# Firewall: Introduction

CSC 154

Partial slides adapted from Peng Liu, College of IST, Penn State University with other content sources specified

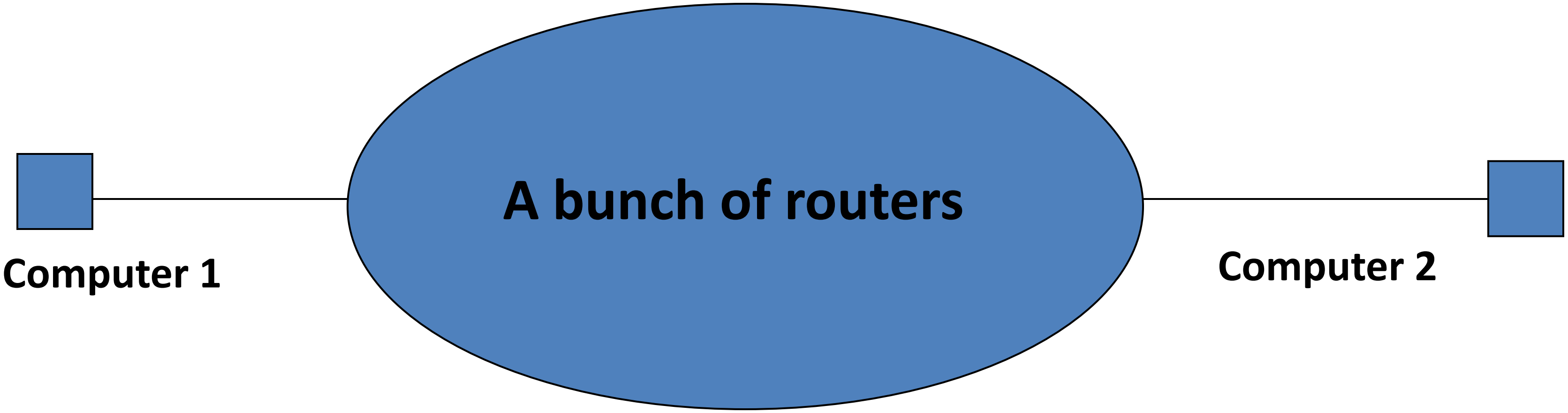
# Roadmap

- Network
  - Hosts, routers, packets
- Packet
  - DL header
  - IP header
  - TCP header
    - Port #
      - Popular network services and vulnerabilities
    - Flags
      - TCP3-way handshake
- Why we need a firewall

# What is the Internet?

- Purpose of the Internet: to enable any two computers to talk to each other
  - Online chat/video
  - Instant Message

- The Simplest Internet = Computer 1 (sender) + (a bunch of routers) + Computer 2 (receiver)

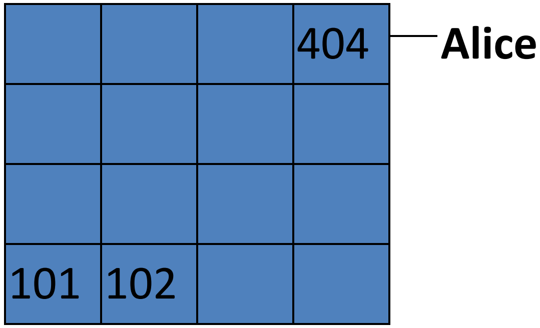


# How does the Internet work?

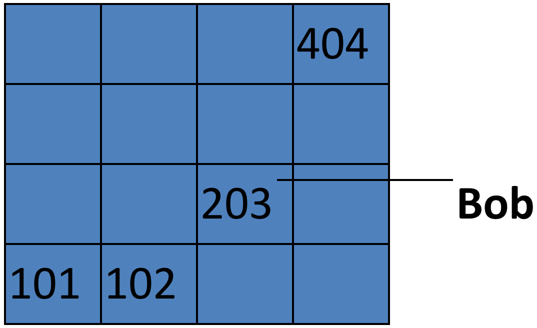
- The routers **relay** the message from the sender to the receiver
- The message is contained in a **packet**
  - which could be viewed as an *envelope*
- The packet could get **lost** or **corrupted** during transmission
- For reliability, the receiver will typically send an **acknowledgement** note back
- There may exist multiple **routes** (or **paths**) from the sender to the receiver

# How to use TCP ports and IP addresses (1)

When Alice sends a letter to Bob,
what to put on the envelope?



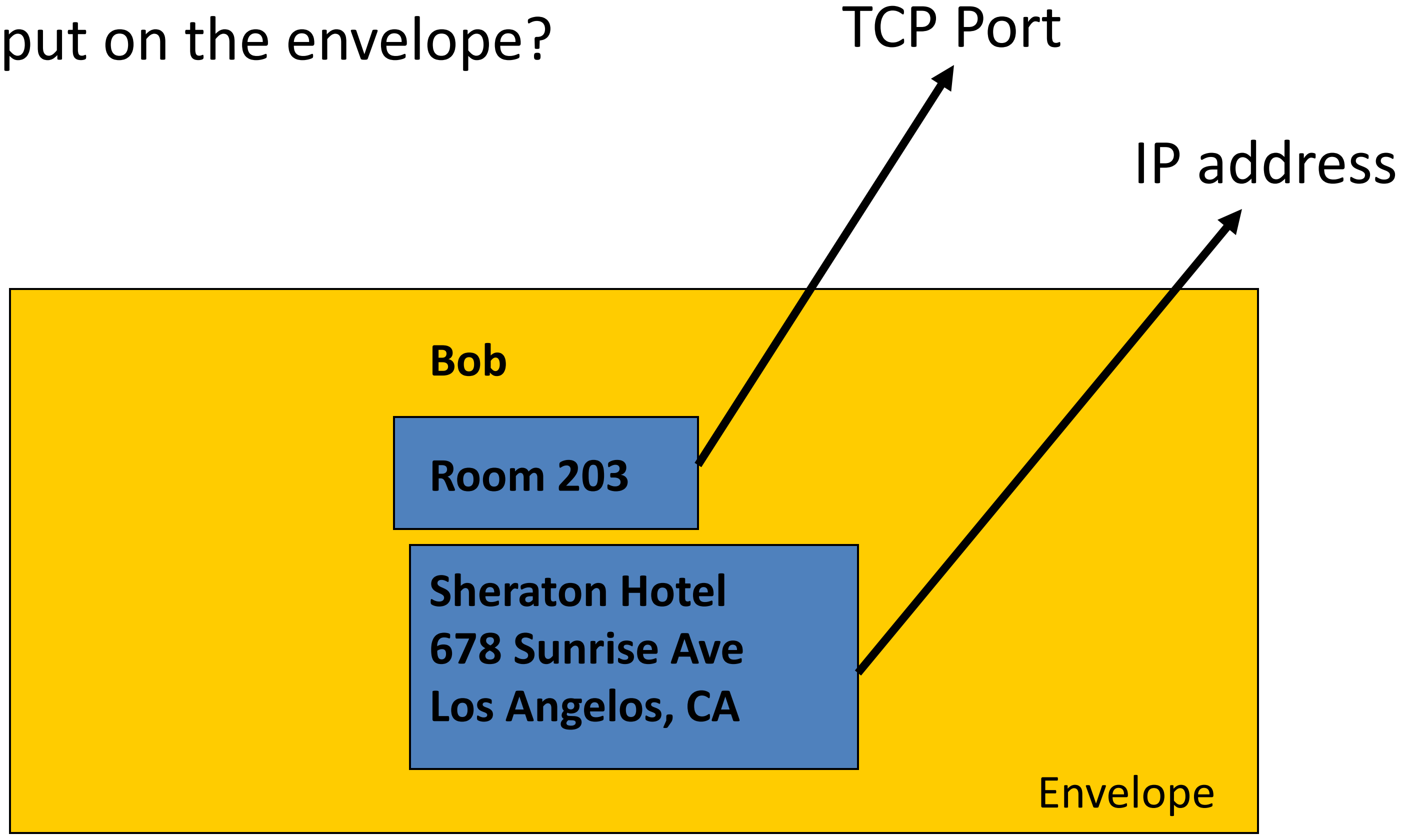
**Hilton Hotel**
**234 Moonrise Ave**
**Boston, MA 02116**



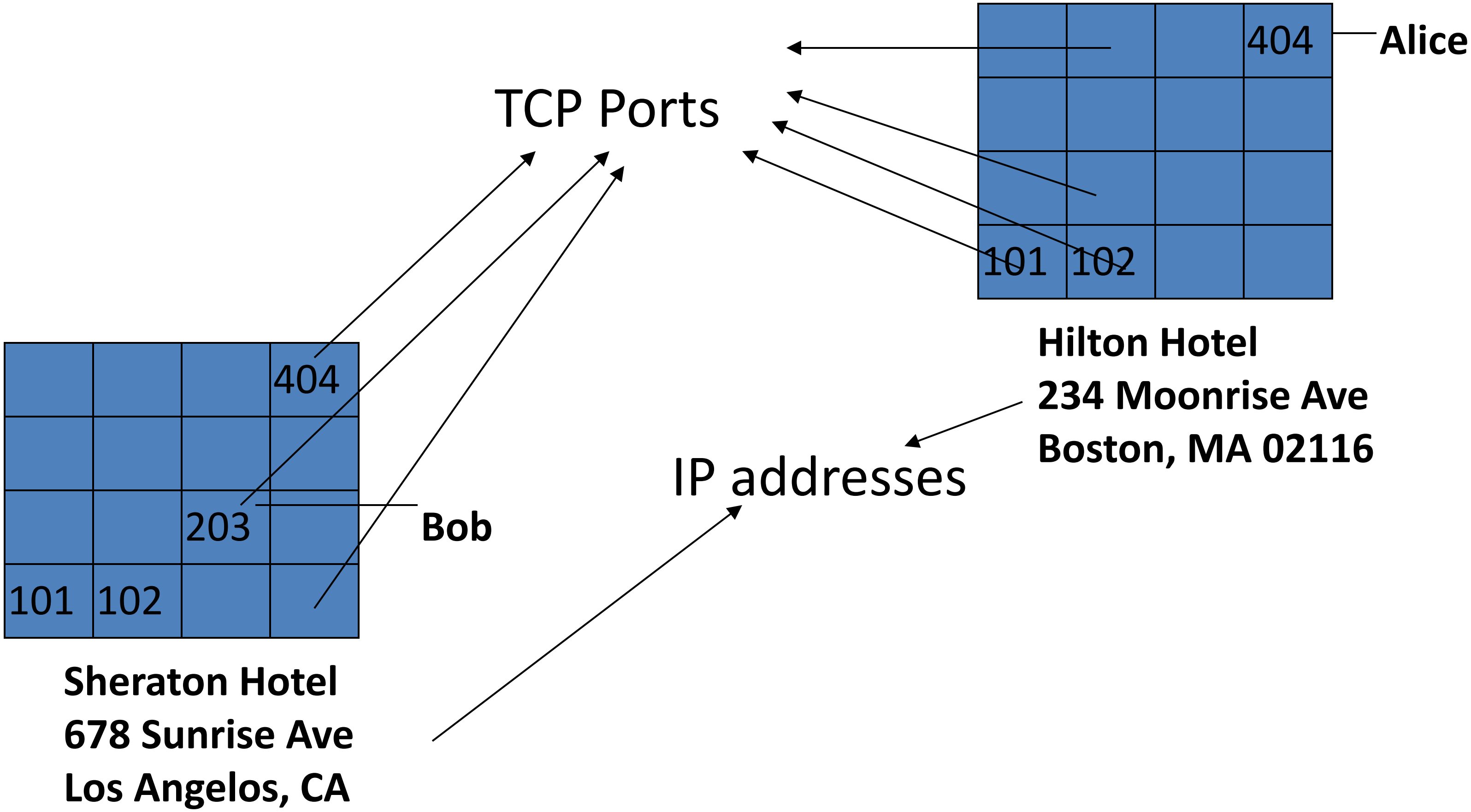
**Sheraton Hotel**
**678 Sunrise Ave**
**Los Angelos, CA**

# How to use TCP ports and IP addresses (2)

When Alice sends a letter to Bob,
what to put on the envelope?



# How to use TCP ports and IP addresses (3)



Alice

TCP Ports

Hilton Hotel
234 Moonrise Ave
Boston, MA 02116

IP addresses

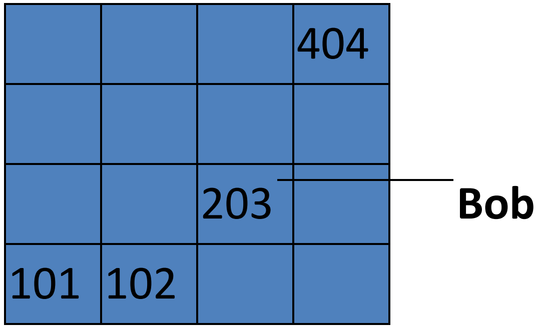
Bob

Sheraton Hotel
678 Sunrise Ave
Los Angelos, CA

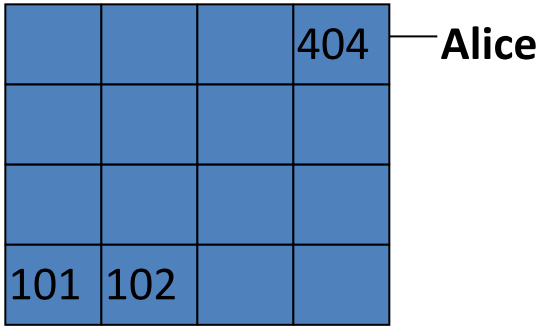
# How to use TCP ports and IP addresses (4)

When Alice calls Bob, what number to dial?

| | |
|---|---|
| (310) 642-1111 | Ext. 203 |
| IP address | TCP port number |



**Sheraton Hotel**
**678 Sunrise Ave**
**Los Angelos, CA**
**(310) 642-1111**



**Hilton Hotel**
**234 Moonrise Ave**
**Boston, MA 02116**

# How to use TCP ports and IP addresses (5)

What does a TCP port mean in computer world?

-- Multiple programs are running on a single computer

-- -- We assign a port number to **each program**

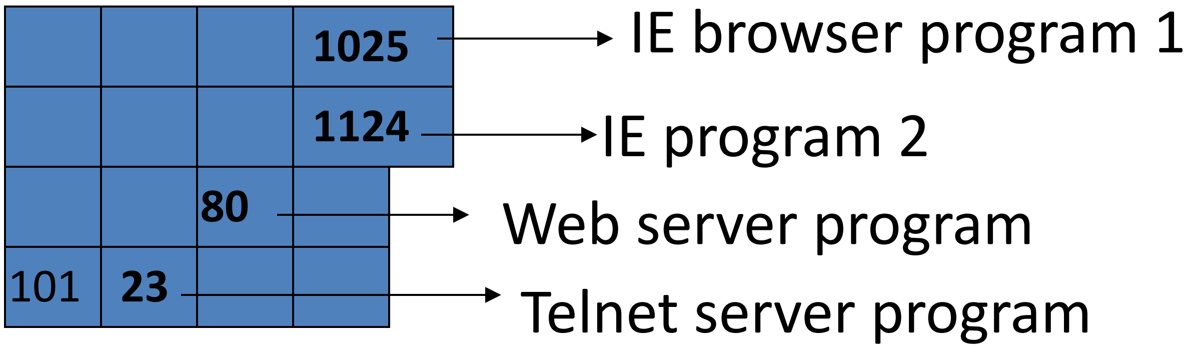
-- Two types of programs:

-- **Service**-providing programs

-- Service-requesting programs

-- The first 1,024 ports re reserved for **services**

-- From OS perspectives: **processes**, **sockets**



1025 → IE browser program 1

1124 → IE program 2

80 → Web server program

101

23 → Telnet server program

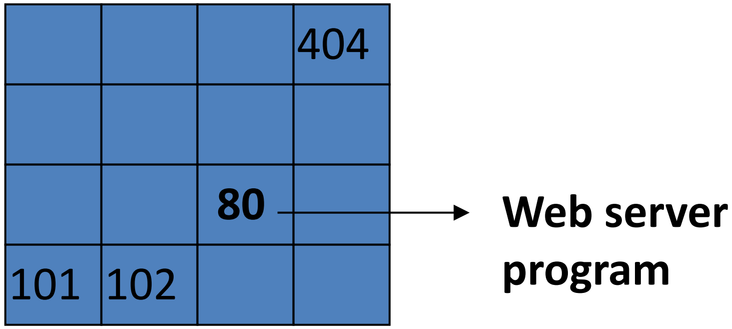
**Receiver PC**
**IP address: 130.126.2.45**

# How to use TCP ports and IP addresses (6)

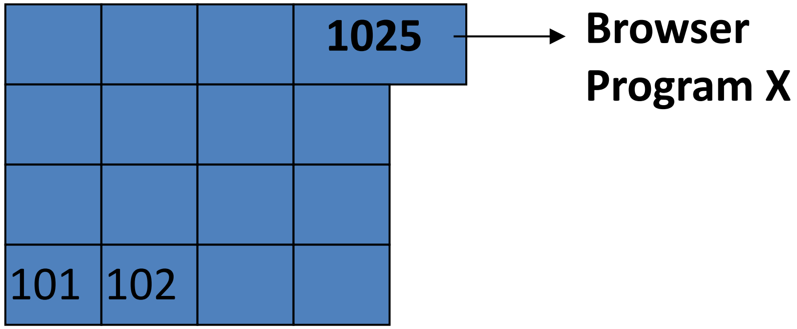
When browser X sends a message to the Web server:

| | | |
|---|---|---|
| **130.126.2.45**<br>IP address | **80**<br>TCP port number | **Message content** |

The simplest packet



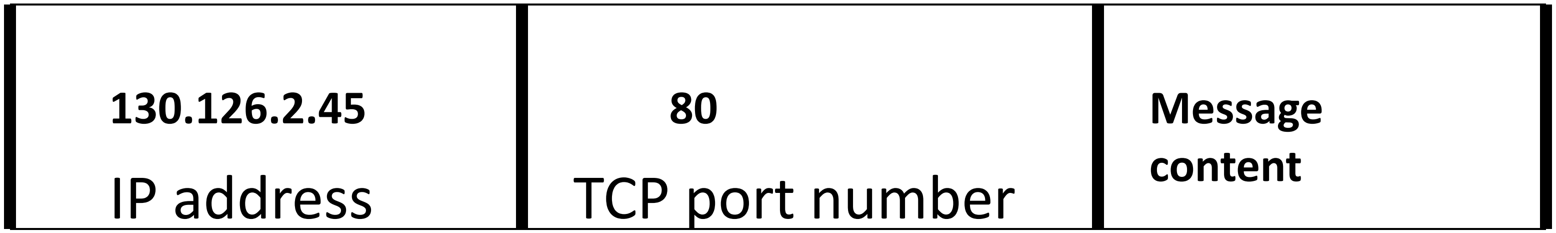
**Receiver computer**
**IP address: 130.126.2.45**



**Sender computer**
**IP address: 162.11.200.5**

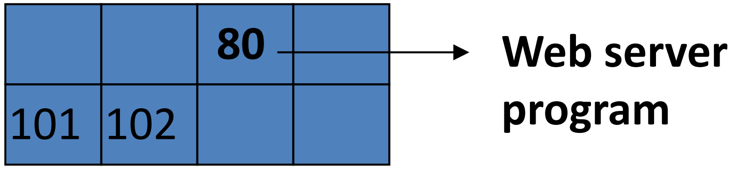
# The simplest packet has a drawback! (7)

When browser X sends a message to the Web server:

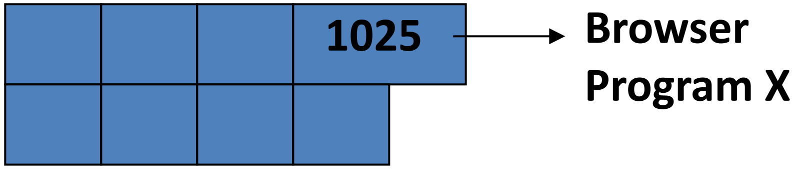


The simplest packet

Drawback: there is NO return address!!! When the receiver sends back the acknowledgement, the receiver will cry – because …



**Receiver computer**
**IP address: 130.126.2.45**



**Sender computer**
**IP address: 162.11.200.5**

# How to use TCP ports and IP addresses (8)

- Preamble | Destination Mac | Source Mac | Source IP | Destination IP | Source Port | Destination Port | payload | CRC |
  - Transport Layer Header
    - TCP Header
    - UDP Header
  - Network Layer/Internet Protocol Header
  - Data Link Layer Header

# What elements are inside a TCP header? What is the size for each such element?

- Source Port
- Destination Port
  - 16bits each

- ACK Flag
- SYN Flag
  - 1bit each

# What elements are inside an IP header? What is the size for each such element?

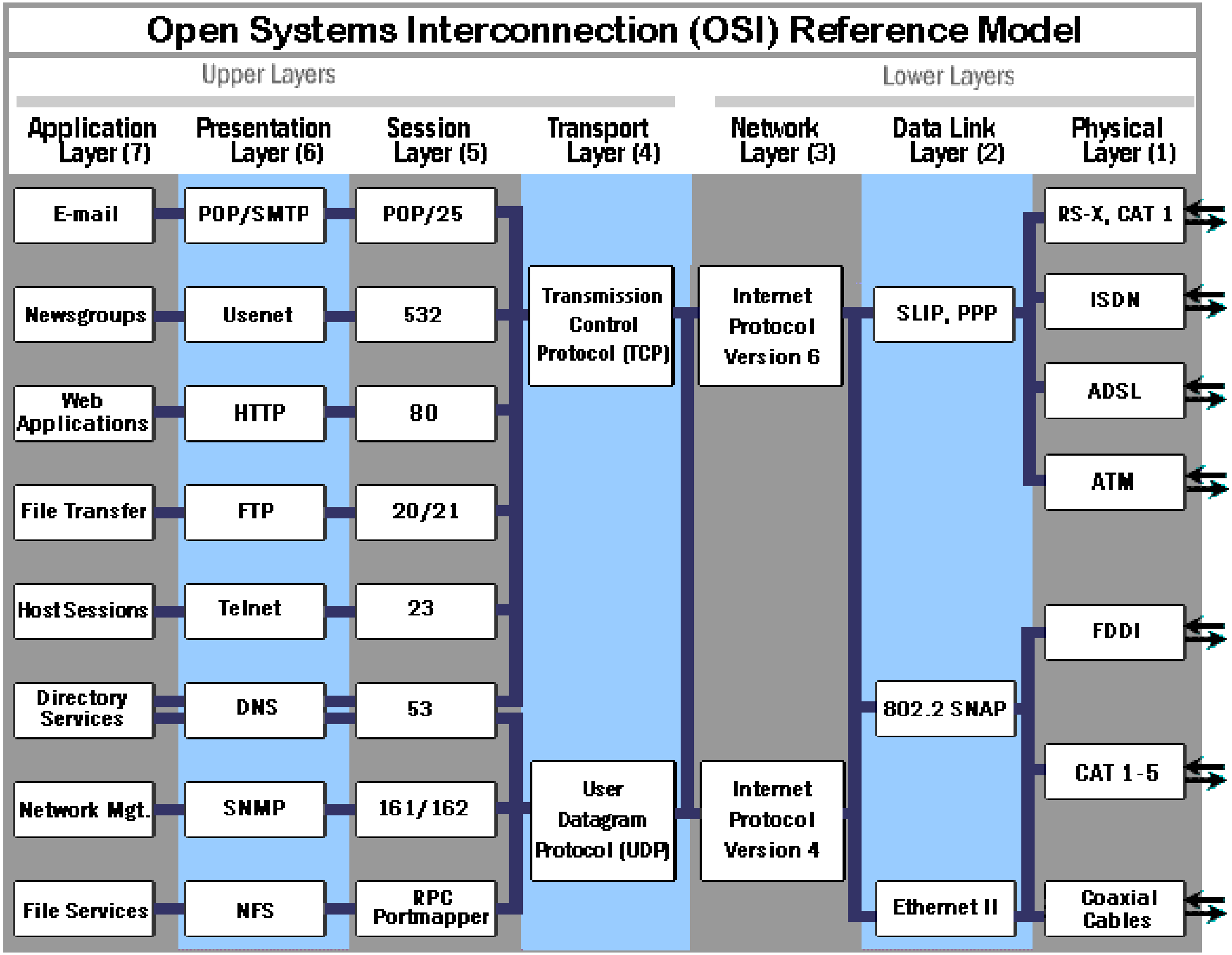
- Protocol
  - 8 bits
  - TCP? UDP?
  - Protocol can also be ICMP

- Source IP
- Destination IP
  - both 32 bits

# Popular Internet services and their port numbers (1)

- 80 – HTTP
- 23 – Telnet
- 25 – SMTP
- 20/21 - FTP
- 53 – DNS
- 22 - SSH

# Popular Internet services and their port numbers (2)



Open Systems Interconnection (OSI) Reference Model

Upper Layers

Lower Layers

Application Layer (7)

Presentation Layer (6)

Session Layer (5)

Transport Layer (4)

Network Layer (3)

Data Link Layer (2)

Physical Layer (1)

E-mail — POP/SMTP — POP/25

Newsgroups — Usenet — 532

Web Applications — HTTP — 80

File Transfer — FTP — 20/21

Host Sessions — Telnet — 23

Directory Services — DNS — 53

Network Mgt. — SNMP — 161/162

File Services — NFS — RPC Portmapper

Transmission Control Protocol (TCP)

User Datagram Protocol (UDP)

Internet Protocol Version 6

Internet Protocol Version 4

SLIP, PPP

802.2 SNAP

Ethernet II

RS-X, CAT 1

ISDN

ADSL

ATM

FDDI

CAT 1-5

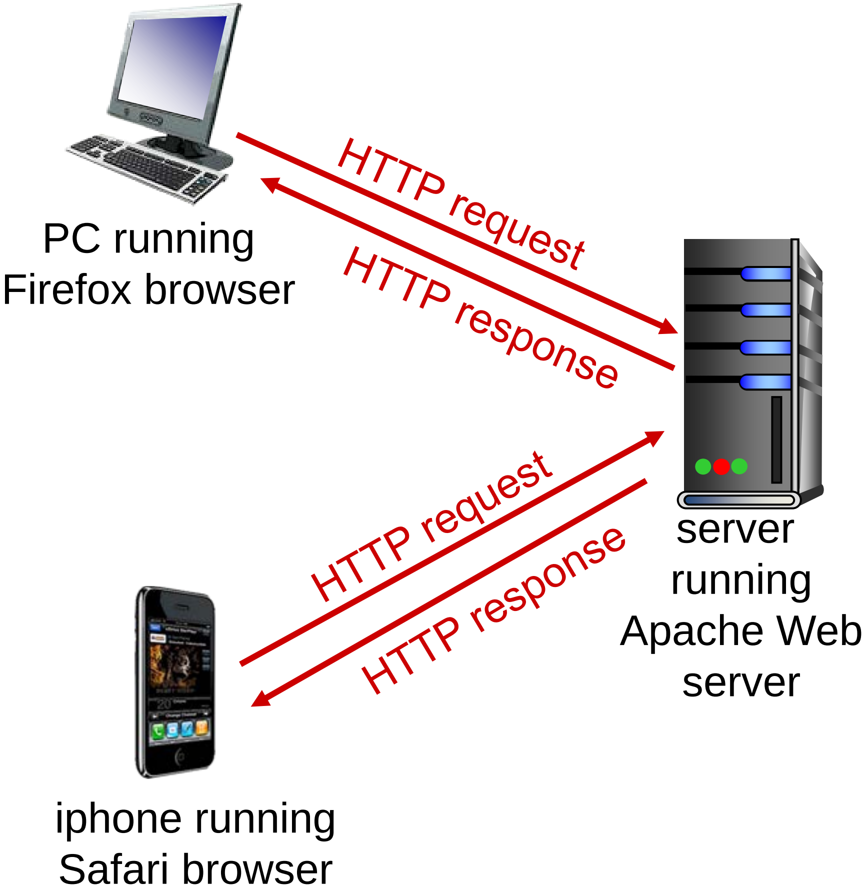
Coaxial Cables

Picture from http://homepages.uel.ac.uk/u0116278/osi.htm

# HTTP overview

**HTTP: hypertext transfer protocol**

- Web's application layer protocol
- Port: 80 on servers
- client/server model
  - *client:* browser that requests, receives, (using HTTP protocol) and "displays" Web objects
  - *server:* Web server sends (using HTTP protocol) objects in response to requests



PC running Firefox browser
HTTP request
HTTP response
server running Apache Web server
HTTP request
HTTP response
iphone running Safari browser

adapted from Jim Kurose, Keith Ross,
*Computer Networking: A Top Down Approach*, 6th edition, with content from other specified sources

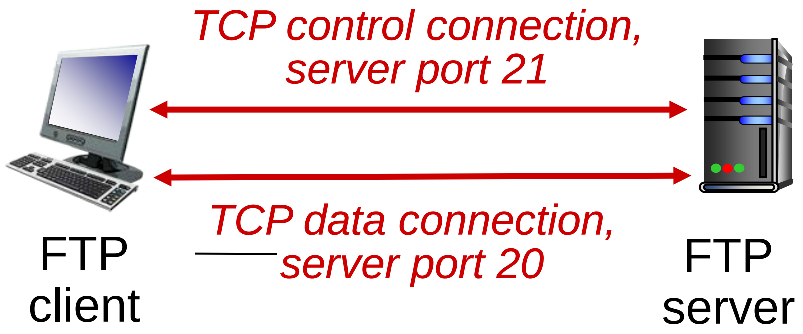
# Browser-side Vulnerability

- Malicious (java) applets
- Buffer overflow vulnerabilities of browser plug-ins
- Cross Site Scripting vulnerabilities

# FTP: separate control, data connections

***Sharing or isolation: what if the uploaded file is a trojan horse?***

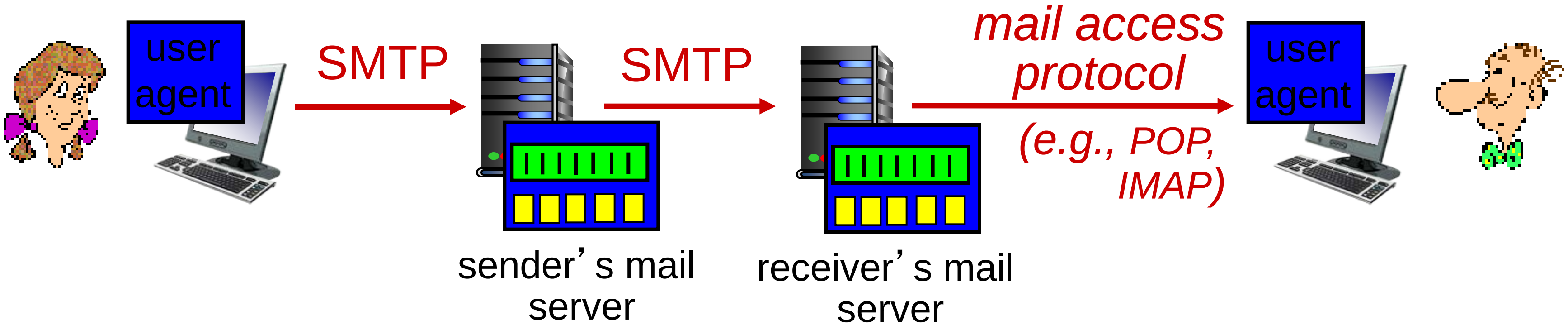
- FTP client contacts FTP server at port 21, using TCP
- client authorized over control connection
- client browses remote directory, sends commands over control connection
- when server receives file transfer command, *server* opens *2nd* TCP data connection (for file) *to* client
- after transferring one file, server closes data connection



- server opens another TCP data connection to transfer another file
- control connection: *"out of band"*
- FTP server maintains "state": current directory, earlier authentication

adapted from Jim Kurose, Keith Ross,
*Computer Networking: A Top Down Approach*, 6th edition, with content from other specified sources

# Mail access protocols



- SMTP: delivery/storage to receiver's server
- mail access protocol: retrieval from server
  - POP: Post Office Protocol [RFC 1939]: authorization, download
  - IMAP: Internet Mail Access Protocol [RFC 1730]: more features, including manipulation of stored msgs on server
  - HTTP: gmail, Hotmail, Yahoo! Mail, etc.

- Sniffing: Plain text login/password in SMTP
- Virus, SPAM

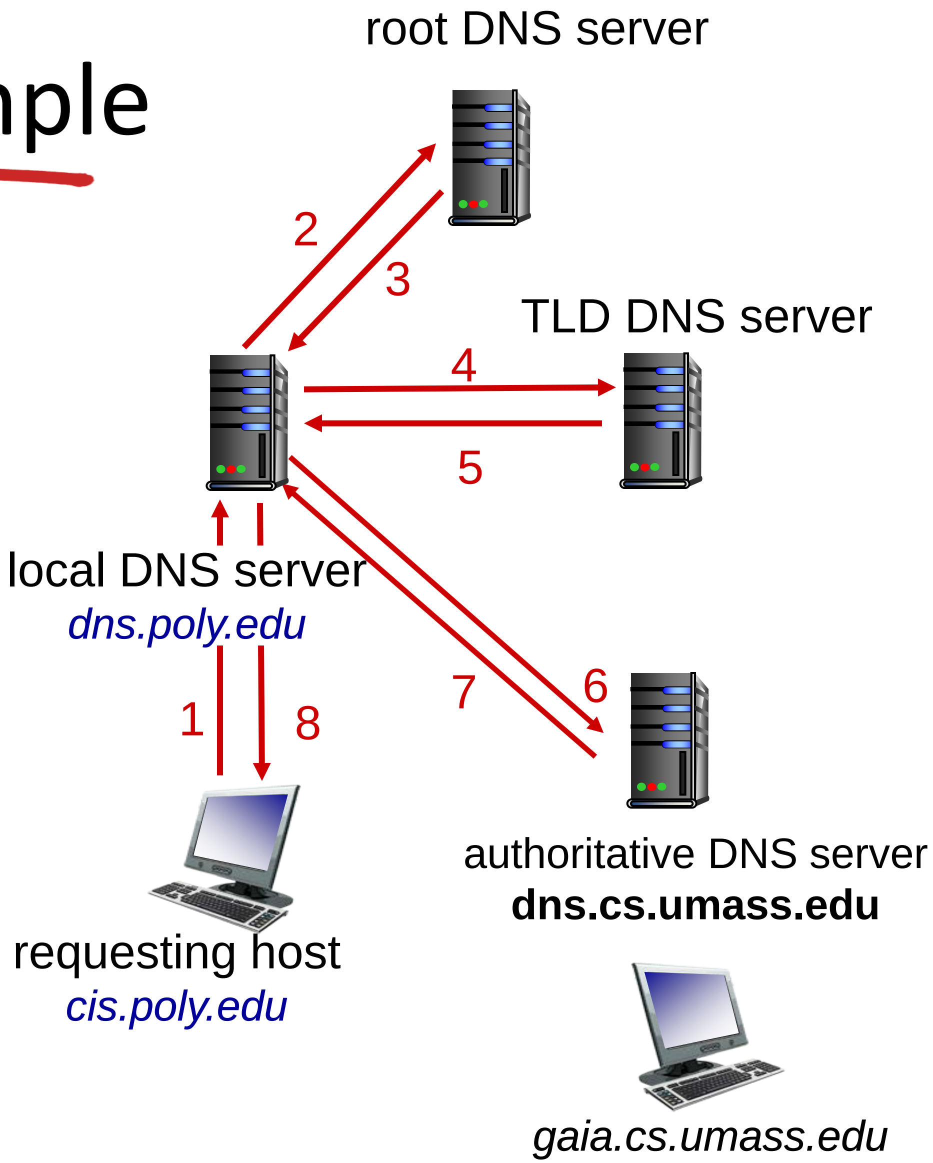
adapted from Jim Kurose, Keith Ross,
*Computer Networking: A Top Down Approach*, 6th edition, with content from other specified sources

# DNS name resolution example

- host at cis.poly.edu wants IP address for gaia.cs.umass.edu

*iterated query:*

- contacted server replies with name of server to contact
- "I don't know this name, but ask this server"



adapted from Jim Kurose, Keith Ross,
*Computer Networking: A Top Down Approach*, 6th edition, with content from other specified sources

# Attacking DNS

## DDoS attacks

- Bombard root servers with traffic
  - Not successful to date
  - Traffic Filtering
  - Local DNS servers cache IPs of TLD servers, allowing root server bypassed
- Bombard TLD servers
  - Potentially more dangerous

## Redirect attacks

- Man-in-middle
  - Intercept queries
- DNS poisoning
  - Send bogus replies to DNS server, which caches

## Exploit DNS for DDoS

- Send queries with spoofed source address: target IP
- Requires amplification

adapted from Jim Kurose, Keith Ross,
*Computer Networking: A Top Down Approach*, 6th edition, with content from other specified sources

# How does Telnet work? Vulnerabilities?

- For users to remotely access system over the net
  - unencrypted login/password
  - Port 23

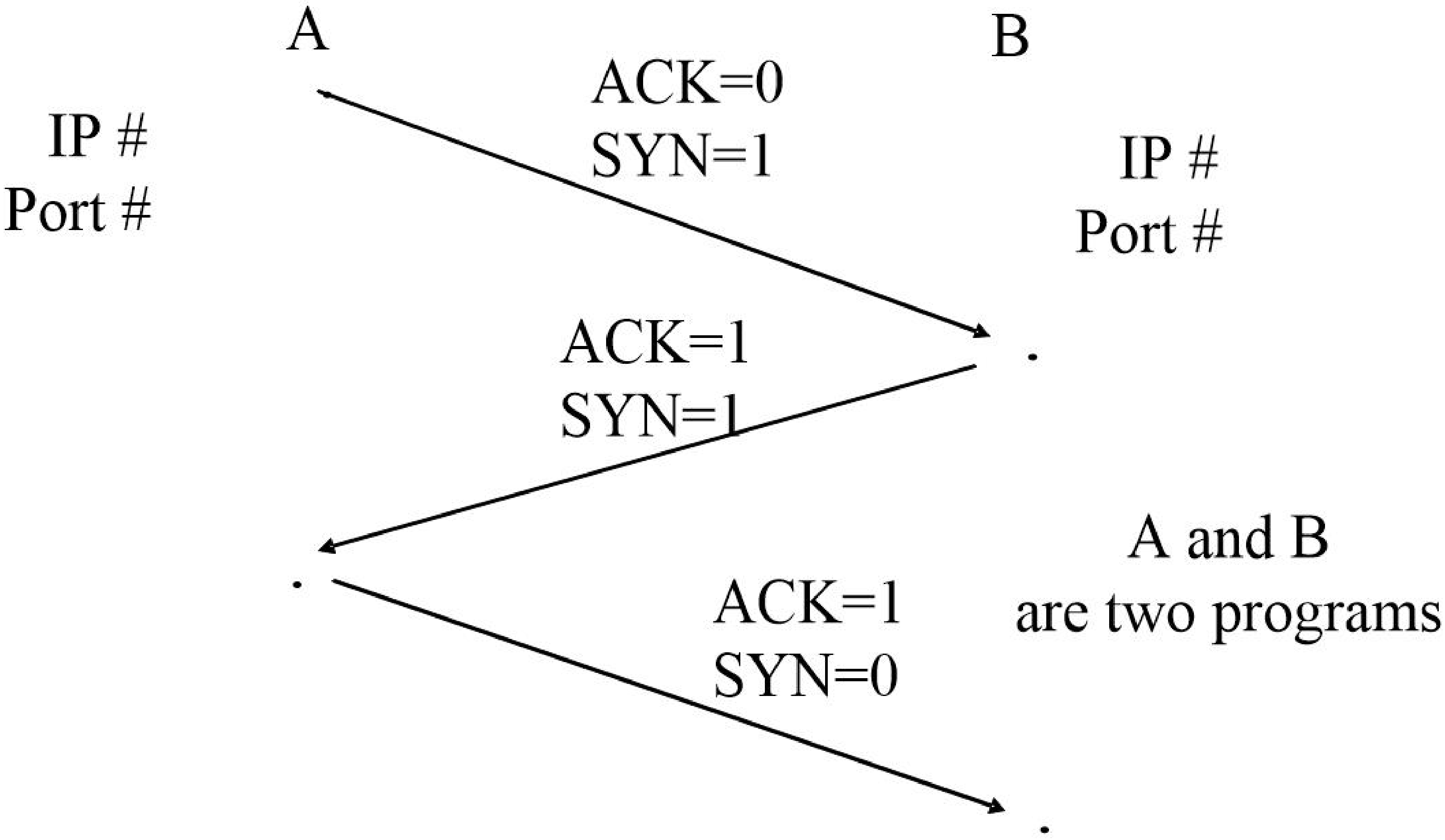
- Vulnerable because of lack of encryption:
  - Password sniffing
  - Hijack a session already in progress

# How to establish a TCP connection between two hosts?

- A TCP connection
  - Step 1: establish the connection
  - Step 2: send the messages back and forth
  - Step 3: terminate the connection

- A phone call session
  - Step 1: Dial the phone number
  - Step 2: do talking
  - Step 3: hang up

# TCP 3-way handshake



A

B

IP #
Port #

ACK=0
SYN=1

IP #
Port #

ACK=1
SYN=1

A and B
are two programs

ACK=1
SYN=0

# TCP 3-Way Handshake (step-by-step)

- Assume client A with (5.6.7.8:xxxx) wants to establish a TCP connection with server B (1.2.3.4: yyyy)
  - 1st Step: Connection request by client A : Contains source IP(5.6.7.8), source port(xxxx), and destination IP (1.2.3.4), destination Port(yyyy) . Also has SYN flag is set to 1, ACK is 0
  - 2nd Step: the ip 1.2.3.4 will acknowledge by sending ACK flag (set to 1).
    - Source 1.2.3.4 destination is 5.6.7.8
  - 3rd Step: the original IP of 5.6.7.8 sends back to 1.2.3.4 ACK 1 SYN 0.

- Additional comments: SYN flag is used to setup TCP connection, ACK flag is used to acknowledge receipt of a packet.

# UDP Has No Handshaking

UDP: no "connection" between client & server

- no handshaking before sending data
- sender explicitly attaches IP destination address and port # to each packet
- rcvr extracts sender IP address and port# from received packet

UDP: transmitted data may be lost or received out-of-order

adapted from Jim Kurose, Keith Ross,
*Computer Networking: A Top Down Approach*, 6th edition, with content from other specified sources

# Compare TCP with UDP

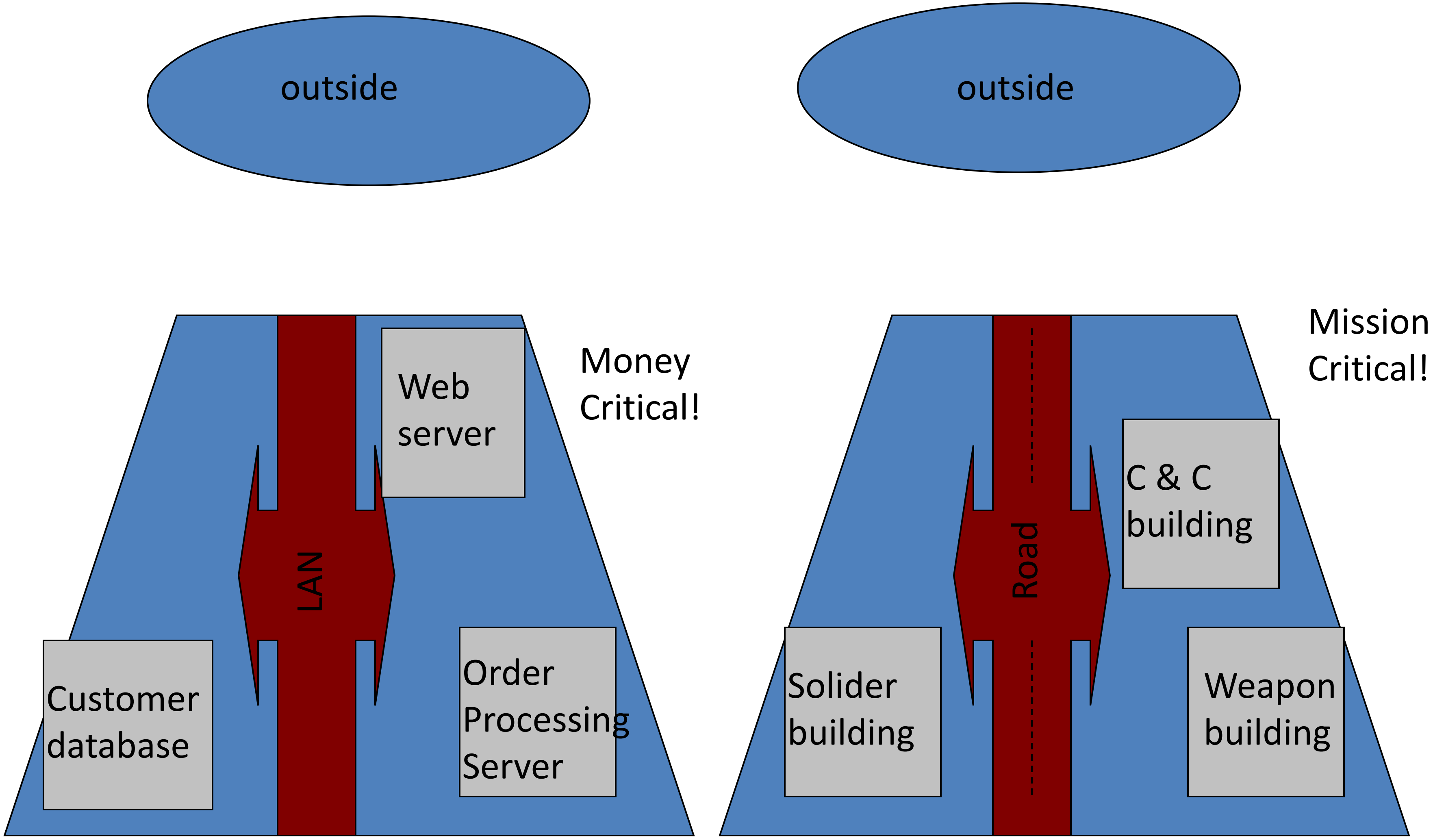
- Header information exactly same
  - TCP connection-oriented, reliable, flags
    - TCP provides reliable, in-order byte-stream transfer (“pipe”) between client and server
  - UDP connectionless, no flags
    - UDP provides unreliable transfer of groups of bytes (“datagrams”) between client and server
  - All packets of the session must be blocked if the goal is to block a UDP session; however, TCP session packets to be blocked can be distinguished by ACK.

# Why do we need a firewall?

- Internet/network
- Packet

- The attacker hacks a network via packets
  - Why a packet can enable the attacker to break into the network?
- Firewalls can protect a network
  - Why?

# A network = a military base



outside

outside

Web server

Money Critical!

LAN

Customer database

Order Processing Server

Mission Critical!

Road

C & C building

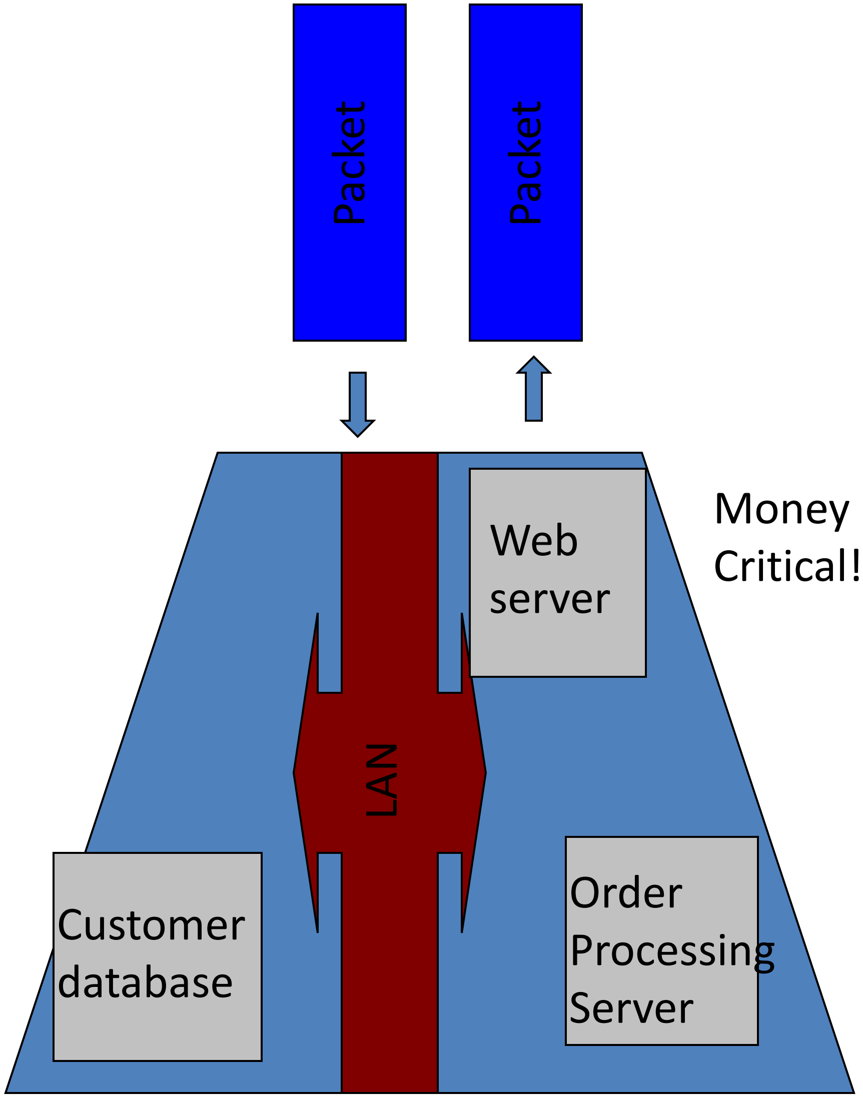
Solider building

Weapon building

Network

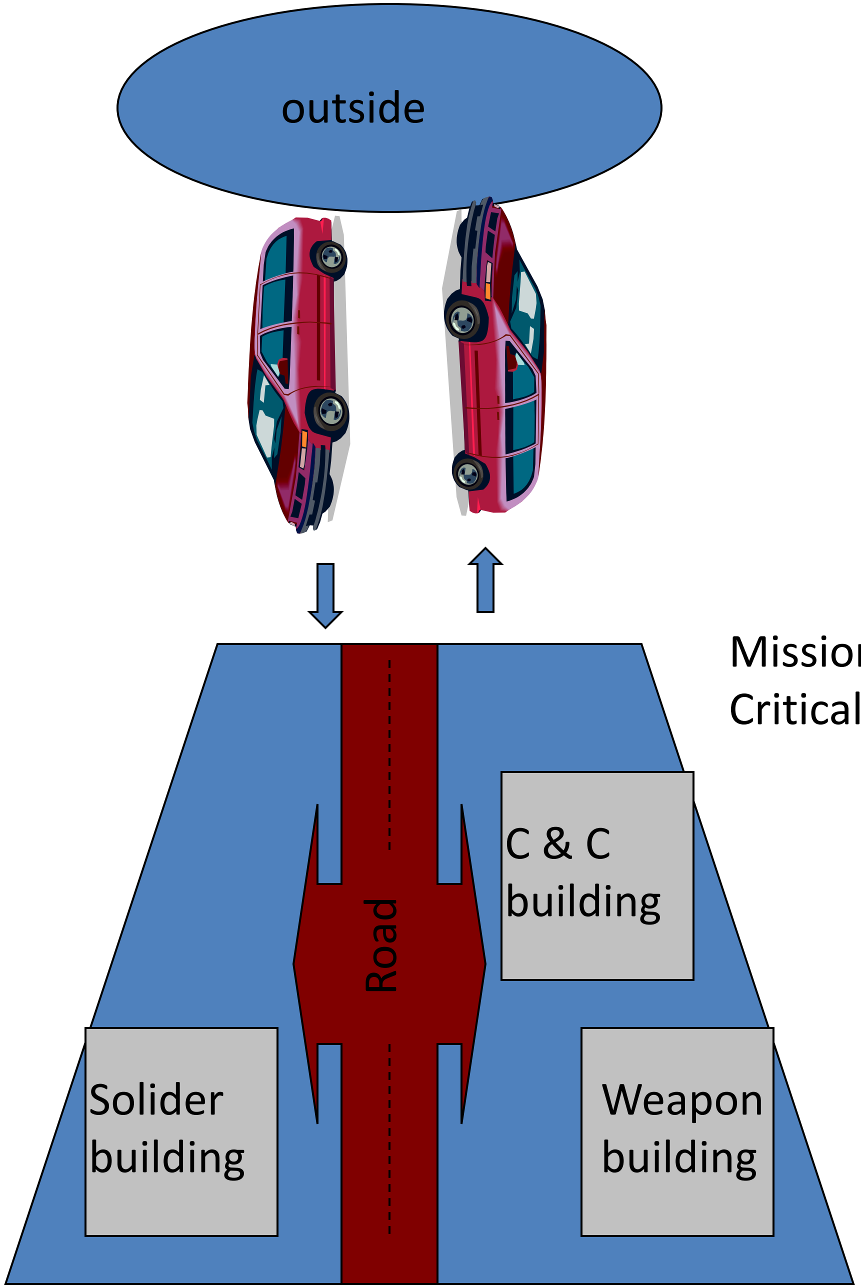
Military base

# A packet = a van



Money Critical!

Network

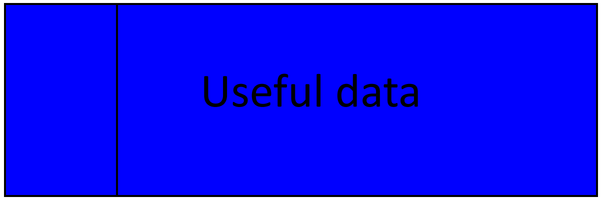


Mission Critical!

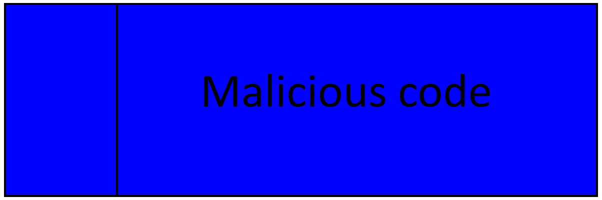
Military base

# Good packets vs. bad packets

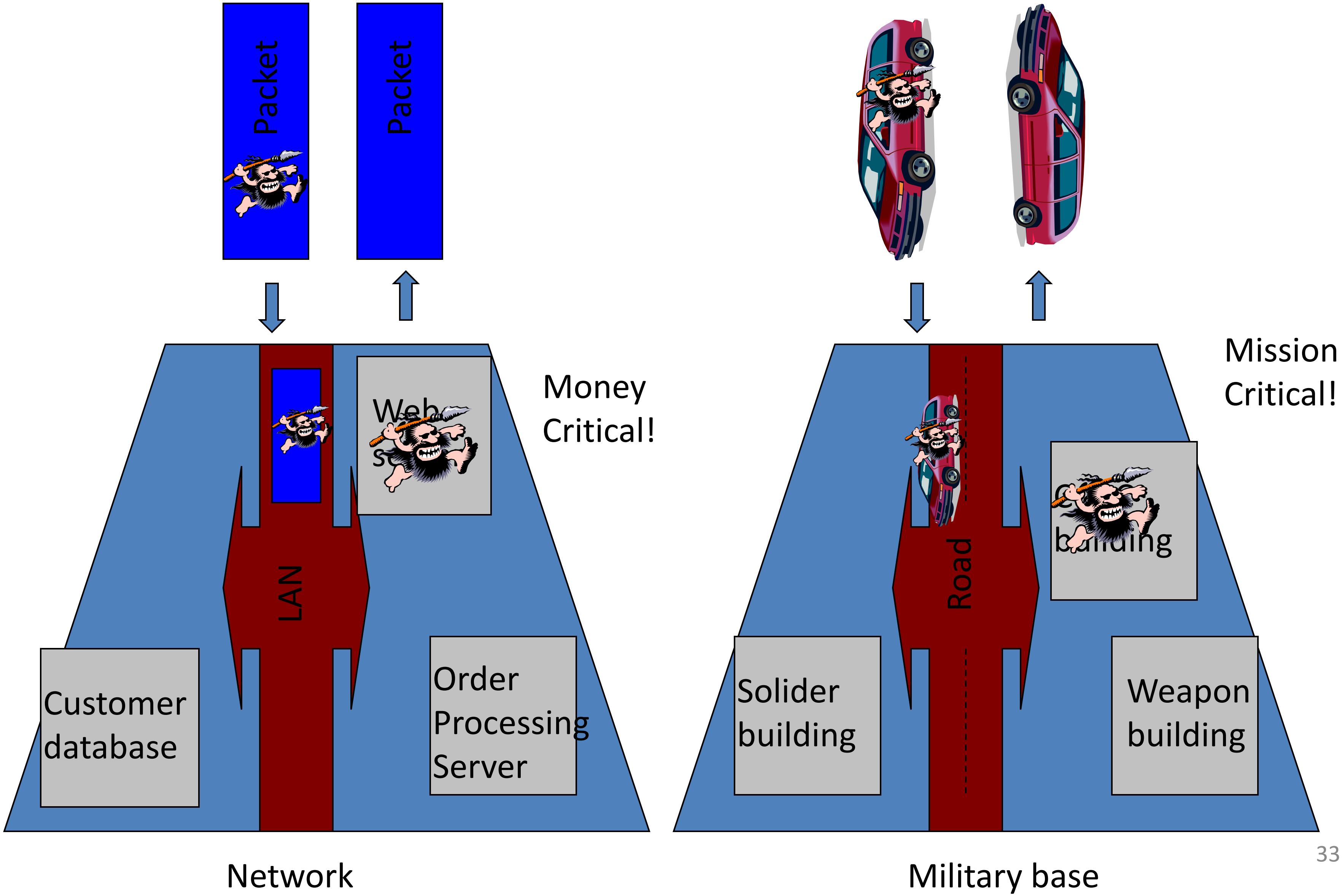
A good packet = a truck with chocolate


Useful data

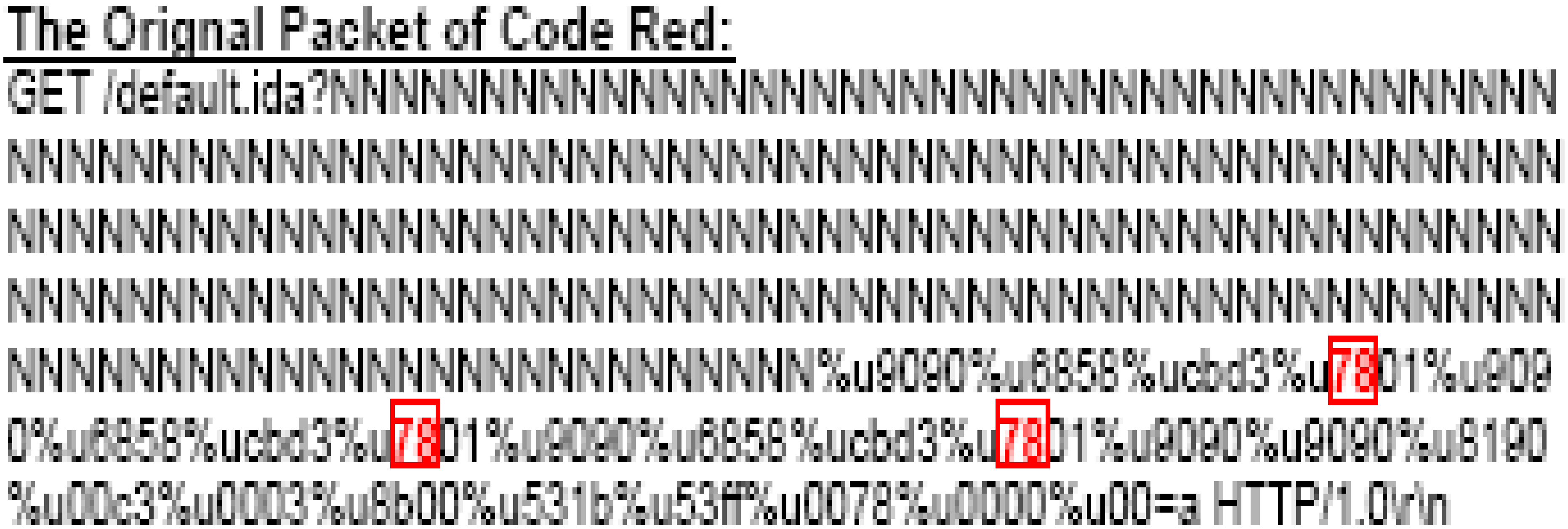
A bad packet = a truck with terrorists and bombs


Malicious code

# A bad packet can hack the network if you let it in!



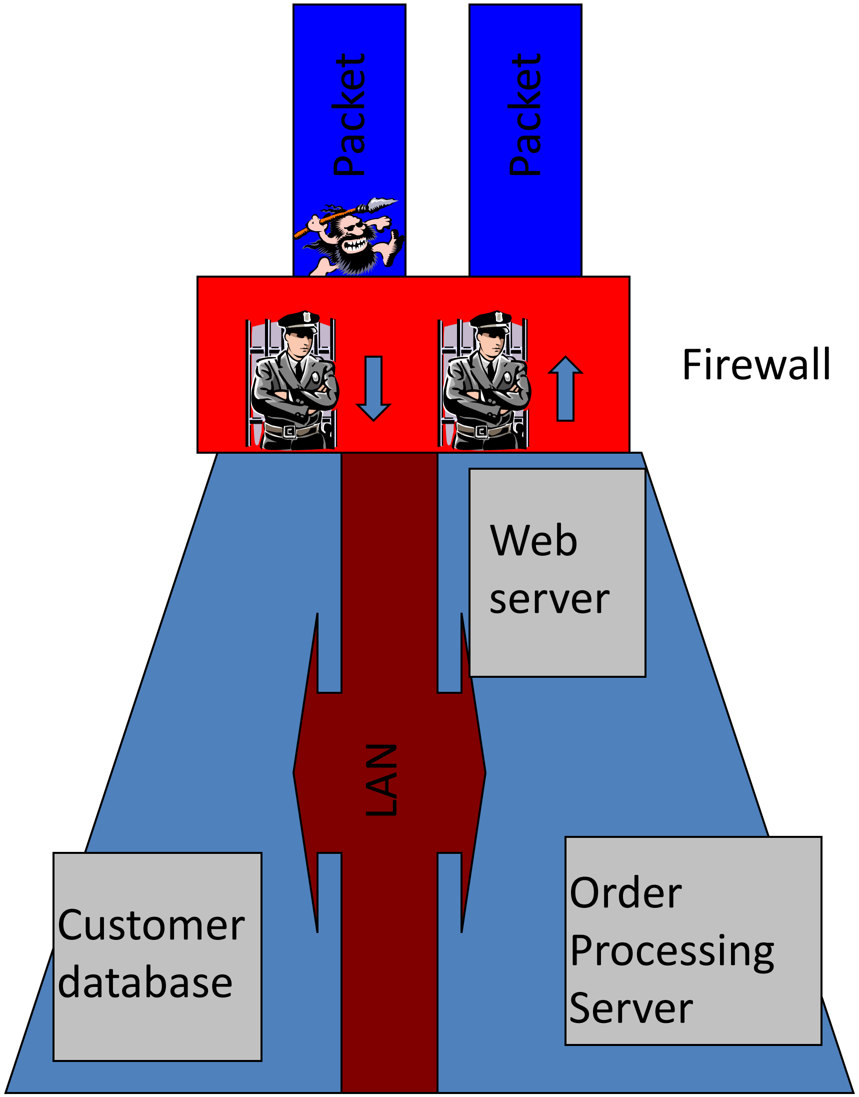
# Look Into a Bad Packet



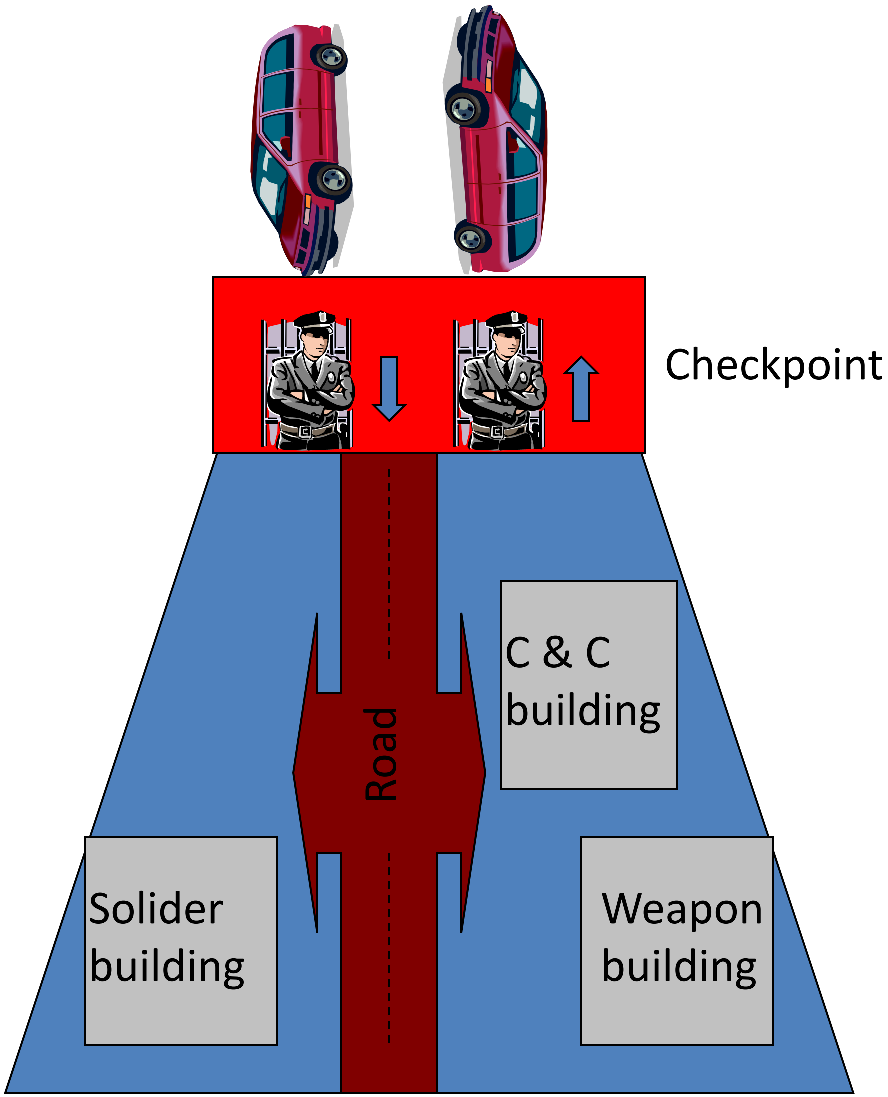
**The Orignal Packet of Code Red:**

```
GET /default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNN%u9090%u6858%ucbd3%u7801%u909
0%u6858%ucbd3%u7801%u9090%u6858%ucbd3%u7801%u9090%u9090%u8190
%u00c3%u0003%u8b00%u531b%u53ff%u0078%u0000%u00=a HTTP/1.0\r\n
```

# A network needs a firewall = a military base needs a guard

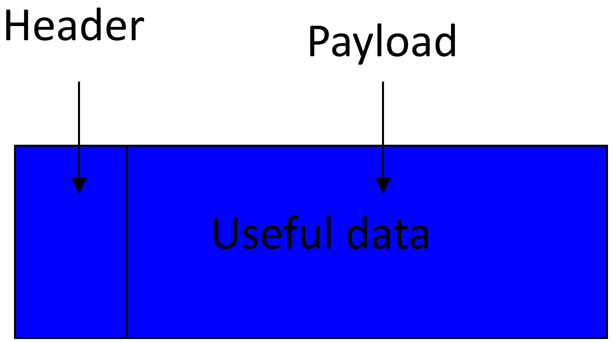


Network



Military base

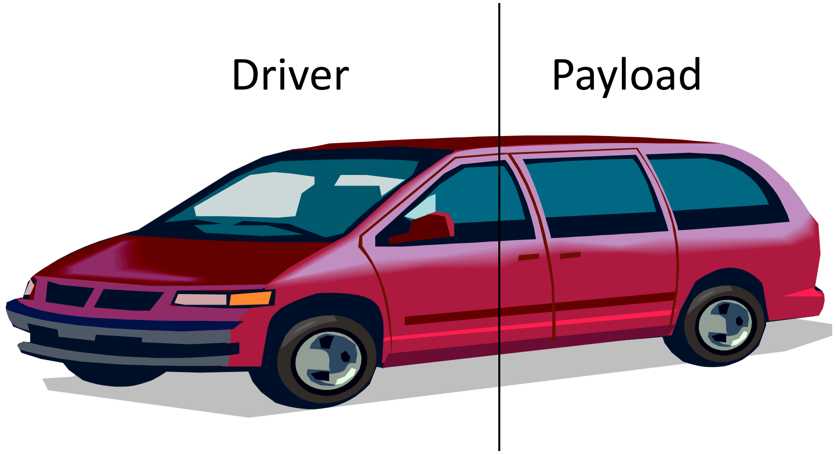
# A packet filtering firewall = a guard that only checks the driver



Header

Payload

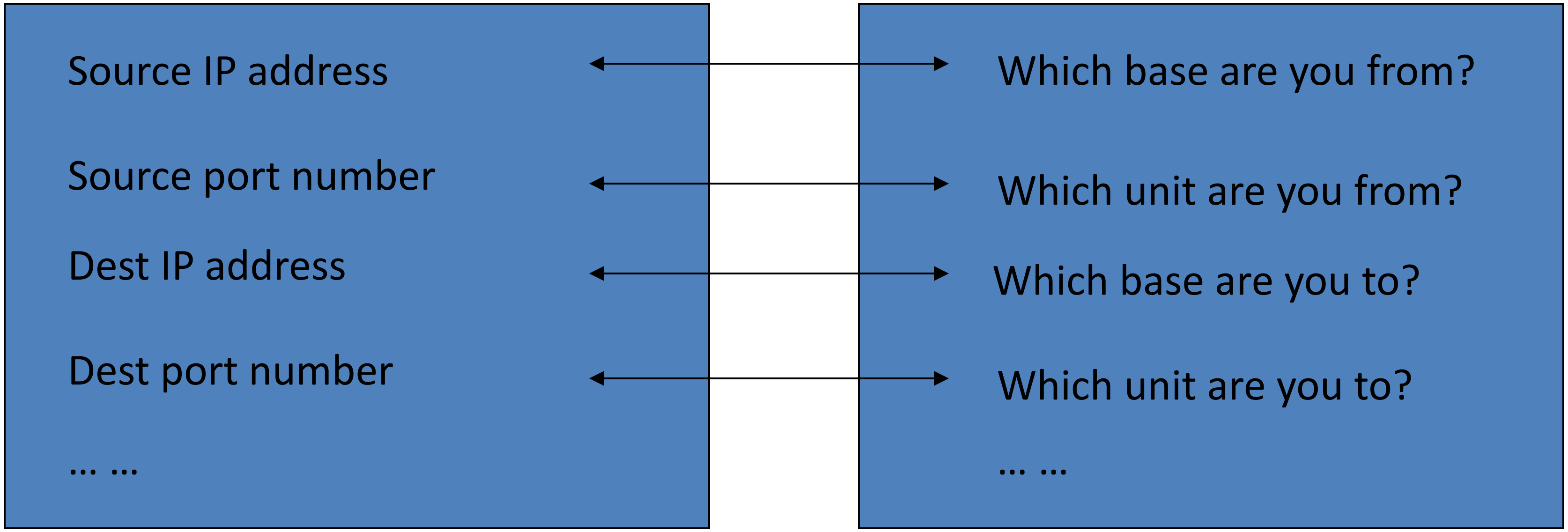
Useful data



Driver

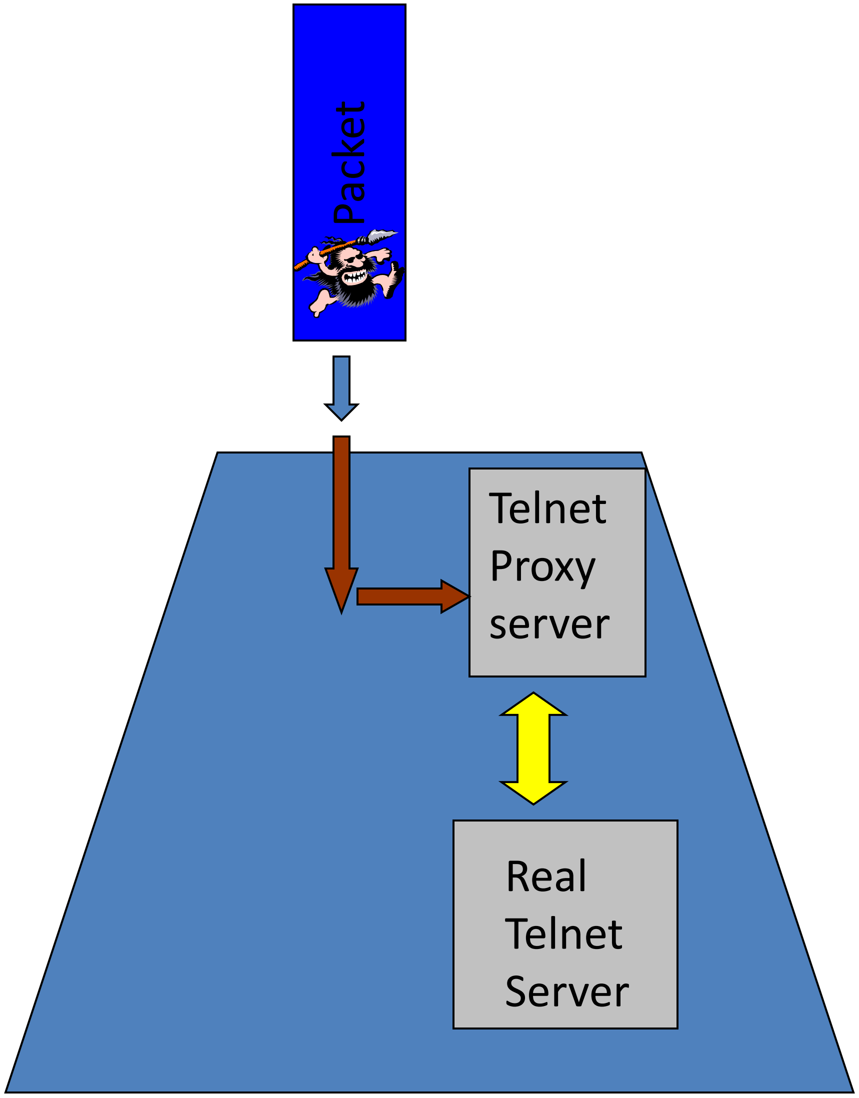
Payload

## So a packet filtering firewall only checks the header



| | |
|---|---|
| Source IP address | Which base are you from? |
| Source port number | Which unit are you from? |
| Dest IP address | Which base are you to? |
| Dest port number | Which unit are you to? |
| … … | … … |

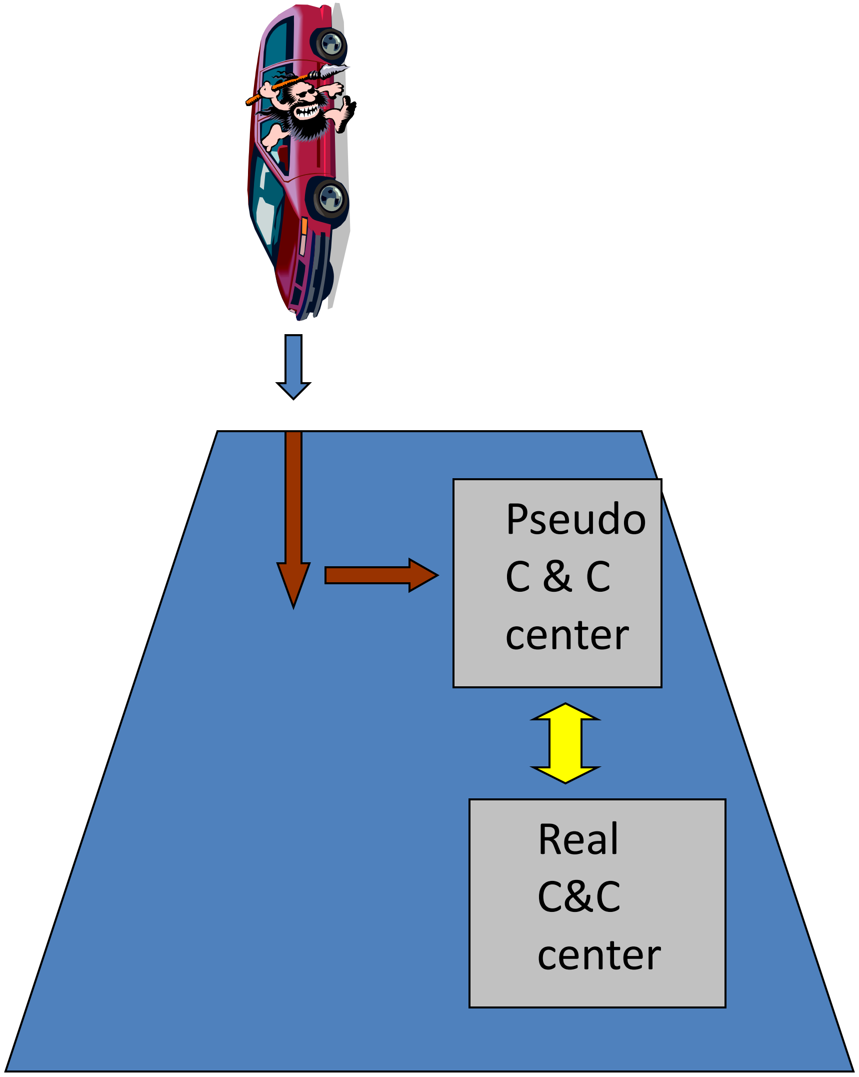
# A proxy firewall = a pseudo center



Telnet Proxy server

Real Telnet Server

Network



Pseudo C & C center

Real C&C center

Military base