Computer Networking (Recap)

- Local Area Networks
- MAC Addresses
- IP Addresses (IPv4 and IPv6)
- Switches vs Routers
- Protocols: ARP, ICMP, DHCP
- Sockets
- TCP vs UDP
- Wireshark

COMP 177, in 45 minutes?

Dive Right In!

Local Area Networks (LANs)

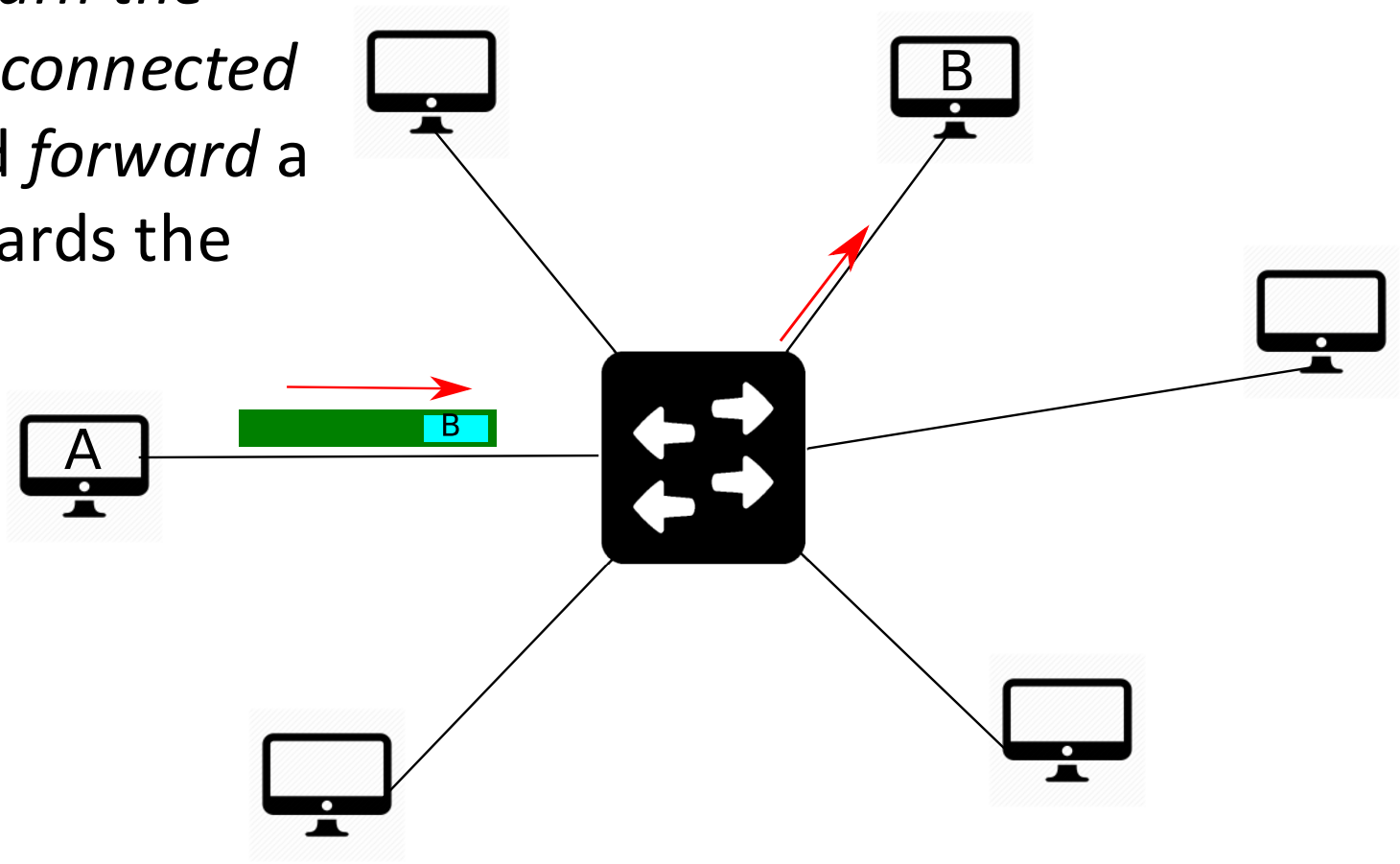
- Physically connected devices that can communicate with each other “directly”
  - All computers in your home?
  - All computers in your office building? (Or floor? Or room?)
- Devices in a LAN use *physical addresses* (property of the hardware) to reach each other
  - **MAC Address** (“Media Access Controller”)
  - 48 bits, 6 groups of 2 hex digits
    - Example: **`08:00:27:A8:69:6C`**

Network Devices: Switches

- ***Switches*** are used to physically interconnect multiple nodes within a LAN
- Packets include the physical addresses of both the sender and the recipient that resides in the LAN
- Switches use these addresses to *learn the location of devices in the LAN* and send the received packet to that recipient
- **Ethernet Switches** use **Ethernet (MAC) Addresses**

Network Devices: Switches

Switches *learn the location of connected devices* and *forward* a packet towards the destination

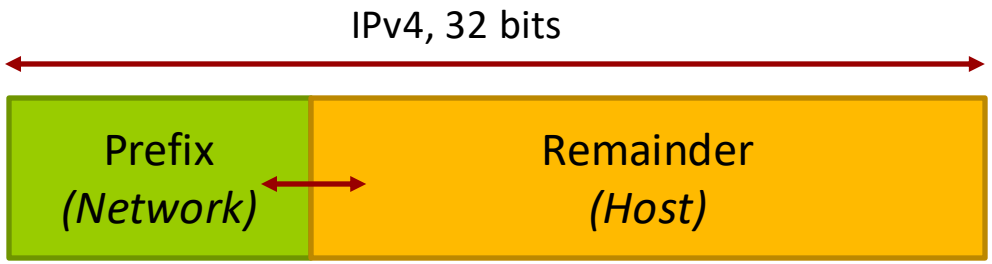


Internet Protocol (IP)

- Ethernet is **not scalable** for global communication
- Internet Protocol (**IP**) is used to join multiple LANs together into a larger *(global)* network
- IP provides for
  - *Universal addressing* of nodes using IP addresses
  - *Routing* of packets using *routing protocols*

IP Addresses

- An IP address consists of two parts:
  - *Network* part (prefix)
  - *Host* (interface) part (remainder)



- The size of the prefix can **vary!**
  - The network part is assigned by the *ISP*
  - The host part is usually configured by the *network administrator*

Multiple Addresses

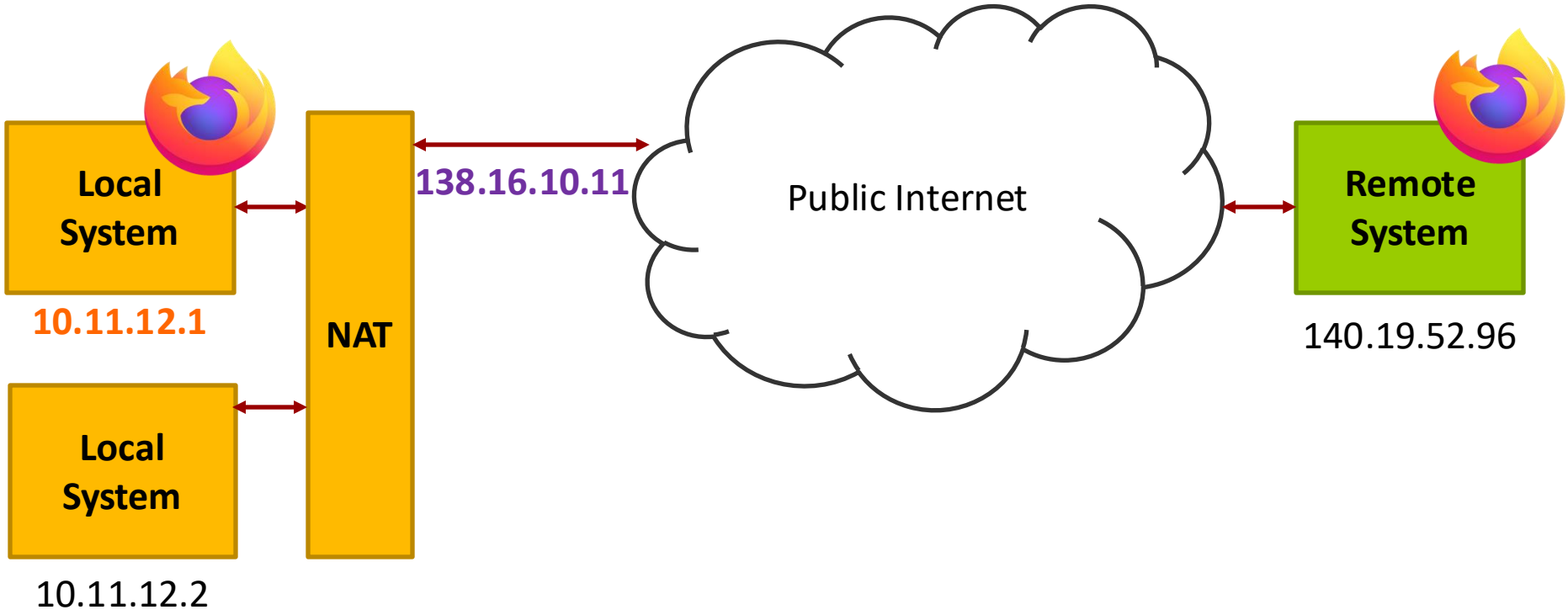
- Sending a packet from one host to another?
  - 4 addresses in play
- Sender
  - Source IP address (e.g., **`50.60.70.80`**)
  - Source MAC address (e.g., **`08:00:27:A8:69:6C`**)
- Receiver
  - Destination IP address (e.g., **`135.80.13.96`**)
  - Destination MAC address (e.g., **`25:00:88:C3:22:75`**)

Private IP Addresses

- 32-bit address space for IPv4 is not enough for today's Internet ($2^{32} \simeq$ 4billion addresses)
- Many of the IP addresses are for internal / private use
  - The address for a corporate file server
  - The address of each interfaces of an internal router
  - The address of the PCs and laptops handed to employees
- Private IP addresses can be used *arbitrary number of times* in different networks
- **Public routers (e.g., ISP's) cannot forward packets with a destination address in the private ranges**
- Private IP address blocks
  - `10.0.0.0/8`
  - `172.16.0.0/12`
  - `192.168.0.0/16`

Network Address Translation (NAT)

- Network Address Translation (NAT) is a capability of routers that enables multiplexing large number of individual hosts behind a single IPv4 public address
- Benefits of NAT
  - Conserves *limited address space of IPv4*
  - Enables a form of *firewall-based security* in LANs
    - Internal devices can *initiate* connections to external devices, but not vice versa



The WhatIsMyIP.com site (and similar) will only see the public IP of your NAT (gateway), not the private IP of your internal device.

Sometimes this is what you want.
Other times it is not.

Network Address Translation (NAT)

- NAT can also be used *within* a single computer system
- Common example?
  Running multiple virtual machines
- Your computer has the IP address assigned by the network
  - Your virtual machines have NAT'ed IP addresses assigned by VMware/VirtualBox
  - VMs are only accessible *from within* your computer

Private IP Addresses

http://xkcd.com/742/

IP Versions

Version	Description
0-3	Unused: Development versions of IP
4	**Current network-layer protocol**
5	Unused: Experimental stream protocol – ST
6	**New network-layer protocol (1996)**
7-9	Unused: Experimental protocols – TP/IX, PIP, TUBA
10-15	Not allocated

Why Replace IPv4?

- The problem
  - IPv4 has ~4.3 billion addresses
  - World has ~6.6 billion people!
    - How many internet-capable devices per person?
- IP address exhaustion
  - Internet will not “collapse”, but new devices / networks will not be able to join(*)

(*) Except via address translation…

Comparison – IPv4 vs IPv6

	IPv4	IPv6
Deployed	1981 *[RFC 791]*	1999 *[RFC ~~2460~~, 8200]*
Address Size	32-bit number	128-bit number

Address Format	Dotted Decimal Notation:
	`192.149.252.76`
		Hexadecimal Notation:
		`3FFE:F200:0234:AB00:`
		`0123:4567:8901:ABCD`

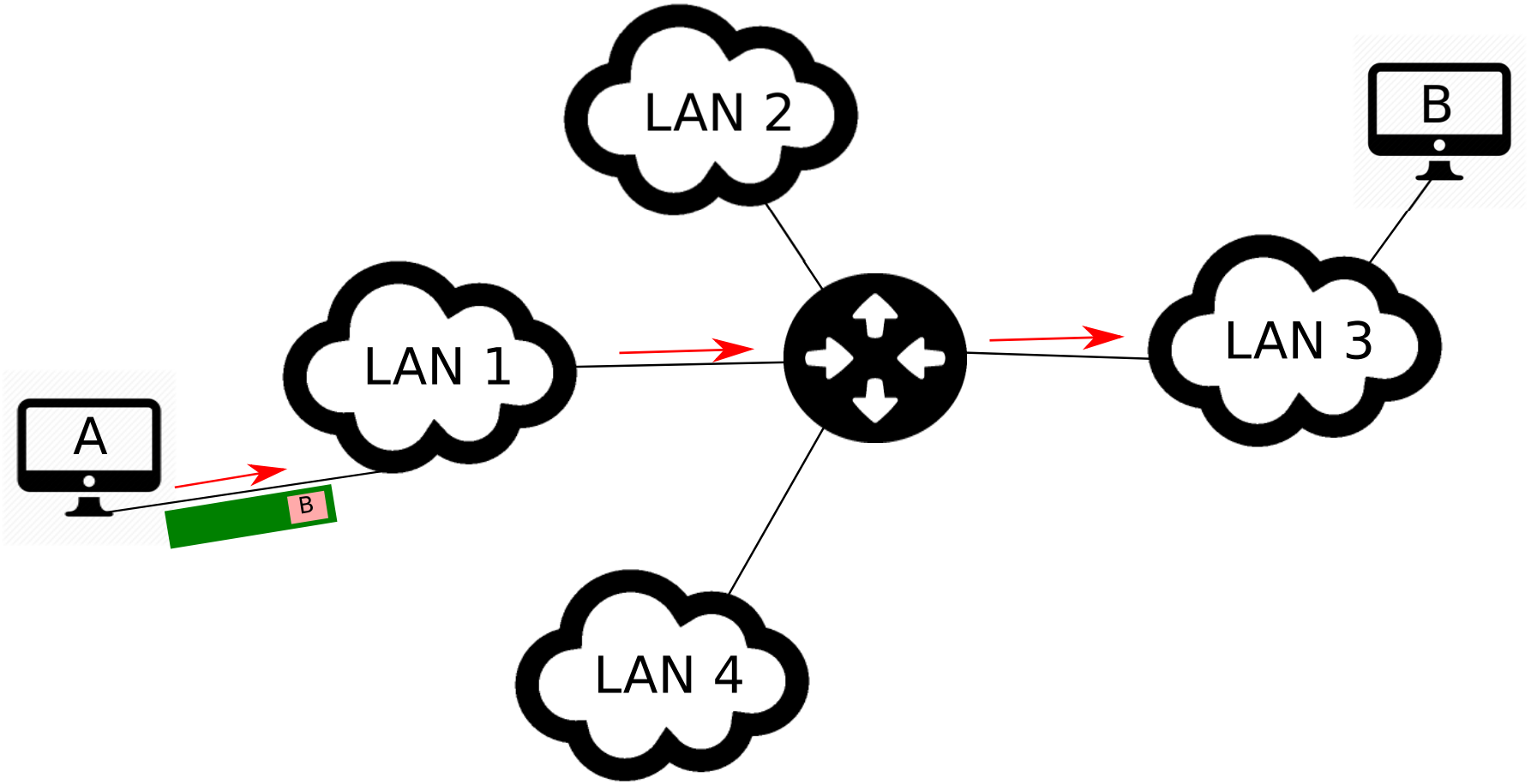
Prefix Notation	`192.149.0.0/24`	`3FFE:F200:0234::/48`
Number of Addresses	2^{32} = ~4,294,967,296 (~4 billion)	**2^{128} = ~340,282,366,** **920,938,463,463,374,** **607,431,768,211,456**

https://biotech.law.lsu.edu/blog/ipv4_ipv6.pdf (ARIN Fact Sheet)

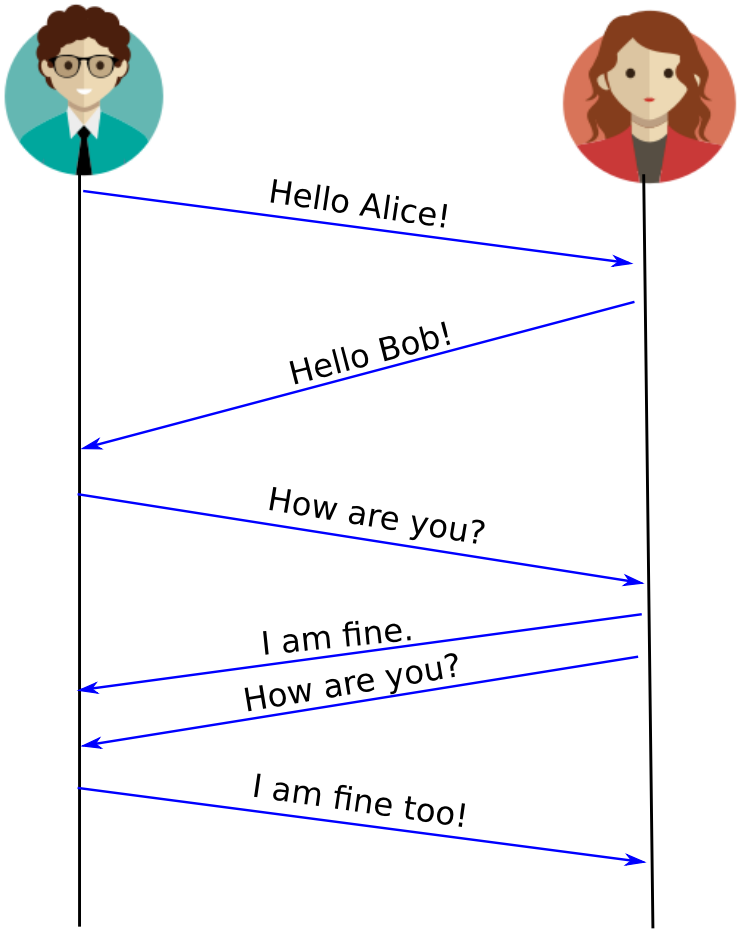
Network Devices: Routers

- ***Routers*** are used to interconnect multiple LANs to each other
- Packets include the logical address of the recipient that resides in a potentially remote LAN
- Routers use this address to forward the received packet toward the recipient
- **IP Routers** use **IP Addresses**
  - *Compare this to switches, which use MAC addresses*

Network Devices: Routers



Protocols



- A protocol is a set of guidelines according to which two entities communicate
- A *networking protocol* is a set of guidelines according to which two devices communicate with each other through the network

Example Protocols – Low Level

Protocol	Purpose
Ethernet	Wired communication across local area network
WiFi	Wireless communication across local area network
Address Resolution Protocol (**ARP**)	Obtaining MAC address from an IP address
Internet Control Message Protocol (**ICMP**)	Diagnostic and error reporting messages
Transmission Control Protocol (**TCP**)	*Reliable* transmission of stream data from one host to another
User Datagram Protocol (**UDP**)	*Unreliable* transmission of packet data from one host to another

Example Protocols – Application Level

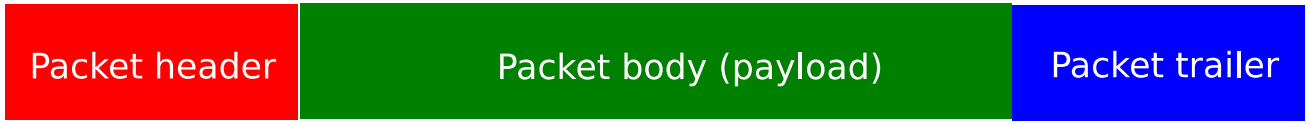
Protocol	Purpose
Domain Name System (**DNS**)	Obtaining an IP address given a hostname
Dynamic Host Configuration Protocol (**DHCP**)	Host can obtain its IP address, subnet, and next-hop router dynamically
Hyper Text Transport Protocol (**HTTP**/HTTPS)	Communication between web browsers and web servers
Secure Shell (**SSH**)	Encrypted command line and file transfers with a remote system

Network Protocols

- Protocols need to be standardized for interoperability
- Two major sources for networking protocol specifications:
  - **"Request For Comments"** documents (**RFCs**) administered by Internet Engineering Task Force (IETF)
    - Freely available at ietf.org
    - Examples: TCP, UDP, IP, HTTP
  - Institute of Electrical and Electronics Engineers (**IEEE**) standards
    - Accessible to members ($$)
    - Examples: 802.3 (Ethernet), 802.11 (WiFi)

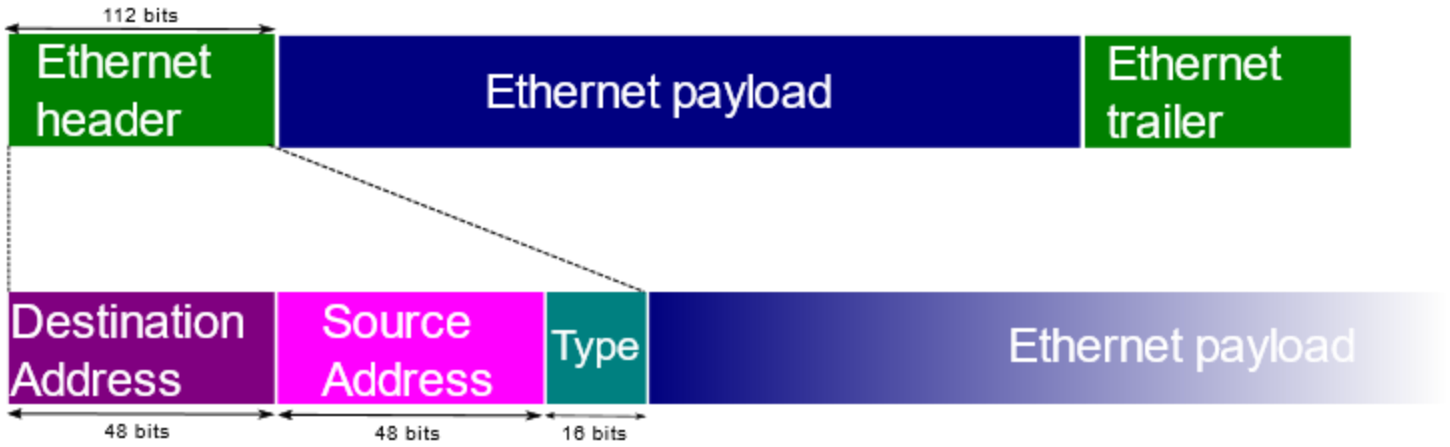
Protocol Packets

- Each protocol defines the structure (syntax) of the packets that need to be communicated
- Different protocols have different packet structure
- Generally, a packet may consist of the following components:
  - Packet header
  - Packet body (payload)
  - Packet trailer



Example Packet – Ethernet

- Ethernet header has three fields:
  - *Dest. MAC address* (48 bits) – physical addr. of NIC in *receiving* host
  - *Source MAC address* (48 bits) – physical addr. of NIC in *sending* host
  - *Type* (16 bits) stores the upper layer protocol, i.e., the protocol used in the Ethernet payload
    - IPv4: `0x0800` IPv6: `0x86DD` ARP: `0x0806`



Address Resolution Protocol

- Find link layer address given a network layer address
  - What is the **Ethernet address** for a given **IP address**?
- Every IP node (hosts and routers) has an ARP table
  - Mapping from IP to Ethernet addresses on their LAN
  - May be incomplete
  - Can include both static and dynamic entries

Dynamic ARP Entries

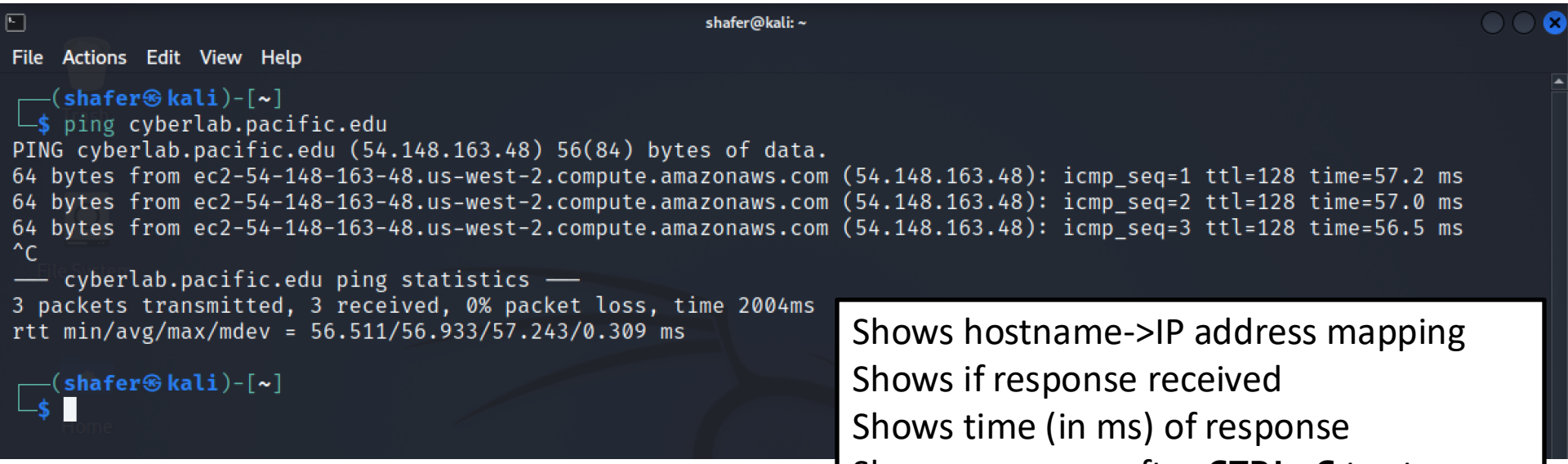
- Systems "discover" IP → Ethernet address mappings, as needed
- Each entry has an IP address, an Ethernet address, and a timeout (typically around 1 minute)
- ARP messages are **broadcast** on the LAN to discover mappings
  - All computers on the network receive the ARP requests
- Hosts learn IP → Ethernet address mappings
  - ARP responses are stored in ARP tables
  - ARP requests are stored in ARP tables (whether the host is the target or not!)

Internet Control Message Protocol

- One of the core protocols in the Internet
- Primarily used to communicate errors among routers and hosts
  - IP datagram errors
  - Communicate routing information/errors
  - Communicate diagnostics
- Not (typically) used by applications
  - Applications communicate application-level errors using higher level protocols
  - Ping and traceroute are the exceptions

Example Application - Ping

- Purpose: Does the hostname exist? Is the target up? Is the network working?
- Usage: `ping [hostname/IP]`



Shows hostname->IP address mapping
Shows if response received
Shows time (in ms) of response
Shows summary after **CTRL-C** to stop

Network Configuration

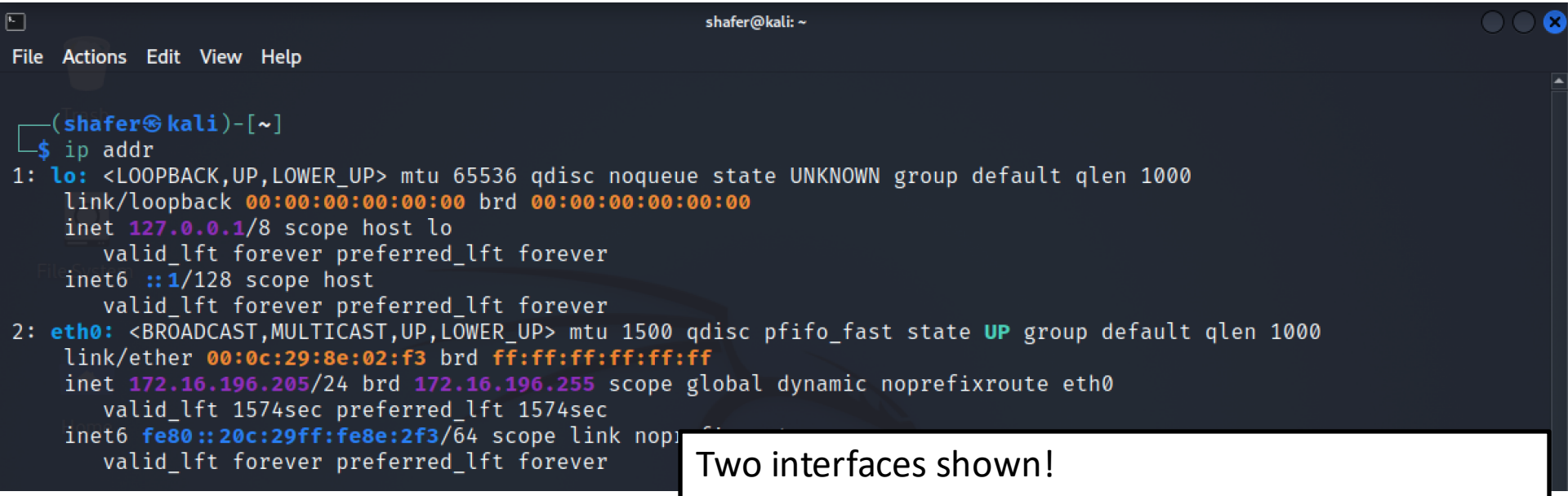
- How does a host get its network interface configured?
  - IP address
  - Network mask
  - Default gateway
  - DNS servers
  - …

Dynamic Host Configuration Protocol (DHCP)

- Goals of DHCP
  - Plug and play!
  - Allow host to *dynamically* obtain its IP address from network server when it joins network
  - Allow host to renew its lease on in-use address
  - Allow reuse of addresses (if you disconnect your host, someone else can use that address)

Example Application - IP

- Purpose: What is my IP address?
- Usage: `ip addr`



Two interfaces shown!

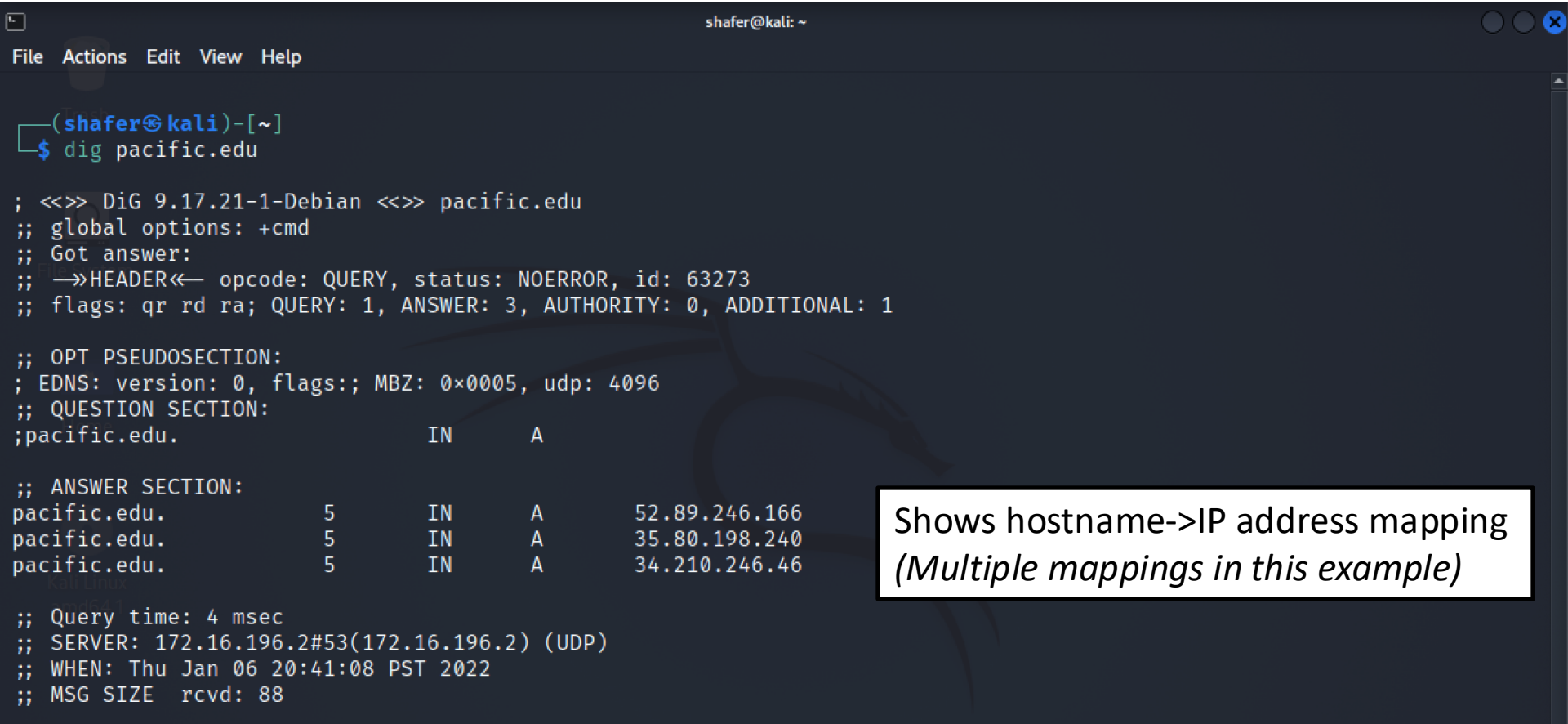
- "**lo**" (local loopback) with address **127.0.0.1**
- "**eth0**" with address **172.16.196.205**

Domain Name System (DNS)

- Global distributed database
- Primary goal: Convert hostname (`engineering.pacific.edu`) to IP address
  - .edu is top-level domain
  - “pacific” belongs to .edu
  - “engineering” belongs to “pacific”
  - Hierarchical! Read from right to left

Example Application - Dig

- Usage: `dig [hostname]`



Internet Transport Protocols

TCP Service

- Connection-oriented
  - Setup required between client and server processes
- Reliable transport between sending and receiving process
- Flow control
  - Sender won't overwhelm receiver
- Congestion control
  - Throttle sender when network overloaded
- Does not provide
  - Timing, minimum throughput guarantees, security

UDP Service

- Unreliable data transfer between sending and receiving process
- Does not provide
  - Connection setup
  - Reliability
  - Flow control
  - Congestion control
  - Timing
  - Throughput guarantee
  - Security

Why bother with UDP then?

Application-Layer Protocol

- Sockets just allow us to send raw messages between processes on different hosts
  - Transport service takes care of moving the data
- **What** exactly is sent is up to the application
  - An application-layer protocol
  - HTTP, IMAP, Zoom, Discord, etc…

Wireshark

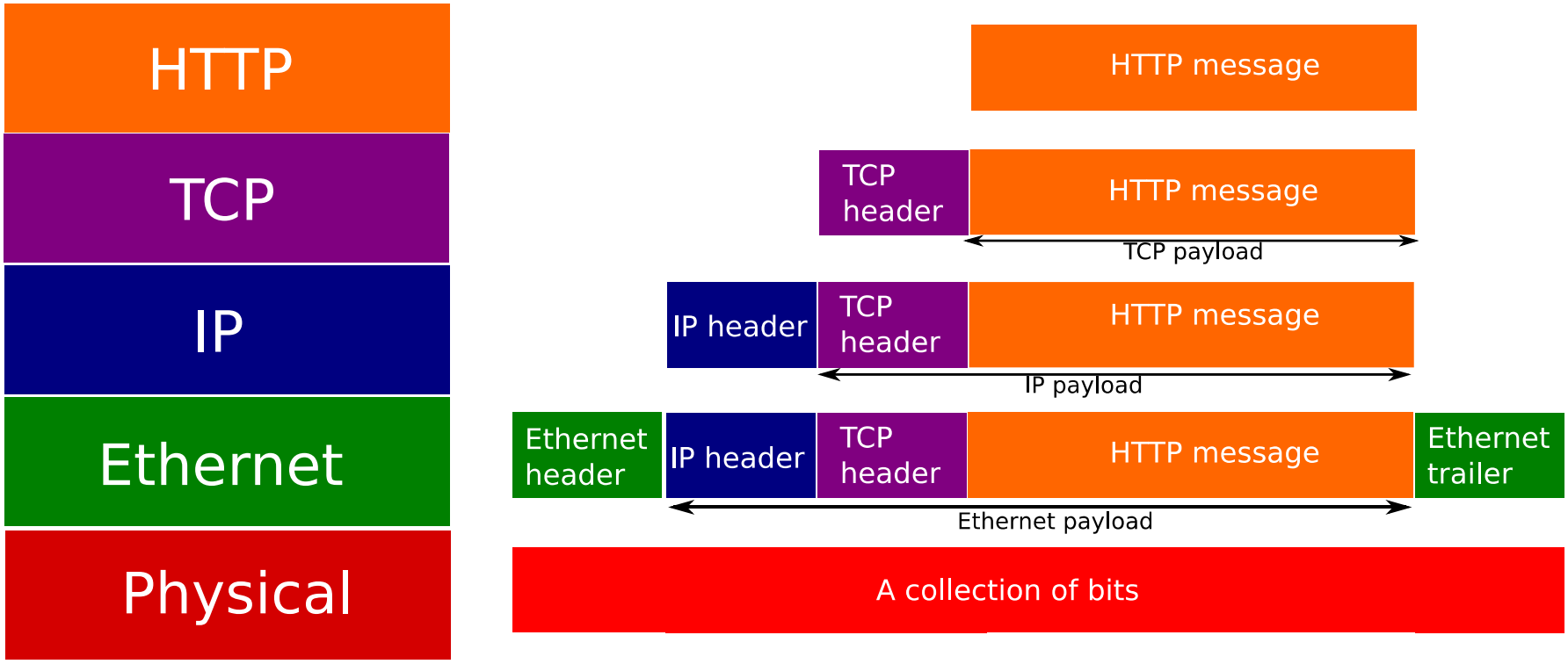
- To *understand how protocols* work, it is helpful to
  - Observe the sequence of packets communicated between network nodes
  - Study the packet details and how protocols work in practice
  - Cause the protocol to do a specific action and check out the result
- *Wireshark* is a free tool that provides such services
  - Supports all major operating systems

https://www.wireshark.org/

Reading & Parsing Packets

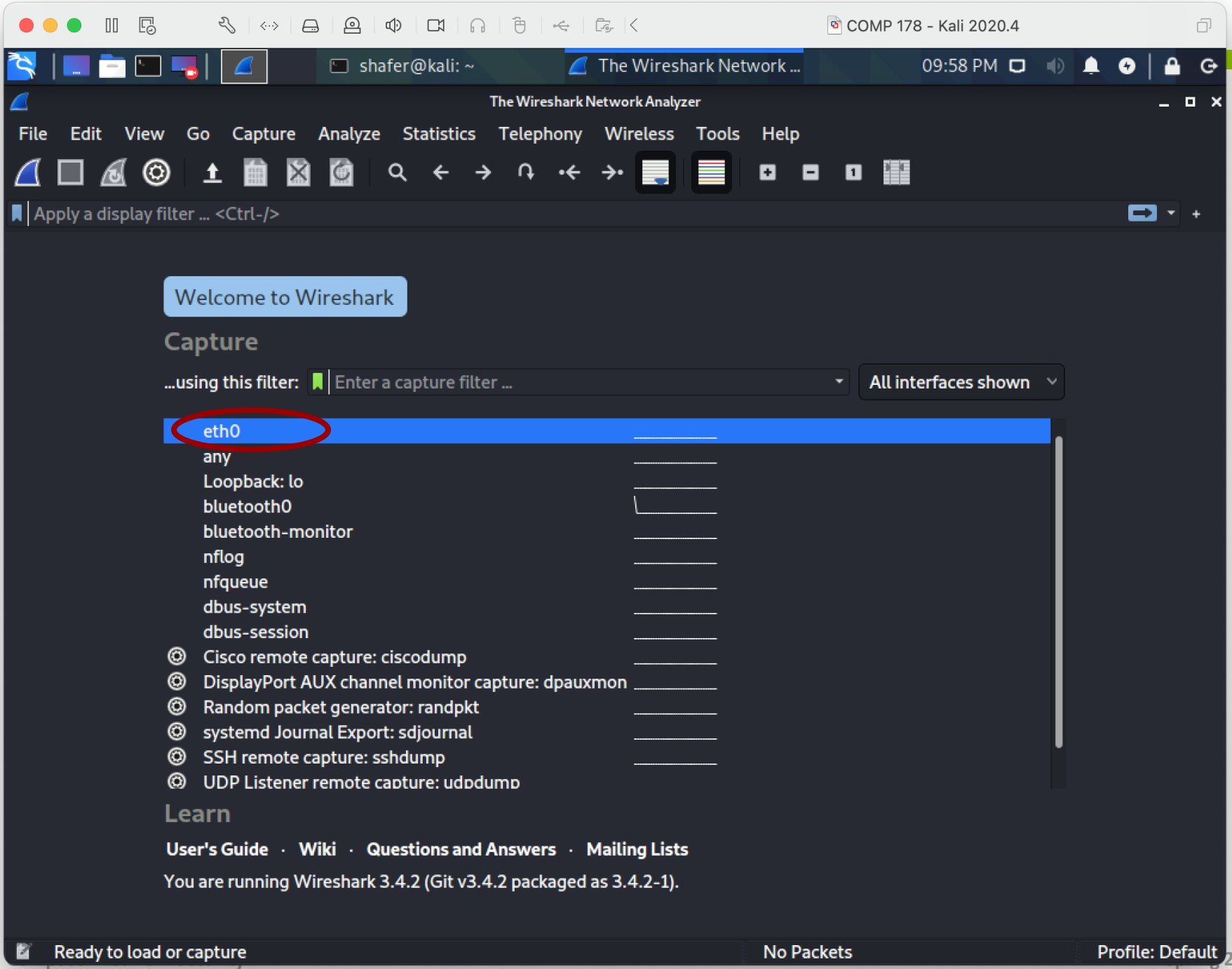
- Wireshark can capture a packet and parse it into
  - Different protocols headers
  - Different fields in each protocol header
  - … *plus* report some *meta-data* about the fields
    - This does not appear directly in the packet header but is based on Wireshark analysis of this packet or even a sequence of packets

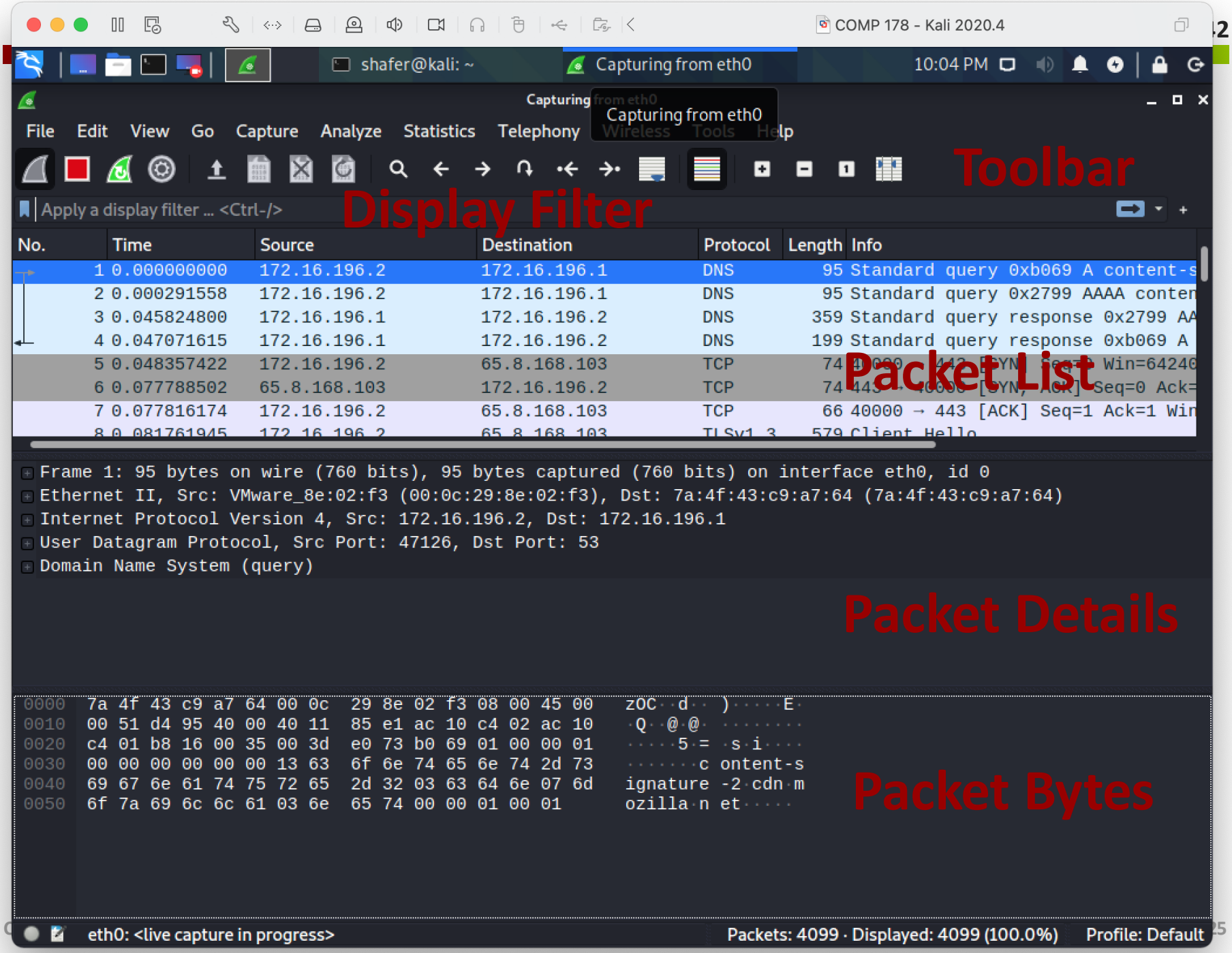
Reading & Parsing Packets



Wireshark Packet Capture (Kali)

- Wireshark is pre-installed in Kali
- Configure so non-root users have permission to capture packets (insert your Linux username into the second command)
  - `$ sudo dpkg-reconfigure wireshark-common`
    `$ sudo usermod -a -G wireshark USERNAME`
  - `$ sudo reboot`
- Run Wireshark
  - `$ wireshark &`
- Next, interfaces are listed. Selecting one of the interfaces will start capturing packets on that interface.


COMP 178 - Kali 2020.4
shafer@kali: ~
The Wireshark Network ...
09:58 PM
The Wireshark Network Analyzer
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Welcome to Wireshark
Capture
...using this filter: Enter a capture filter ...
All interfaces shown
eth0
any
Loopback: lo
bluetooth0
bluetooth-monitor
nflog
nfqueue
dbus-system
dbus-session
Cisco remote capture: ciscodump
DisplayPort AUX channel monitor capture: dpauxmon
Random packet generator: randpkt
systemd Journal Export: sdjournal
SSH remote capture: sshdump
UDP Listener remote capture: udpdump
Learn
User's Guide · Wiki · Questions and Answers · Mailing Lists
You are running Wireshark 3.4.2 (Git v3.4.2 packaged as 3.4.2-1).
Ready to load or capture
No Packets
Profile: Default


COMP 178 - Kali 2020.4
shafer@kali: ~
Capturing from eth0
10:04 PM
Capturing from eth0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Toolbar
Apply a display filter ... <Ctrl-/>
Display Filter
No. Time Source Destination Protocol Length Info
1 0.000000000 172.16.196.2 172.16.196.1 DNS 95 Standard query 0xb069 A content-s
2 0.000291558 172.16.196.2 172.16.196.1 DNS 95 Standard query 0x2799 AAAA conten
3 0.045824800 172.16.196.1 172.16.196.2 DNS 359 Standard query response 0x2799 AA
4 0.047071615 172.16.196.1 172.16.196.2 DNS 199 Standard query response 0xb069 A
5 0.048357422 172.16.196.2 65.8.168.103 TCP 74 40000 → 443 [SYN] Seq=0 Win=64240
6 0.077788502 65.8.168.103 172.16.196.2 TCP 74 443 → 40000 [SYN, ACK] Seq=0 Ack=
7 0.077816174 172.16.196.2 65.8.168.103 TCP 66 40000 → 443 [ACK] Seq=1 Ack=1 Win
8 0.081761945 172.16.196.2 65.8.168.103 TLSv1.3 579 Client Hello
Packet List
Frame 1: 95 bytes on wire (760 bits), 95 bytes captured (760 bits) on interface eth0, id 0
Ethernet II, Src: VMware_8e:02:f3 (00:0c:29:8e:02:f3), Dst: 7a:4f:43:c9:a7:64 (7a:4f:43:c9:a7:64)
Internet Protocol Version 4, Src: 172.16.196.2, Dst: 172.16.196.1
User Datagram Protocol, Src Port: 47126, Dst Port: 53
Domain Name System (query)
Packet Details
0000 7a 4f 43 c9 a7 64 00 0c 29 8e 02 f3 08 00 45 00 zOC··d··)·····E·
0010 00 51 d4 95 40 00 40 11 85 e1 ac 10 c4 02 ac 10 ·Q··@·@· ····
0020 c4 01 b8 16 00 35 00 3d e0 73 b0 69 01 00 00 01 ·····5·= ·s·i····
0030 00 00 00 00 00 00 13 63 6f 6e 74 65 6e 74 2d 73 ·····c ontent-s
0040 69 67 6e 61 74 75 72 65 2d 32 03 63 64 6e 07 6d ignature -2·cdn·m
0050 6f 7a 69 6c 6c 61 03 6e 65 74 00 00 01 00 01 ozilla·n et·····
Packet Bytes
eth0: <live capture in progress>
Packets: 4099 · Displayed: 4099 (100.0%)
Profile: Default

Deadlines

- **Lab 1 – VM Setup**
  - Due Jan 21st at 11:59pm
- **Lab 2 – Reconnaissance**
  - Due Jan 29th at 11:59pm
- **TA**
  - Vivek Kumar Maheshwari
  - SOECS Student Support Center
  - Hours TBD