

CSCI369 Ethical Hacking

Week 3 –TCP/IP Basics & Capturing Traffic

Instructor: Dr. Manoj Kumar

Faculty of Engineering and Information Sciences

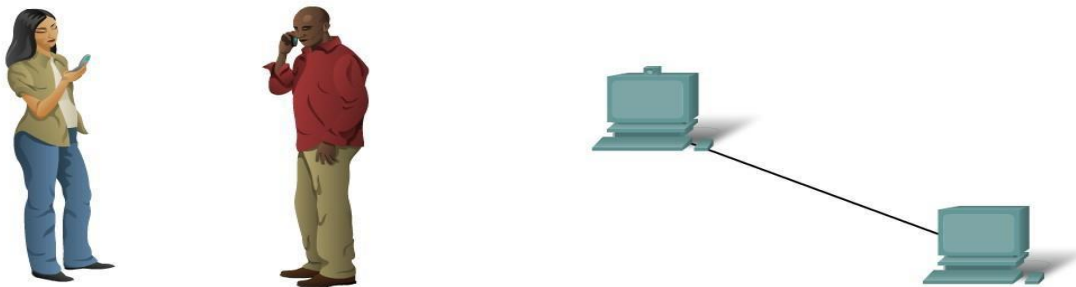
These slides are based on the lecture slides prepared by Dr. Joonsang Baek and Dr. Manoj Kumar



UNIVERSITY
OF WOLLONGONG
IN DUBAI

Network Structure

- Define the elements of communication
 - 3 common elements of communication
 - message source
 - the channel
 - message destination



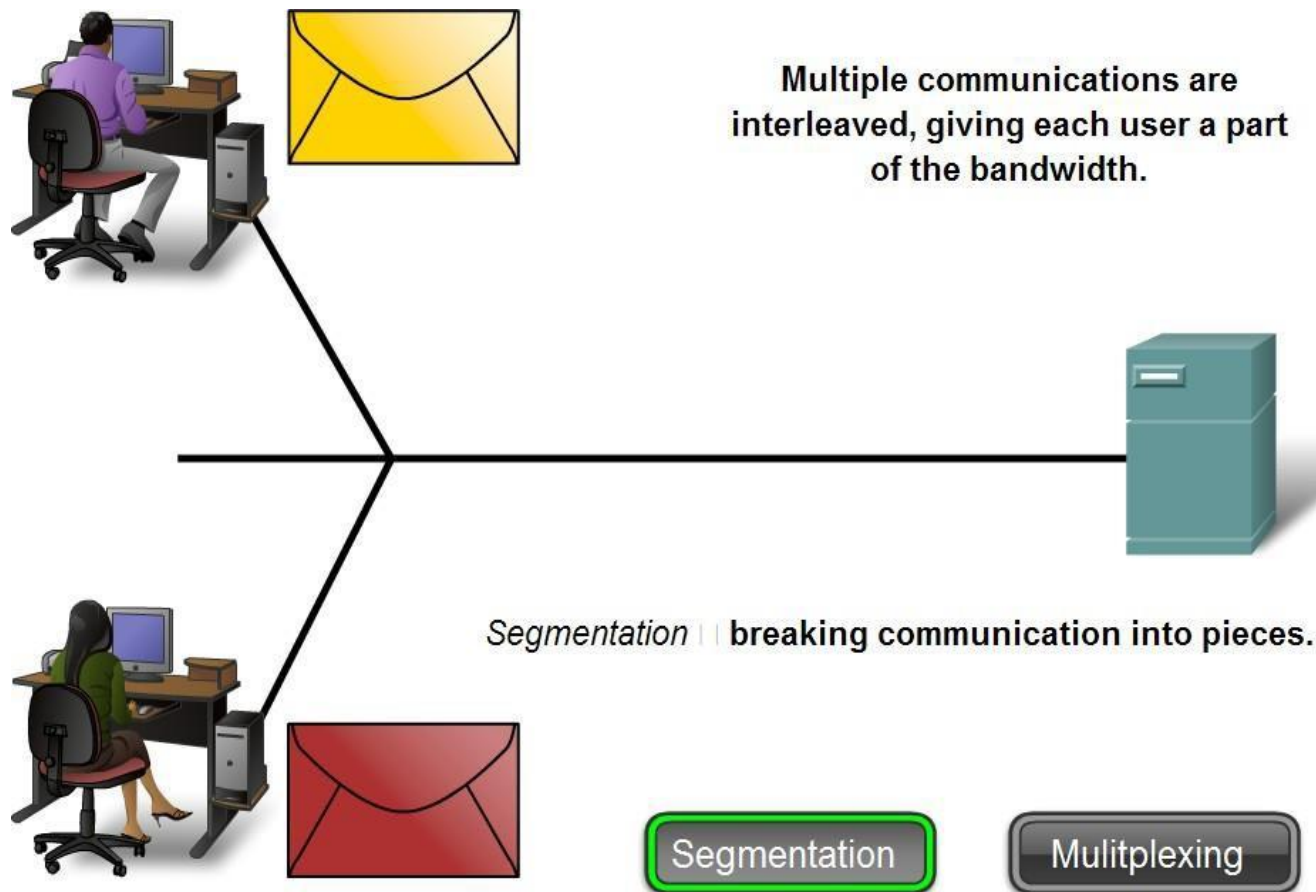
- Define a network

data or information networks capable of carrying many different types of communications

Network Structure

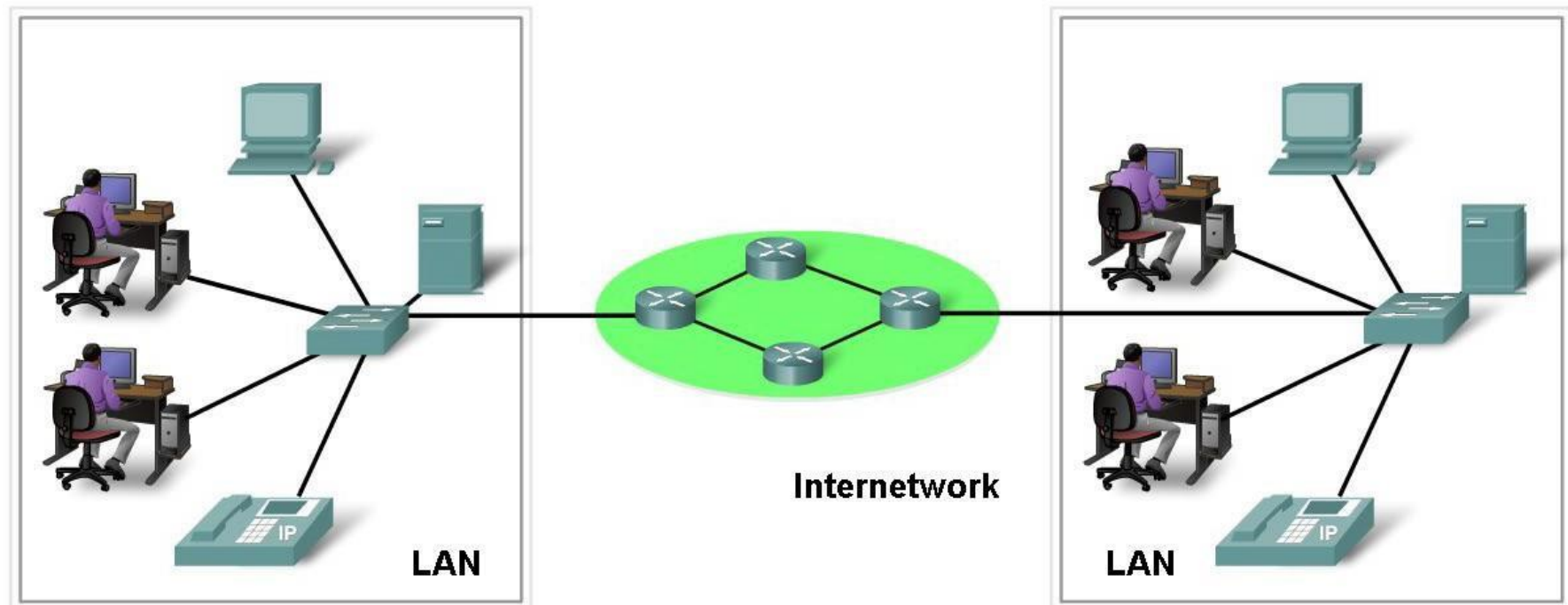
- Describe how messages are communicated

Data is sent across a network in small “chunks” called segments



Network Structure

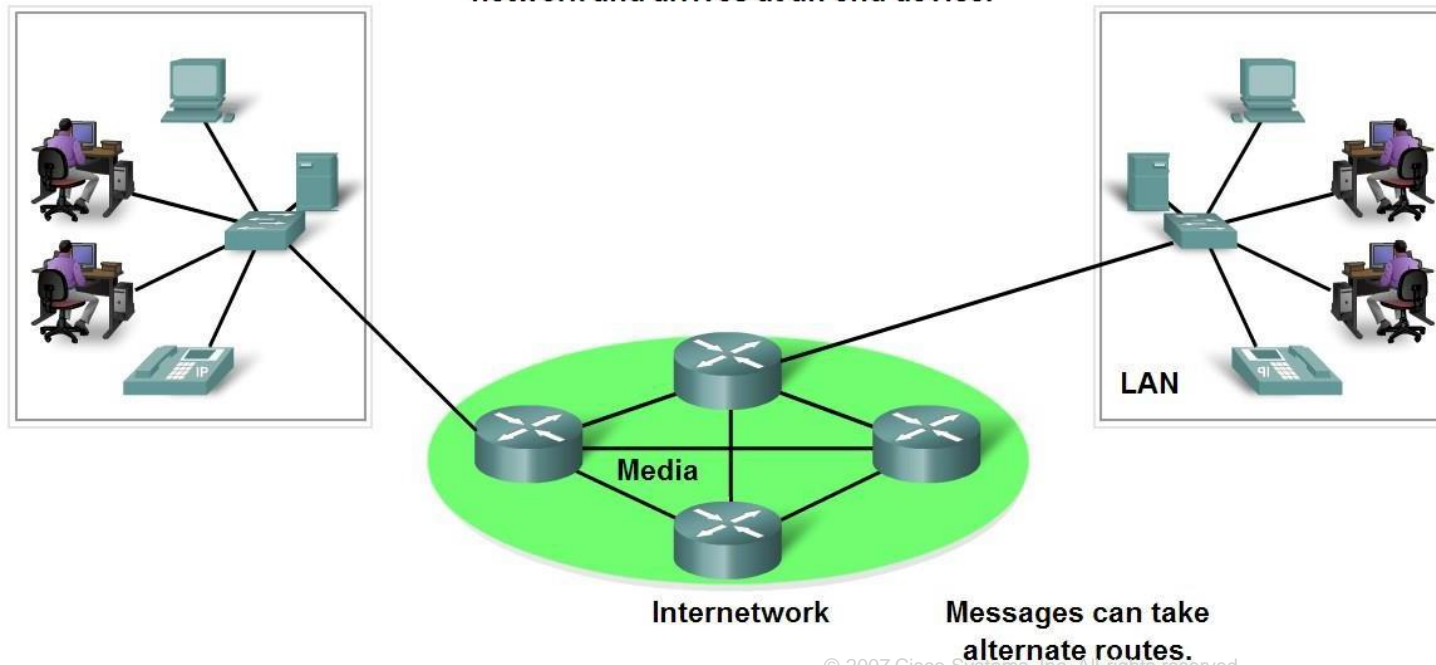
- Define the components of a network
 - Network components
 - hardware
 - software



Network Structure

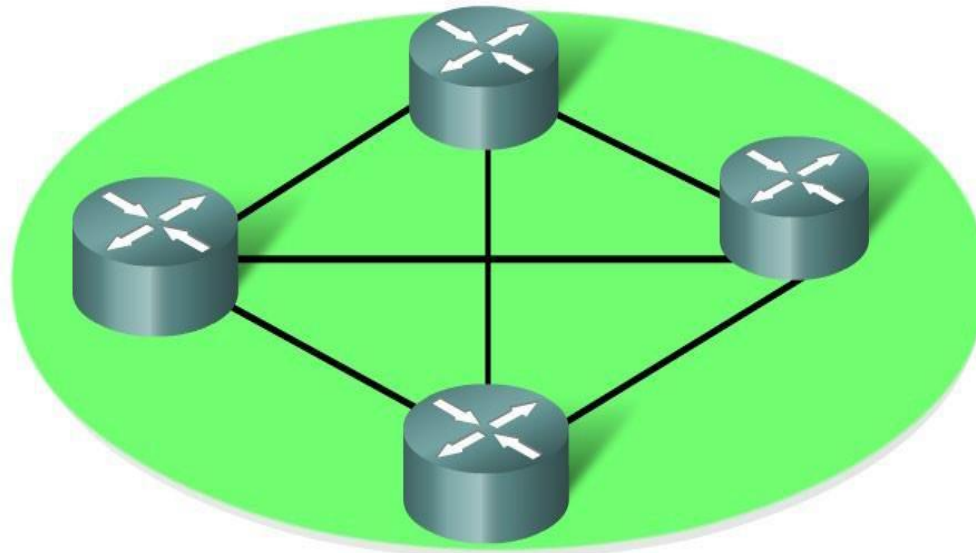
- End Devices and their Role in the Network
 - End devices form interface with human network & communications network
 - Role of end devices:
 - client
 - server
 - both client and server

Data originates with an end device, flows through the network and arrives at an end device.



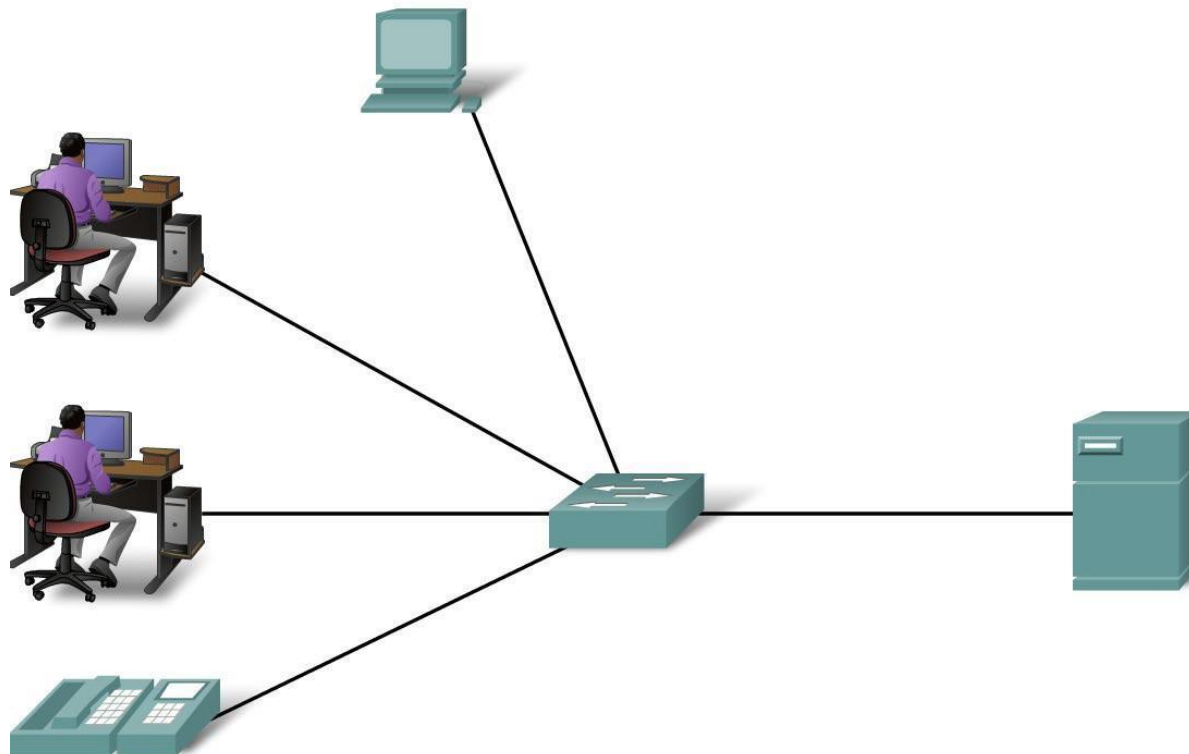
Network Structure

- Identify the role of an **intermediary device** in a data network and be able to contrast that role with the role of an end device
 - Role of an intermediary device
 - provides connectivity and ensures data flows across network



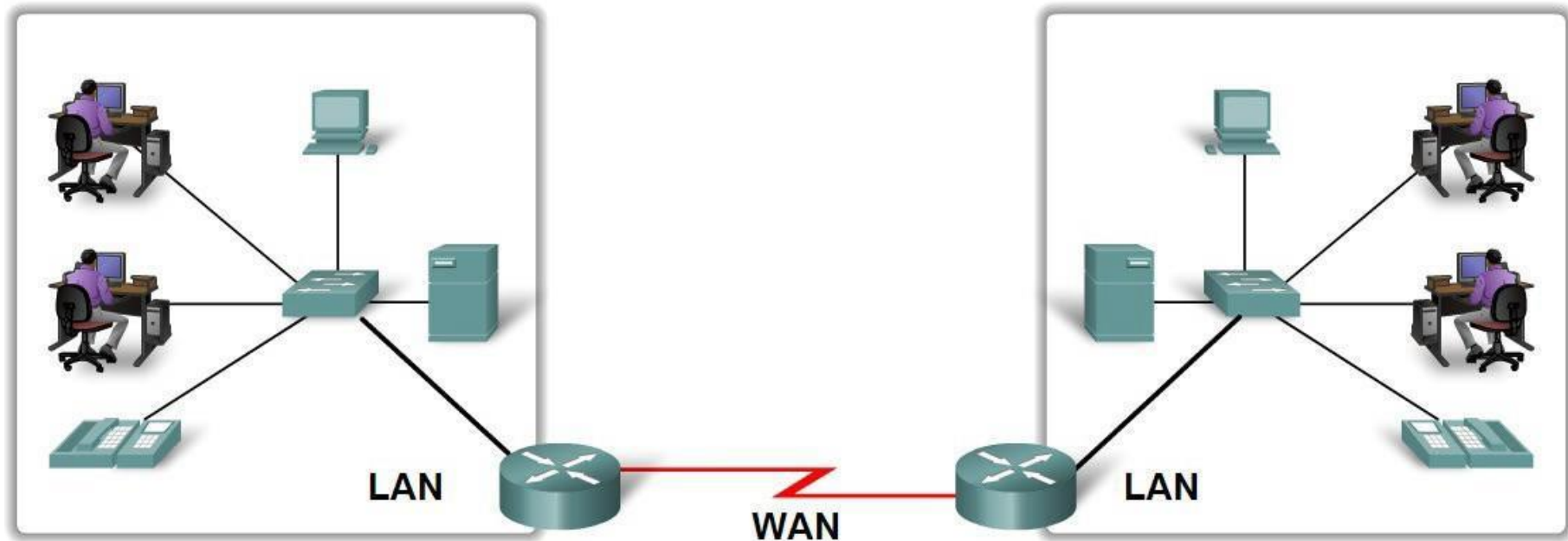
Network Types

- Define Local Area Networks (LANs)
 - A network serving a home, building or campus is considered a Local Area Network (LAN)



Network Types

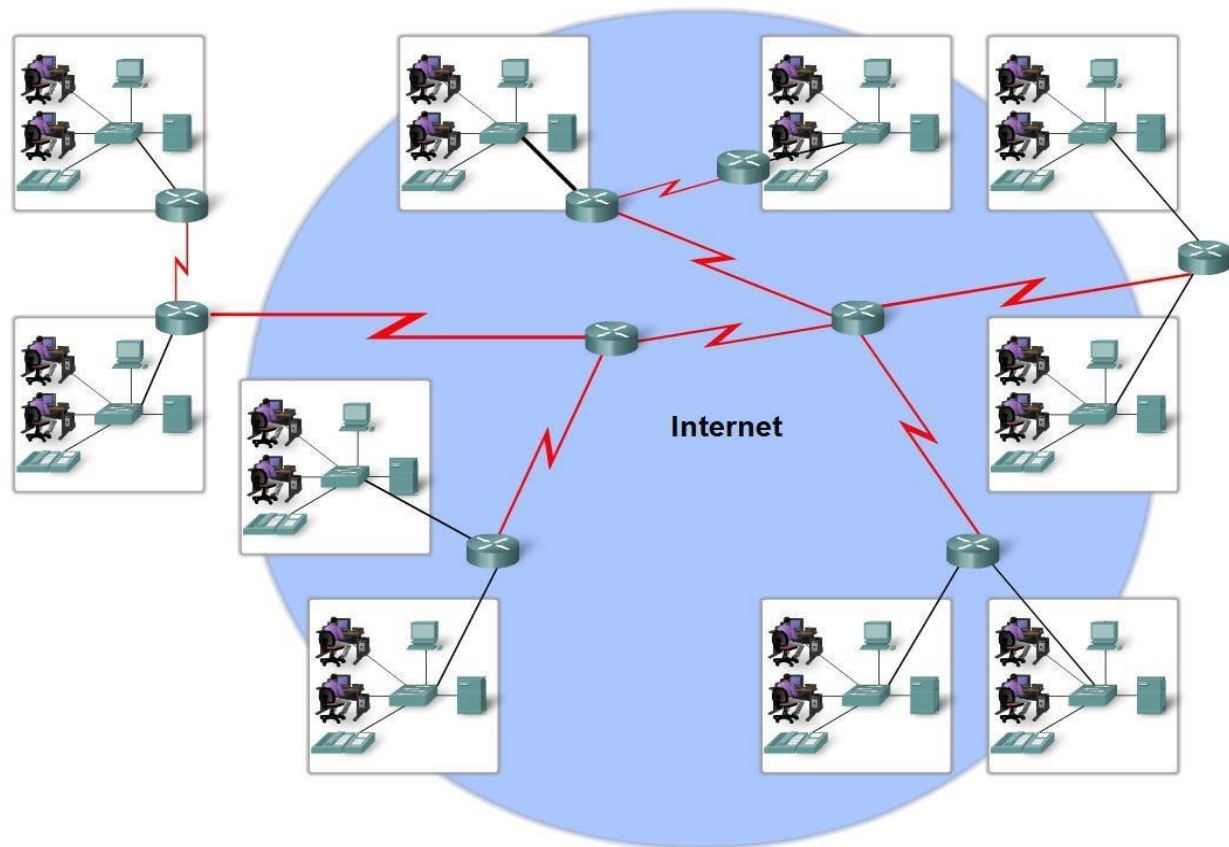
- Define Wide Area Networks (WANs)
 - LANs separated by geographic distance are connected by a network known as a Wide Area Network (WAN)



Network Types

- Define the Internet

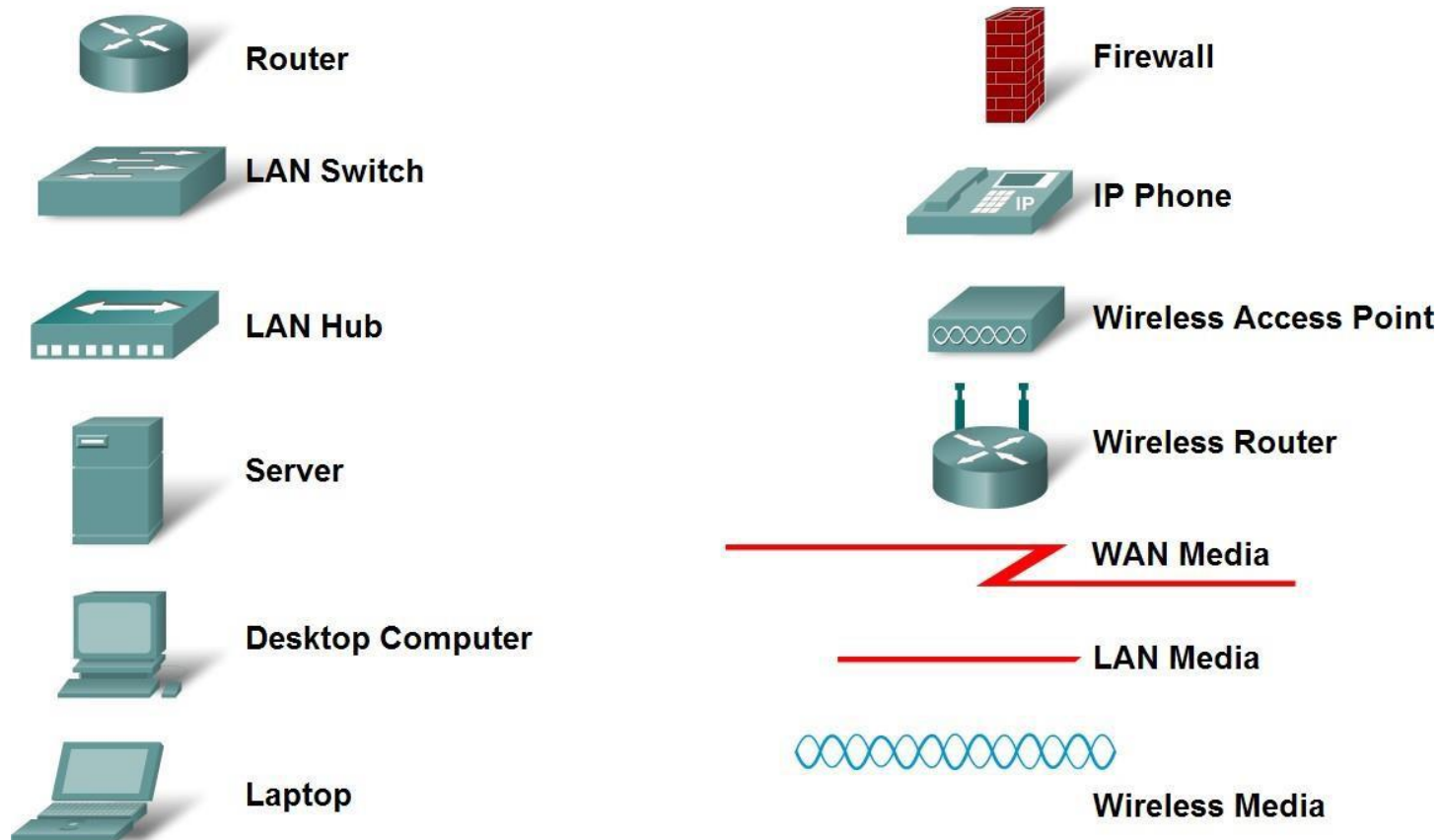
The internet is defined as a
global mesh of interconnected networks



Network Devices

- Describe network representations

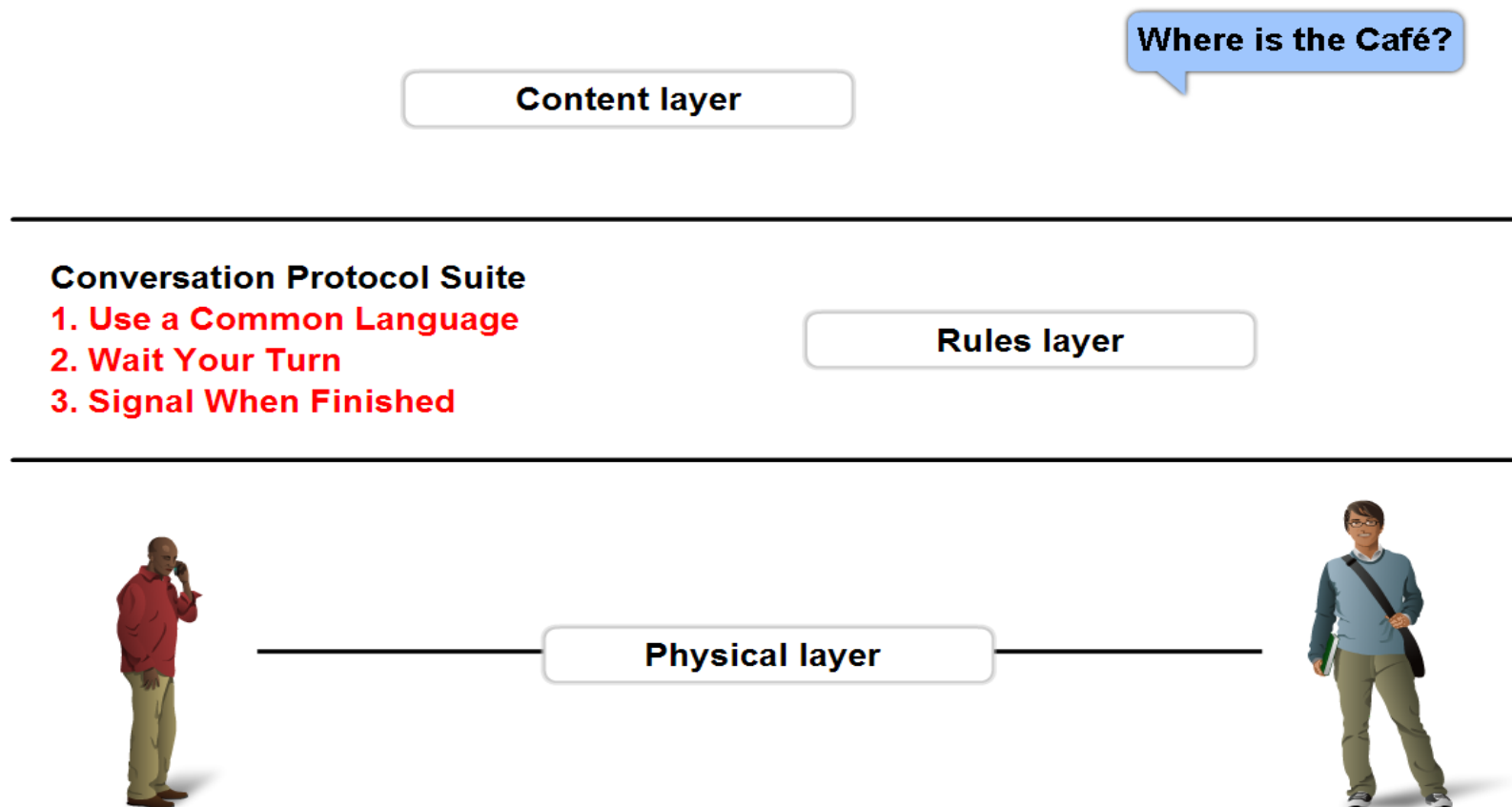
Common Data Network Symbols



Function of Protocol in Network Communication

- The importance of protocols and how they are used to facilitate communication over data networks

A protocol is a set of predetermined rules



Function of Protocol in Network Communication

- Explain network protocols

Network protocols are used
to allow devices to
communicate
successfully

Protocols provide:

The format or structure of the message

The process by which networking devices share information about pathways to other networks

How and when error and system messages are passed between devices

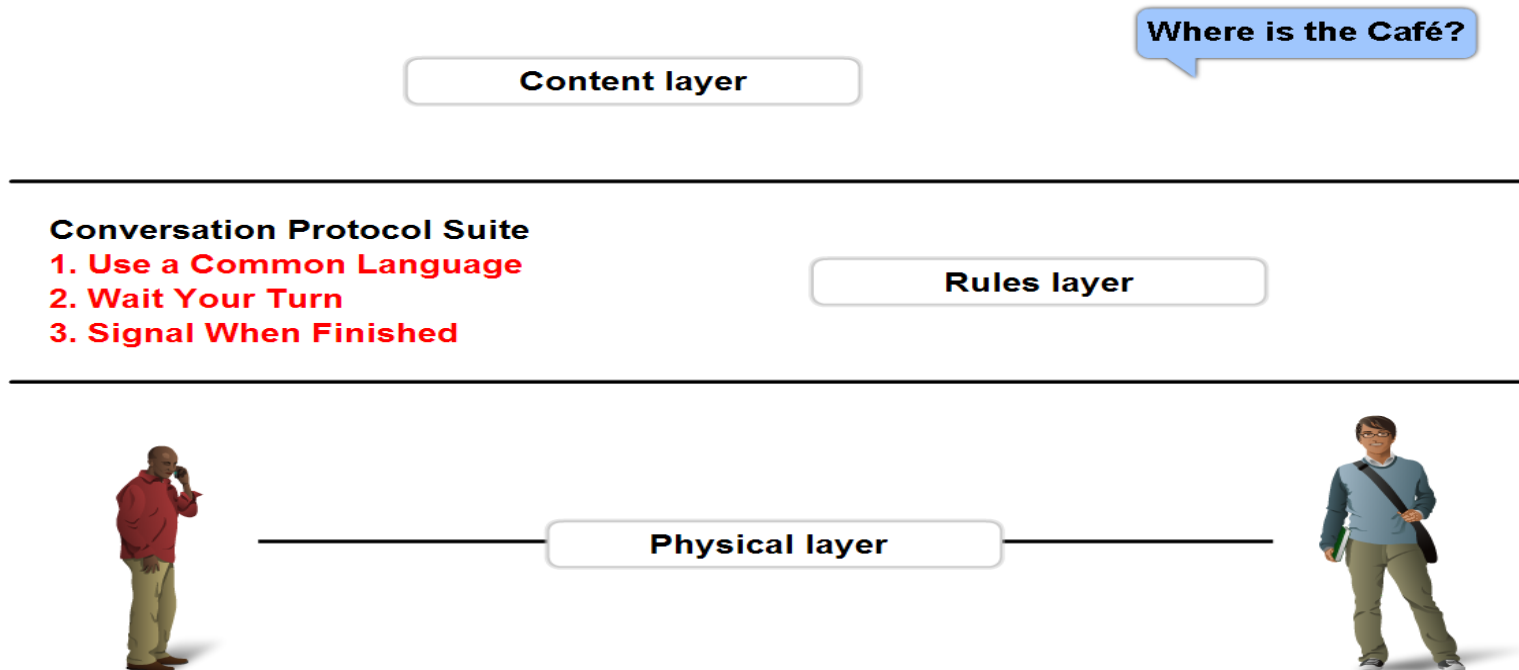
The setting up and termination of data transfer sessions



Function of Protocol in Network Communication

- Describe Protocol suites and industry standards

Protocol Suites are sets of rules that work together to help solve a problem.



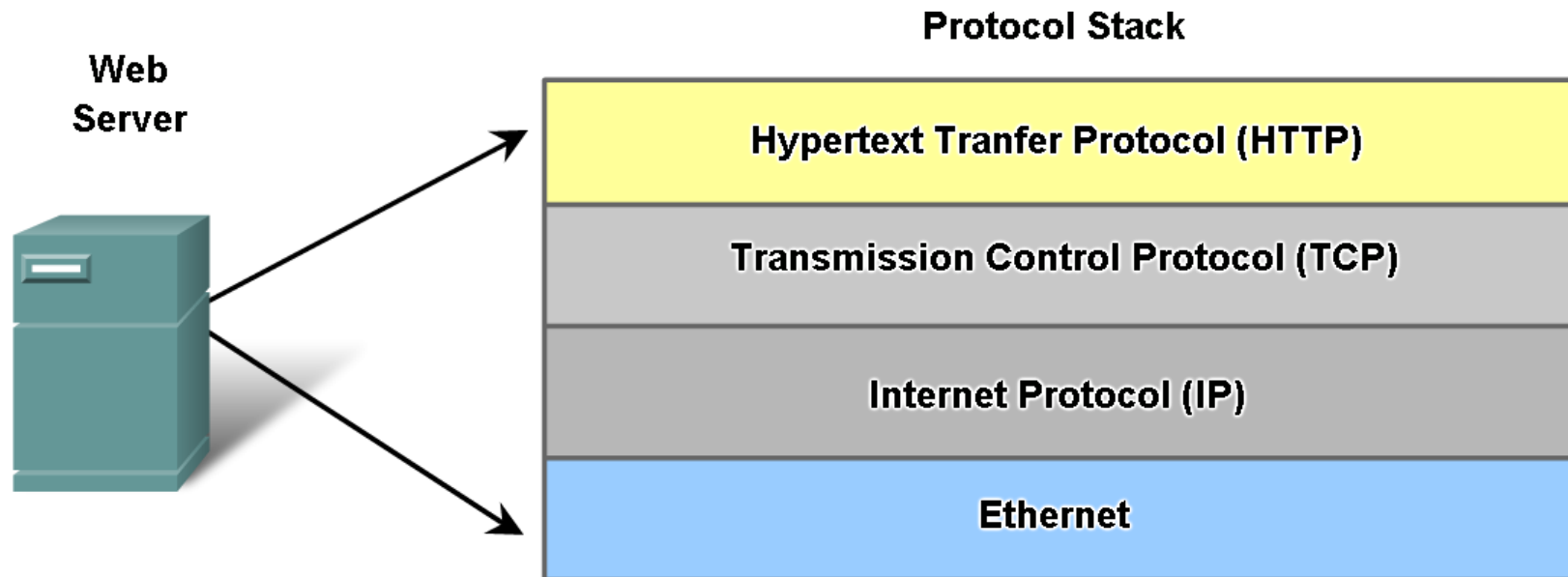
A standard is

a process or protocol that has been endorsed by the networking industry and ratified by a standards organization



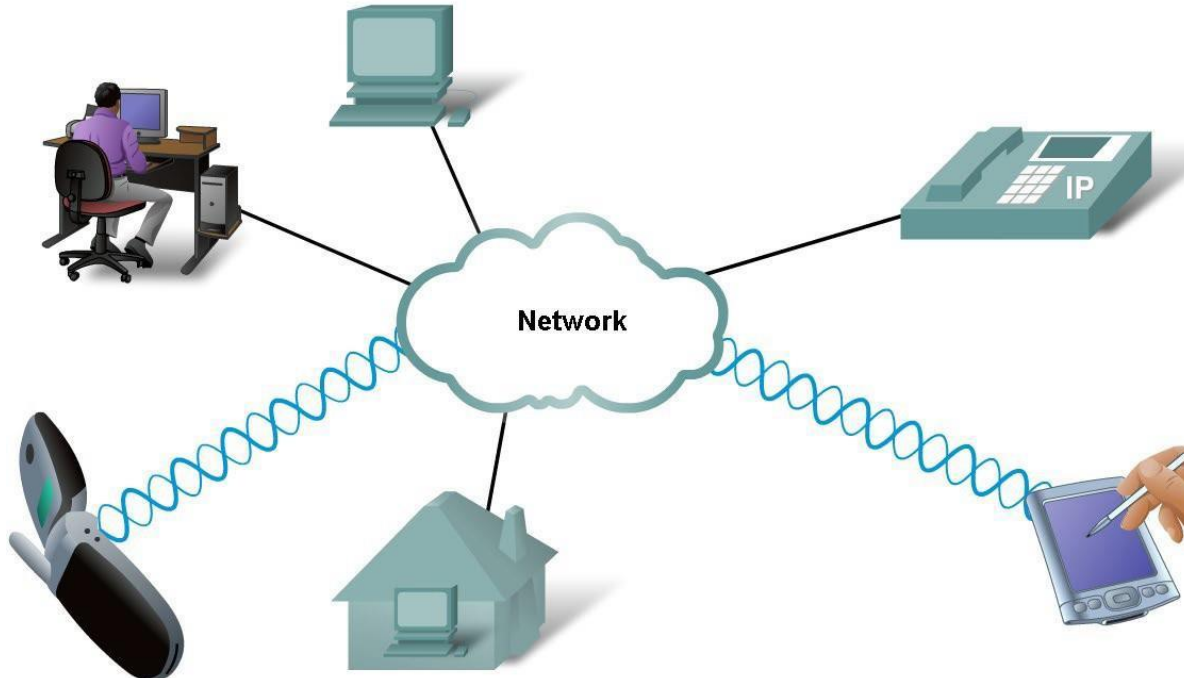
Function of Protocol in Network Communication

- Define different protocols and how they interact



Function of Protocol in Network Communication

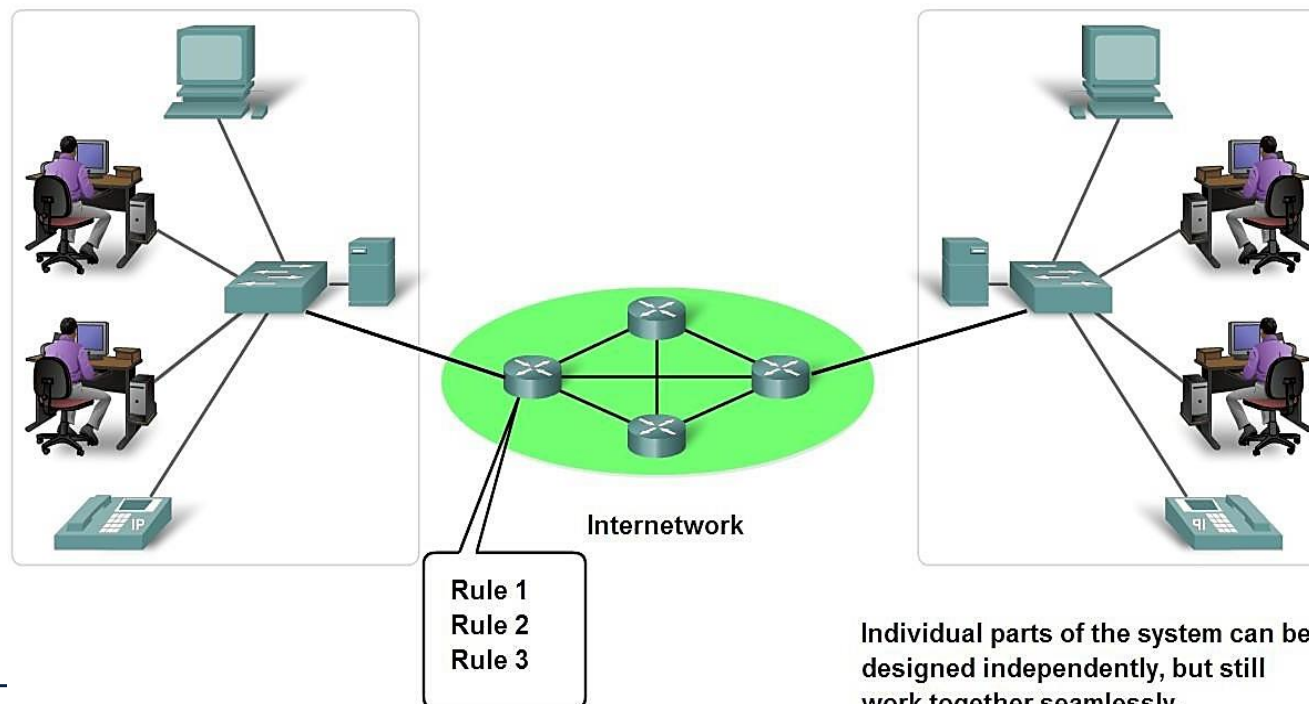
- Technology independent Protocols
 - Many diverse types of devices can communicate using the same sets of protocols. This is because protocols specify network functionality, not the underlying technology to support this functionality.



Layers with TCP/IP and OSI Model

- Explain the benefits of using a layered model
 - Benefits include
 - Assists in protocol design
 - Changes in one layer do not affect other layers
 - provides a common language

Using a layered model helps in the design of complex, multi-use, multi-vendor networks.

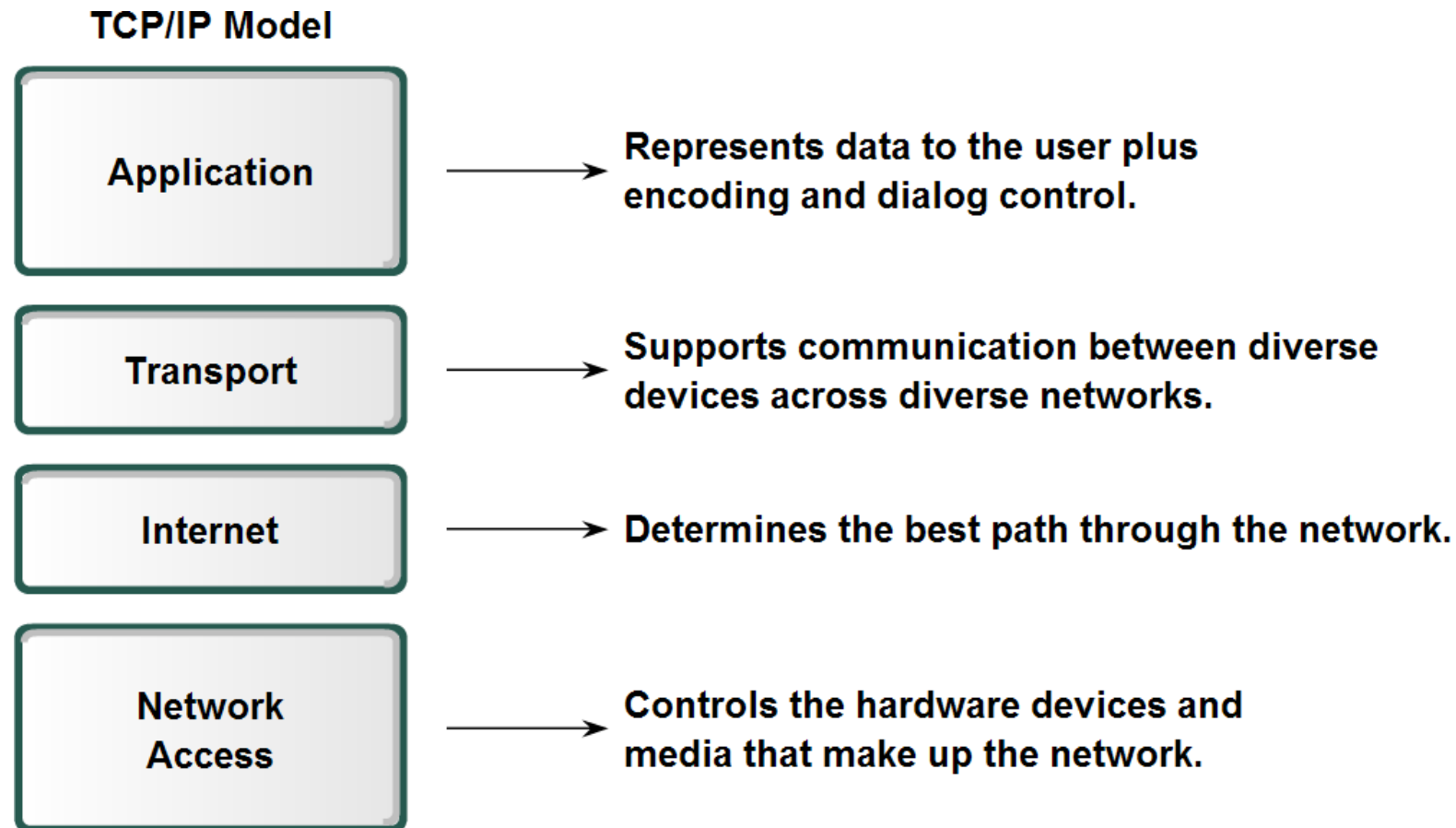


Individual parts of the system can be designed independently, but still work together seamlessly.



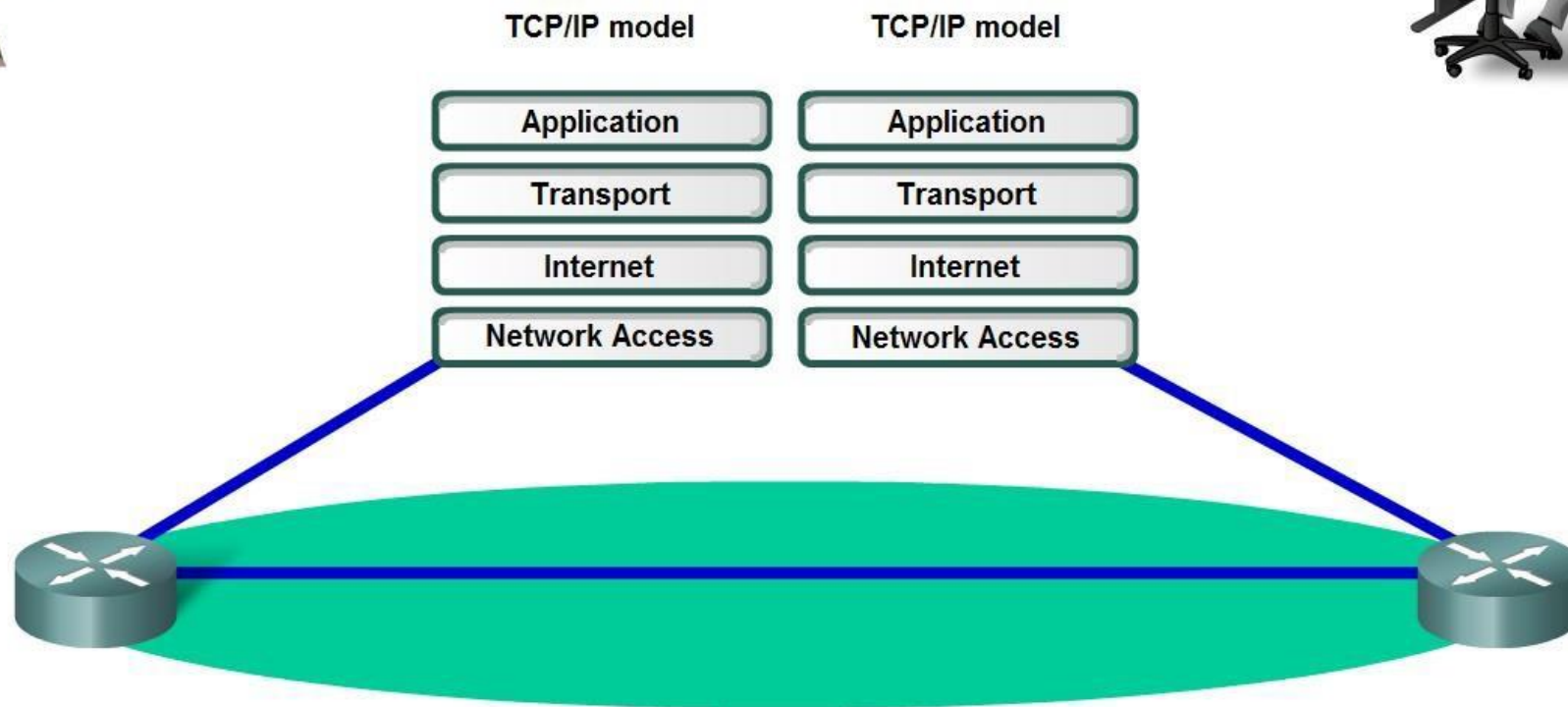
Layers with TCP/IP and OSI Model

- Describe TCP/IP Mode



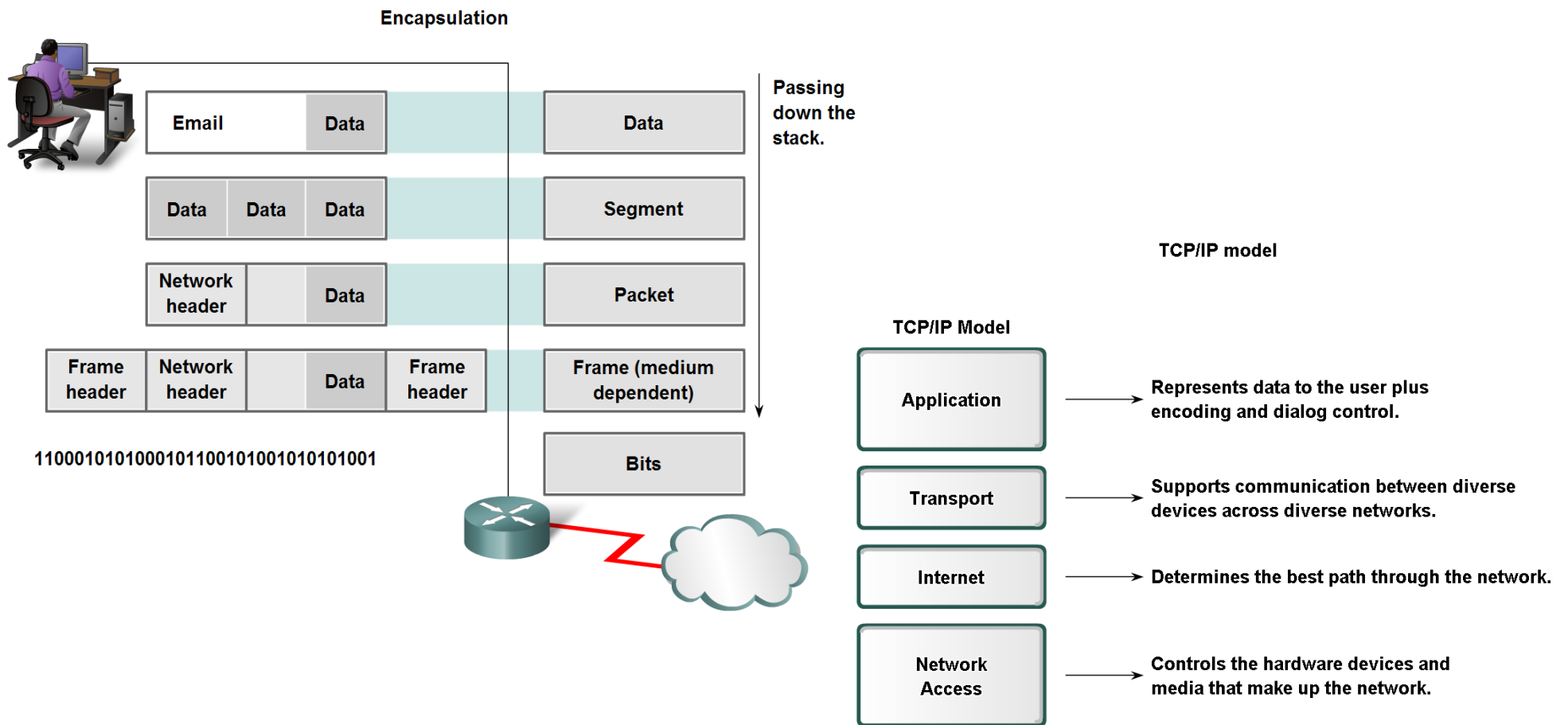
Layers with TCP/IP and OSI Model

- Describe the Communication Process



Layers with TCP/IP and OSI Model

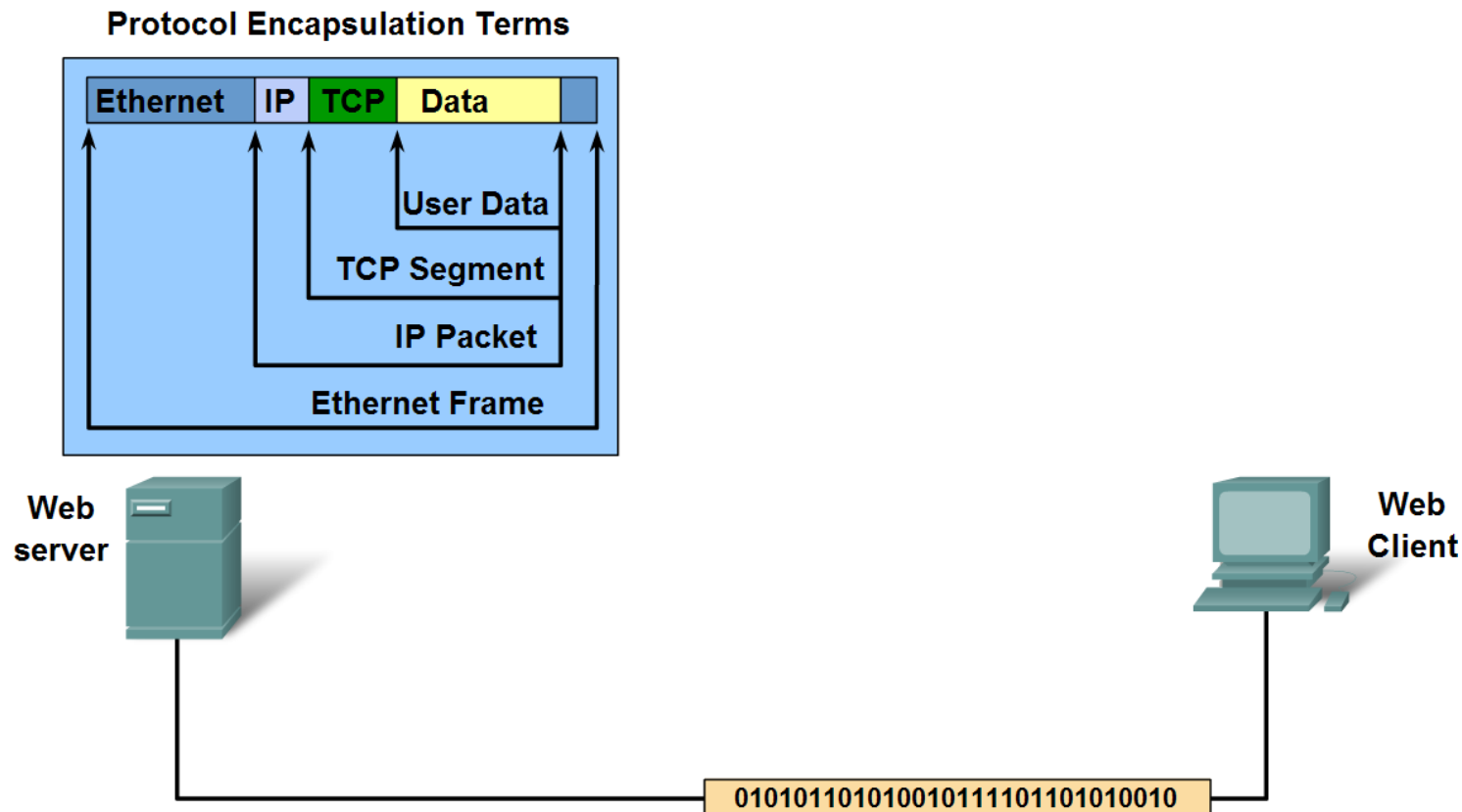
- Explain protocol data units (PDU) and encapsulation



Layers with TCP/IP and OSI Model

- Describe the process of sending and receiving messages

Protocol Operation of Sending and Receiving a Message



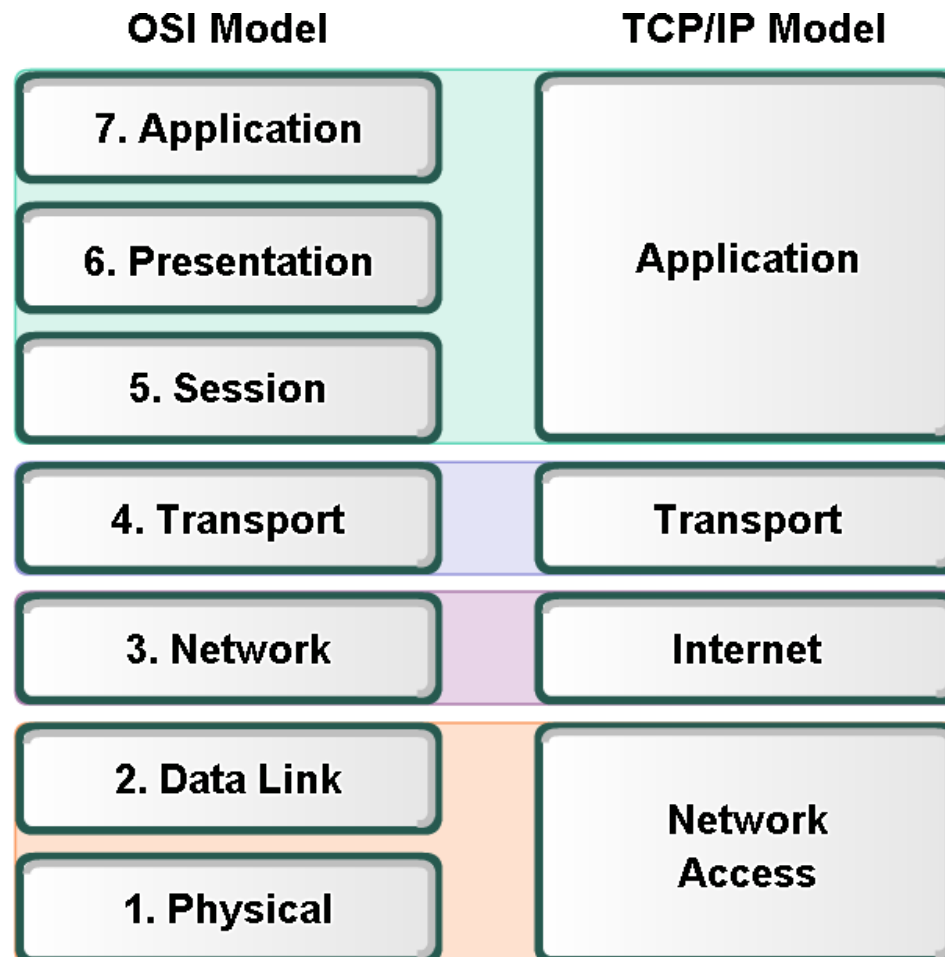
Layers with TCP/IP and OSI Model

- Define OSI



Layers with TCP/IP and OSI Model

- Compare OSI and TCP/IP model

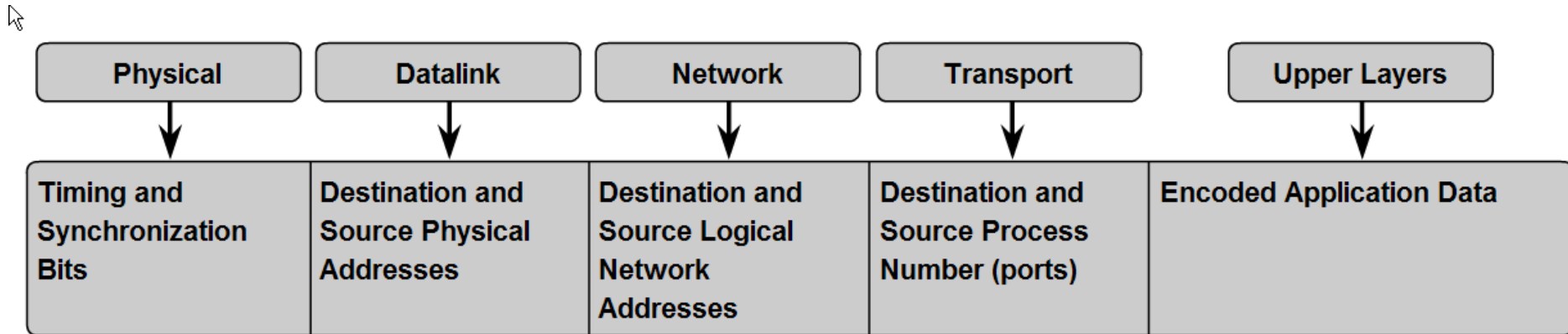


The key parallels are in the Transport and Network layers.



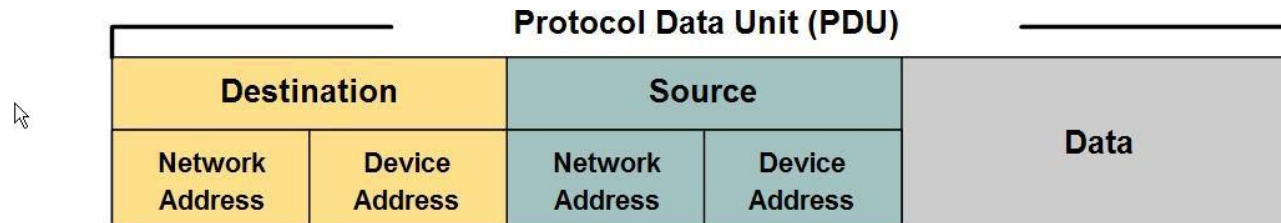
Addressing and Naming Schemes

- Explain how labels in encapsulation headers are used to manage communication in data networks

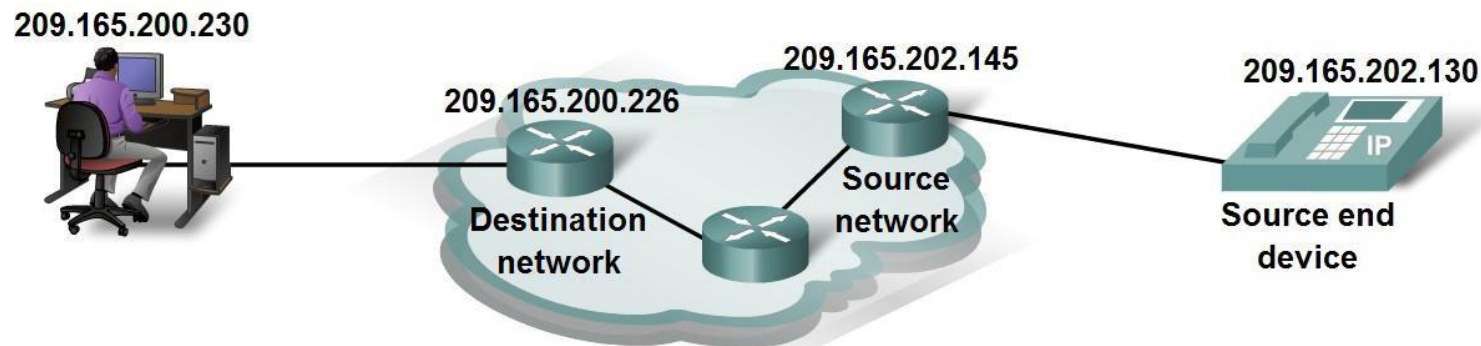


Addressing and Naming Schemes

- Explain how labels in encapsulation headers are used to manage communication in data networks



The Protocol Data Unit header also contains the network address.



The Application Layer

- Some important main applications
 - **HTTP**: The primary protocol used to communicate over the web
 - **FTP**: Allows different OSs to transfer files between one another
 - **SMTP** (Simple Mail Transfer Protocol) : Transmits email messages across the Internet
 - **SNMP** (Simple Network Management Protocol): Primarily used to monitor devices on a network, such as monitoring a router's state remotely → Should be disabled (Security problem)



The Application Layer

- Some important main applications

➤ **SSH**: Enables users to securely log on to a remote server and issue commands interactively

➤ **Telnet**: Enables users to insecurely log on to a remote server and issue commands interactively.



The Transport Layer

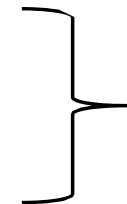
- Main function

- Data is encapsulated into segments
- Segments can be TCP or UDP

- TCP

- **Connection-oriented protocol**: Sender does not send any data to the destination until the destination node acknowledges that it's listening to the sender

- ✓ The sender sends a SYN packet to the receiver
- ✓ The receiver sends SYN-ACK to the sender
- ✓ The sender sends ACK to the receiver

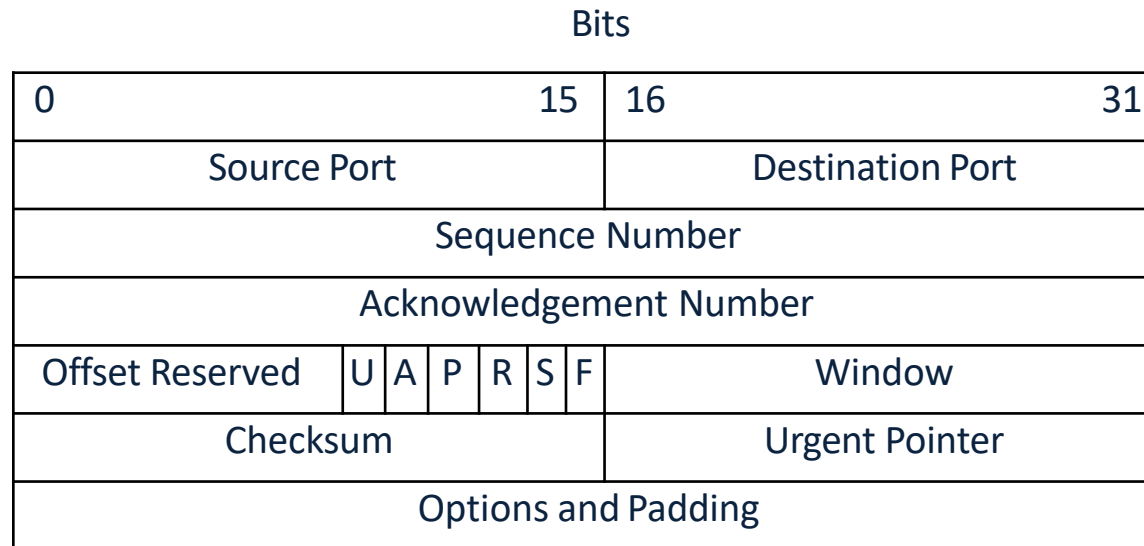


Three-way
handshake



The Transport Layer

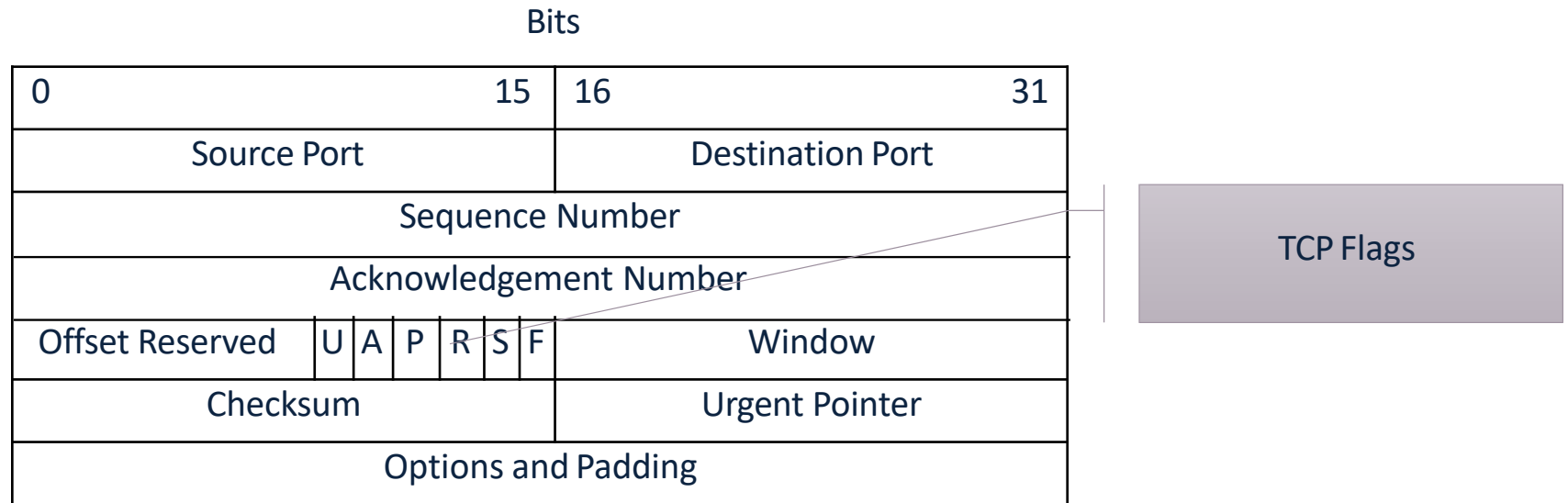
- TCP Segment Header



TCP Header Diagram

The Transport Layer

- TCP Segment Header



TCP Header Diagram

The Transport Layer

- TCP Flags

- Can be set 0 (off) or 1 (on)

- ✓ SYN flag – This signifies the beginning of a session
 - ✓ ACK flag – This acknowledges a connection request
 - ✓ PSH flag – This flag is used to deliver data directly to an application (Data is not buffered; it is sent immediately)
 - ✓ URG flag – This flag is used to signify urgent data
 - ✓ RST flag – This resets or drops a connection
 - ✓ FIN flag – The indicates that the connection is finished



The Transport Layer

- Sequence Number (SN)

- The SN is a 32-bit number, which allows sending receiving nodes to keep track of data transmission. (Data ordering/missing packets)

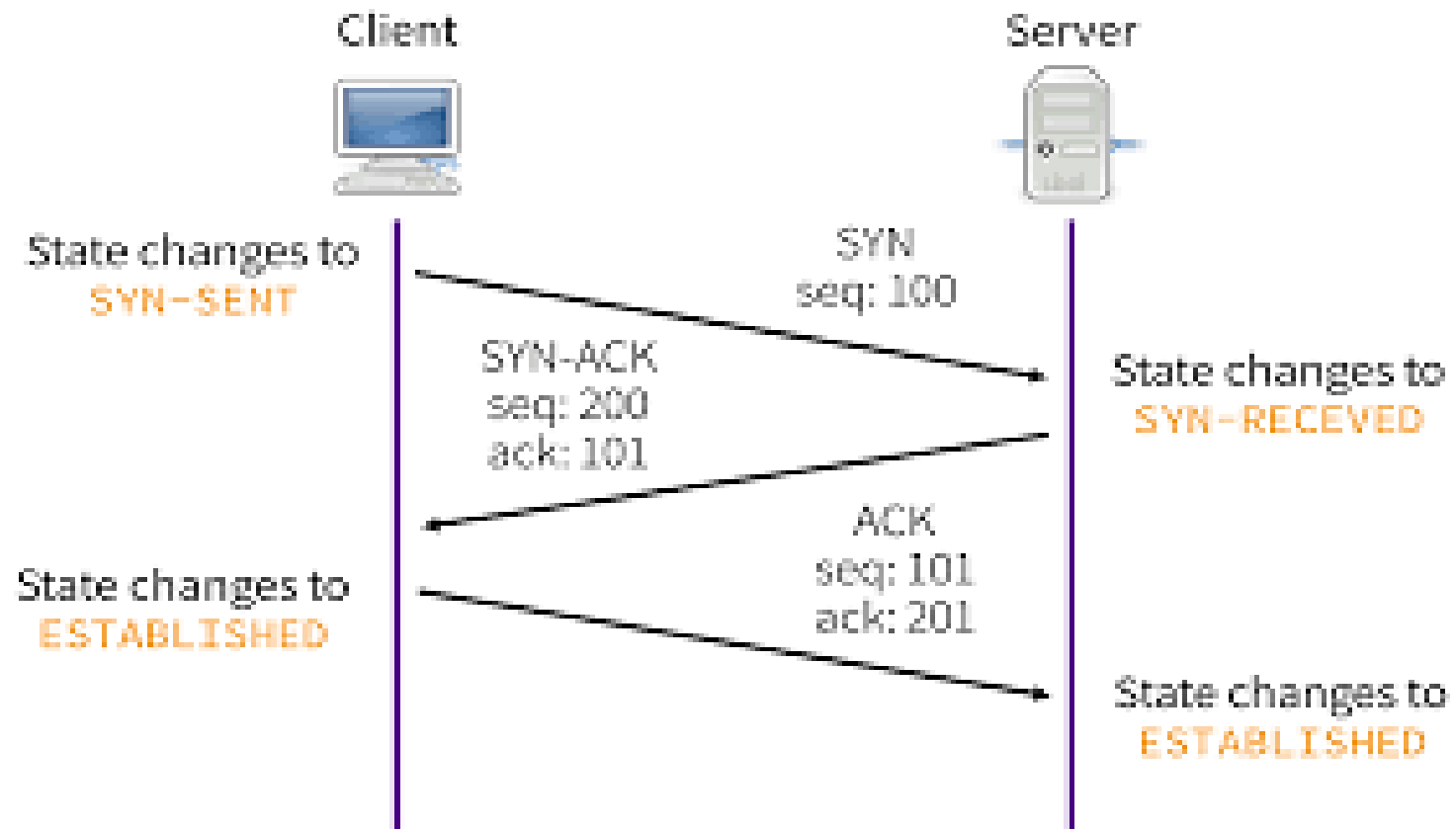
- The three-way handshake with SNs:

1. A sequence number from the sending node (SENDER_SN) is put in the Sequence Number field of a SYN segment and is sent to the receiving node.
2. The receiving nodes extracts the SENDER_SN from the SYN segment and puts SENDER_SN+1 in the Acknowledgement Number field; also the receiving nodes selects a RECEIVER_SN and puts it in the Sequence Number field of a SYN-ACK segment, which is sent to the sending node.
3. The sending nodes extracts the RECEIVER_SN from the SYN-ACK segment and puts RECEIVER_SN+1 in the Acknowledgement Number field; also the receiving nodes puts SENDER_SN+1 in the sequence number field of a ACK segment, which is sent to the receiving node.



The Transport Layer

- Illustration (Three-way handshake with SN)



The Transport Layer

- TCP Ports

- 16 bits for both source and destination ports
- A port is a logical component of TCP connection assigned to a process requiring network connectivity
- A port is the way a client program specifies a specific server program
- Some important ports (to remember)
 - ✓ 20 and 21: FTP, 20 – data transfer, 21 – control
 - ✓ 25: SMTP
 - ✓ 53: DNS
 - ✓ 80: HTTP
 - ✓ 443: HTTPS (Secure HTTP)
 - ✓ 110: POP3 (Post Office Protocol 3)



The Transport Layer

- UDP

- UDP does not need to verify whether the receiver is listening or ready to accept the packets

- ✓ It has no handshaking dialogues

- ✓ No guarantee of delivery, ordering and/or duplicate protection

- **Connectionless** transmission

- Unreliable but fast

- ✓ Used in applications in which queries must be fast and only consist of a single request such as DNS, SNMP and DHCP



The Internet Layer

- Main function
 - Responsible for routing a packet (datagram) to a destination address
 - Routing is done by using an IP address
 - Connectionless transmission like UDP



The Internet Layer

- IP address basics (IPv4)

- IP address: 4 numbers separated by decimals; each number represents 1 byte = 8 bits = 1 *octet*

- ✓ Ex) 128.214.18.16 = 1000000.11010110.00010010.00010000 (Binary)

- Each IP address is separated into a network address and a host address ➔ One part of the IP address represents a network prefix and the other part represents a host

- ✓ EX) 192.168.1.231 belongs to Class C address, which uses the first three numbers to identify the network and the last number to identify the host



The Internet Layer

- IP address classes

- Class A address – network.host.host.host

- ✓ The first byte is reserved for the network address (network prefix) and the last three bytes are assigned to host computers

- ✓ Range: 1.x.x.x to 126.x.x.x (127.x.x.x is reserved for loopback IP)

- Class B address – network.network.host.host

- ✓ The first two bytes are reserved for the network address and the last two bytes are assigned to host computers

- ✓ Range: 128.0.x.x to 191.255.x.x

- Class C address – network.network.network.host

- ✓ The first three bytes are reserved for the network address and the last one byte is assigned to host computers

- ✓ Range: 192.0.0.x to 223.255.255.x

- Class D for multicast, Class E for R&D and study



The Internet Layer

- The number of IP addresses available for hosts
 - $2^x - 2$, where x is the number of bits in an octet of host
 - For example in Class C address, x = 8, so the number of hosts = $2^8 - 2 = 254$
 - The reason why 2 is subtracted is that x.x.x.0 is not used and x.x.x.255 is reserved as broadcast address
 - How many IP addresses are available for hosts in Class B address?
 $2^{16} - 2 = 65534$



The Internet Layer

- Subnet mask

- A bitmask that yields the network prefix (network address) when applied by a **bitwise AND operation with any IP address** in the network

- Example

- ✓ Network prefix of an IP address 192.168.54.3 with a subnet mask 255.255.255.0 is 192.168.54
 - ✓ An IP address 192.168.54.3 with a subnet mask 255.255.0.0 produces a network prefix 192.168 ➔ This means that to be on the same network, two machines must have IP addresses starting with 192.168



The Internet Layer

- Default gateway (First-hop router)

- The default gateway is a router to which a host computer sends traffic if it does not know where the destination IP address
- If a computer tries to communicate with another computer on the same network it sends the data directly to it
- If the computer is on a separate network it forwards the data to whatever IP address is specified in the default gateway. Because a router will be attached to multiple networks it knows where these other networks are so it can route traffic to them
- Routers also have default gateways so that if they do not know where the destination is it also sends the data onto its own default gateway.
- This continues up the IP network hierarchy until it eventually finds a router that is part of the destination network.



The Internet Layer

- IPv6 addressing

- Developed to increase the space of IP address (The size of IPv4 address space = 2^{32})
- IPv6 uses 16 bytes address: The size of IPv6 address space = 2^{128}
- Example:

✓ 1111:0cb7:75a2:0110:1234:3a2e:1113:7777

1111 → 0001 0001 0001 0001

0cb7 → 0000 1100 10110111

75a2 → 0111 0101 10100010

0110 → 0000 0001 0001 0000

1234 → 0001 0010 0011 0100

3a2e → 0011 1010 0010 1110

1113 → 0001 0001 0001 0011

7777 → 0111 0111 0111 0111

128-bit representation of Ipv6
address



The Internet Layer

- ICMP (Internet Control Message Protocol)
 - It is used by network devices, including routers, **to send error messages and operational information** such as “Requested service is not available” or “Destination network unreachable”
 - It is used in the `ping` tool
 - ✓ A source sends ICMP `ECHO_REQUEST` to a destination system
 - ✓ If the system is live, it will respond by sending ICMP `ECHO_REPLY`
 - It is also used in the `traceroute` tool
 - ✓ In UNIX-like OS, `traceroute` has an option (`-I`) to use ICMP (By default, UDP is used)



Introduction to Wireshark

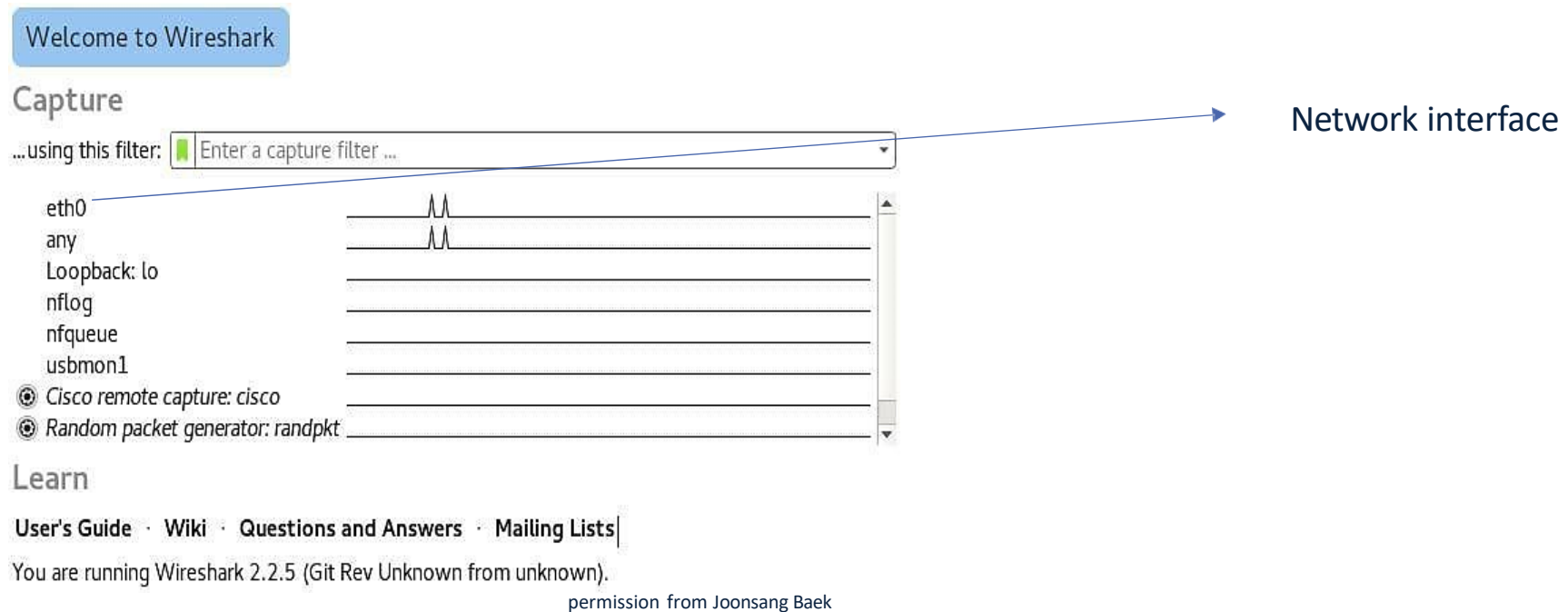
- Wireshark
 - A well-known network analysis tool previously known as Ethereal.
 - It captures packets in real time and displays them in human-readable format.
 - Main features include filters and color coding to analyse network traffic and inspect individual packets.



Capturing packets

- Selecting interfaces

- After launching Wireshark, a user can select a network interface and start capturing packets on that interface.

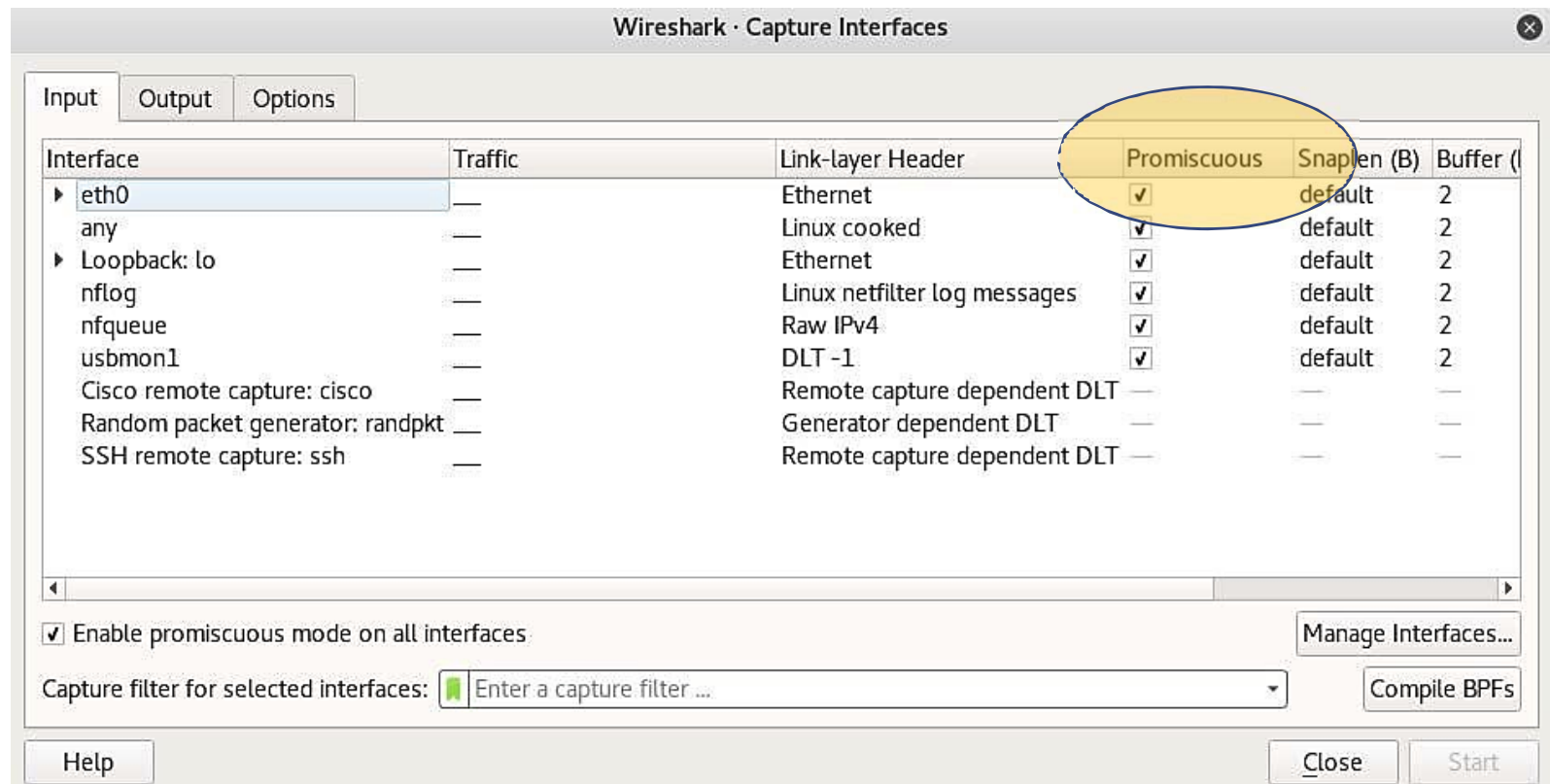


Capturing Traffic

- Promiscuous mode
 - Promiscuous mode is a mode for a wired network interface controller (NIC) or wireless network interface controller (WNIC) that causes the controller to **pass all traffic** it receives to the CPU.
 - This mode is normally used for packet sniffing which may take place on a router or on a computer connected to a hub or a part of a WLAN.
 - By default, Wireshark runs in promiscuous mode, but can be updated in the Capture ➔ Option panel



Capturing Traffic

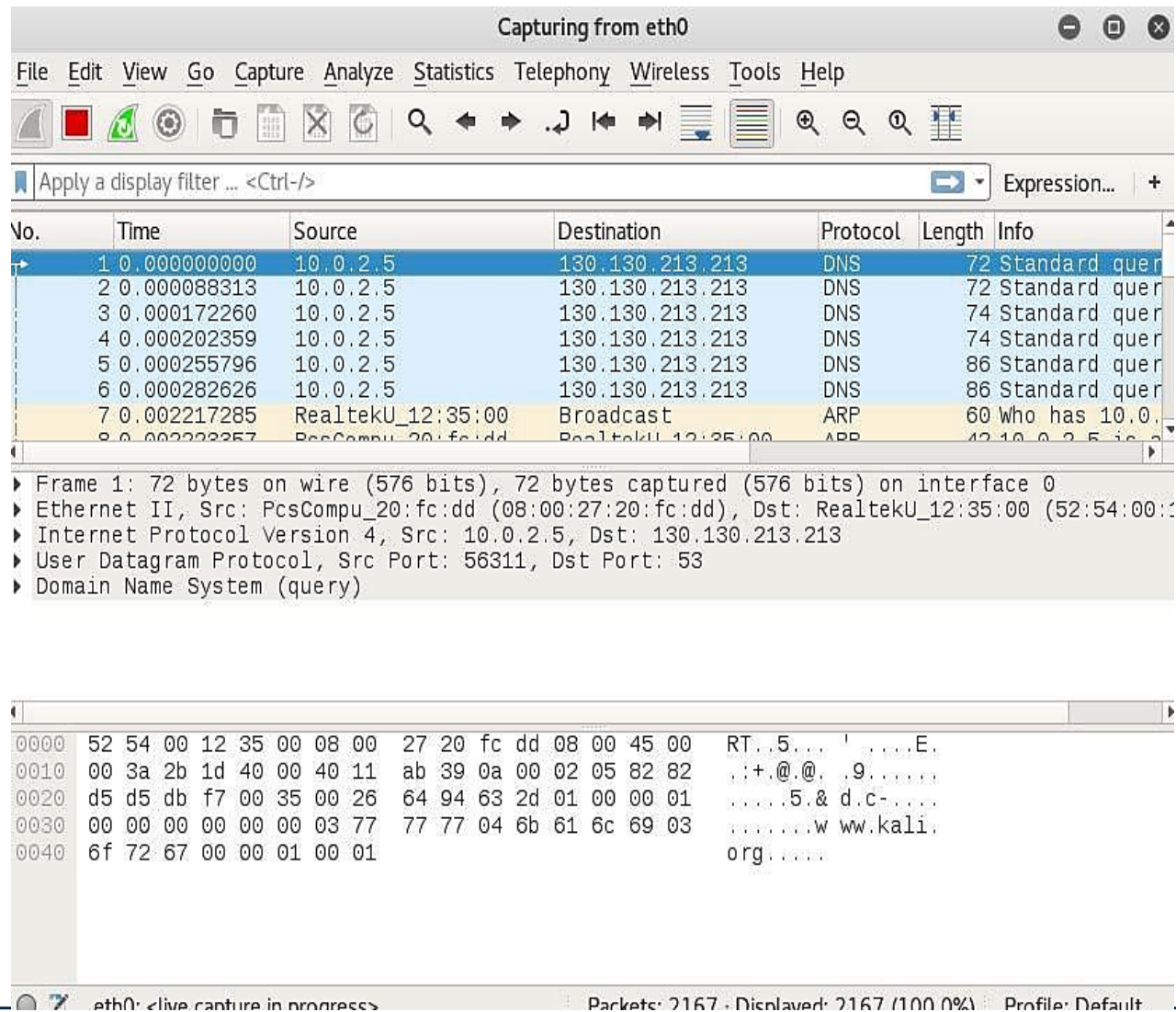


Capturing Traffic

- Packet capturing: Upon selecting a network interface, the packets start to appear in real time; Wireshark captures each packet sent to or from the system.
- In promiscuous mode, a user can see all the other packets on the network instead of only packets addressed to the user's network adapter.



Capturing Traffic



Capturing from eth0

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/> Expression...

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.2.5	130.130.213.213	DNS	72	Standard query
2	0.000088313	10.0.2.5	130.130.213.213	DNS	72	Standard query
3	0.000172260	10.0.2.5	130.130.213.213	DNS	74	Standard query
4	0.000202359	10.0.2.5	130.130.213.213	DNS	74	Standard query
5	0.000255796	10.0.2.5	130.130.213.213	DNS	86	Standard query
6	0.000282626	10.0.2.5	130.130.213.213	DNS	86	Standard query
7	0.002217285	RealtekU_12:35:00	Broadcast	ARP	60	Who has 10.0.
8	0.002222257	PcsCompu_20:fc:dd	RealtekU_12:35:00	ARP	42	10.0.2.5 is a

Frame 1: 72 bytes on wire (576 bits), 72 bytes captured (576 bits) on interface 0

- Ethernet II, Src: PcsCompu_20:fc:dd (08:00:27:20:fc:dd), Dst: RealtekU_12:35:00 (52:54:00:12:35:00)
- Internet Protocol Version 4, Src: 10.0.2.5, Dst: 130.130.213.213
- User Datagram Protocol, Src Port: 56311, Dst Port: 53
- Domain Name System (query)

Offset	Hex	ASCII
0000	52 54 00 12 35 00 08 00 27 20 fc dd 08 00 45 00	RT..5... '....E.
0010	00 3a 2b 1d 40 00 40 11 ab 39 0a 00 02 05 82 82	..+.@.@.9.....
0020	d5 d5 db f7 00 35 00 26 64 94 63 2d 01 00 00 015.& d.c-....
0030	00 00 00 00 00 00 03 77 77 77 04 6b 61 6c 69 03w ww.kali.
0040	6f 72 67 00 00 01 00 01	org.....

eth0: <live capture in progress> Packets: 2167 • Displayed: 2167 (100.0%) Profile: Default



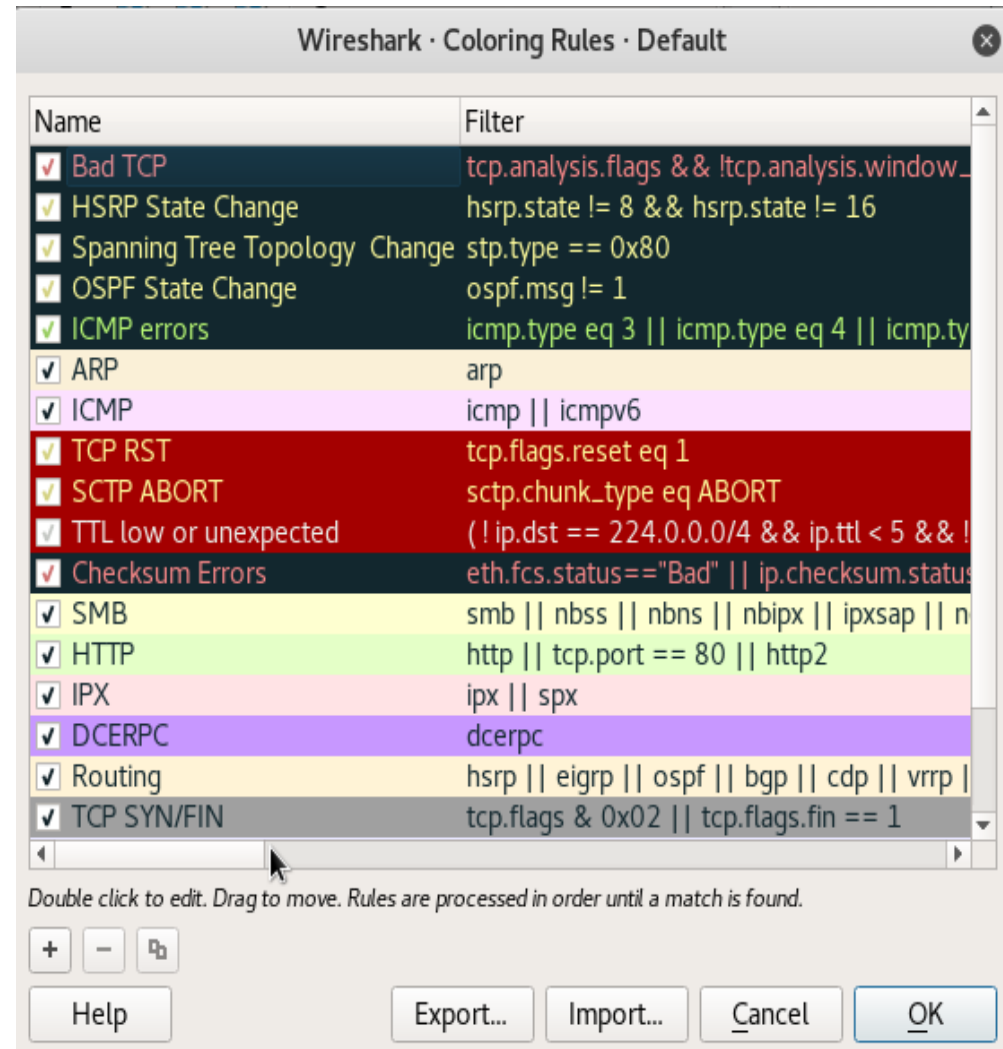
Color Coding

- Wireshark uses colors to identify the types of traffic at a glance.
- Examples (by default):
 - Light purple: TCP traffic
 - Light blue: UDP traffic
 - Black: Packets with errors
- Current (default) color setting can be seen on View
 - ➔ Coloring Rules (Modification is possible)

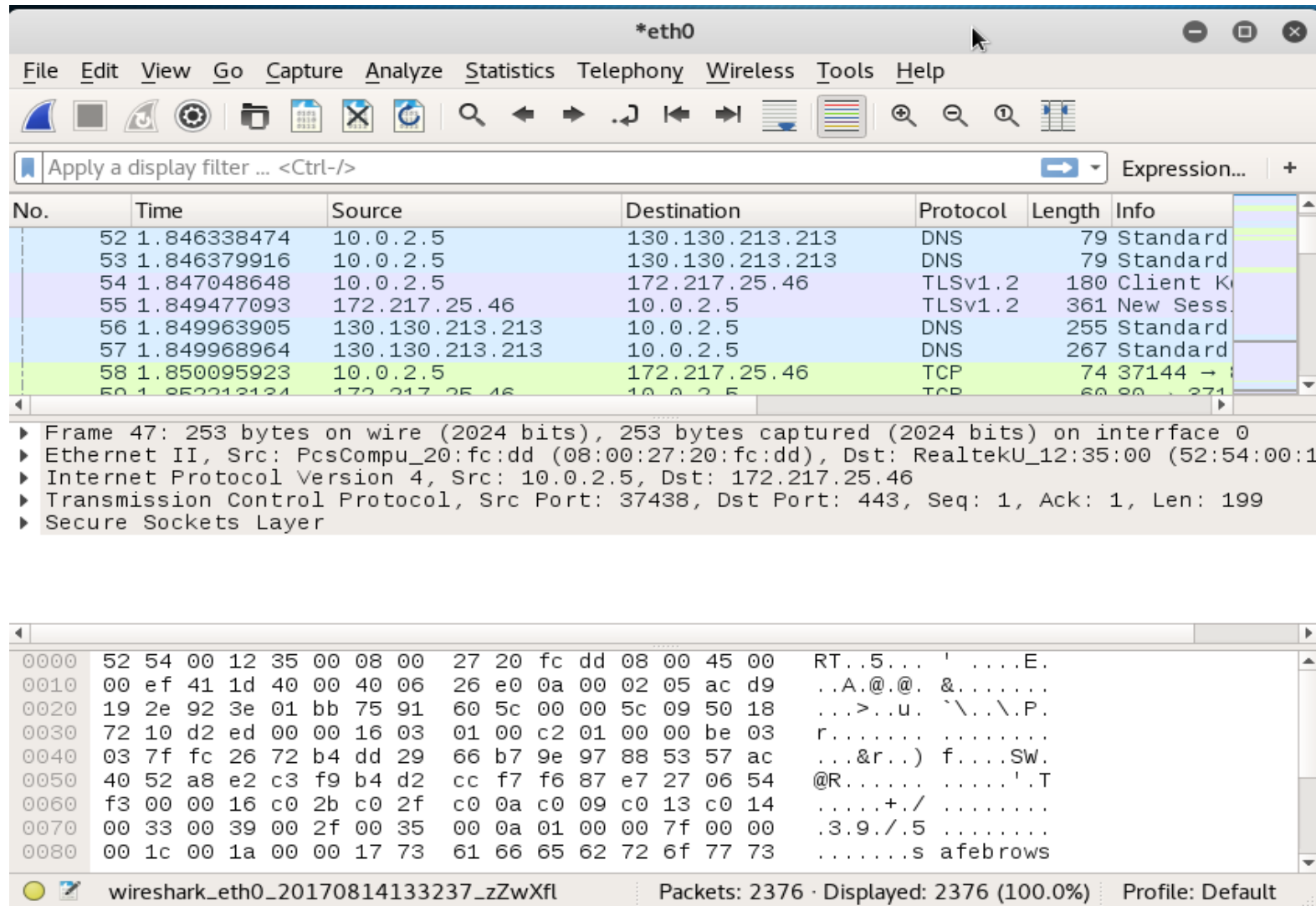


Color Coding

- Default color coding rules



Color Coding



The image shows a Wireshark packet capture window titled '*eth0'. The interface includes a menu bar (File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, Help) and a toolbar with various icons. A display filter bar at the top shows 'Apply a display filter ... <Ctrl-/>' and 'Expression... +'. The main packet list table has columns: No., Time, Source, Destination, Protocol, Length, and Info. Packets are color-coded: blue for DNS, purple for TLSv1.2, and green for TCP. Packet 58 is highlighted in green. Below the packet list, the packet details pane shows the structure of packet 47: Ethernet II, Internet Protocol Version 4, Transmission Control Protocol, and Secure Sockets Layer. At the bottom, the packet bytes pane shows the raw data in hexadecimal and ASCII. The status bar at the bottom indicates 'wireshark_eth0_20170814133237_zZwXfl', 'Packets: 2376 · Displayed: 2376 (100.0%)', and 'Profile: Default'.

No.	Time	Source	Destination	Protocol	Length	Info
52	1.846338474	10.0.2.5	130.130.213.213	DNS	79	Standard
53	1.846379916	10.0.2.5	130.130.213.213	DNS	79	Standard
54	1.847048648	10.0.2.5	172.217.25.46	TLSv1.2	180	Client K
55	1.849477093	172.217.25.46	10.0.2.5	TLSv1.2	361	New Sess.
56	1.849963905	130.130.213.213	10.0.2.5	DNS	255	Standard
57	1.849968964	130.130.213.213	10.0.2.5	DNS	267	Standard
58	1.850095923	10.0.2.5	172.217.25.46	TCP	74	37144 →
59	1.852212124	172.217.25.46	10.0.2.5	TCP	60	80 → 271

▶ Frame 47: 253 bytes on wire (2024 bits), 253 bytes captured (2024 bits) on interface 0
▶ Ethernet II, Src: PcsCompu_20:fc:dd (08:00:27:20:fc:dd), Dst: RealtekU_12:35:00 (52:54:00:12:35:00)
▶ Internet Protocol Version 4, Src: 10.0.2.5, Dst: 172.217.25.46
▶ Transmission Control Protocol, Src Port: 37438, Dst Port: 443, Seq: 1, Ack: 1, Len: 199
▶ Secure Sockets Layer

0000 52 54 00 12 35 00 08 00 27 20 fc dd 08 00 45 00 RT..5... 'E.
0010 00 ef 41 1d 40 00 40 06 26 e0 0a 00 02 05 ac d9 ..A.@.@. &.....
0020 19 2e 92 3e 01 bb 75 91 60 5c 00 00 5c 09 50 18 ...>..u. \...\P.
0030 72 10 d2 ed 00 00 16 03 01 00 c2 01 00 00 be 03 r.....
0040 03 7f fc 26 72 b4 dd 29 66 b7 9e 97 88 53 57 ac ...&r..) f....SW.
0050 40 52 a8 e2 c3 f9 b4 d2 cc f7 f6 87 e7 27 06 54 @R..... ' .T
0060 f3 00 00 16 c0 2b c0 2f c0 0a c0 09 c0 13 c0 14+./
0070 00 33 00 39 00 2f 00 35 00 0a 01 00 00 7f 00 00 .3.9./ .5
0080 00 1c 00 1a 00 00 17 73 61 66 65 62 72 6f 77 73s afebrows

wireshark_eth0_20170814133237_zZwXfl Packets: 2376 · Displayed: 2376 (100.0%) Profile: Default

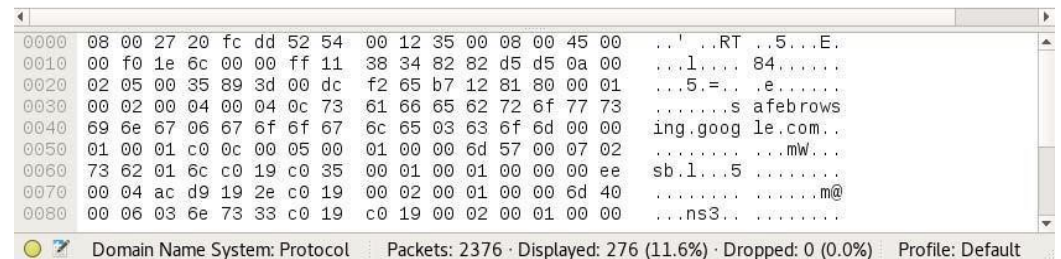
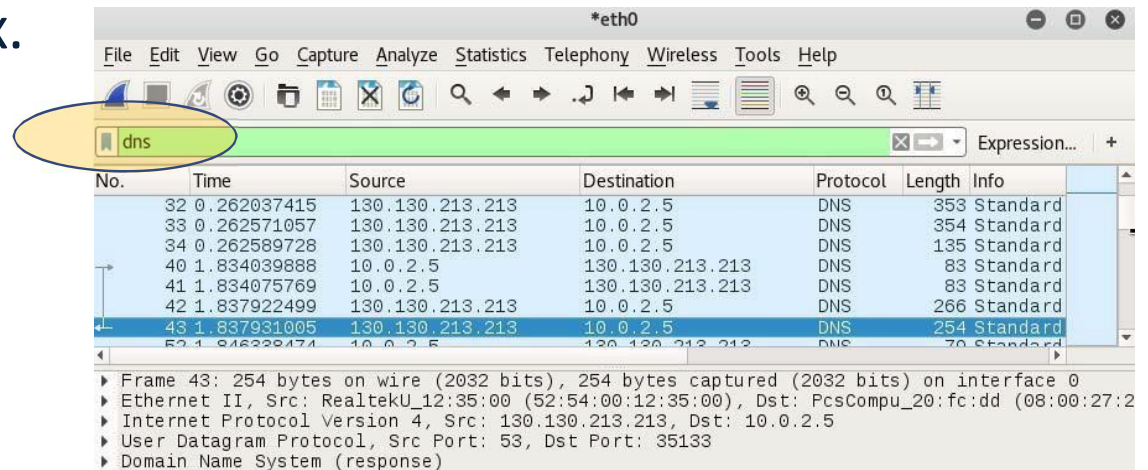
Packets captures
by Wireshark



Filtering Packets

- Filter Box

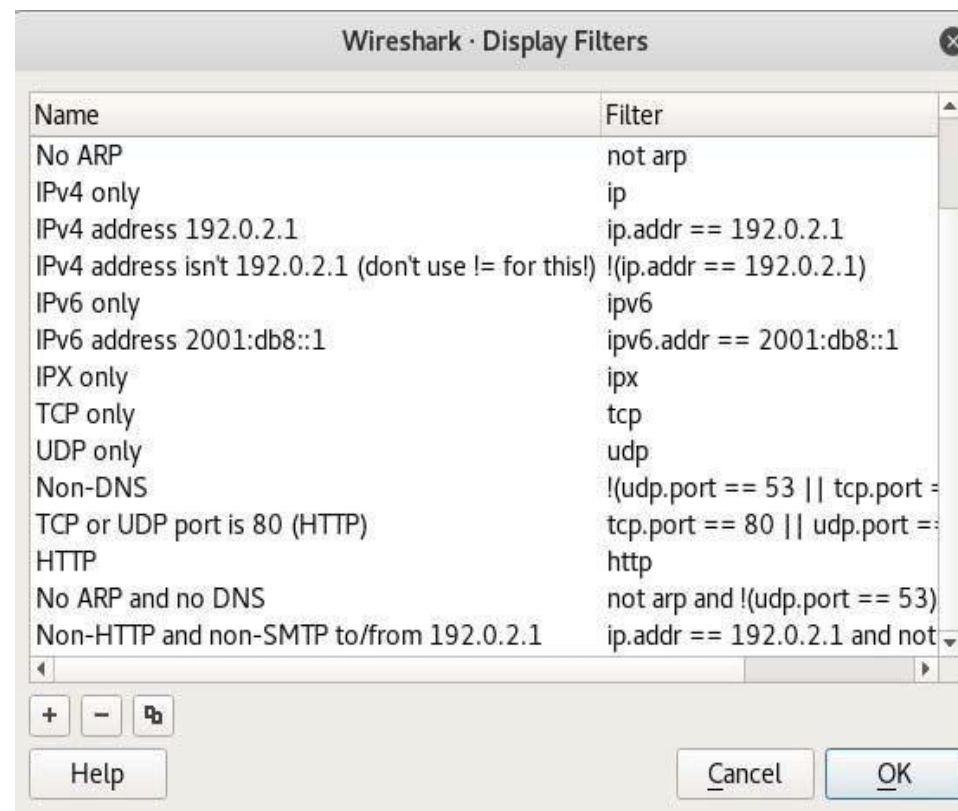
- The most basic way to filter packets in Wireshark is to enter keywords in the Filter Box.



Filtering Packets

- Default filters

➤ Analyze ➔ Display Filters will list default filters included in Wireshark.



Inspecting Packets Example (1)

- Another interesting thing you can do is right-click a packet and select **Follow → TCP Stream**.
 - You'll see the full TCP conversation between the client and the server. You can also click other protocols in the Follow menu to see the full conversations for other protocols, if applicable.
 - Next slide (Screen shot)



Inspecting Packets Examples (1)

The image displays the Wireshark network protocol analyzer interface. The main window is titled "Wireshark · Follow TCP Stream (tcp.stream eq 2) · wireshark_eth0_2017...". The left pane shows the packet list with a selected packet (37144) and its details. The right pane shows the packet details for the selected packet, including the "Info" section and the "Request" section. The "Info" section shows the packet details: 37144 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS. The "Request" section shows the HTTP request details: POST /ocsp HTTP/1.1, Host: clients1.google.com, User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:45.0) Gecko/20100101 Firefox/45.0, Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8, Accept-Language: en-US,en;q=0.5, Accept-Encoding: gzip, deflate, Content-Length: 75, Content-Type: application/ocsp-request, Connection: keep-alive. The bottom pane shows the packet bytes in hexadecimal and ASCII.

Wireshark · Follow TCP Stream (tcp.stream eq 2) · wireshark_eth0_2017...

POST /ocsp HTTP/1.1
Host: clients1.google.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:45.0) Gecko/20100101 Firefox/45.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Length: 75
Content-Type: application/ocsp-request
Connection: keep-alive

0IOG0E0C0A0 ...+.....j.....p.I.#z...
(~d..J.....h.v....b..Z./...PN.....1HTTP/1.1 200 OK
Content-Type: application/ocsp-response
Date: Mon, 14 Aug 2017 04:15:09 GMT
Expires: Fri, 18 Aug 2017 04:15:09 GMT
Cache-Control: public, max-age=345600
Server: ocsp_responder
Content-Length: 463
X-XSS-Protection: 1; mode=block
X-Frame-Options: SAMEORIGIN

0...
.....0.....+.....
0.....0...0.....J.....h.v....b..Z./...
20170813192729Z0k0i0A0 ...+.....j.....p.I.#z...

6 client pkts, 6 server pkts, 9 turns.

Entire conversation (4700 by ▾ Show and save data as ASCII ▾ Stream 2 ▾

Find: Find Next

Info

37144 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS
80 → 37144 [SYN, ACK] Seq=0 Ack=1 Win=3276
37144 → 80 [ACK] Seq=1 Ack=1 Win=29200 Len
Request
80 → 37144 [ACK] Seq=1 Ack=430 Win=32339 L
Response
37144 → 80 [ACK] Seq=430 Ack=747 Win=30586
Request

bytes captured (592 bits) on interface 0
27:20:fc:dd), Dst: RealtekU_12:35:00 (52:54:
Dst: 172.217.25.46
144, Dst Port: 80, Seq: 0, Len: 0

00 45 00 RT...5...!...E.
05 ac d9 .<..@..@...
00 a0 02P4.....
0a 00 24 r.....\$
!C.....



Inspecting Packets Example (2)

- In fact, Wireshark will enable us to capture a username and a password entered to an insecure website
- A filter you need in this regard is
 - `http.request.method == "POST"`
 - Once you locate a packet right-click the packet and select Follow
 - ➔ TCP Stream
 - And examine the content



Inspecting Packets Example (3)

- By using Wireshark, connections based on the secure application protocol like SSL can be analysed.

3141	36.398756	10.9.26.59	162.125.34.129	TLSv1.2	237 Client Hello
3186	36.556681	162.125.34.129	10.9.26.59	TLSv1.2	1514 Server Hello
3187	36.556682	162.125.34.129	10.9.26.59	TLSv1.2	1514 Certificate [TCP segment of a reassembled PDU]
3188	36.556684	162.125.34.129	10.9.26.59	TLSv1.2	156 Server Key Exchange, Server Hello Done
3190	36.561857	10.9.26.59	162.125.34.129	TLSv1.2	180 Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
3238	36.716422	162.125.34.129	10.9.26.59	TLSv1.2	296 New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
3239	36.719132	10.9.26.59	162.125.34.129	TLSv1.2	473 Application Data

An example of connection based on SSL



Any Questions?

Let's do the activity using Wireshark for capturing and analyzing network packet logs 😊

