Network Security - 1

Kehuan Zhang

IERG4130 2022

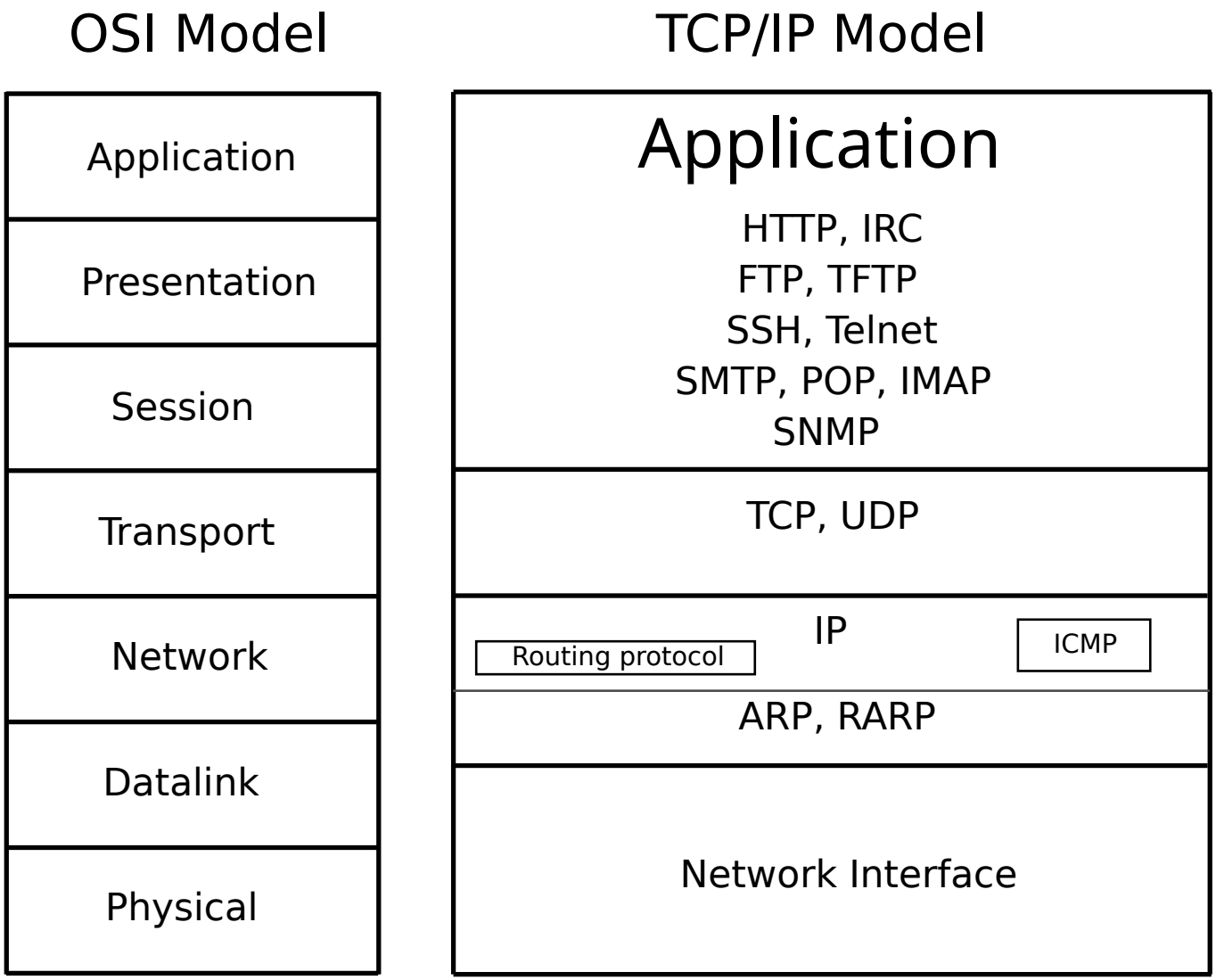
Topics to Be Covered in Network Security

- Network Basics
- Various Network Attacks
  - Spoofing, Sniffing, DoS, Hijacking, Man-in-the-Middle
- Various Network Defenses
  - Firewall, IDS, IPS
- Web Security
- Secure Network Protocols
  - SSL/TLS, IPSec, DNSSEC, etc.
- **Pay attention to CIA and attacking surface, and how they are used in network security**
- Try to use some tools to help you
  - Wireshark: network traffic capture and analyzer `https://www.wireshark.org/download.html`
  - Fiddler: for HTTP/HTTPS `https://www.telerik.com/fiddler`
  - mitmproxy: another HTTP/S proxy `https://mitmproxy.org/`

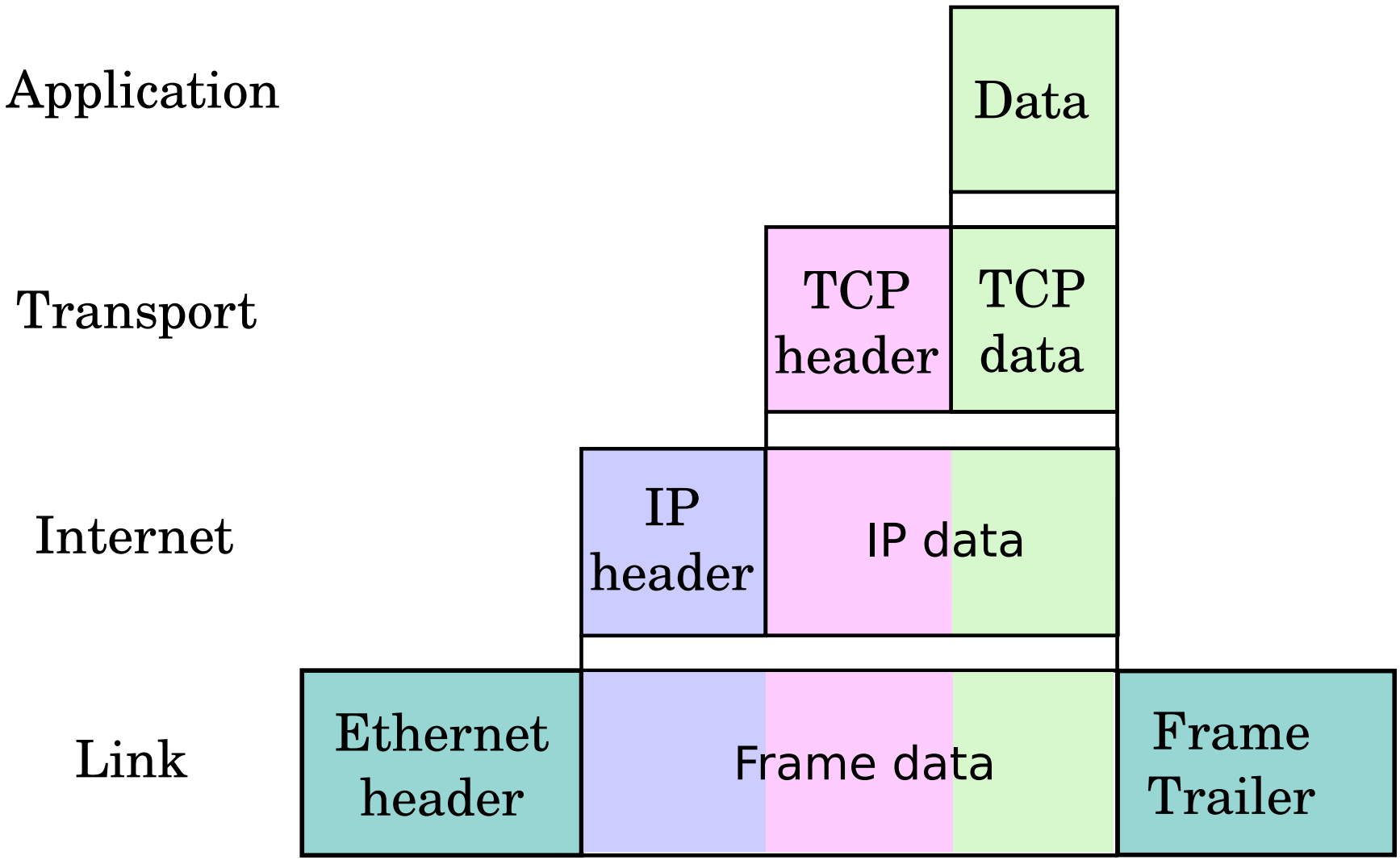
Network Basics

Layered Structure of Networks

- ISO OSI (Open System Interconnection) Model has defined seven layers
- But in pratice, the TCP/IP model is mostly used



Packet Encapsulation: Headers and Payloads



Different Equipment Types (at Different Layers)

- Physical layer: Hub
- Data link layer: Switch
- Network layer: Router, Layer-3 Switch, Gateway
- Transport layer: SOCKS 5 Proxy
- Application layer: HTTP Proxy

Concept of Address in Network

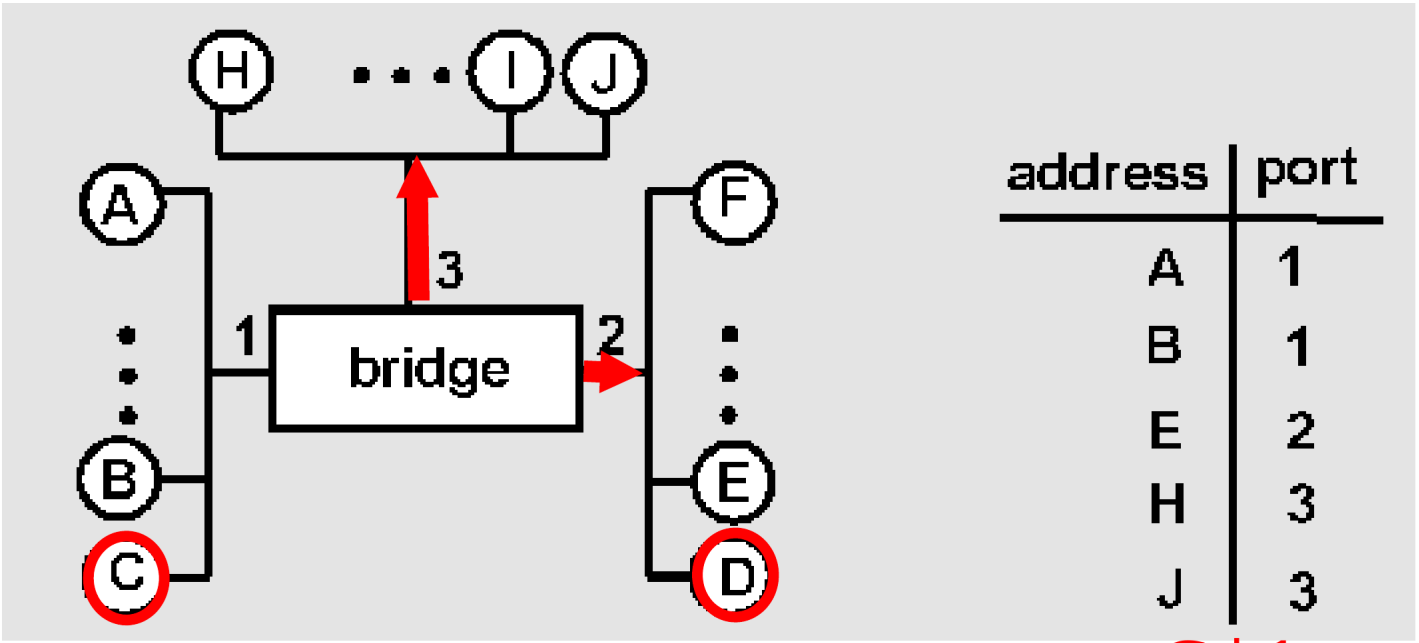
- Addresses are needed to deliver packets correctly
- There are some sorts of addresses at almost every layer
  - Physical layer: interface ID
  - Data Link Layer: MAC (Media Access Control) address
    - ⋆ On Ethernet: 48-bit (demo)
  - Network Layer: IP address
  - Transport Layer: Port Number
  - Application Layer: URL

Translations among Different Addresses

- Physical Layer → Data Link Layer
  - Lookup table (in Switches)
- Data Link Layer → Network Layer
  - ARP (Address Resoluiton Protocol)
- Network Layer → Application Layer
  - DNS (Domain Name Service)
- Transport layer address is a little bit special
  - It is specified in network packets (together with IP address)
  - Network protocol stack (in OS) will forwrad the packet to the corresponding application process who is listening on that port

Switch vs. Hub

- In Hub: packets will be broadcasted to all ports
- Switch: forwarding packets based on **lookup table**
  - Every entry in the lookup table will map MAC to Interface
  - Most Switch will learn and build up such a table dynamically

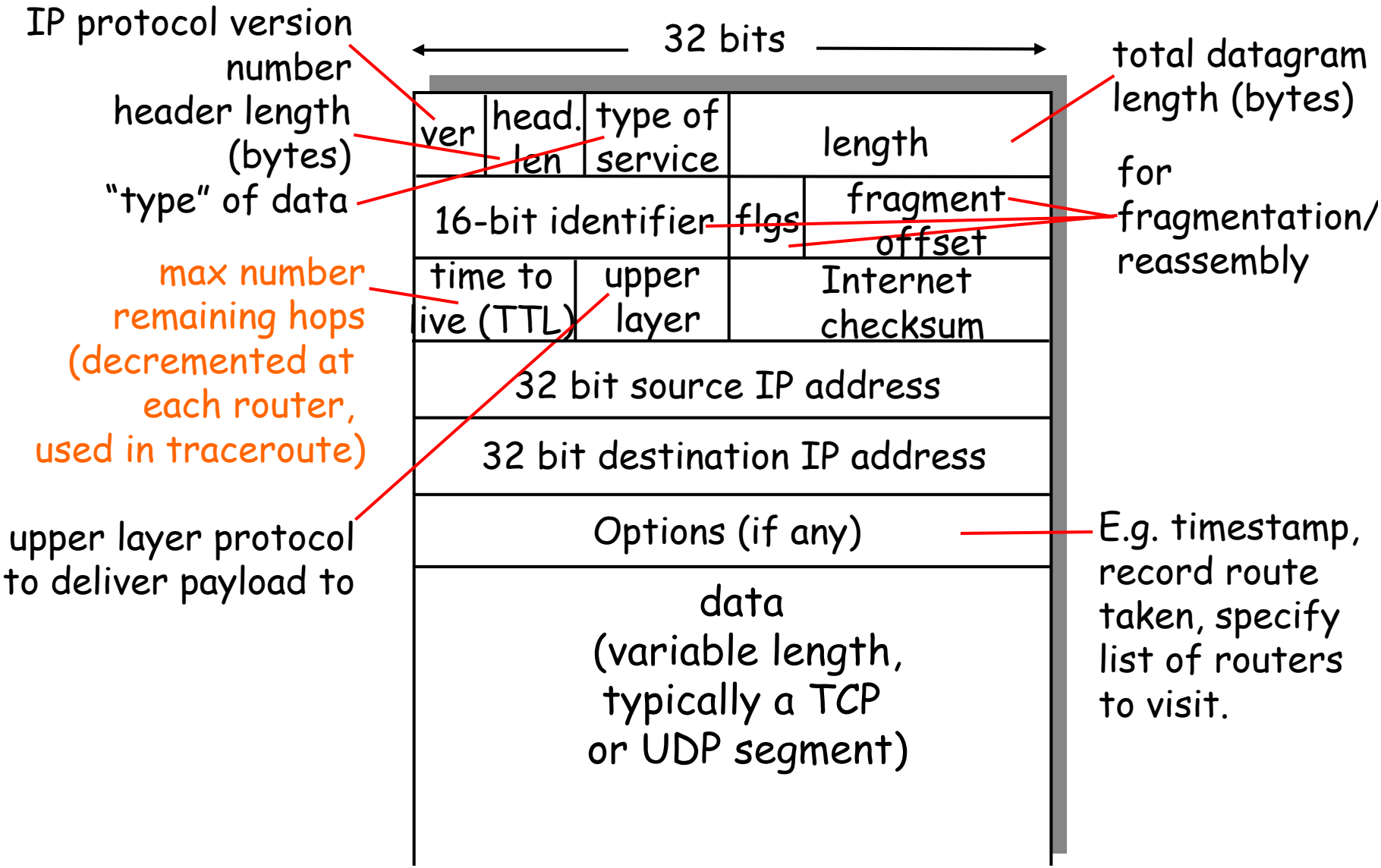


address	port
A	1
B	1
E	2
H	3
J	3
C	1

How Does ARP Work?

- Each host will maintain and ARP table (also called ARP cache)
  - Every entry contains:
    - ★ IP address, MAC address, TTL
    - ★ TTL: Time To Live
- ARP table will be built dynamically during communication process
  - If target IP address is not in table, then:
    - ★ Build and broadcast an ARP request packet ("who has x.x.x.x?")
    - ★ The ARP request will also contain (src_IP, src_MAC) information
  - When received the broadcasted ARP request, every node will:
    - ★ Update its own ARP table suing (src_IP, src_MAC) info
    - ★ Check if target IP address belongs to it
    - ★ If not, then discard the packet
    - ★ If yes, then build and send back an ARP reply packet by filling in its MAC address
    - ★ The ARP reply packet will **NOT** be broadcasted. Why?

IP Header


IP protocol version number
header length (bytes)
"type" of data
max number remaining hops (decremented at each router, used in traceroute)
upper layer protocol to deliver payload to
32 bits
ver
head. len
type of service
length
16-bit identifier
flgs
fragment offset
time to live (TTL)
upper layer
Internet checksum
32 bit source IP address
32 bit destination IP address
Options (if any)
data (variable length, typically a TCP or UDP segment)
total datagram length (bytes)
for fragmentation/ reassembly
E.g. timestamp, record route taken, specify list of routers to visit.

IP Address and Subnet

- 32-bit for IPv4, normally written as dotted decimal format of four bytes
  - Each byte has the range of $0 \cdots 255$, e.g., 127.0.0.1
- Subnets
  - The 32 bits can be divided into **network** and **host** using a **subnet mask**
    - ★ subnet mask: 255.255.0.0, 255.255.255.0
    - ★ 137.189.97.1 & 255.255.0.0 $\rightarrow$ 137.189.0.0 (which is subnet address)
  - Four categories: Class A/B/C, and CIDR (Classless Inter-Domain Routing)
    - ★ Class A/B/C have fixed length of subnet mask at 8/16/24-bit
    - ★ CIDR is more flexible as subnet mask is not fixed
    - ★ E.g., with CIDR, the subnet address can also be written as 136.192.0.0/18
- IPv6: 128-bit $\rightarrow$ assign an address for every sand on earth
  - More security features by design
  - Out of the scope of this course

IP address Types

- Unicast: unique to a certain host or network node
- Broadcasting: packet will be sent to all possible destinations
  - Typical broadcasting address is 255.255.255.255
  - But all-one host address is also broadcasting address
    - ★ E.g., in subnet 192.168.0.0/24, 192.168.0.255 is broadcasting address
- Multicasting: packet will be delivered to a group of interested receivers
  - In IPv4, addresses in 224.0.0.0 - 239.255.255.255 is for multicasting
- Private address
  - Three ranges: 192.168.0.0/16, 10.0.0.0/8, 172.16.0.0/12
  - Routers can handle and forward packets within local network
  - Internet routers will discard packets with such addresses
- Link-local address: only valid on point-to-point or local connection
  - Routers will **NOT** forward packets with these addresses
  - E.g., 169.254.0.0./16
- Loopback address: packets will be send back to the host itself
  - 127.0.0.0/8 $\rightarrow$ 127.0.0.1 is most common

IP Address Assignment and Translation

- Assignment
  - Static assignment vs. dynamic assignment
  - Simple dynamic assignment: BOOTP (Bootstrap Protocol)
    - ★ Simple Mapping from MAC address to IP address
  - Dynamic: DHCP (Dynamic Host Configuration Protocol)
    - ★ More flexible and controls: e.g., valid time period
- NAT (Network Address Translation)
  - Due to the exhaustion of IPv4 address
  - Mainly used to let multiple private addresses to share a single (public) IPv4 address
  - Based on a translation table to do the mapping
    - ★ Every table entry contains: private IP & associated port, public IP and associated port
    - ★ For every outgoing packet, will lookup that table to get public IP and port
    - ★ If not found, then a new entry will be created
    - ★ For incoming packet , will get private IP and port from the lookup table
    - ★ Modify the IP headers according using the private (or public) IP & port
    - ★ Question: while will happen if no entry was found for incoming packet?

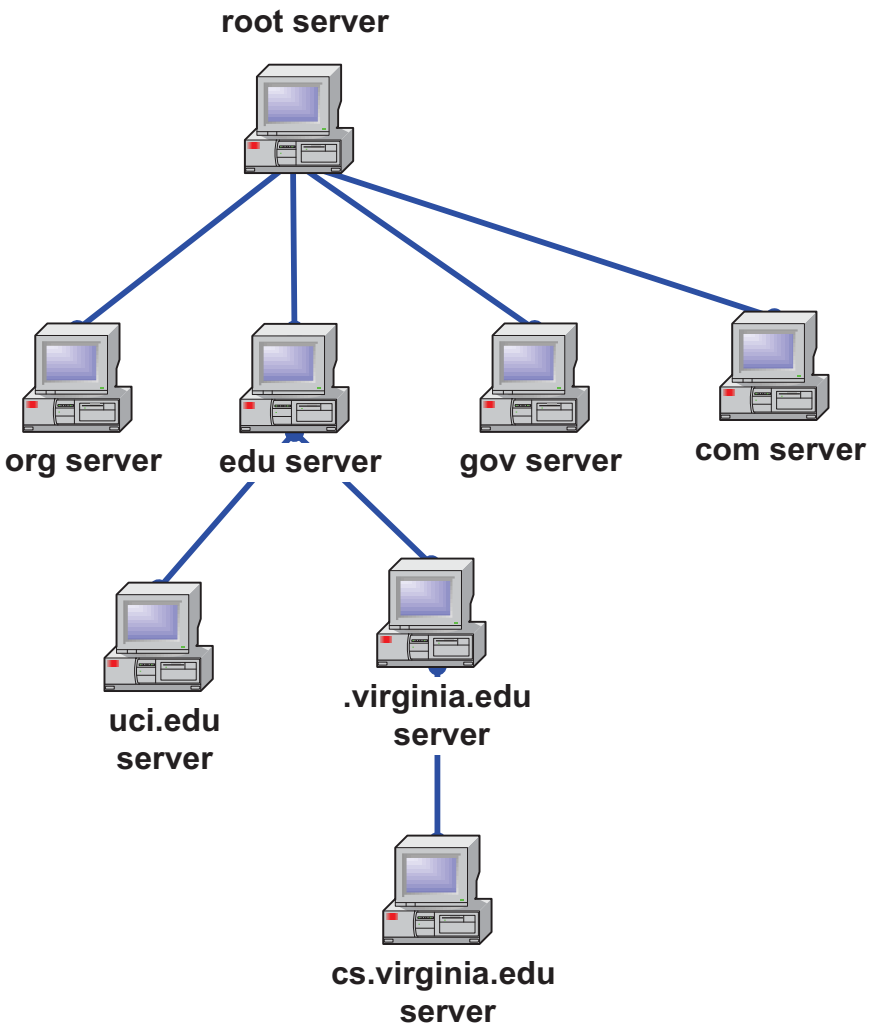
How Does Router Work?

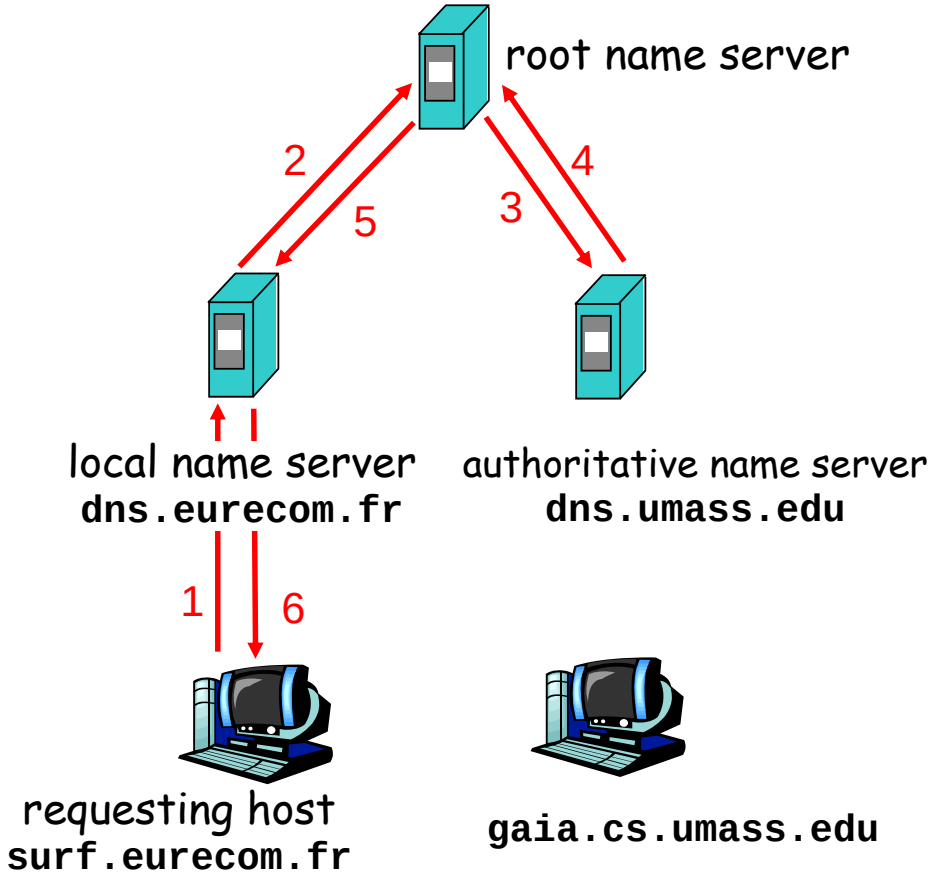
- Router will:
  - Running routing protocol to maintain a routing table
  - For an incoming packet, lookup the routing table to forward it to destination interface
    - ⋆ Using a technique call "TCAM" (Ternary Content Addressable Memory): IP $\rightarrow$ mem address
    - ⋆ Modify the IP header accordingly, e.g., TTL, phyiscal and data link layer headers
  - Queue and Traffic Management, e.g, QoS (Quality of Service), congestion control

How Does DNS Work?

- Purpose: a protocol to translate name to IP address (dynamically)
  - Static approach: using host.txt file (e.g., /etc/hosts on Linux)
- Concept of Domain Namespace
  - Root: represented with a "." (but normally omitted)
  - Top-level domains: three types
    - ★ Contry Code Top Level Domains (ccTLD): 2-char, e.g., hk, jp, cn
    - ★ Generic Top Level Domains (gTLD): 3-char indicating functions, e.g, edu, gov, com
    - ★ Reserved (skipped)
  - Hostname or subdomain name (recursive): e.g., gTLD, organization name, or specific machine name
- Domain Name Servers
  - Each server will maintain a table mapping from domain name to IP address
  - Such servers are organized into a hierarchy structure (based on the namespace)

Domain Name Resolution Process



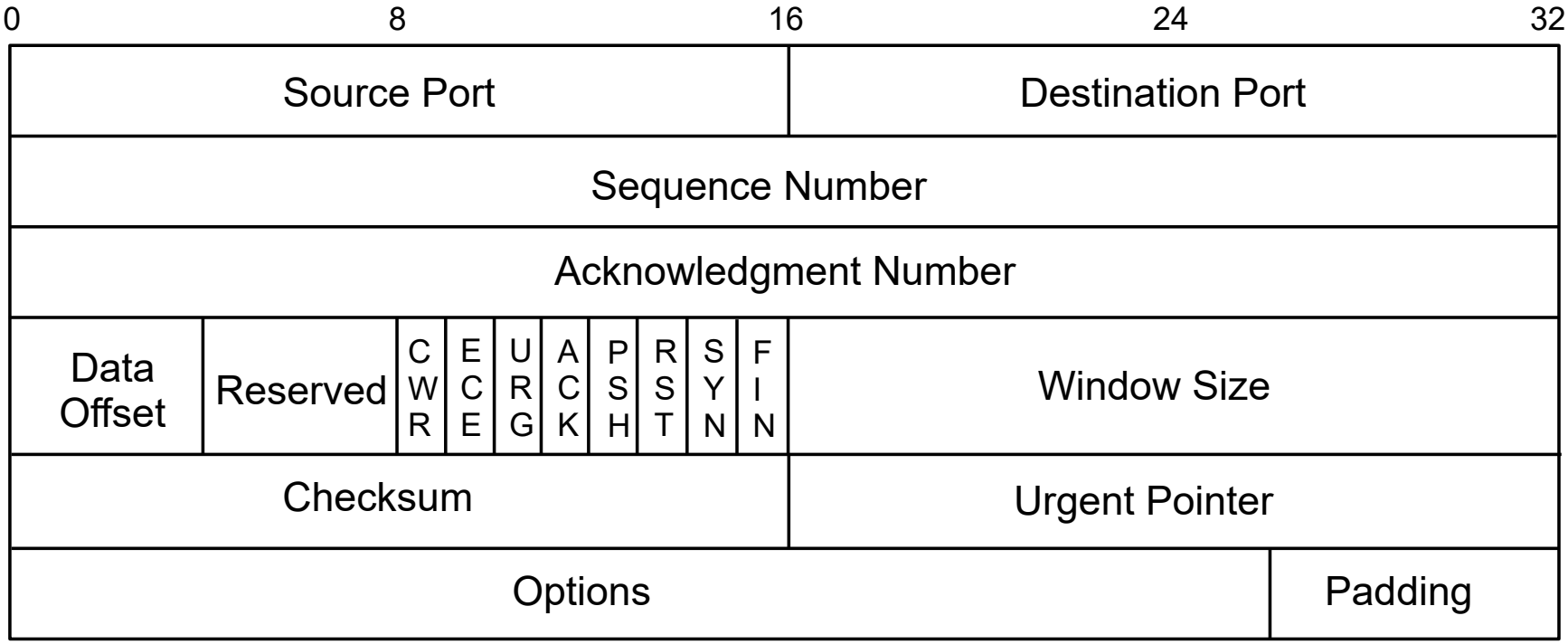


- To reduce DNS traffic, responses can be cached for a period of time locally and/or by name server

Use IP Address to Identify Network Problems

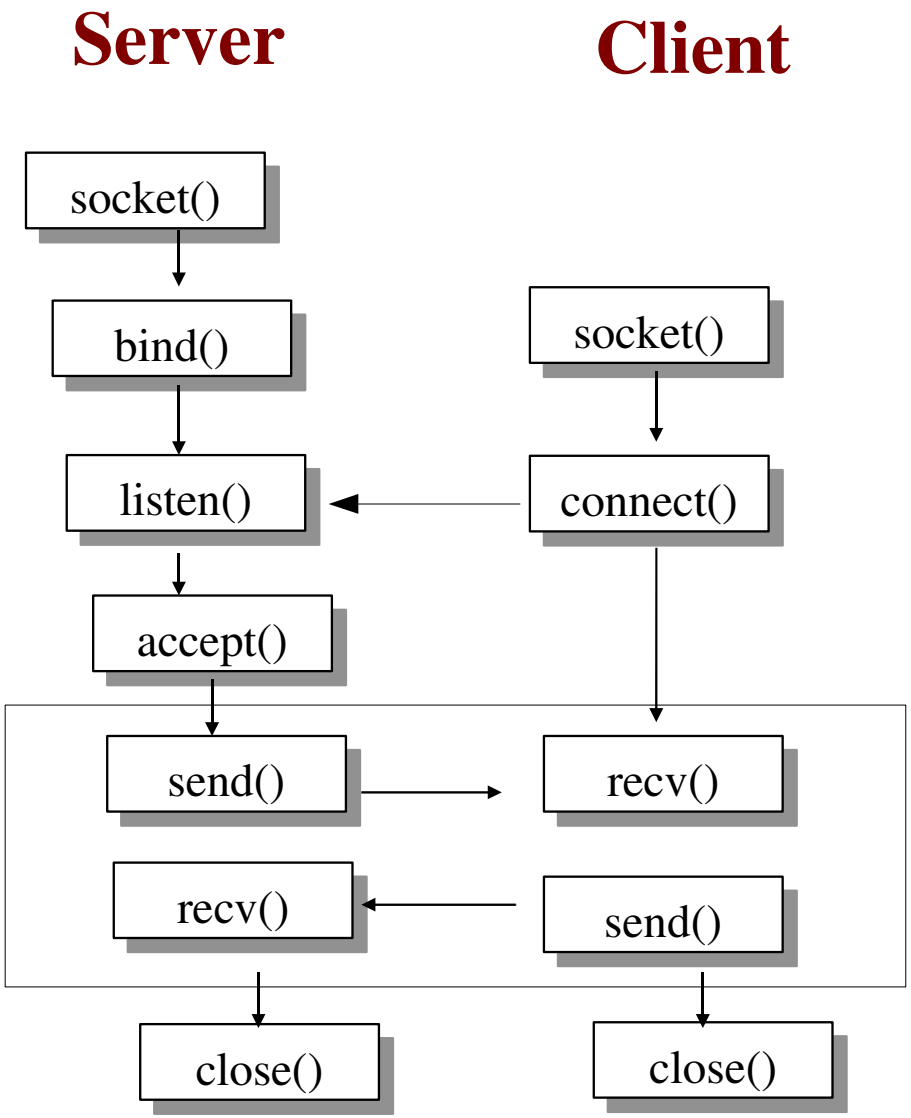
- If you could not visit your favorate web site, say Google.com, how can you know what is wrong?
  - Strategy: checking the protocol stack, either bottom-up or top-down
- Bottom-up approach:
  1. Check physical interface: (use command *ifconfig* or *ipconfig*)
     - ★ Is interface available, is it up, has it get an IP address, what type of IP address, etc.
     - ★ e.g., link-local address means it has not been configurated yet.
  2. Check local reachability, (use command *ping*)
     - ★ Can we get replies from gateway, router, or other hosts inside the same subnet?
  3. Check the domain name resolving (a common problem)
  4. Check the routing path, (use command *traceroute* or *tracert*)
     - ★ Trace the routing of your packet and see if it can reach destination host
  5. You can also use *ping* to check the reachability of destination host
     - ★ Pay attention to the *RTT (round-trip-time)* (to detect possible DoS)
  6. If all are fine, then network is OK, so it may be other problems
     - ★ E.g., used HTTP or SOCKS proxy in browser, or blocked by firewall, etc.

TCP Service Model and TCP Header



- TCP provide a point-to-point, connection-oriented byte-stream service
  - Connection-oriented: hand-shaking and disconnection (SYN, ACK, FIN)
  - Has congestion control (sliding window and slow start on window size)
  - Incorrect sequence number and ack number will be discarded
    - ★ But packets with correct numbers will be accepted!!
    - ★ Sometime it could be problemtic (e.g., hijacking attack)

Socket Programming (Connection Oriented)



Network Attacks

Typical Network Attacks

- Spoofing
- Sniffing
- Denial of Service
- TCP Session Hijacking
- Man-in-the-Middle

Attack 1: Spoofing

- Spoofing means using a fake address to impersonate another host/node
- Spoofing can happen at multiple layers
  - Physical layer: normally not possible, except can access the hardware
  - Link layer: Spoofing the MAC address $\rightarrow$ No problem
    - ★ Via ARP protocol: by generating a fake ARP reply (draw a figure)
    - ★ By changing OS configuraion directly: e.g, MAC-based authentication
    - ★ What can attackers do with MAC spoofing?
    - ★ Redirect traffic to attacker's node, MITM attack, sniffing, etc.
  - Network layer: e.g., build a packet with fake source IP address
    - ★ Consequences? could bypass protections like Firewall
    - ★ Can be part of a Denial-of-Service attack (more details later)
    - ★ Limitation: could not receive replying IP packet. Why?
    - ★ **Counter Measure**: with source address validation
  - Transport layer: also possible (refer to TCP session Hijacking)
  - Domain Name Spoofing: by generating a fake reply of DNS request
    - ★ Binding the target domain to a malicious IP address
    - ★ Consequences? MITM, sniffing, DoS, etc.

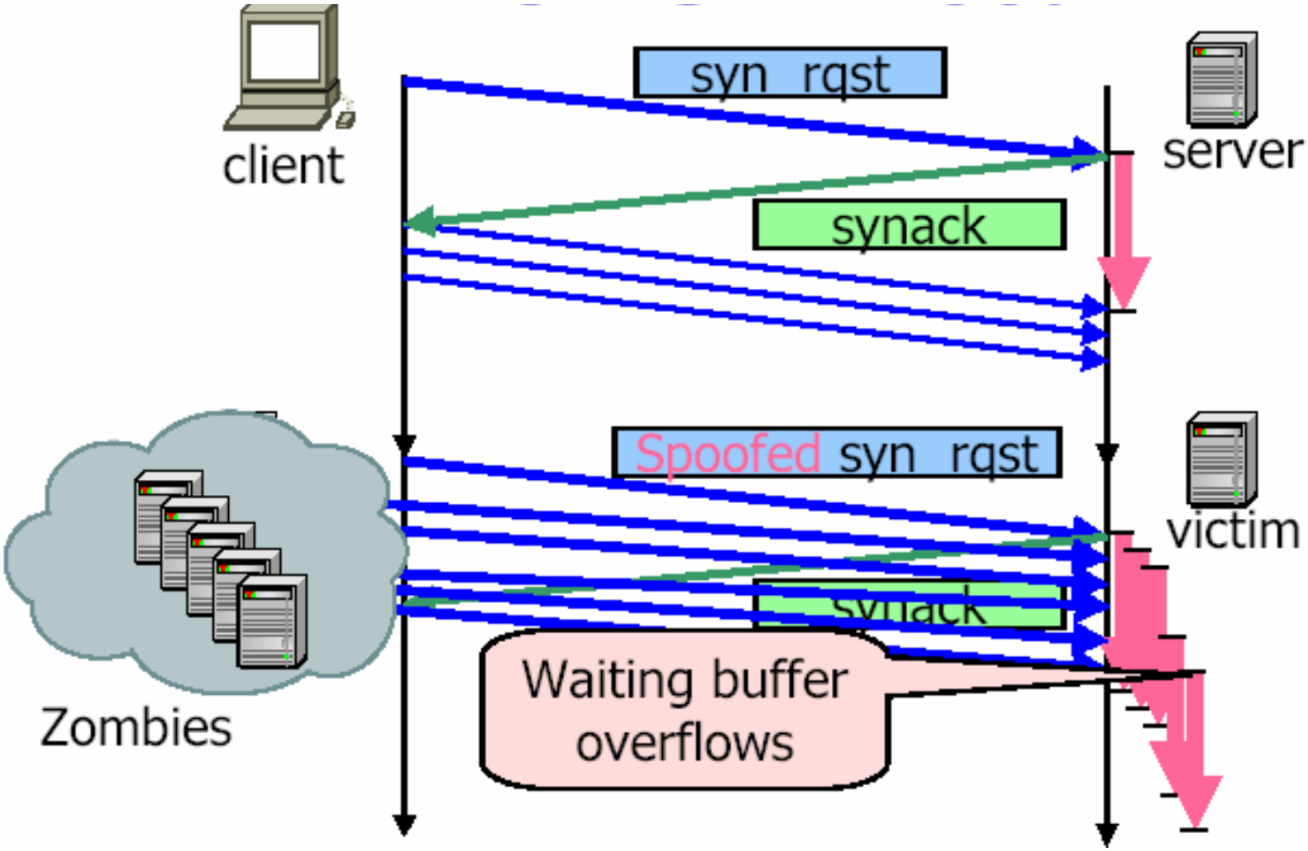
Attack 2: Sniffing

- Sniffing means getting packets stealthily without authorization
- Sniffing can be done at multiple layers
  - Physical: wire-tapping (not only copper wire, but also fiber)
    - ★ Wireless is easier to be attacked
  - Link layer:
    - ★ Hub is vulnerable by nature (due to broadcasting)
    - ★ Sniffing is also possible on Switch? (by MAC flooding) (page 9)
  - Network layer
    - ★ Port mirroring at router (can be selectively with filtering option)
    - ★ Rogue Access Point (for WiFi)
- Defense? $\rightarrow$ traffic encryption

Attack 3: Denial of Service (DoS)

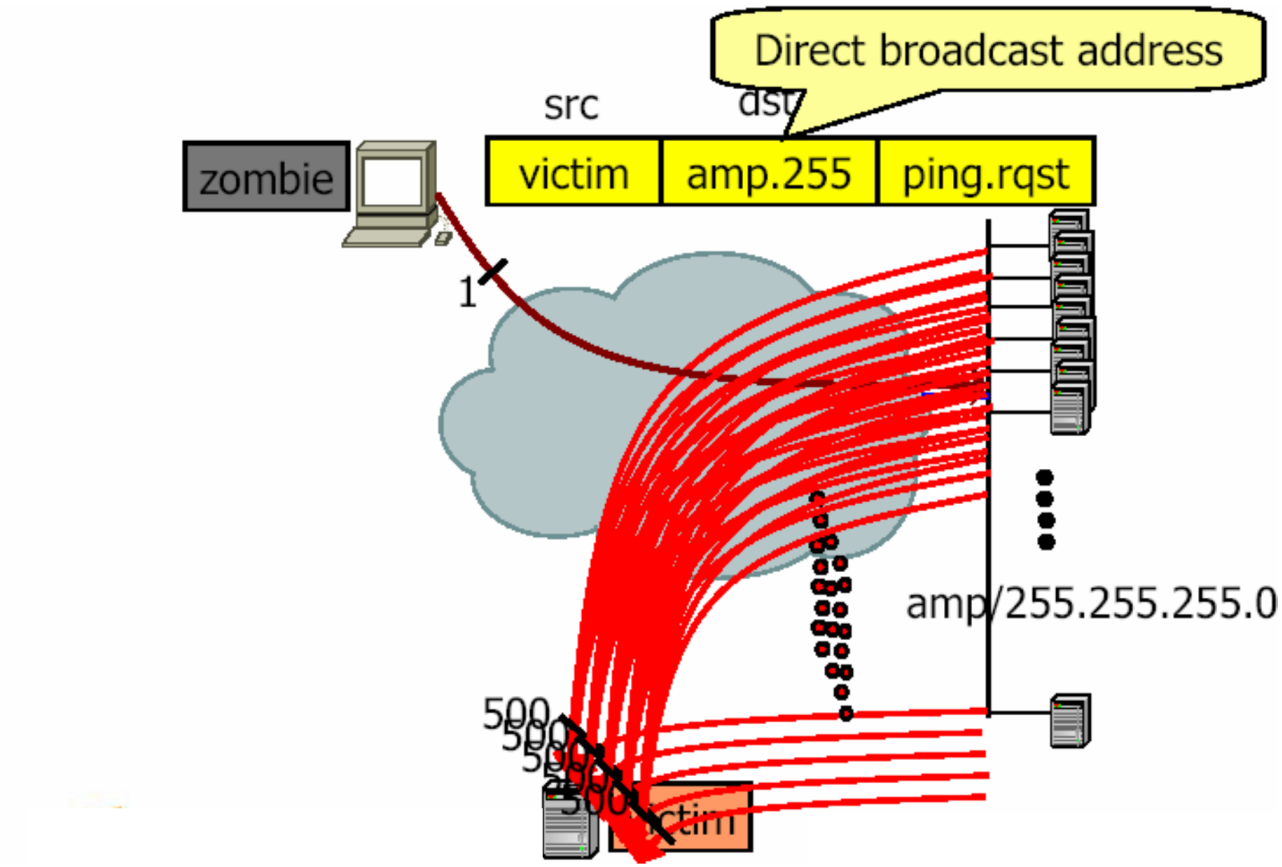
- Overloading and exhaust the available resources on certain node
  - Can happen at link layer (exceeding the bandwidth)
  - Can be the router (exceeding to forwarding capacity)
    - ★ Packets would be lost from queues
    - ★ Activate congestion control algorithm
  - Can be the target Server
    - ★ E.g., exaust CPU, kernel memory object, or other resources
  - Can be a specific application
    - ★ E.g., exaust heap or stack space, using JavaScript to occupy CPU time
- DoS at TCP level
  - TCP SYN Flood
  - Smurf Amplification
  - Reflectors
  - Distributed Denial of Service (DDoS)

TCP SYN Flood



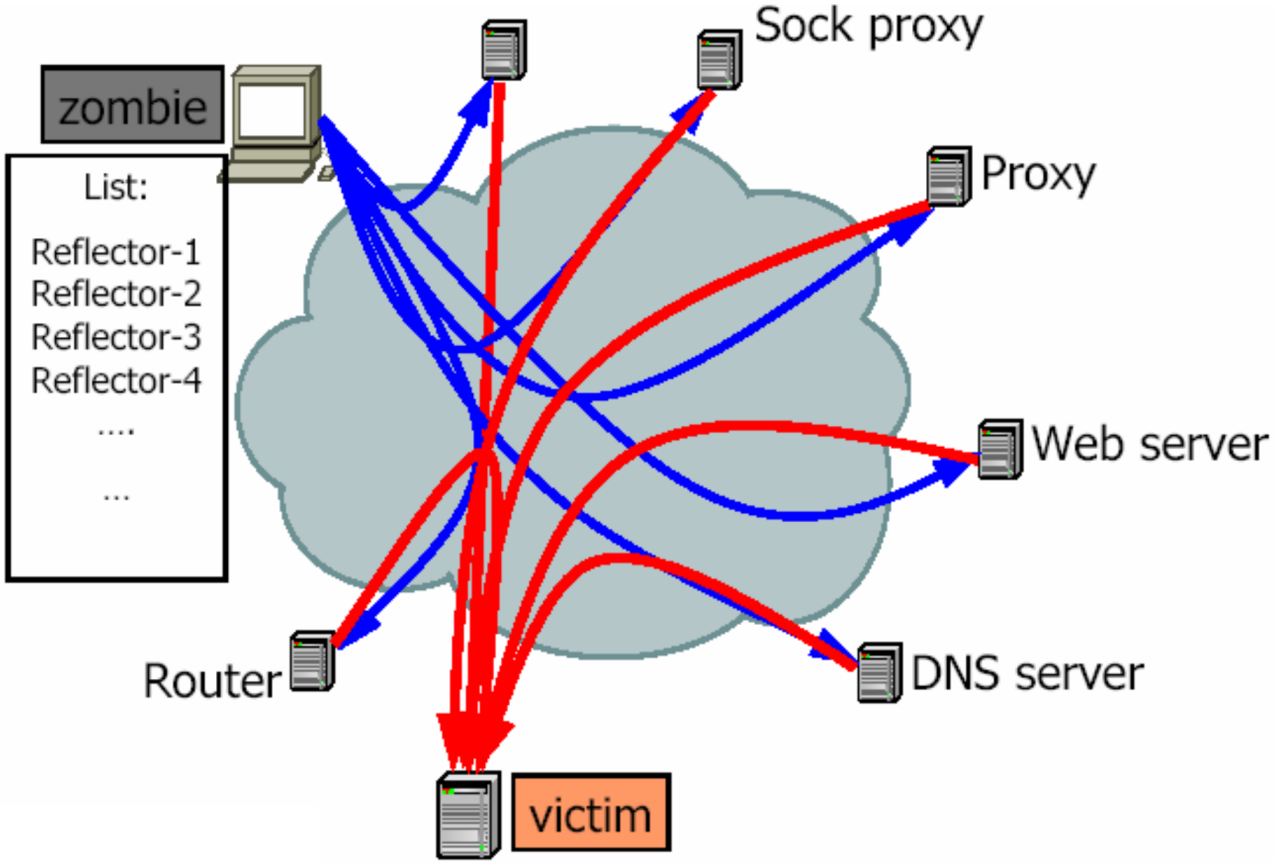
- TCP is connection-oriented $\rightarrow$ need to remember connection status
  - Some memory to manage connection status of **three-way handshake**

Smurf Amplification

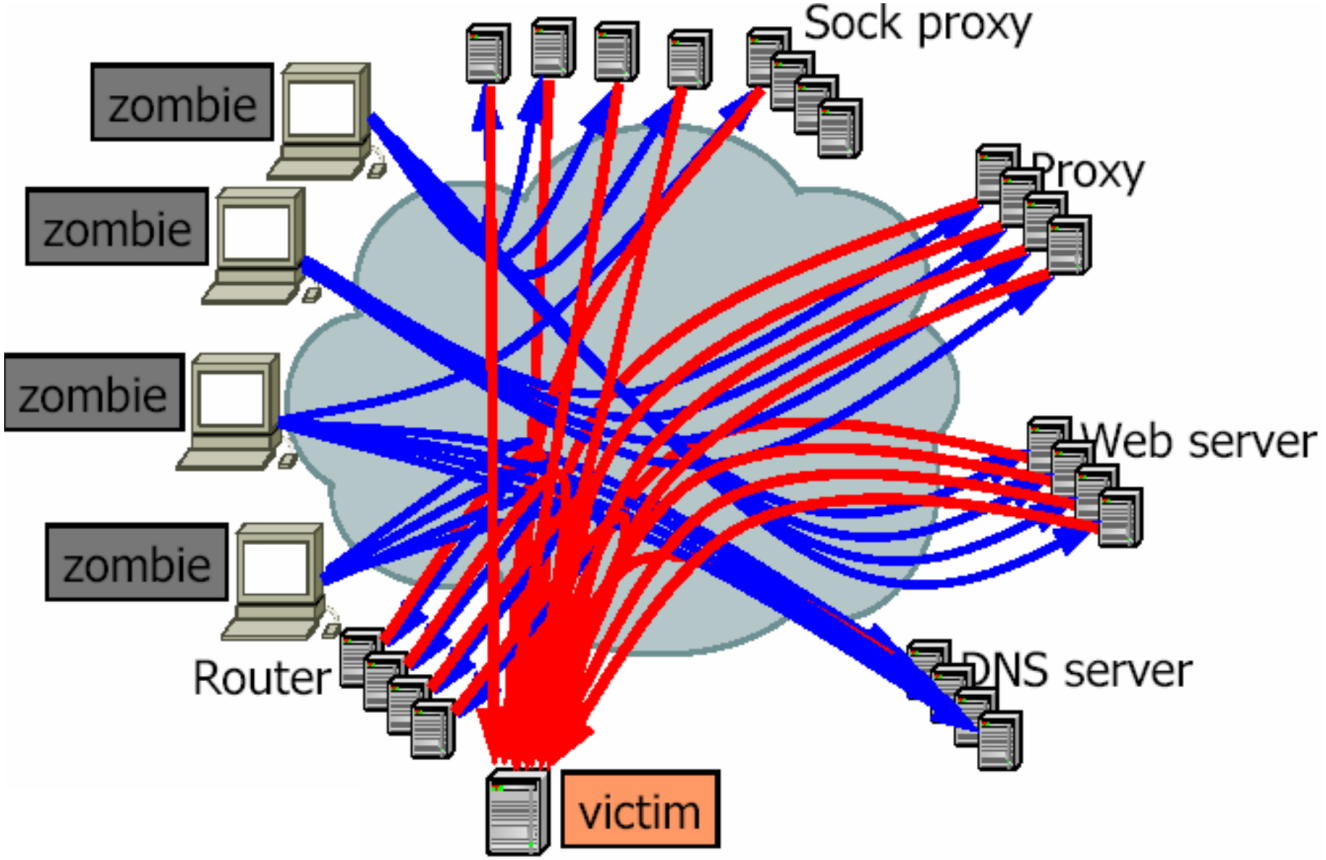


- Defense? $\rightarrow$ disable direct boradcast across subnets

Reflectors

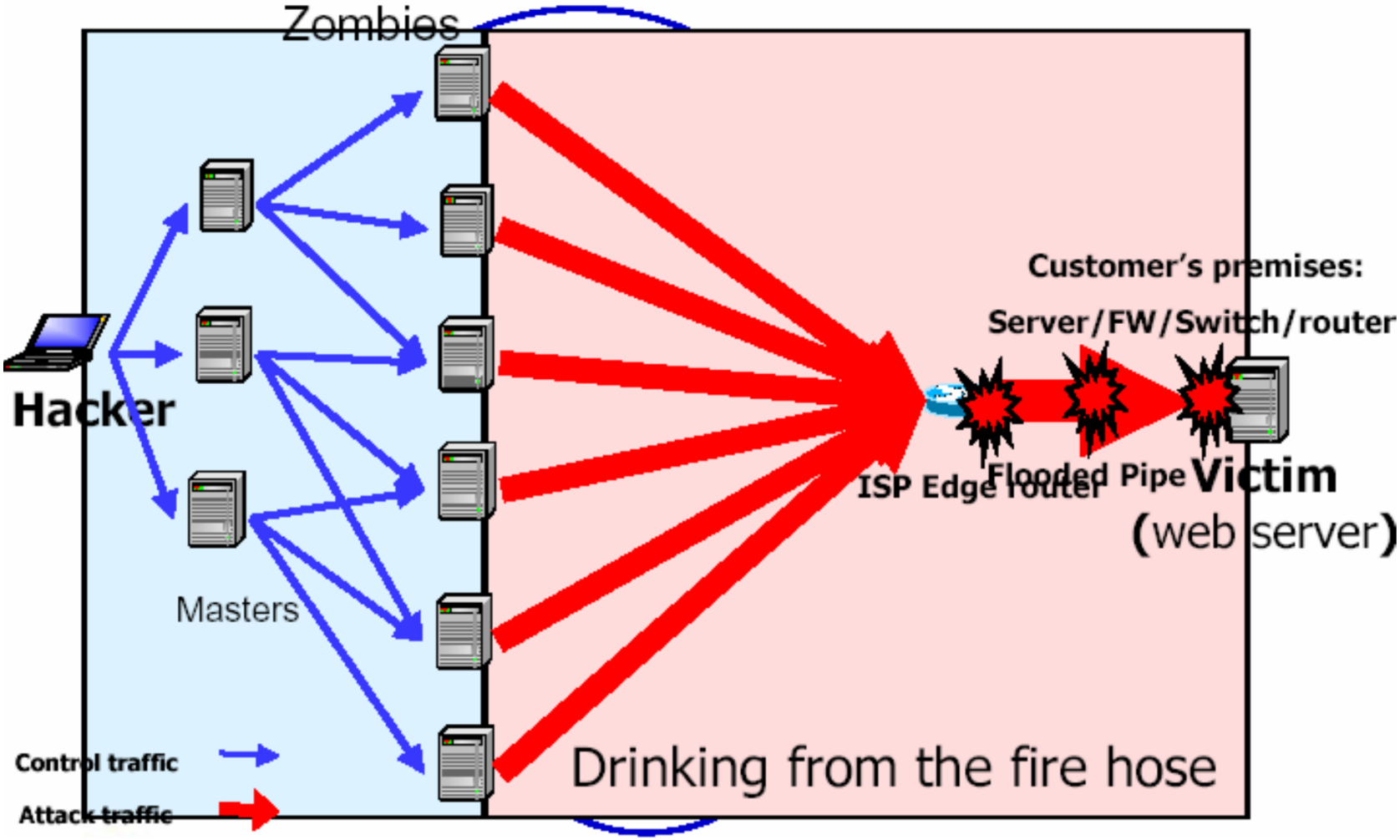

zombie
List:
Reflector-1
Reflector-2
Reflector-3
Reflector-4
....
...
Sock proxy
Proxy
Web server
DNS server
Router
victim

Reflector Attack from Multiple Nodes



- Defense? $\rightarrow$ Use IPSec (or similar techniques) to authenticate connections

Distributed Denial of Service (DDoS)



- The traffic could be over 600Gbps (Giga bits per second)

Defense to DDoS?

1. Increase servre capacity
   - Deploy more hardware and purchase more bandwidth
2. Detect and filtering attacking traffic
3. Use 3rd party services (from ISP, or Cloud Service providers)
   - E.g., Akamai: much larger pool of resource to do traffic detection and rinsing

Attack 4: TCP Session Hijacking

- What is TCP hijacking?
  - Attackers can take over and control victim's TCP connection
- Why is this attack possible?
  - TCP connection is identified by four elements in TCP header
    - ★ Source Port, Destination Port, Sequence Number, ACK number
  - Same session if a packet has expected values of above fields
    - ★ May also require the match of source IP and dest IP
    - ★ But it is not a problem due to IP address spoofing

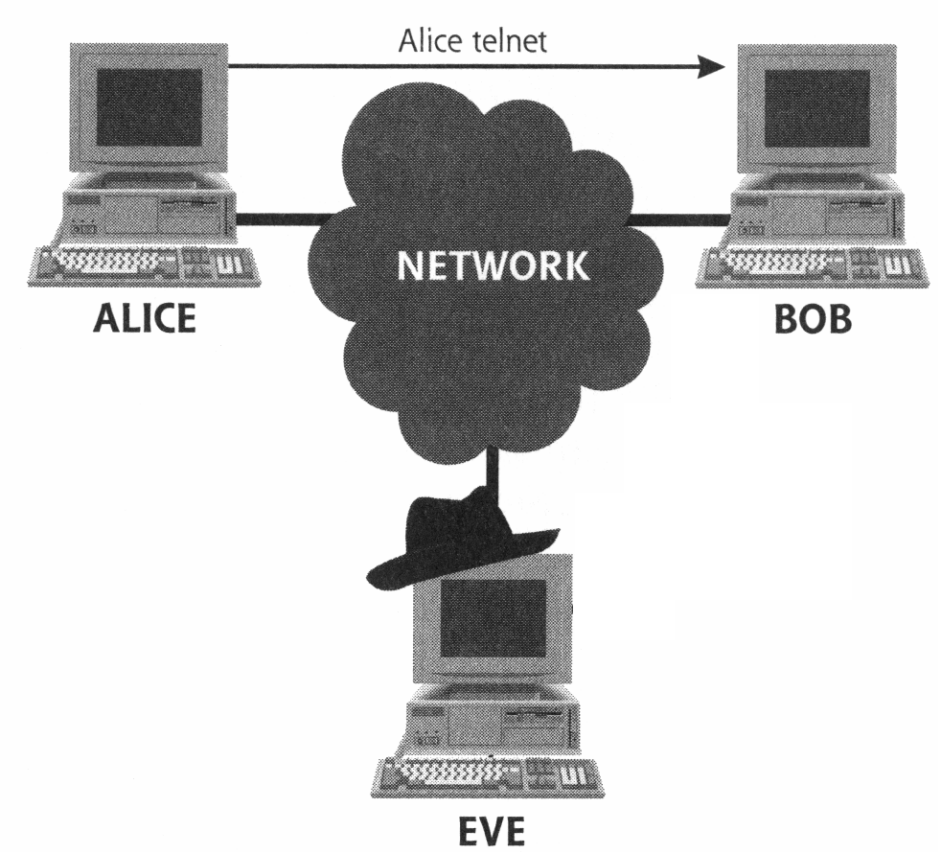
How does TCP Hijacking Work?

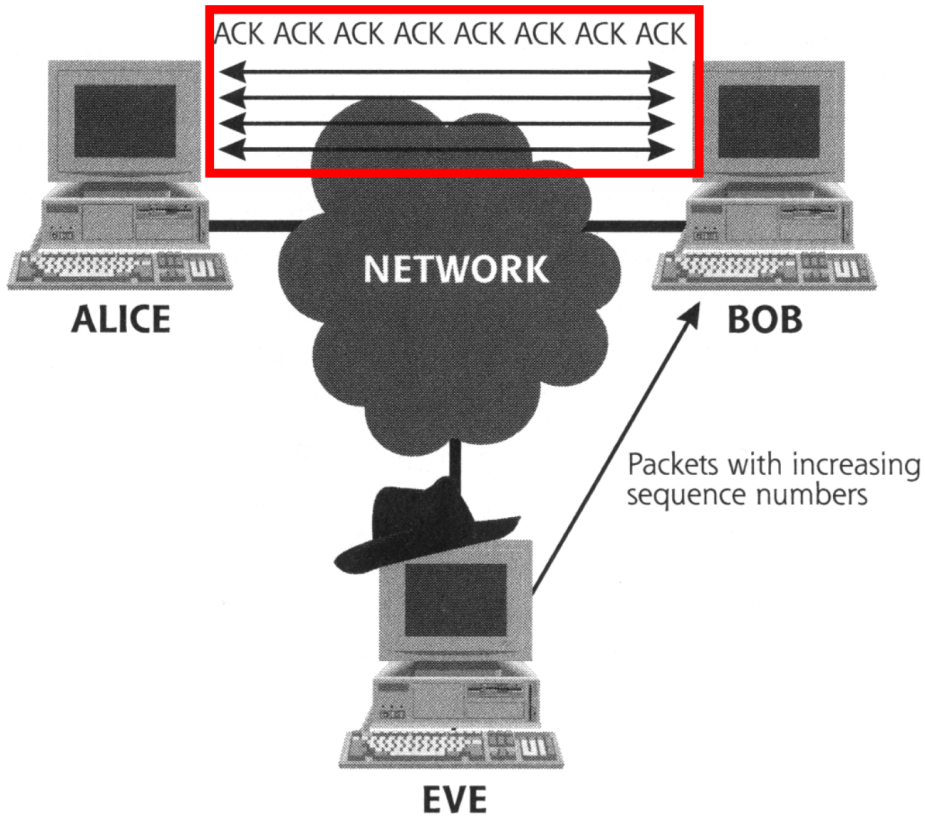
1. Eve first sniffs at the Network to discover an ongoing TCP connection
   - Then recover the corresponding TCP ACK sequence numbers being used
2. Eve creates an IP packet to carry its command of choice
   - Using Alice IP's address of the packet's source IP address
   - Also use the right TCP ACK sequence numbers

- Potential Threat of TCP Hijacking Attack
  - Attack to confidentiality
    - ★ Man-in-the-Middle: take over the TCP connection
    - ★ Sniffing and injecting fake packets
  - Attack to availability $\rightarrow$ abrupt existing connection

3. One limitation: ACK-storm to one victim
   - To avoid this problem, Eve can choose to bring Alice down by launching a DOS attack against Alice
   - Or Eve can perform an ARP or DNS spoofing on Alice and Bob to redirect their outgoing packets to a blackhole

Problem of ACK-Storm in TCP Hijack





Attack 5: Man-in-the-Middle

- Attacks can manipulate network traffic in the middle
- We have already seen lots of examples
  - Address Spoofing
    - ARP (MAC-IP address binding)
    - DNS spoofing (Domain Name to IP address binding)
  - Physically
    - Via wire-tapping, or Rogue WiFi Access Point
- Defenses?
  - Message Authentication Code
  - Message Encryption
  - Use CA (Certificate Authority) and PKI (Public Key Infrastructure)

Summary

- In this lecture, we have covered
  - Basic concepts of TCP/IP network
  - Different attacks (spoofing, sniffing, DoS, session hijacking, Man-in-the-Middle)
- Next Lecture:
  - Defense techniques, like Firewall, IDS, etc.