ITP20004 – Open-Source Software Labs

Computer Networks

Charmgil Hong

charmgil@handong.edu

Spring, 2023

Handong Global University

Announcements

- Weekly schedule

Week	Mon		Week	Thur	
1	Course overview, motivation, administrivia		1	CPR: C Programming Reinforcement - Functions	
2	Computer organization and Linux environment (1)		2	CPR: C Programming Reinforcement - Strings	
3	Computer organization and Linux environment (2)		3	CPR: C Programming Reinforcement - User-defined types, and memory allocation	
4	Basic Linux commands + Writing code on Linux (vim)		4	Getting started with Linux / Hands-on Linux command-line tools	
5	More Linux commands		5	CPR: C Programming Reinforcement - Understanding compilation and build process	
6	Project management (1)	**Proj 1 출제**	6	Project management (2)	
7	-		7	Project: BASIC interpreter	Project 1
8	-	Tuesday night: Midterm exar	8	Project QnA (Optional)	
9	CPR: C Programming Reinforcement - Accessing files and		9	Debugging with GDB + Unit testing with gtest	**Project 1 due: Week#9 Saturday**
10	Shell script			Shell script	
11	Code review GNU utilities	**Proj 2 출제**	12	Writing an application in C	
12	Github and open-source community			Using Github	**Project 2 due: Week#12 Saturday**
13	Computer network basics		10	Linux network commands	
14	Project: Text-based Game		14	Socket programming	
15	Project: Multi-user game	**Proj 3 출제**	15	Project: Multi-user game	
16	Final exam		16		**Project 3 due: Week#16 Saturday**

Announcements

- Team assignment for Weeks 11-16

학번	이름	학번	이름	학번	이름	학번	이름
22	곽철호	20	유승준	18	정현준	21	쉘튼
18	임건호	18	조성민	21	송영은	22	서종현
20	비보시놉 아잣	21	김연희	20	정성호	19	이지명
18	최정겸	20	정지원	19	유건민	21	서준예
20	이상현	20	이준형	18	마석재	22	반대준
21	조유진	21	사우 지아 유인	17	김홍찬	22	이채연
22	윤유원	18	현승준	20	윤예람	20	김가현
22	이온유	20	송산	18	김두환	22	황찬영
18	송민준	20	나예원	21	이선환	20	김승환
20	방석민	21	최지안	18	박현우	20	김유겸

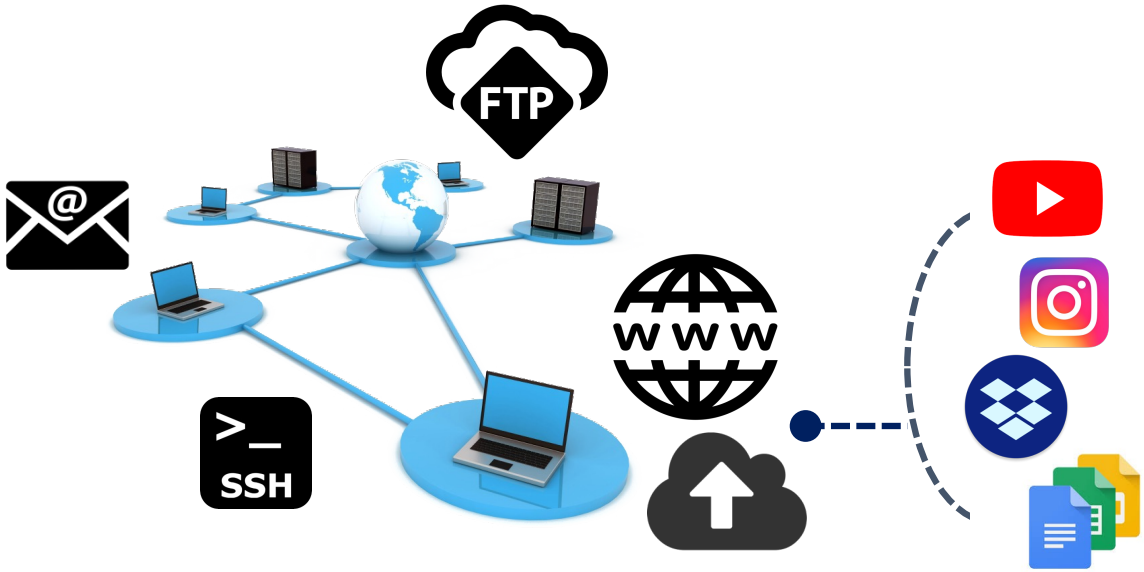
Announcements

- For each lab
  - Before a lab, every student submits a pre-lab report (worksheet-type assignment) – **individual work**
  - After a lab, each team sees and reports to the TA with the results – **team work**

- Up-coming schedule
  - We have a post-lab session on May 23 (Week #13) and May 30 (Week #14)

Network Services

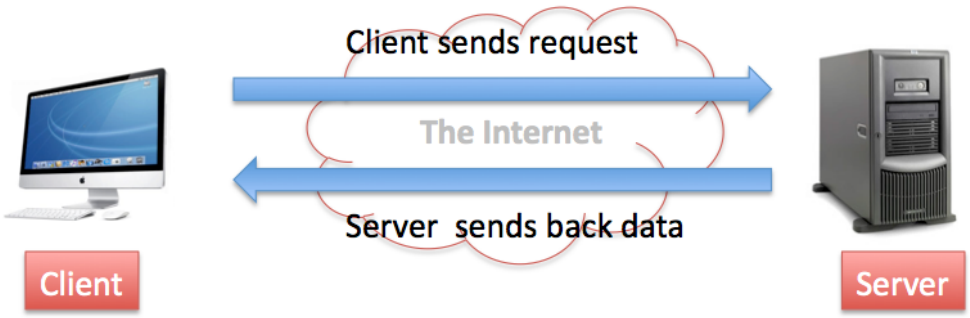
- Network services: Applications running over computer networks that provide data storage, manipulation, presentation, communication or other capability
  - Often implemented using a client-server or peer-to-peer architecture based on network protocols
  - *E.g.*, World Wide Web (WWW), E-mails, file transfer, cloud services, ssh, …



Network Services

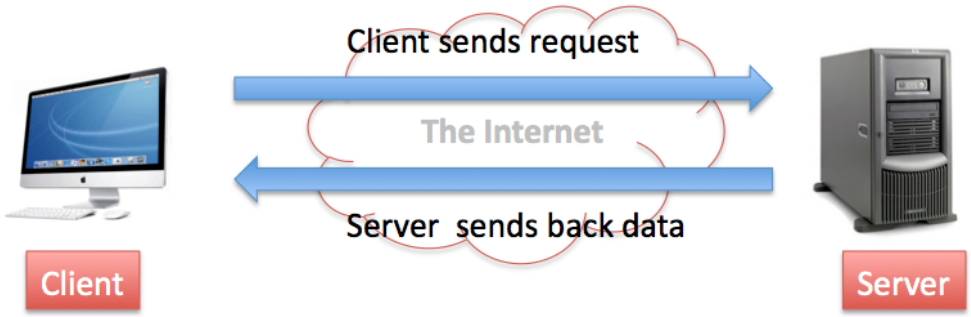
- Informal definitions
  - Applications running on networks
  - A capability that facilitates a network operation

- Client-server model
  - Centralized communication model
  - A network service is typically provided by a server
  - Clients request and receive the service
  - Client and server systems are communicating over a computer network or on the same computer



Communication Model

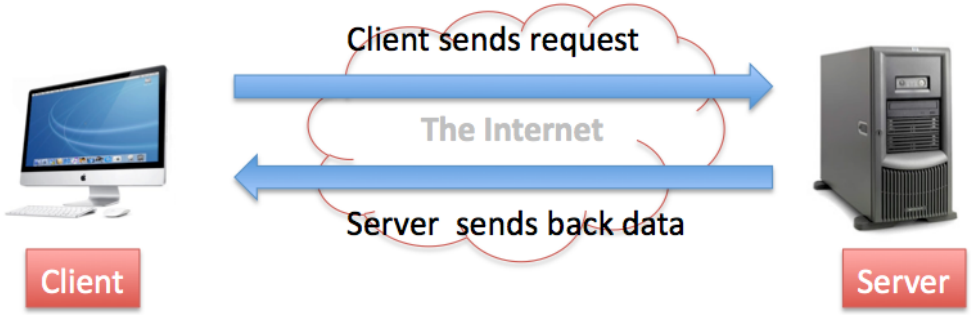
- Client-server model



Clients	Server
- Always initiates requests to servers - Waits for replies - Receives replies - Usually connects to a small number of servers at one time - Usually interacts directly with end-users	- Always wait for a request from a client - Serves clients' requests and replies to the clients - A server may communicate with other servers to serve a client request - End-users typically do not interact directly with a server

Communication Model

- Client-server model



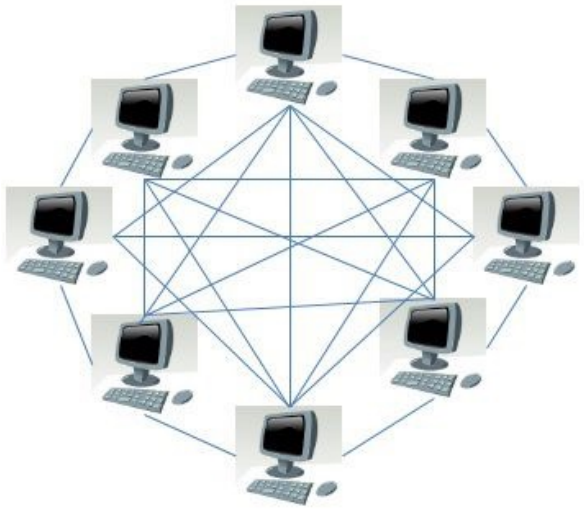
  - Clients: Web browsers, E-mail clients, only chat clients, …
  - Servers: Web servers, database servers, E-mail servers, file servers, print servers, …

Communication Model

- Client-server model
  - Advantages
    - Easier to maintain
      - *E.g.*, possible to replace, repair, upgrade, or relocate a server while its clients remain unaware and unaffected
    - Easier to provide better security
      - All the data is stored on the servers
  - Disadvantages
    - Increased network traffic
    - Unbalanced workload distribution (servers take care of all tasks)
    - Servers could be single points of failure
  - *C.f.*, Standalone machine, Peer-to-peer (P2P) model

Communication Model

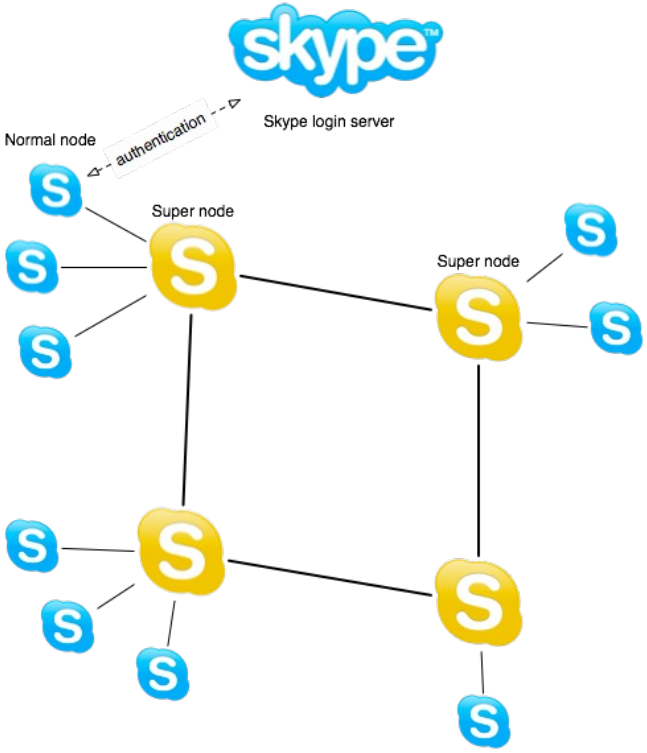
- Peer-to-peer (P2P) model
  - Decentralized communication model
  - Every node (device) in a P2P network can function as both a server and a client
  - Increasing number of clients/servers does not create a bottleneck
    - *C.f.*, Client-server model



Peer-to-Peer Network Model

Communication Model

- Peer-to-peer (P2P) model
  - Frequently used for:
    - Communication in massive parallel computing environments
    - Distributed storage and other functions
    - Media sharing – often associated with software piracy and copyright violation
      - *E.g.*, Napster, Soribada, Skype



* Image src: https://crypto.stanford.edu/cs294s/projects/skype.html

A Network Service Example – Domain Name System

- When you type in *google.com…*
  - Domain name systems translate domain names to IP addresses

Google

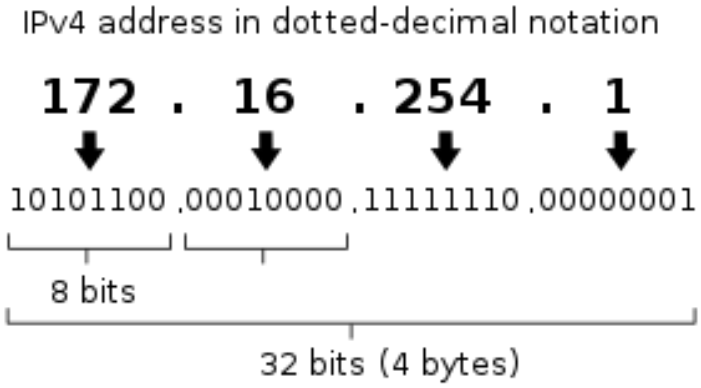
How do I get to google.com?

Computer A

Primary DNS of A

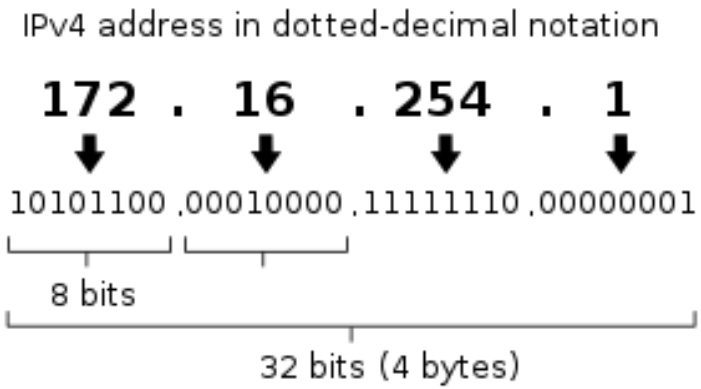
A Network Service Example – Domain Name System

- IP address (Internet Protocol address): a numerical label assigned to each device on network
  - An identifier of a device on network



A Network Service Example – Domain Name System

- IP address (Internet Protocol address): a numerical label assigned to each device on network
  - An identifier of a device on network



A Network Service Example – Domain Name System

- When you type in *google.com…*
  - Domain name systems translate domain names to IP addresses

Google

How do I get to google.com?

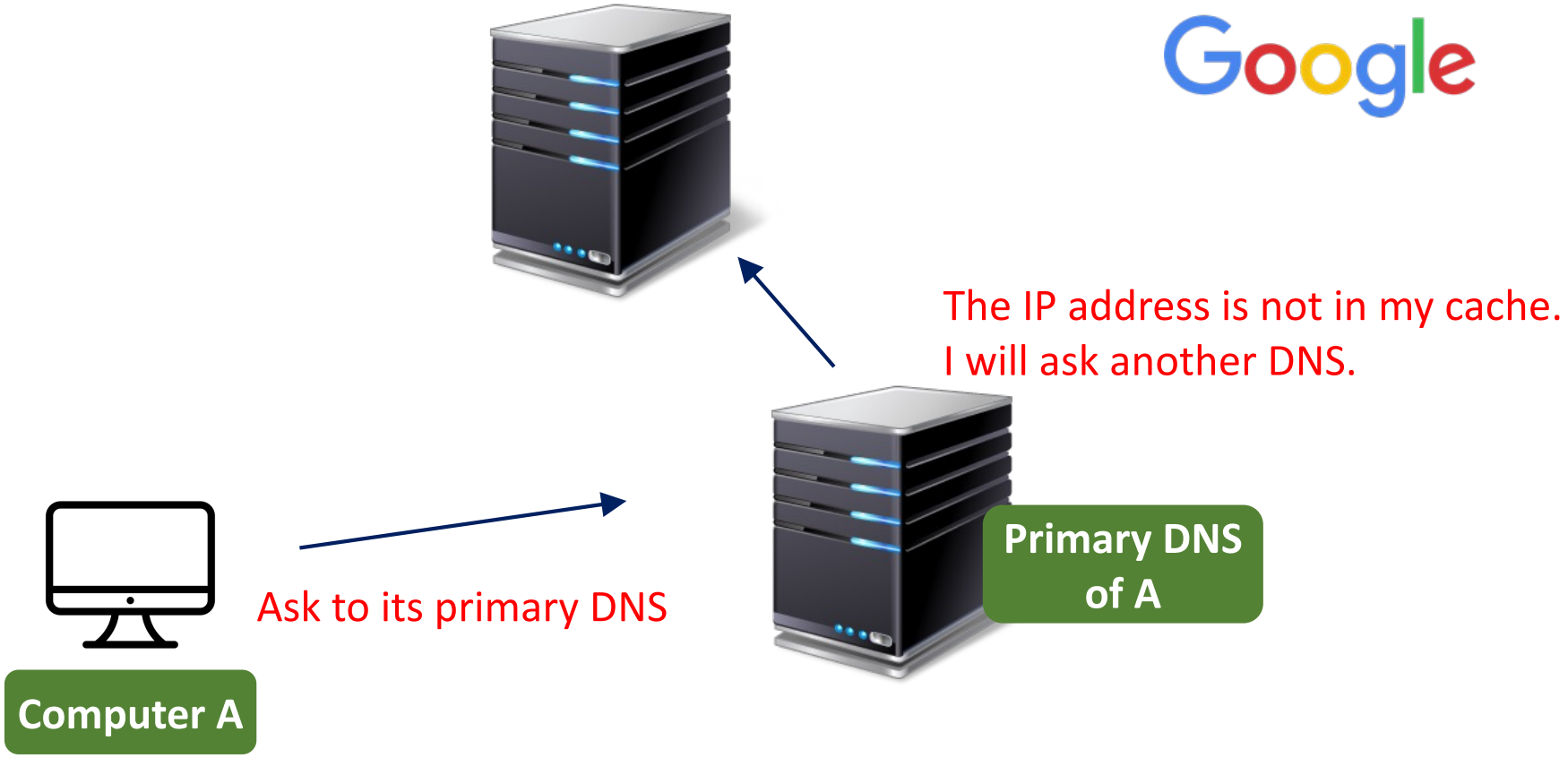
Computer A

Ask to its primary DNS

Primary DNS of A

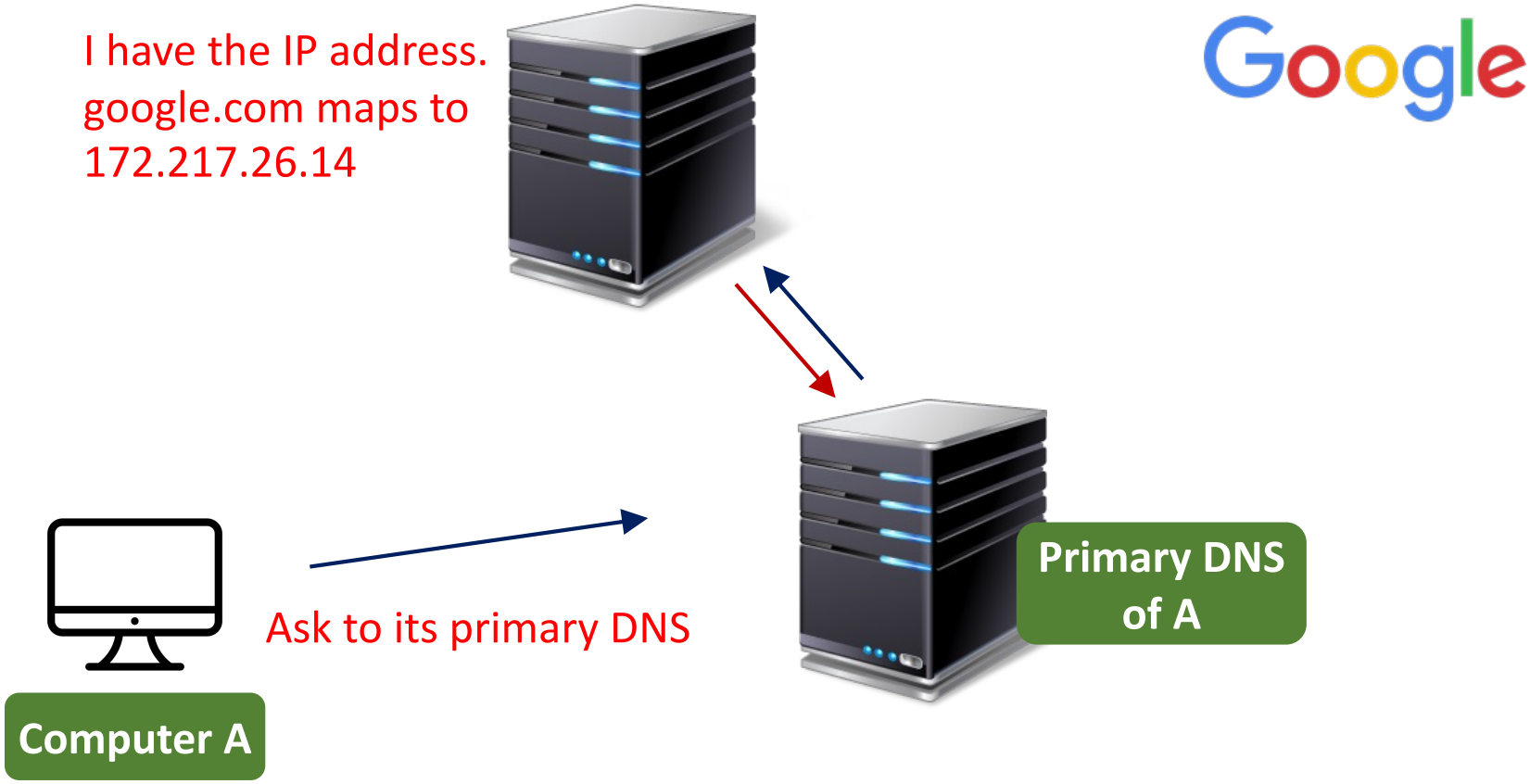
A Network Service Example – Domain Name System

- When you type in *google.com…*
  - Domain name systems translate domain names to IP addresses



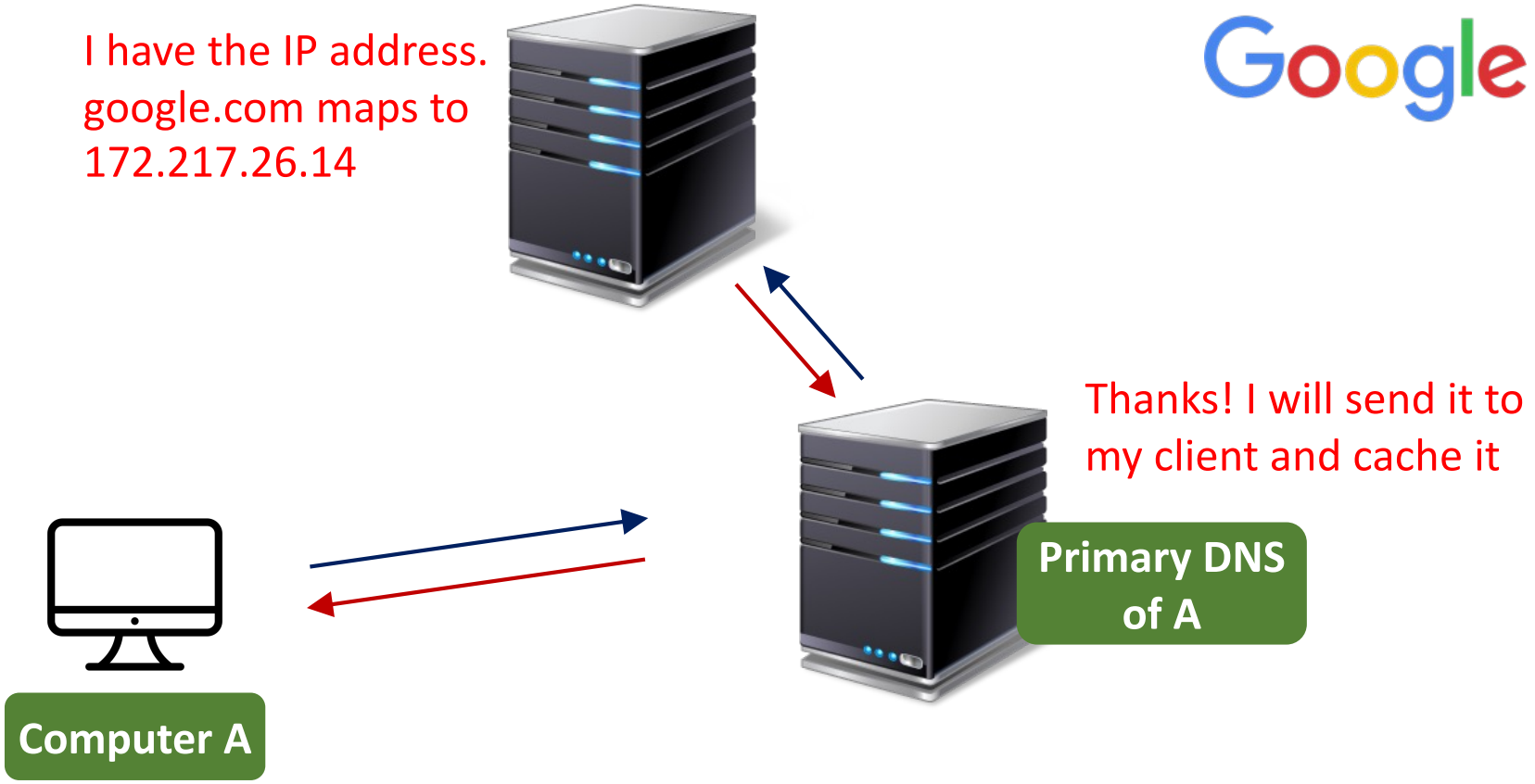
A Network Service Example – Domain Name System

- When you type in *google.com…*
  - Domain name systems translate domain names to IP addresses



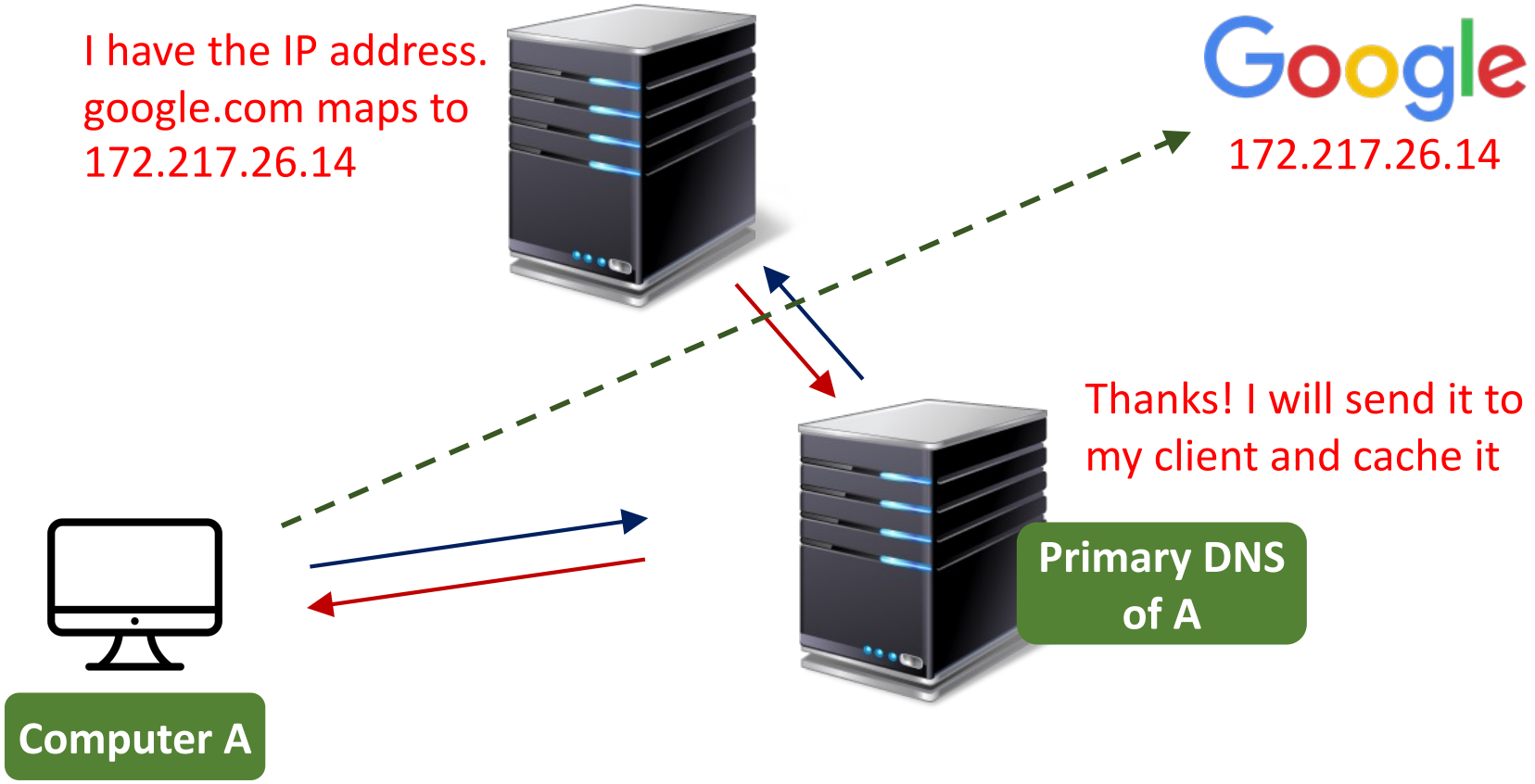
A Network Service Example – Domain Name System

- When you type in *google.com…*
  - Domain name systems translate domain names to IP addresses



A Network Service Example – Domain Name System

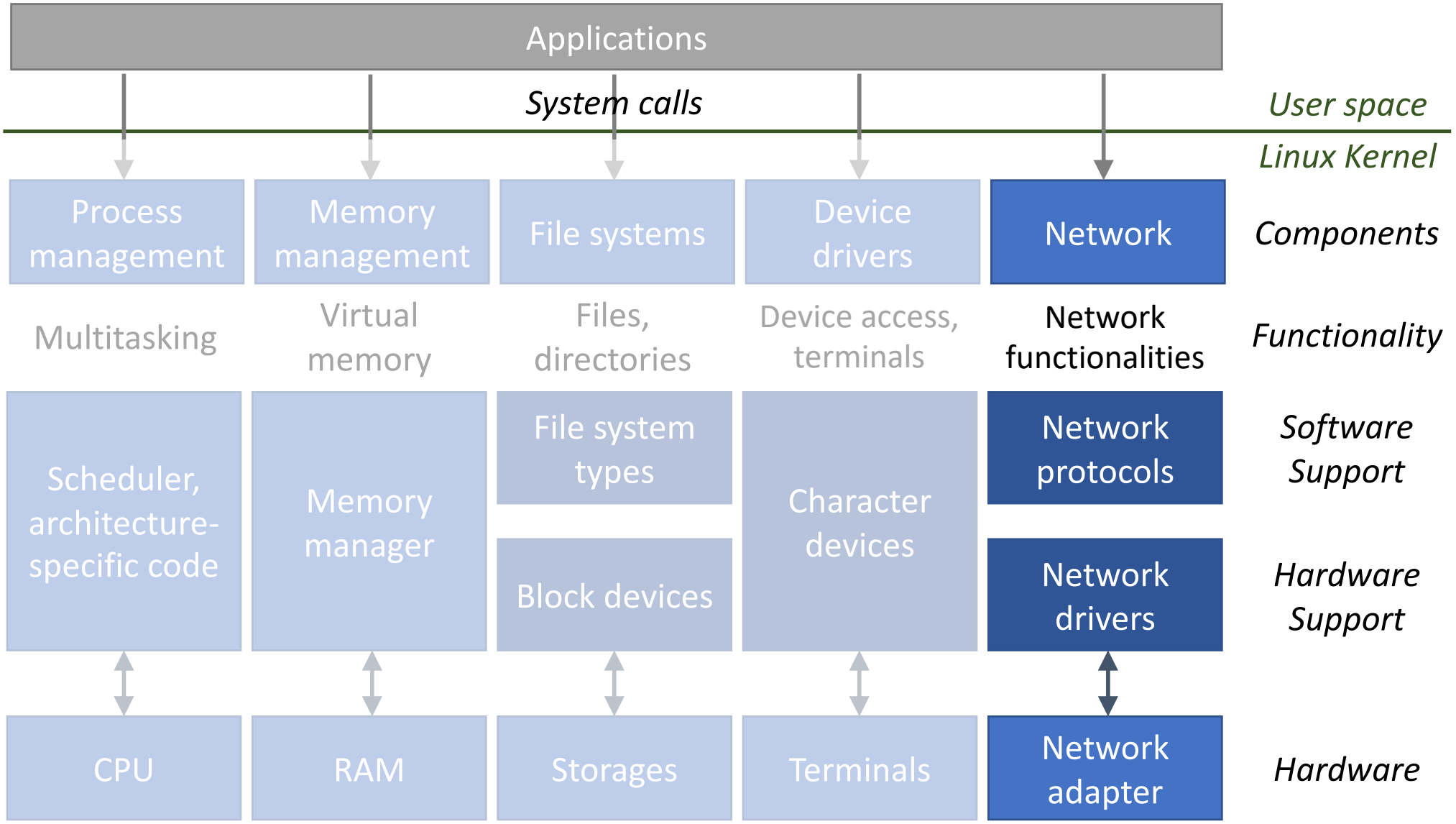
- When you type in *google.com…*
  - Domain name systems translate domain names to IP addresses



Agenda

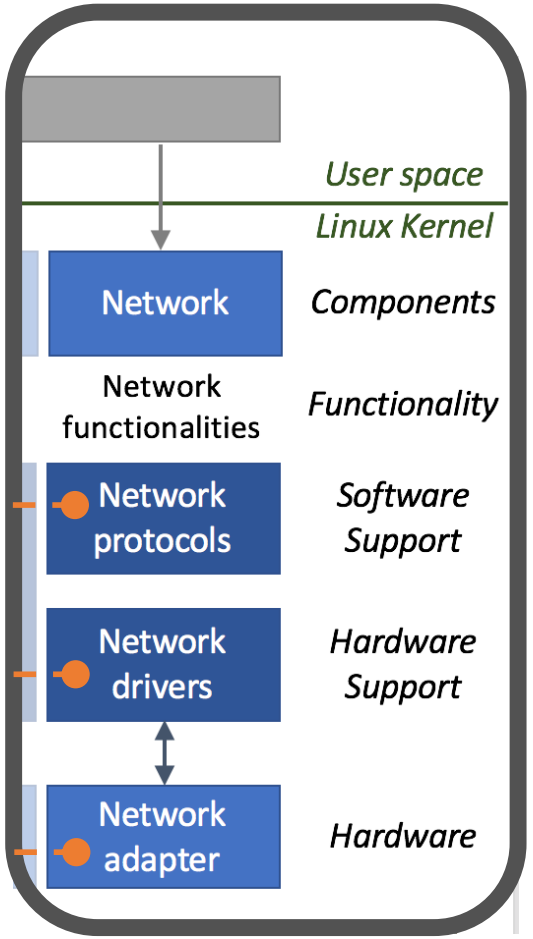
- Network Abstraction in Linux
- Linux Networking Commands
  - ip
  - netstat
  - ping
  - traceroute
  - nslookup
  - nmap
  - tcpdump

Linux Kernel Hierarchy



Network Abstraction in Linux

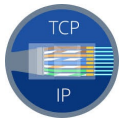
- User programs: web browser, ssh client, …
- Protocols: TCP/IP, …
- Drivers
- Physical connection
  - Ethernet, Wireless Network Cards, Cables
  - Virtual connections (Tunnel, VPN)



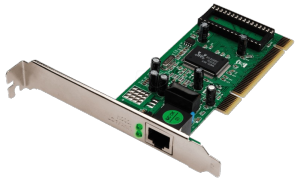
Network Abstraction in Linux

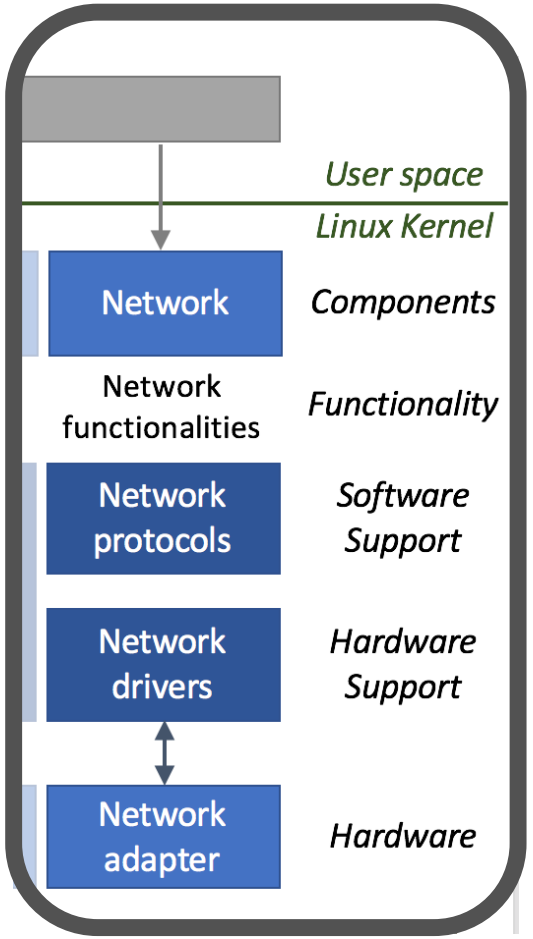
System calls (libc interface)

Protocols



Device Drivers





OSI 7-Layer Model

- Open Systems Interconnection (OSI) Reference Model Layers
  - Conceptual framework that characterizes and standardizes how different SW and HW components should interact with one another

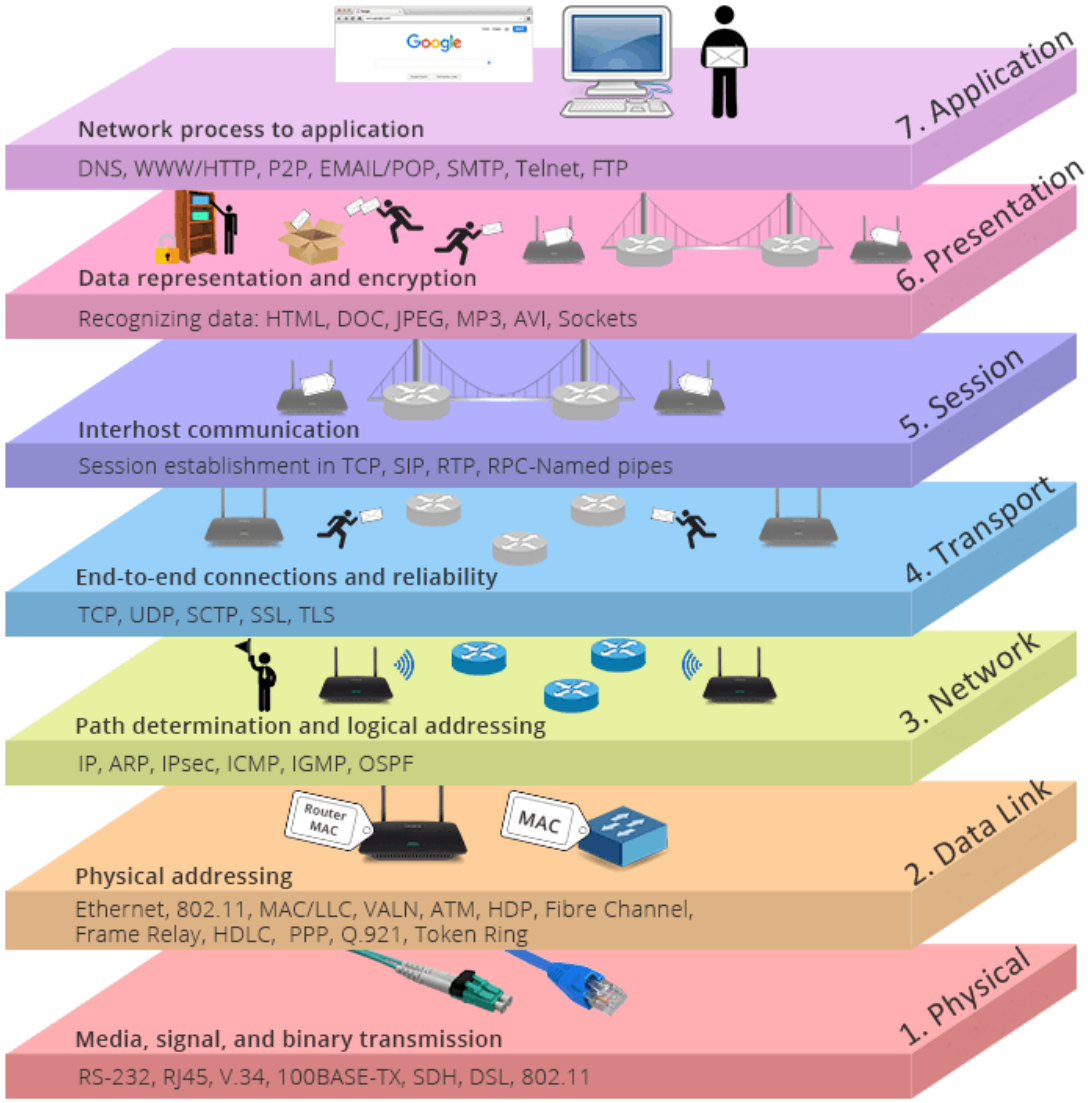
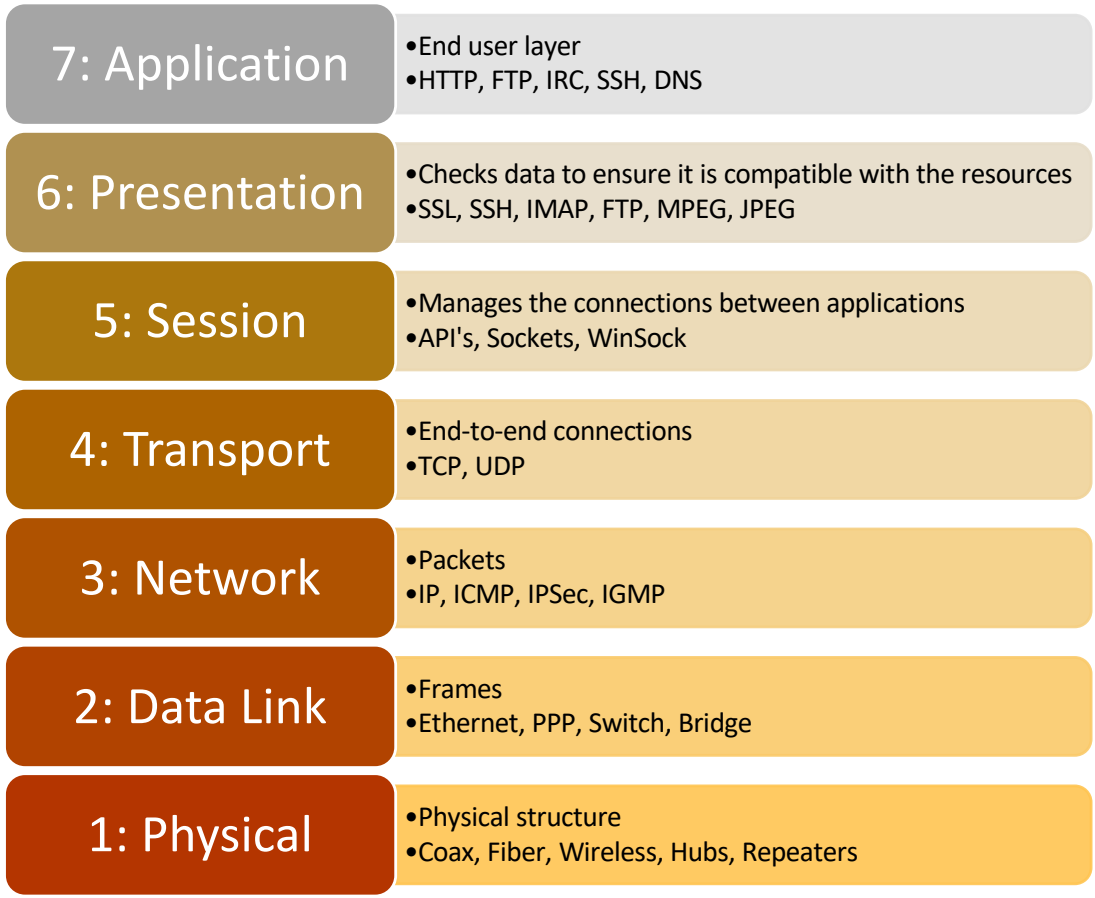


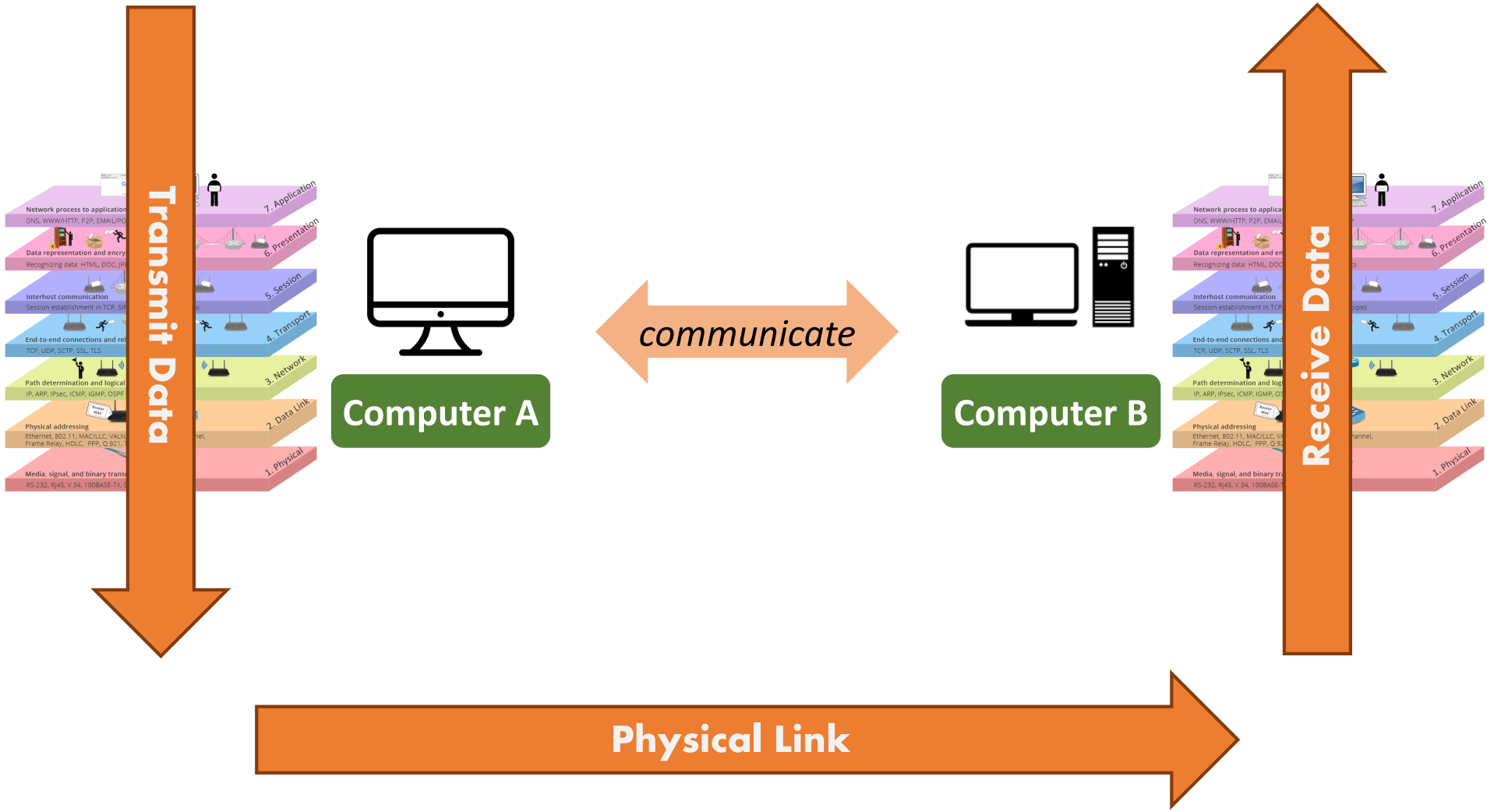
Image obtained from: https://community.fs.com/blog/tcpip-vs-osi-whats-the-difference-between-the-two-models.html

OSI 7-Layer Model

- Divides data communication into seven abstraction layers
- Standardizes protocols into appropriate groups of networking functionalities



OSI 7-Layer Model



OSI 7-Layer Model

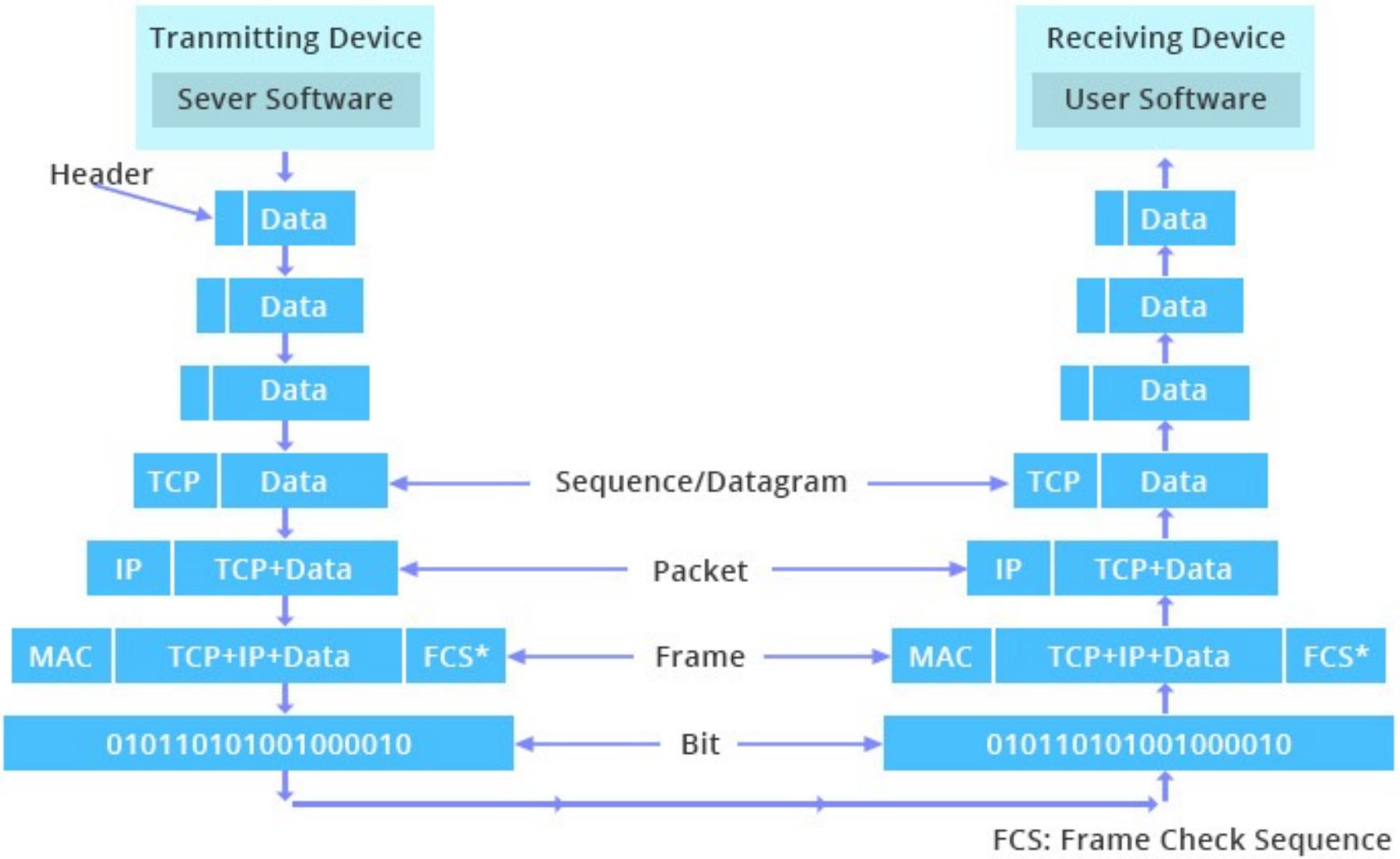
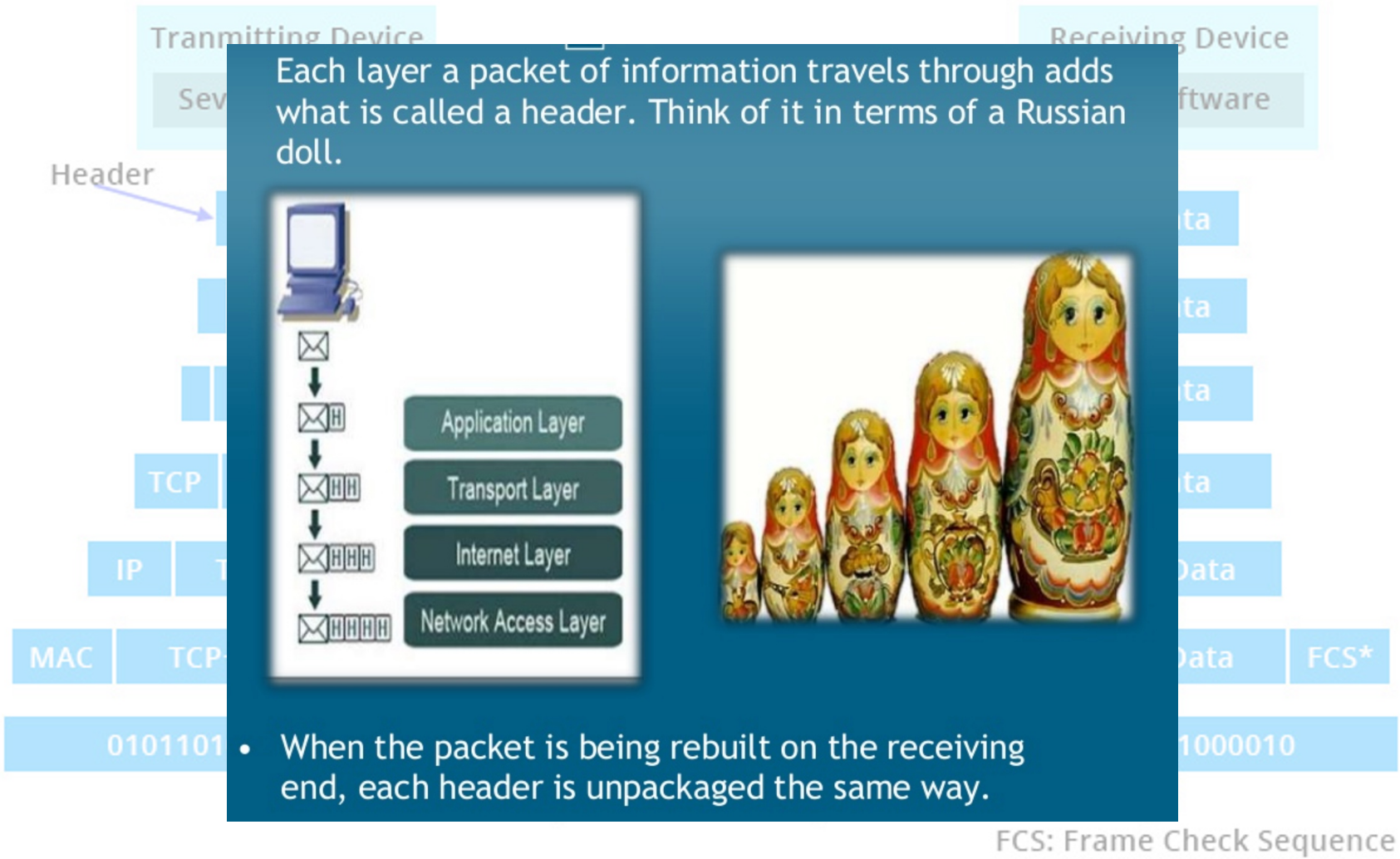


Image obtained from: https://community.fs.com/blog/tcpip-vs-osi-whats-the-difference-between-the-two-models.html

OSI 7-Layer Model



Slide obtained from: https://www.slideshare.net/723323MubarikAli/osi-and-tcp-models

OSI 7-Layer Model

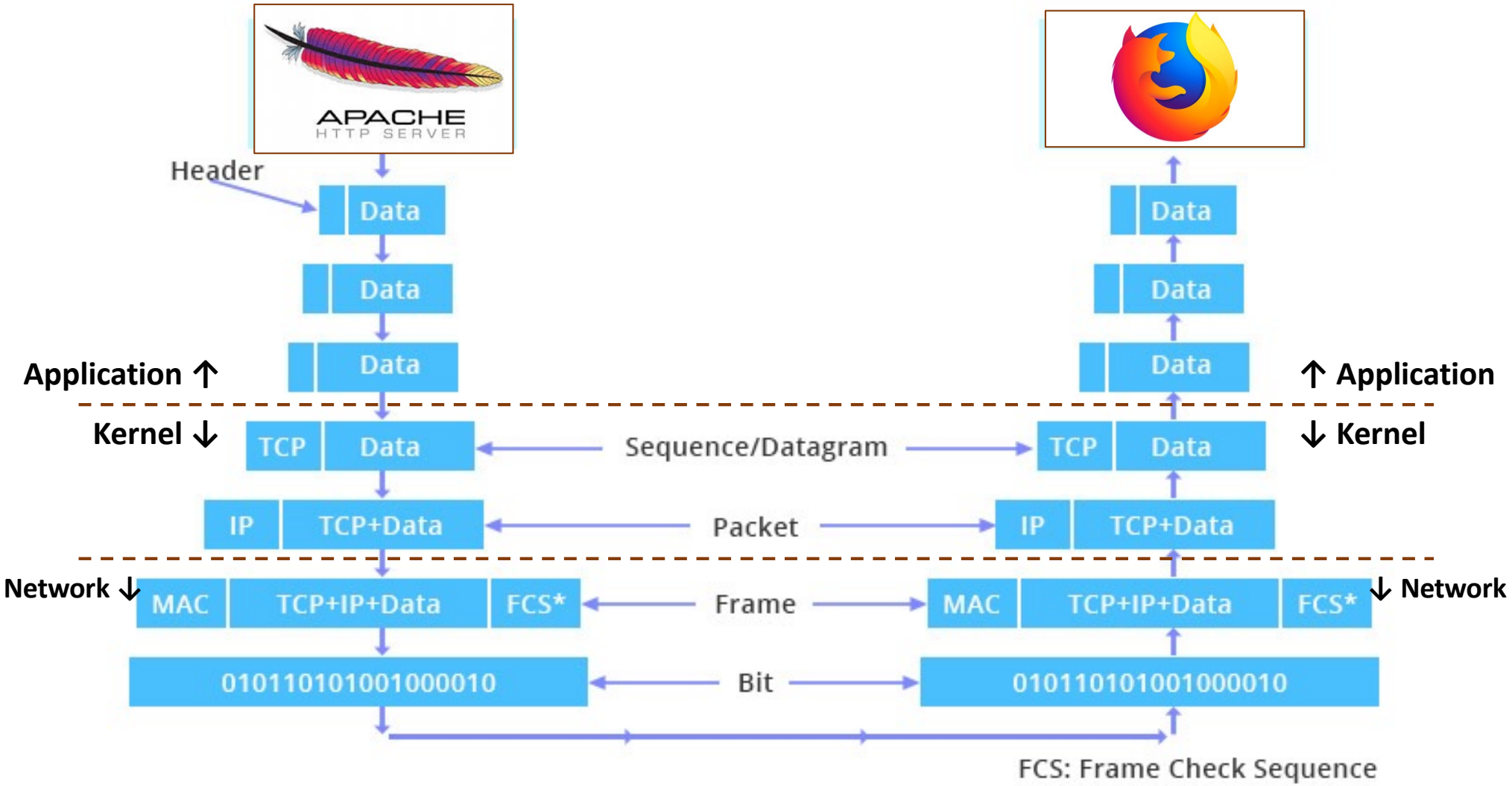
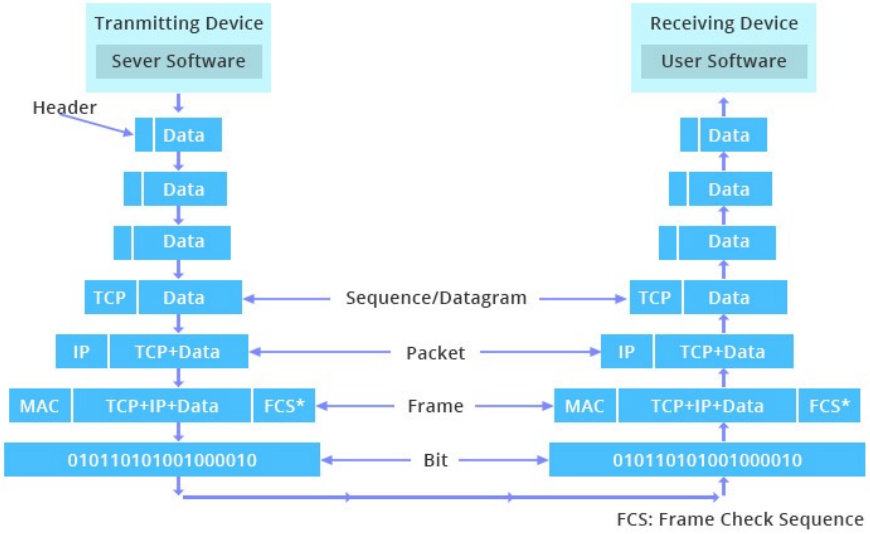
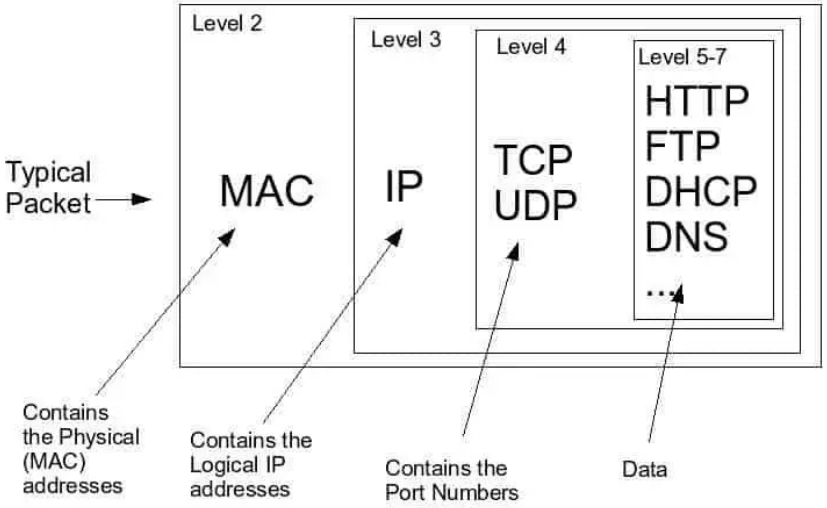


Image obtained from: https://community.fs.com/blog/tcpip-vs-osi-whats-the-difference-between-the-two-models.html

- Data sent over computer networks is divided into packets
  - A packet is a small segment of a larger message
  - Packets are recombined by the computer or device that receives them





TCP/IP

- The OSI model is a conceptual model
  - Used for describing, discussing and understanding individual network functions

- The TCP/IP protocol suite is a set of communication protocols that implements the protocol stack
  - Named after the *two* most important protocols in the suite
    - TCP: Transmission Control Protocol
    - IP: Internet Protocol
  - Developed before the OSI model; TCP/IP can be loosely mapped to the OSI model

OSI 7 – TCP/IP Relationship

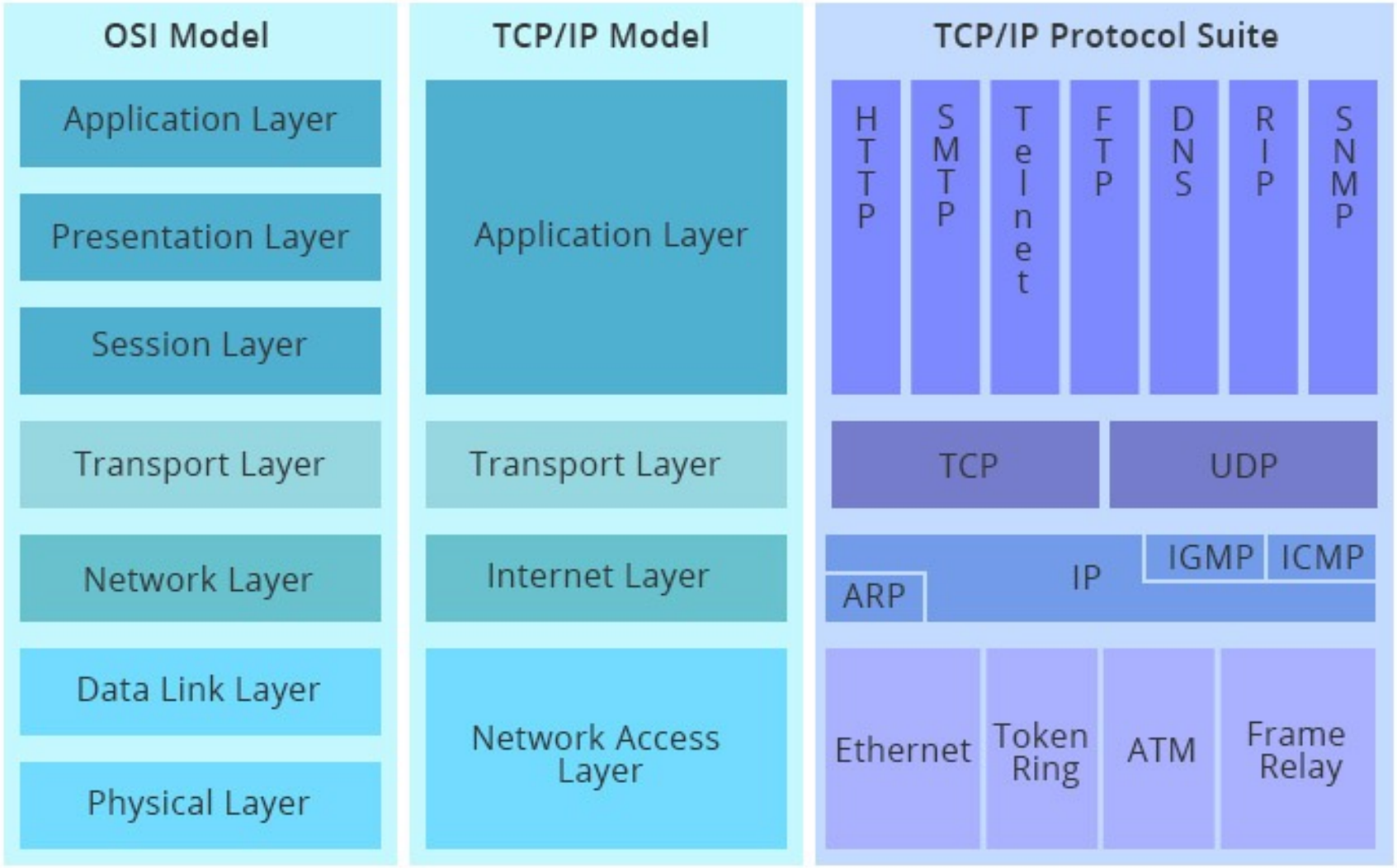
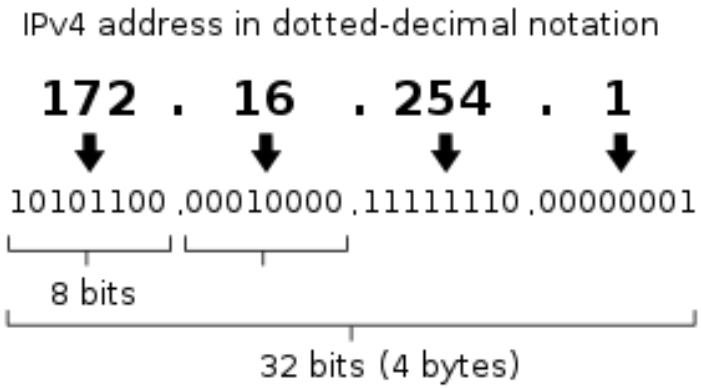


Image obtained from: https://community.fs.com/blog/tcpip-vs-osi-whats-the-difference-between-the-two-models.html

Internet Protocol (IP)

- Addressing/routing protocol – locates hosts and transport data packets
- Uses the IP address (Internet Protocol address): a numerical label assigned to each device on network
  - An identifier of a device on network
  - IPv4 (32-bit), IPv6 (128-bit)



ip – Show / Manipulate Routing, Network Devices

```
$ ip addr

1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group
default qlen 1000

    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00

    inet 127.0.0.1/8 scope host lo

       valid_lft forever preferred_lft forever

    inet6 ::1/128 scope host

       valid_lft forever preferred_lft forever

2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP
group default qlen 1000

    link/ether 08:00:27:f2:ca:17 brd ff:ff:ff:ff:ff:ff

    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3

       valid_lft 76103sec preferred_lft 76103sec

    inet6 fe80::2f11:790b:4b7e:1045/64 scope link noprefixroute

       valid_lft forever preferred_lft forever
```

ip – Show / Manipulate Routing, Network Devices

```
$ ip -s link

1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode
DEFAULT group default qlen 1000

    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00

    RX: bytes  packets  errors  dropped overrun mcast

    219403     3296     0       0       0       0

    TX: bytes  packets  errors  dropped carrier collsns

    219403     3296     0       0       0       0

2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP
mode DEFAULT group default qlen 1000

    link/ether 08:00:27:f2:ca:17 brd ff:ff:ff:ff:ff:ff

    RX: bytes  packets  errors  dropped overrun mcast

    594960433  525095   0       0       0       0

    TX: bytes  packets  errors  dropped carrier collsns

    8623927    132084   0       0       0       0
```

Transmission Control Protocol (TCP)

- Provides reliable, error-checked delivery of a data stream over networks
  - Not only offers a communication service, but also supports data re-transmission, error recovery, network congestion control, traffic load balancing, *etc*.

TCP/IP and Ports

- In TCP/IP, a port is an endpoint to a logical connection
  - Ports are sub-addressed defined within the IP address

  - Ports 0~1023 system or well-know service ports
  - Ports 1024~49151 user or registered ports
  - Ports > 49151 dynamic/private ports

TCP/IP and Ports

- Common/Popular IANA-registered ports
  - 20-21 FTP
  - 22 SSH
  - 23 Telnet
  - 25 SMTP
  - 53 DNS
  - 80 HTTP
  - 443 HTTPS

Agenda

- Network Abstraction in Linux
- **Linux Networking Commands**
  - ip
  - ping
  - traceroute
  - nmap
  - nslookup
  - netstat
  - tcpdump

ping – send ICMP ECHO_REQUEST to network hosts

- The most basic network test tool for testing network reachability
  - Sends out ICMP packets (ECHO_REQUEST) to a host across the network and notifies you whether there is a response
  - Sometimes blocked by the system administrator for security reasons

ping – send ICMP ECHO_REQUEST to network hosts

```
$ ping google.com -c 4

PING google.com (172.217.161.78) 56(84) bytes of data.

64 bytes from nrt20s09-in-f14.1e100.net (172.217.161.78): icmp_seq=1
ttl=63 time=40.9 ms

64 bytes from nrt20s09-in-f14.1e100.net (172.217.161.78): icmp_seq=2
ttl=63 time=41.3 ms

64 bytes from nrt20s09-in-f14.1e100.net (172.217.161.78): icmp_seq=3
ttl=63 time=40.4 ms

64 bytes from nrt20s09-in-f14.1e100.net (172.217.161.78): icmp_seq=4
ttl=63 time=40.2 ms


--- google.com ping statistics ---

4 packets transmitted, 4 received, 0% packet loss, time 3004ms

rtt min/avg/max/mdev = 40.212/40.738/41.350/0.441 ms
```

nmap – Port Scanner

- Checks the opened ports on the server

```
$ nmap google.com

Starting Nmap 7.60 ( https://nmap.org ) at 2019-05-06 10:42 KST
Nmap scan report for google.com (216.58.197.174)
Host is up (0.043s latency).
Other addresses for google.com (not scanned):
2404:6800:4004:808::200e
rDNS record for 216.58.197.174: nrt12s02-in-f174.1e100.net
Not shown: 998 filtered ports
PORT    STATE SERVICE
80/tcp  open  http
443/tcp open  https

Nmap done: 1 IP address (1 host up) scanned in 4.46 seconds
```

nslookup – Query Internet Name Servers

- nslookup is a program to query Internet domain name servers

```
$ nslookup google.com
Server:     127.0.0.53
Address:    127.0.0.53#53

Non-authoritative answer:
Name:   google.com
Address: 172.217.25.238
Name:   google.com
Address: 2404:6800:4004:800::200e


$ nslookup 172.217.25.238
238.25.217.172.in-addr.arpa   name = nrt12s14-in-f238.1e100.net.
238.25.217.172.in-addr.arpa   name = nrt12s14-in-f14.1e100.net.

Authoritative answers can be found from:
```

netstat – Network Status

```
$ netstat -nr
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
0.0.0.0         10.0.2.2        0.0.0.0         UG        0 0          0 enp0s3
10.0.2.0        0.0.0.0         255.255.255.0   U         0 0          0 enp0s3
169.254.0.0     0.0.0.0         255.255.0.0     U         0 0          0 enp0s3


$ netstat -i enp0s3
Kernel Interface table
Iface      MTU    RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR Flg
enp0s3    1500   525448      0      0 0        132418      0      0      0 BMRU
lo       65536     3514      0      0 0          3514      0      0      0 LRU


$ netstat -tan
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN
tcp6       0      0 ::1:631                 :::*                    LISTEN
```

traceroute – Trace the Route Packets

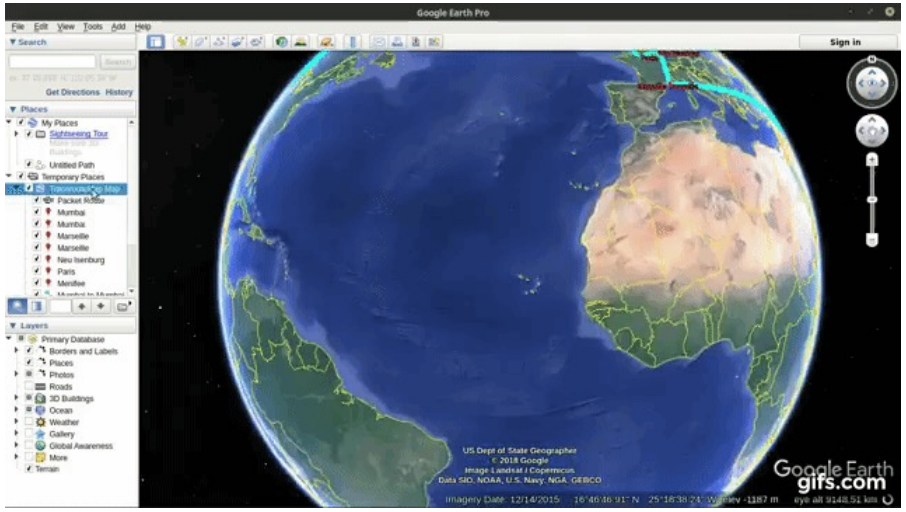
- Prints the route packets take to network host
  - Shows all intermediate nodes between the local system and remote host
- Destination host or IP is mandatory parameter

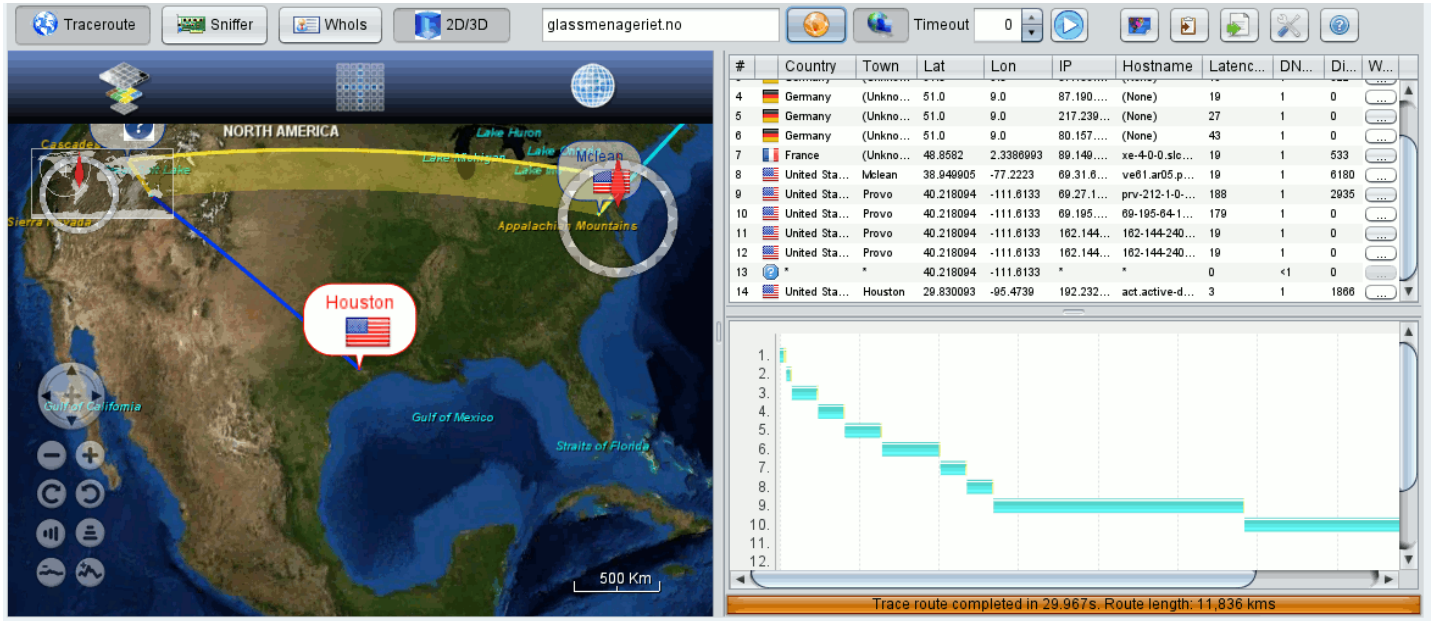
traceroute – Trace the Route Packets

```
$ traceroute google.com
traceroute to google.com (172.217.31.142), 30 hops max, 60 byte
packets
 1  _gateway (10.0.2.2)  8.138 ms  0.154 ms  0.112 ms
 2  192.168.1.1 (192.168.1.1)  2.190 ms  2.762 ms  3.053 ms
 3  222.103.213.254 (222.103.213.254)  3.825 ms  4.067 ms  6.065 ms
 4  59.24.57.13 (59.24.57.13)  6.393 ms  7.134 ms  6.787 ms
 5  112.190.136.209 (112.190.136.209)  2.074 ms  2.335 ms  2.164 ms
 6  112.190.135.77 (112.190.135.77)  2.249 ms 112.190.135.29
(112.190.135.29)  2.708 ms *
 7  112.190.188.101 (112.190.188.101)  3.999 ms 112.190.174.253
(112.190.174.253)  6.035 ms  5.970 ms
 8  * * *
 9  112.174.7.126 (112.174.7.126)  8.847 ms 112.174.7.170
(112.174.7.170)  8.999 ms 112.174.5.126 (112.174.5.126)  8.901 ms
10  74.125.52.16 (74.125.52.16)  38.068 ms  38.279 ms  37.295 ms
11  108.170.242.193 (108.170.242.193)  38.791 ms  38.011 ms  40.282
ms
12  74.125.251.237 (74.125.251.237)  39.308 ms  42.122 ms
74.125.251.235 (74.125.251.235)  42.040 ms
13  nrt20s08-in-f14.1e100.net (172.217.31.142)  38.211 ms  40.292 ms
37.965 ms
```

traceroute – Trace the Route Packets

- Visual traceroute (online tool)
  - https://www.monitis.com/traceroute/





tcpdump – A Packet Sniffing Tool

- Dumps traffic on a network
  - Communication done in cleartext could be observed

tcpdump – A Packet Sniffing Tool

```
$ sudo tcpdump -c 2 -A -i enp0s3
tcpdump: verbose output suppressed, use -v or -vv for full protocol
decode
listening on enp0s3, link-type EN10MB (Ethernet), capture size 262144
bytes
10:45:30.408024 IP vm-VirtualBox.40198 > nrt12s23-in-
f3.1e100.net.http: Flags [.], ack 183872703, win 30143, length 0
E..(..@.@...
......#...P."'.
...P.u.Z&..
10:45:30.408360 IP nrt12s23-in-f3.1e100.net.http > vm-
VirtualBox.40198: Flags [.], ack 1, win 65535, length 0
E..()...@......#
....P..
....''.P.............
2 packets captured
4 packets received by filter
0 packets dropped by kernel
```

tcpdump – A Packet Sniffing Tool

- WireShark – free and open source packet analyzer
  - https://www.wireshark.org
  - Used for network troubleshooting, software development, education