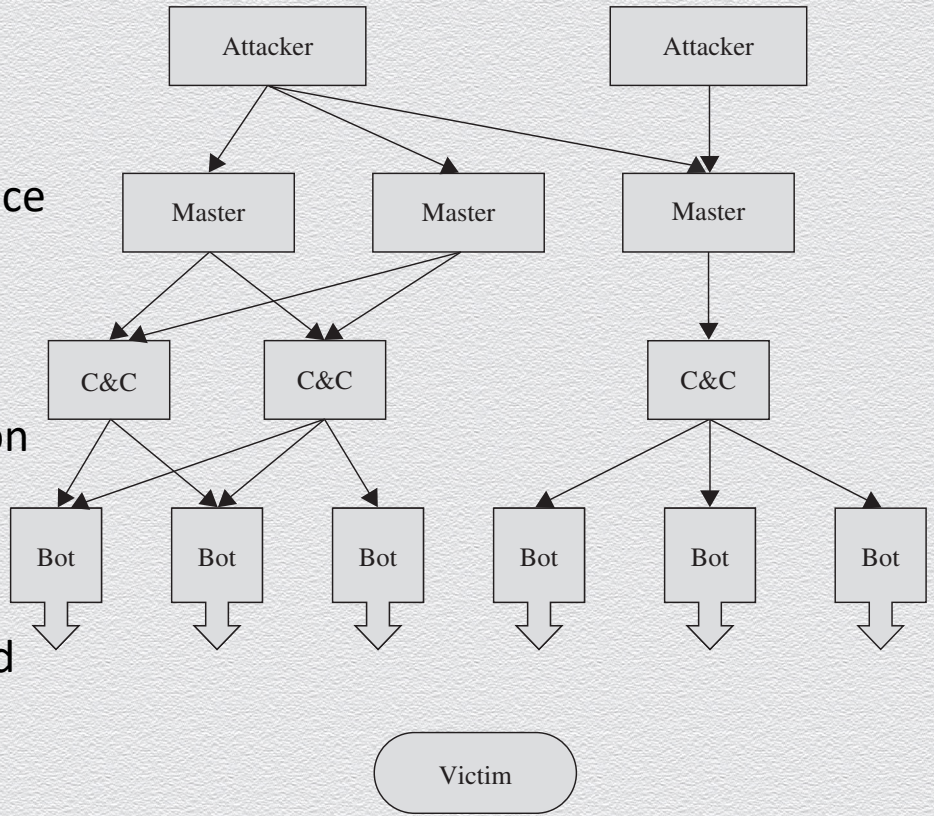
Recall: Distributed Denial of Service (DDoS)



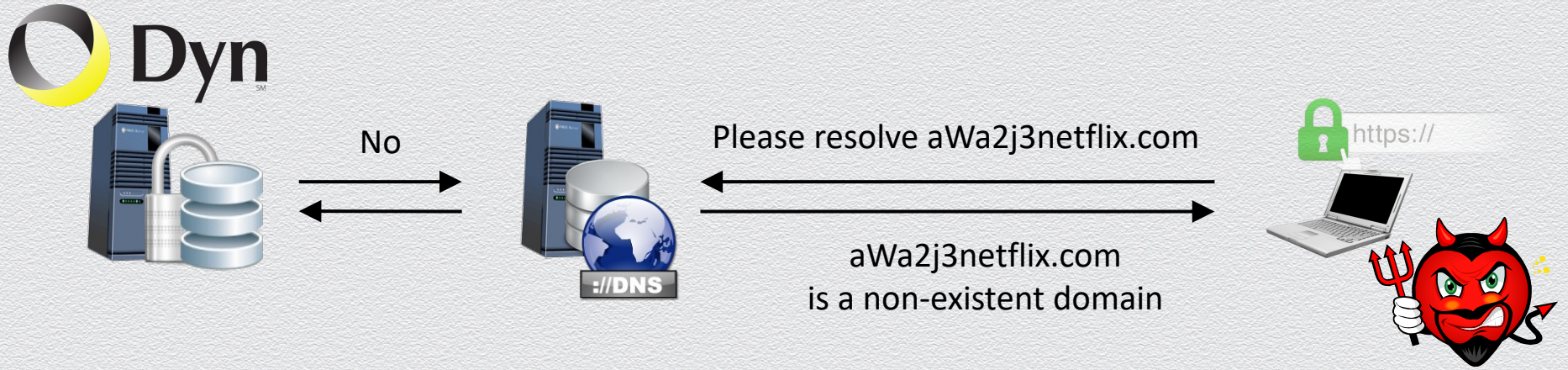
Recall: Botnets

Networks of machines running malicious code under remote control

- massive: scale to million of bots
  - comprise main tool for DDoS attacks
- stealth: remain undetected & difficult to trace
  - do little harm to the host machines
    - users won't likely remove malware
  - multiple-level attacker Vs. bots separation
- resilient: have redundant components
  - even if one master or C&C node is taken down, connectivity is maintained

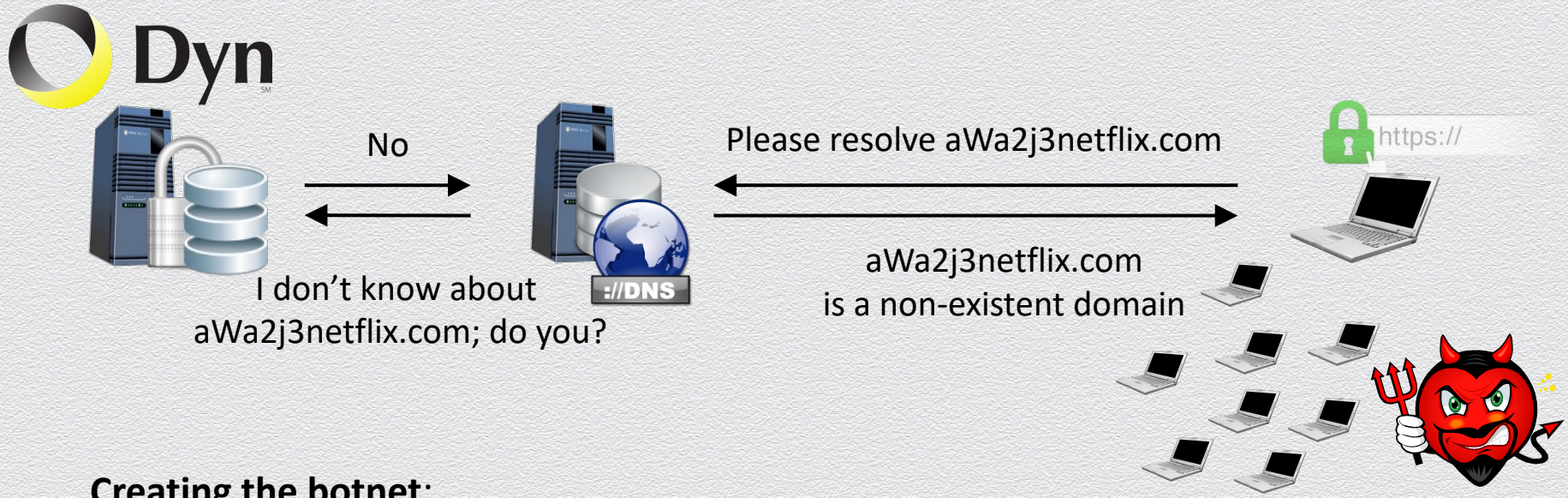


Recall: Why botnets are often behind DoS attacks?



- to avoid effective countermeasures and increase "attack" traffic
  - if the high-volume "attack" traffic comes from few devices, they can be filtered out by blocking their connections to the Dyn servers
  - by employing a large botnet of millions of devices the attacker inflicts a larger, more devastating "attack" traffic against the victim Dyn servers

Recall: Recruiting an army: Internet of Things (IoT)



Creating the botnet:

- compromise easy targets: IoT "thin" devices, e.g., printers, cameras, home routers, …
- how? find a vulnerability on these devices...
  - all such devices used an OS with a static, hard-wired, thus known, admin password...!

The Internet of Things (IoT)

Refers to Internet-connected everyday devices

- comprise a world of so-called smart devices
- examples:
  - smart appliances, such as refrigerators and dishwashers
  - smart home, such as thermostats and alarm systems
  - smart health, such as fitness monitors and insulin pumps
  - smart transportation, such as driverless cars
  - smart entertainment, such as video recorders
- potential downsides
  - loss of privacy
  - loss of control of data
  - potential for subversion
  - mistaken identification
  - uncontrolled access

Smartphones

The control hub of the IoT – important target for malware

- 2013: 143,211 distinct new forms of malware against mobile devices
- 98% targeted Android devices, far in excess of its market share
  - Android: open approach
    - unlike its competitors, does not limit the software users are allowed to install
    - thus, an easier target
  - Apple: locked-down approach
    - in contrast, only allows apps from its app store to be installed on its smartphones
    - all apps go through an approval process, which includes some security review
    - once approved, apps are signed, using a certificate approach

More generally on DoS attacks

DoS attacks are attempts to defeat a system's availability

- volumetric attacks (e.g., flooding)
  - potential weaknesses in the capacity of a computer network

insufficient capacity, overloading

- application-based attacks (e.g., routing malfunction, blocked access)
  - exhaust the resources of an application that services a particular network

blocked access

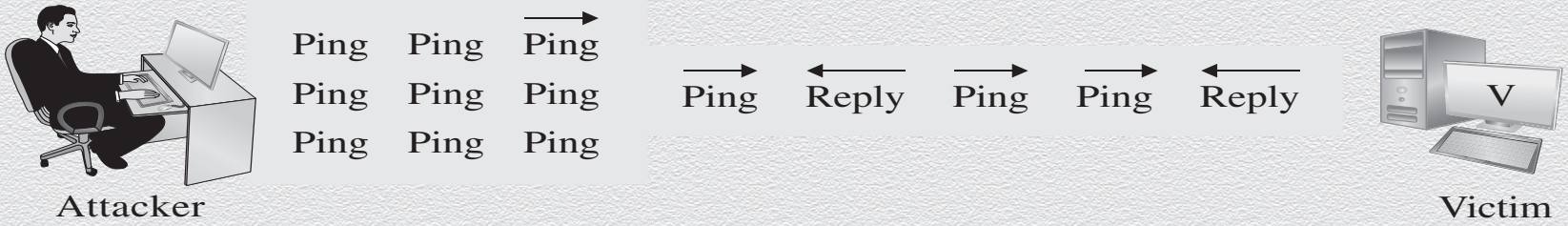
- disabled communications (e.g., access failure)
  - physically disable a communication link or disrupt a single-point of failure
- hardware or software failure (e.g., access failure)
  - failures on machines or programs (without fault-tolerance provisions)

unresponsive asset

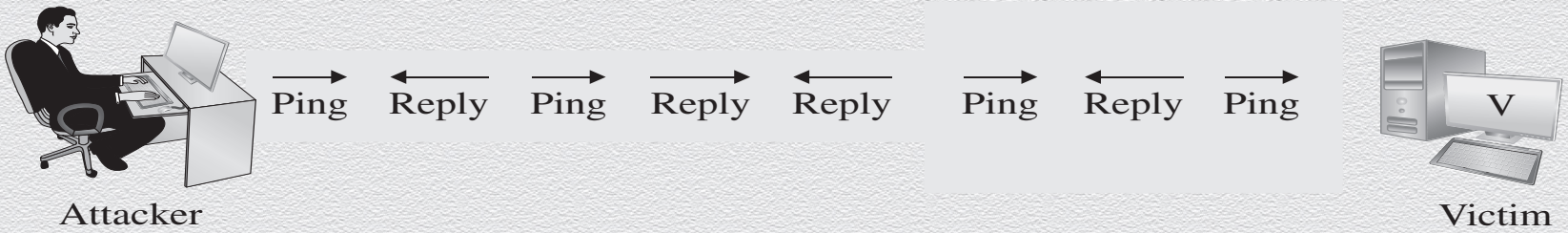
Examples

- Benign errors
  - Beth Israel Hospital system downtime, in 2002, due to mishandling of switches
- Malicious code
  - use vulnerabilities in communication protocols
  - e.g., cause uncontrolled congestion in TCP communications (Vs. UDP)
  - e.g., exploit/misuse Internet Control Message Protocols (ICMP)
    - ping (destination is reachable and functional)
    - echo (the connection between two machines is reliable)
    - destination unreachable (destination address cannot be accessed)
    - source quench (destination is saturated and source should suspend transmissions)

Ping flood (or pink of death)



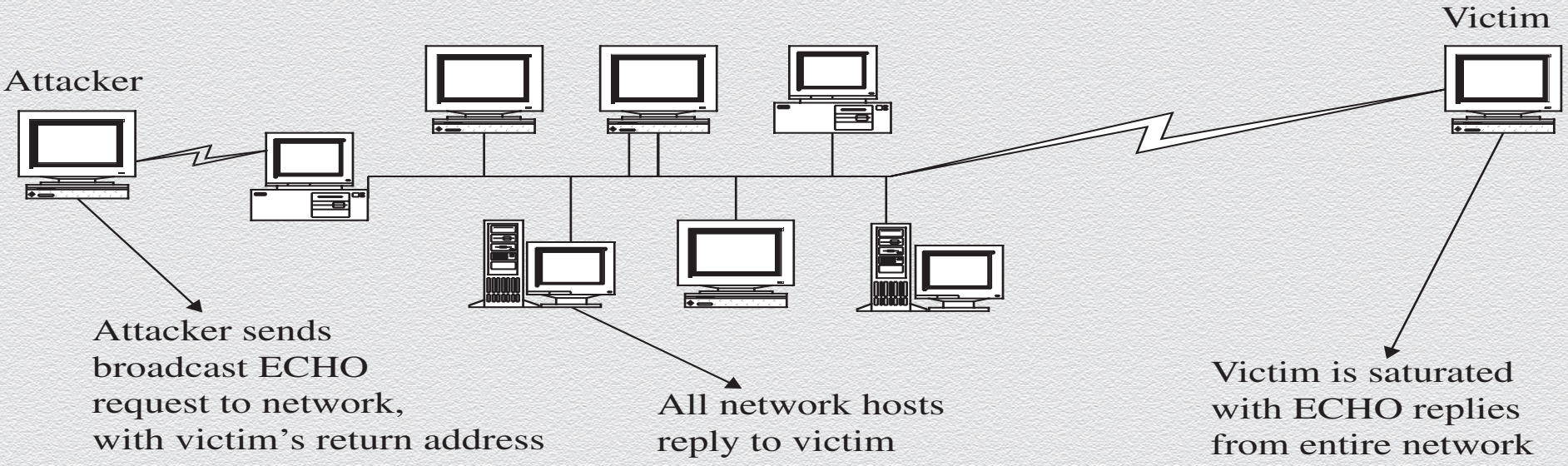
(a) Attacker has greater bandwidth



(b) Victim has greater bandwidth

Smurf Attack

Unwitting victims become accomplices



Echo-Chargen

Victim A

Chargen packet with echo bit on →

← Echoing what you just sent me

Chargen another packet with echo bit on →

← Echoing that again

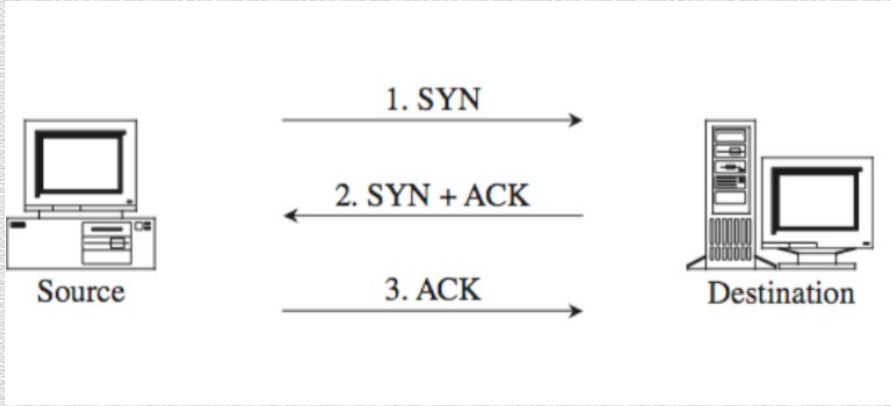
Chargen another packet with echo bit on →

Victim B

SYN flood

Three-way handshake used in TCP to establish a new session

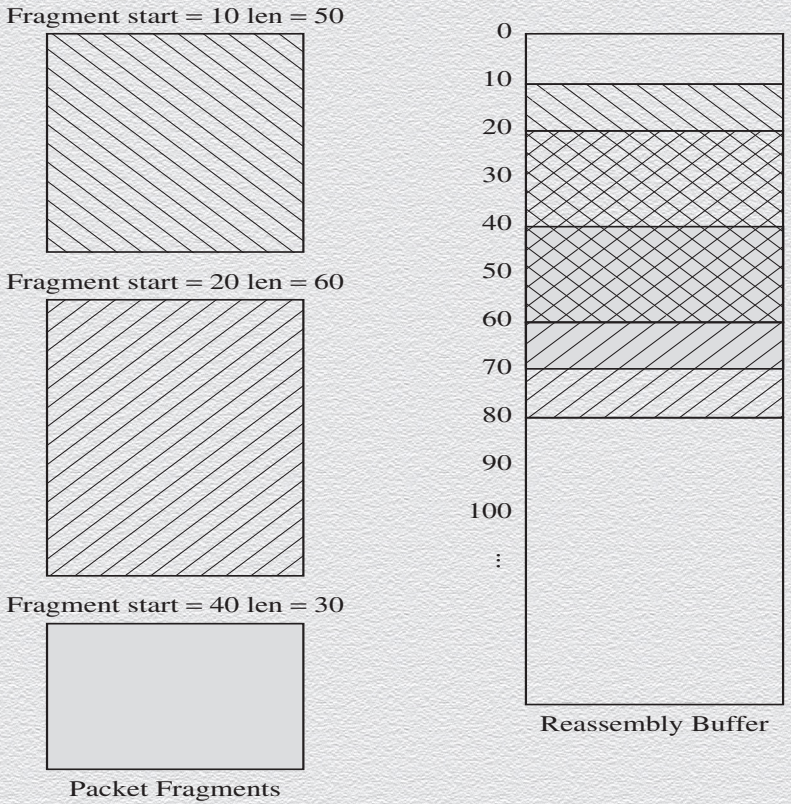
- source & destination exchange control messages to complete session creation
- destination keeps track of incomplete session-creation protocols (SYN-RECV cache)
- attacker spoofs return address of many SYN handshake messages sent to victim
- victim's cache is filled (after ~20) pending incomplete sessions and delays are created



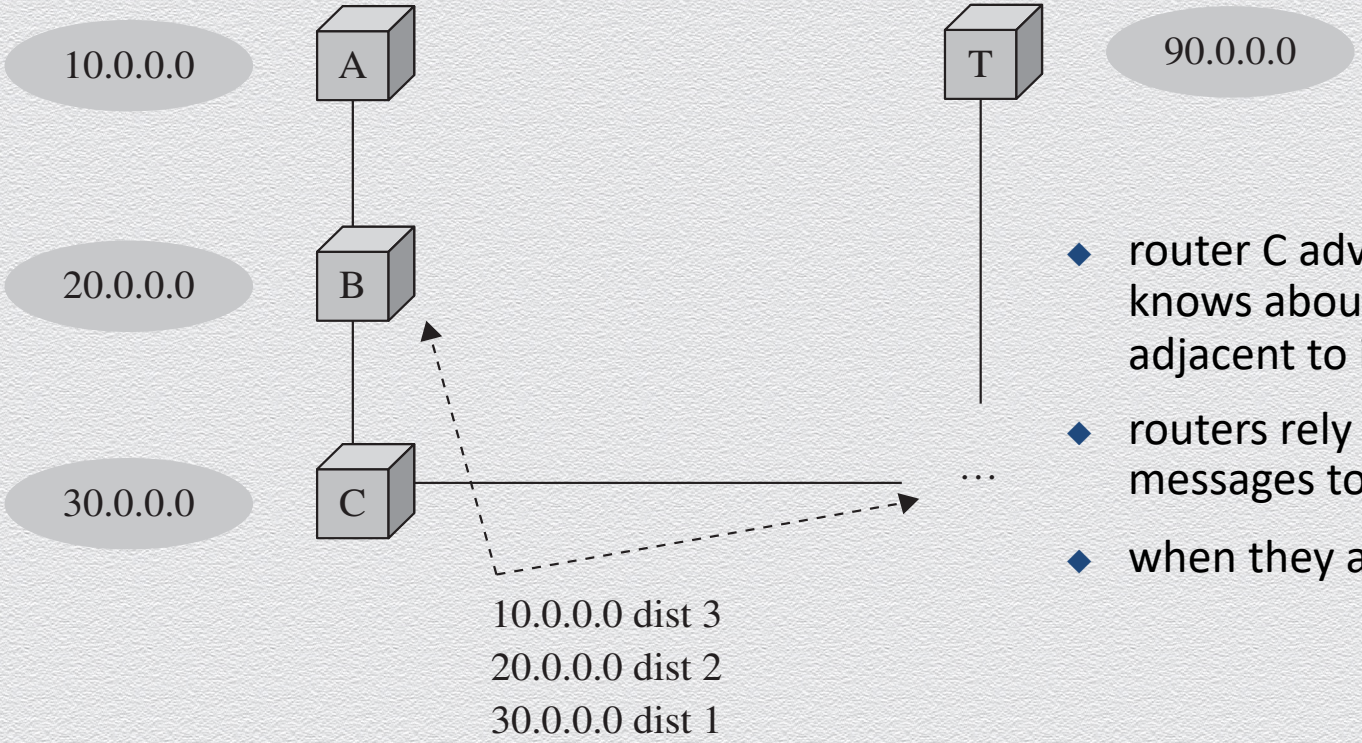
Teardrop Attack

IP datagrams allow to carry variable-length data

- each datagram specifies the length of its data payload and its offset (start-end)
- the attacker can spoof these metadata so that recipient's state remains inconsistent
- sent packets cannot possibly be reassembled, as they conflict instructions
- in extreme cases, this can cause the entire OS to lock up



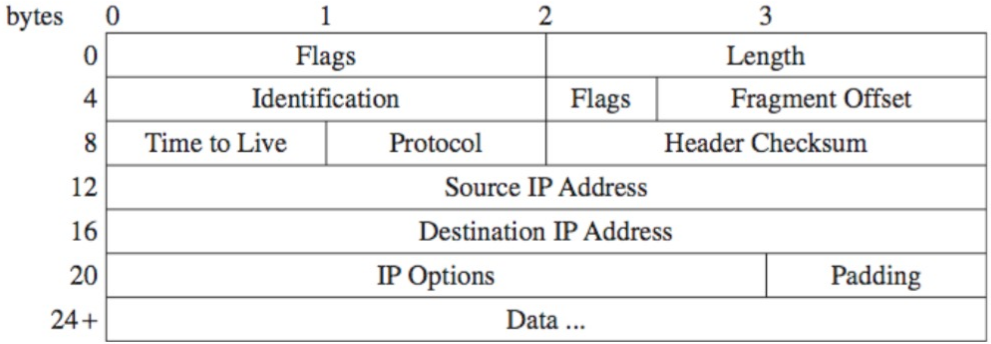
Rerouting routing

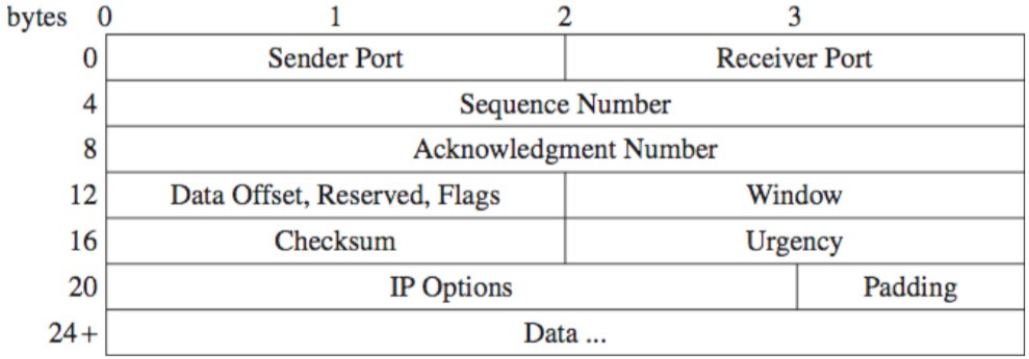


- router C advertises the routes it knows about to the routers adjacent to it
- routers rely on these advertising messages to be accurate
- when they aren't, DoS can ensue

TCP/IP headers

- headers of IP datagram and TCP packet
- IP: right
- TCP: below





Session hijacking

- an attacker is able to synchronize with a receiver while breaking synchronization with the sender and resetting the sender's connection
- the attacker continues the TCP session while the sender thinks the connection just broke off