

ET0730

# Chapter 7

## Transport Layer Protocols

Singapore Polytechnic  
School of Electrical & Electronic Engineering

# Objectives

- Explain why Transport Layer is needed.
- Explain the OSI Seven Layer Model and TCP/IP model.
- Explain the two Transport Layer protocols: TCP and UDP.
- Understand the differences between TCP and UDP.
- Understand what Internet Socket is.
- Describe the types of port numbers.
- Understand the use of port numbers.

# Outline



- Introduction - The Need for Transport Layer
- Layered Network Models
  - OSI Seven Layer Model
  - TCP/IP Model
- Transport Layer Protocols
  - TCP
  - UDP
- Internet Sockets
- Port Number
  - Well-known Ports
  - Registered Ports
  - Dynamic/Private/Ephemeral Ports

# Introduction (1)

## A Busy Laptop

- Many communications may be going on simultaneously for a host.
- Example: you may be doing the following things (different “**applications**”) on your laptop at the same time:
  - browsing the Yahoo web page using Firefox,
  - watching an YouTube video using Chrome,
  - transferring a file from a file server,
  - having an instant messaging session with your friends using Skype, and
  - sending/reading emails using Outlook.

## Introduction (2):

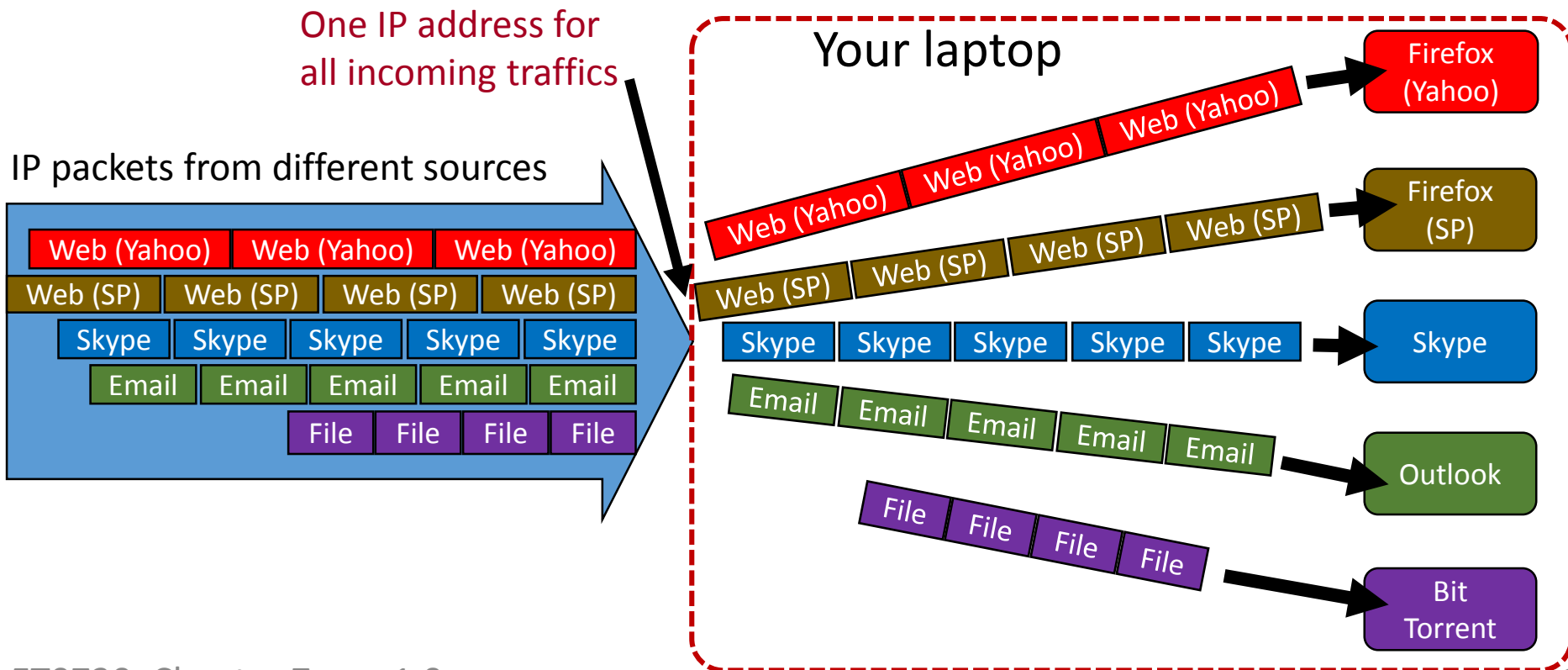
# Which IP Packet is for which application?

- For each communication activity that you are having over the internet, the data are carried in IP packets.
- Many IP packets are leaving your laptop (outgoing traffic) and coming to your laptop (incoming traffic) simultaneously.
- Question:
  - Your laptop has only one IP address.
  - When the IP packets come in, how are the IP packets directed to the correct application or process?

# Introduction (3)

## Sort out the IP Packets

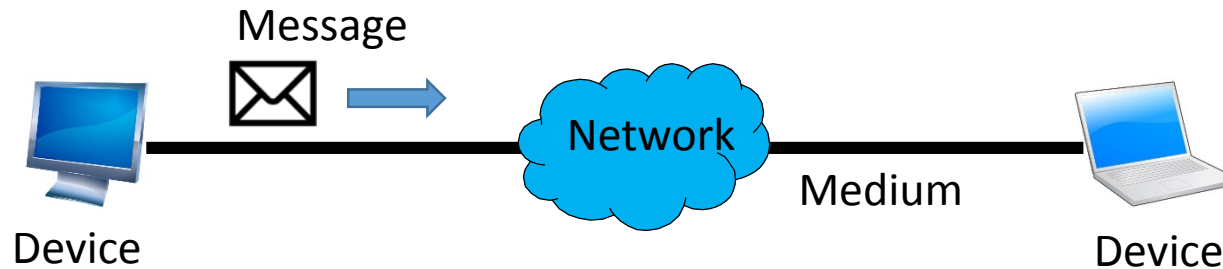
- In order to distinguish the IP packets belonging to different applications or processes, we need to provide **separate data channels** for different applications or processes.



# How do we provide separate data channels?

- Answer: Transport Layer Protocols
- The **Transport Layer** establishes a data channel for an application to achieve end-to-end data exchange.
- To understand how Transport Layer achieves this, we need to know:
  - Network Protocols
  - Layered Network Models
  - Transport Layer and its protocols

# Networking Protocols (1)



## Rules:

- Max. length of message=?
- Must start with ...
- What if fails to send...

- All networks have four basic elements in common:
  - Devices
  - Media
  - Messages
  - Rules
- In any communication system it is necessary to have a set of rules or procedures, which must be obeyed if information transfer is to be implemented successfully.



# Networking Protocols (2)

- Rules” or “Procedures” define how messages are sent, directed, received and interpreted.
- In computer networking, we call these rules or procedures the **Protocols**.
- There are many different protocols, each performs certain networking functions.

# Layered Network Models (1)

- What are Layered Network Models?
  - Layering is a structuring technique that groups network protocols according to their functions.
  - Each group is a layer of a layered network model.
  - Each layer is independent from every other in its purpose and responsibilities.
  - Each layer of the network uses the protocols and software of the layer below it.
  - Each layer communicates with the layer above it so that the higher layer can use the resources it provides.

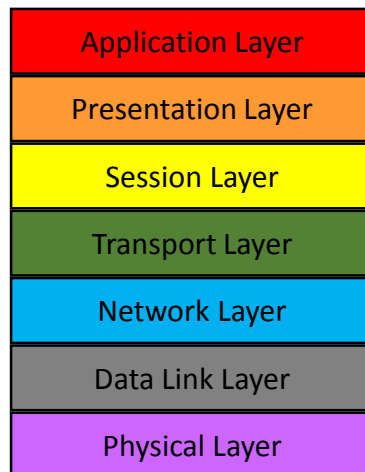
# Layered Network Models (2)

- Reasons for layering
  - Simplifies the network model.
  - Enables programmers to specialize in a particular level or layer of the networking model.
  - Provides design modularity.
  - Encourages interoperability.
  - Allows for standardized interfaces to be produced by networking vendors.

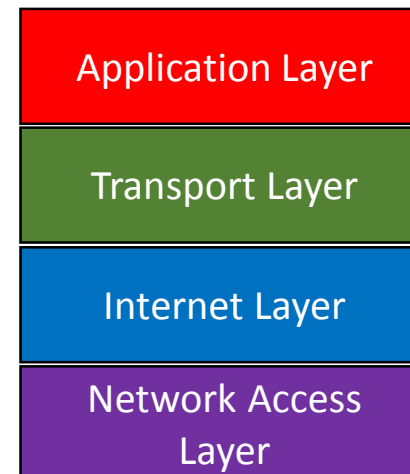
# Layered Network Models (3)

- In this chapter, we will learn about two important layered network models:
  - OSI Seven Layer Model
  - TCP/IP Model

OSI Seven Layer Model



TCP/IP Model



# OSI Seven Layer Model (1)

- Rapid growth in demand for data communication in the 1970's led to each manufacturer developed their own standards and protocols.
- Most manufacturer solutions were different and hence created problems with equipment compatibility.
- In 1977, International Standards Organisation (ISO) created a committee to unify networks.
- The committee established the concept of an "Open System", one in which standard protocols are used and hence **interconnection is easy** to implement.

# OSI Seven Layer Model (2)

- The committee also defined an Interconnection Reference Model, which defines the seven layers of communication protocol with specific functions associated with each layer.
- The “Open System Interconnection (OSI) Reference Model” or “7 Layer Model” is now an accepted International Standard for use in data communication networks.

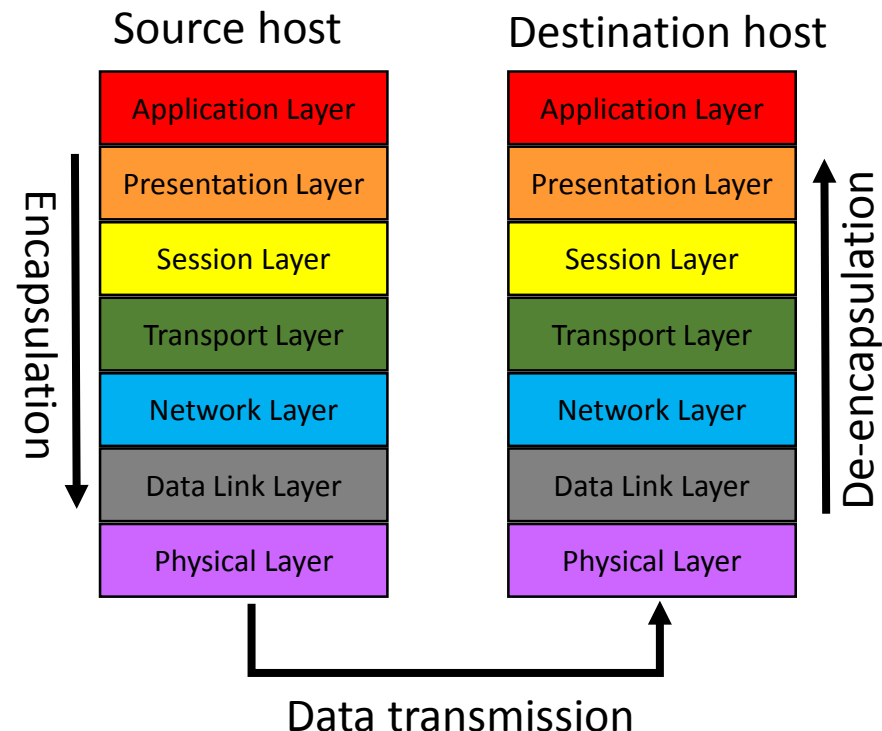
# OSI Seven Layer Model (3)

- In OSI Seven Layer Model, the protocols are organised into **7 layers** according to their **functions**.

<u>Main function of each layer:</u>		
Layer 7	Application Layer	Purpose for communication
Layer 6	Presentation Layer	Syntax conversion
Layer 5	Session Layer	Transmission control and order
Layer 4	Transport Layer	Ensures delivery
Layer 3	Network Layer	Routes data
Layer 2	Data Link Layer	Media access
Layer 1	Physical Layer	Bit conversion and transmission

# Data Encapsulation (1)

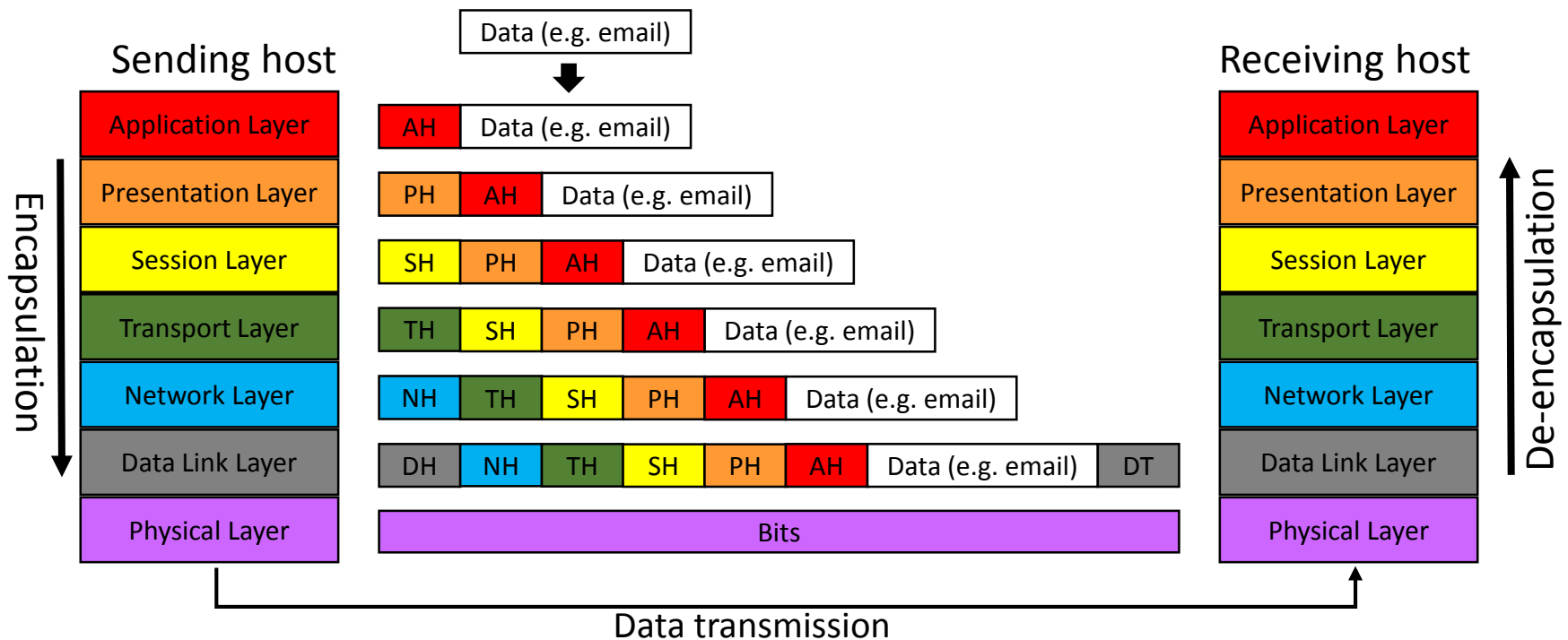
- When data is sent from the source host to the destination host, the data goes through
  - encapsulations at the source host
  - De-encapsulations at the destination host
- Encapsulation is the process of adding headers (packaging) to the data.
- De-encapsulation is the process of removing the headers (unpackaging).





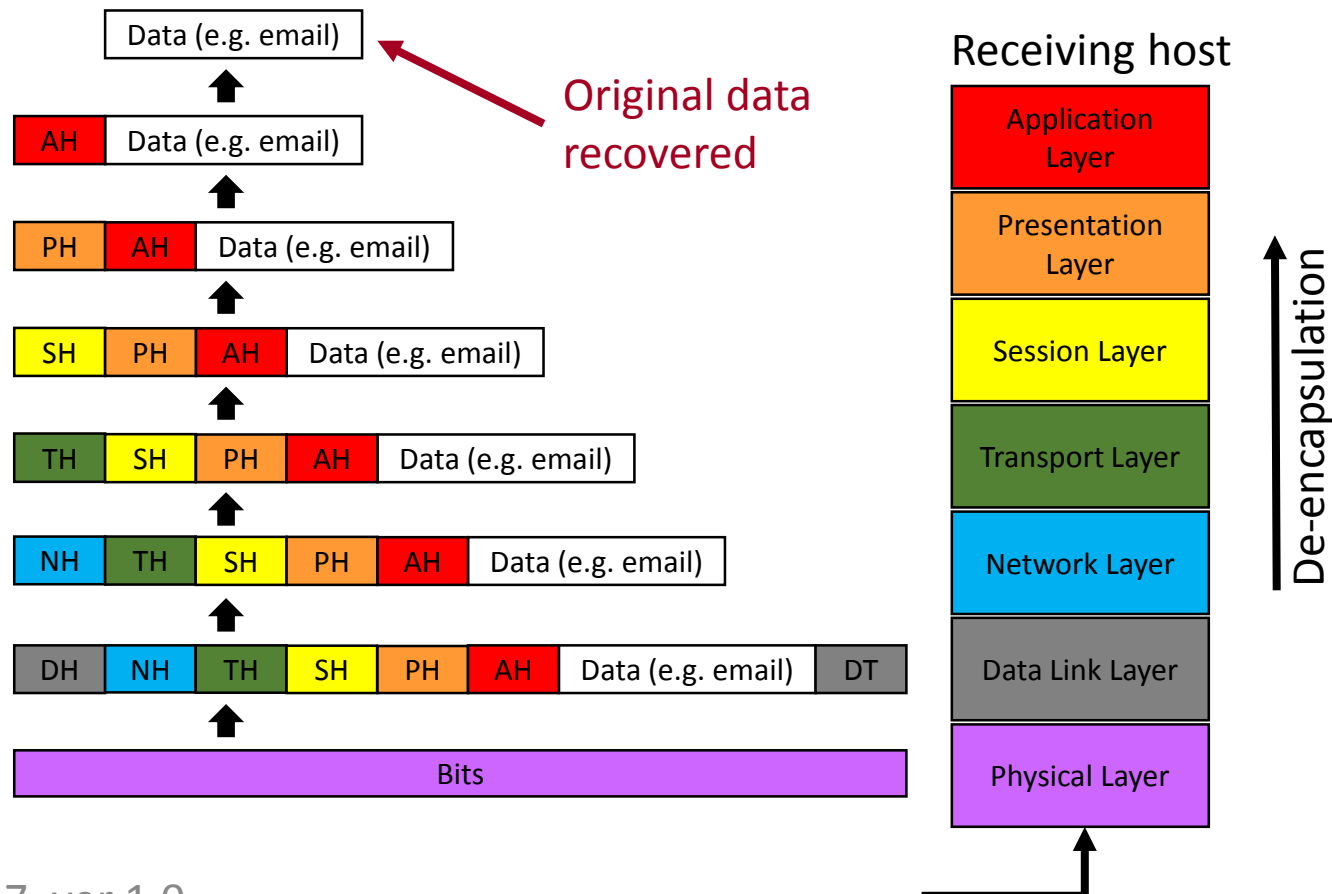
# Data Encapsulation (2)

- At the sending host, as the data (e.g. email) goes down the layers, the encapsulation process adds “header” to the data.
- Data Link Layer may add a “trailer” to the data too.
- Encapsulation process increases the size of the data packet.



# De-Encapsulation

- At the receiving host, as the data goes up the layers, the de-encapsulation process removes “header” from the data.
- Eventually the original data (e.g. email) is recovered.



# TCP/IP Model (1)

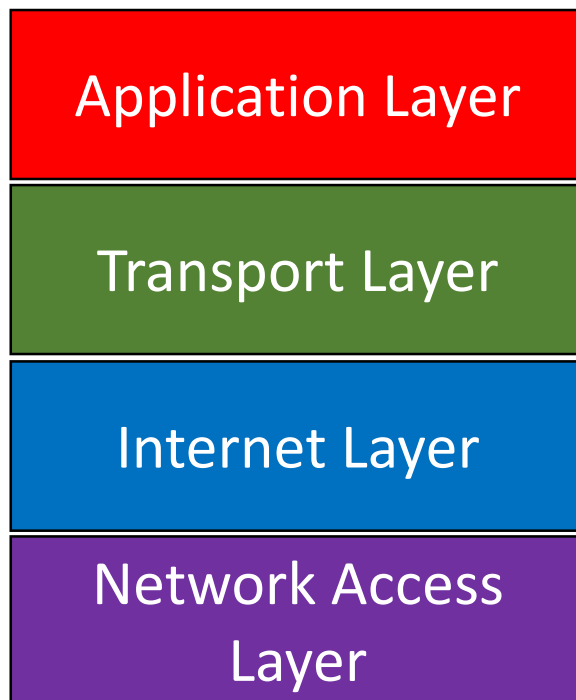
- For internet as well as other similar computer networks, a set of protocols have been developed over the years.
- This set of communication protocols are collectively known as “**Internet Protocol Suite**”
- Internet Protocol Suite is also commonly known as “**TCP/IP**” because of the two most important protocols:
  - TCP (Transmission Control Protocol)
  - IP (Internet protocol)

# TCP/IP Model (2)

- The development of the TCP/IP was funded by DARPA, an agency of the United States Department of Defense in the late 1960s.
- The TCP/IP model and related protocols are maintained by the Internet Engineering Task Force (IETF).
- Documents of IETF standards are call RFCs (Requests for Comments), downloadable from [www.ietf.org](http://www.ietf.org).

# TCP/IP Model (3)

- The protocols in TCP/IP Model are organised into **4 layers**, according to the scope of networking involved.



## Main function of each layer:

Provides process-to-process application data exchange.

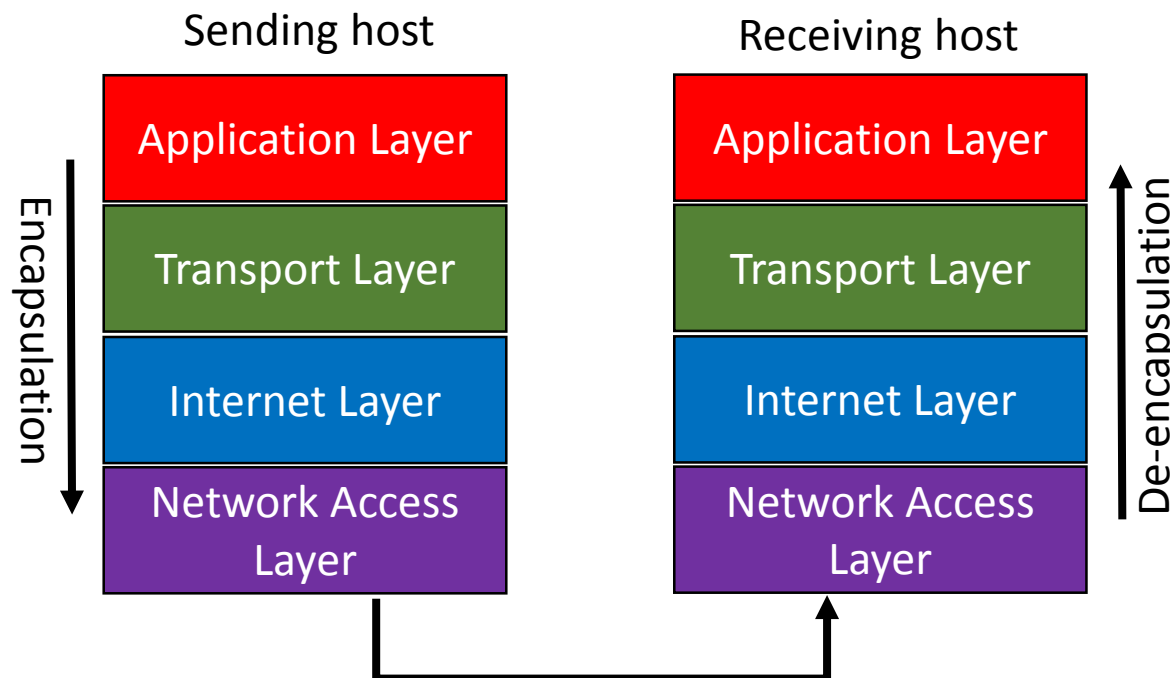
Handles host-to-host communication.

Connects hosts across networks.

Handles communication for a single network segment (link).

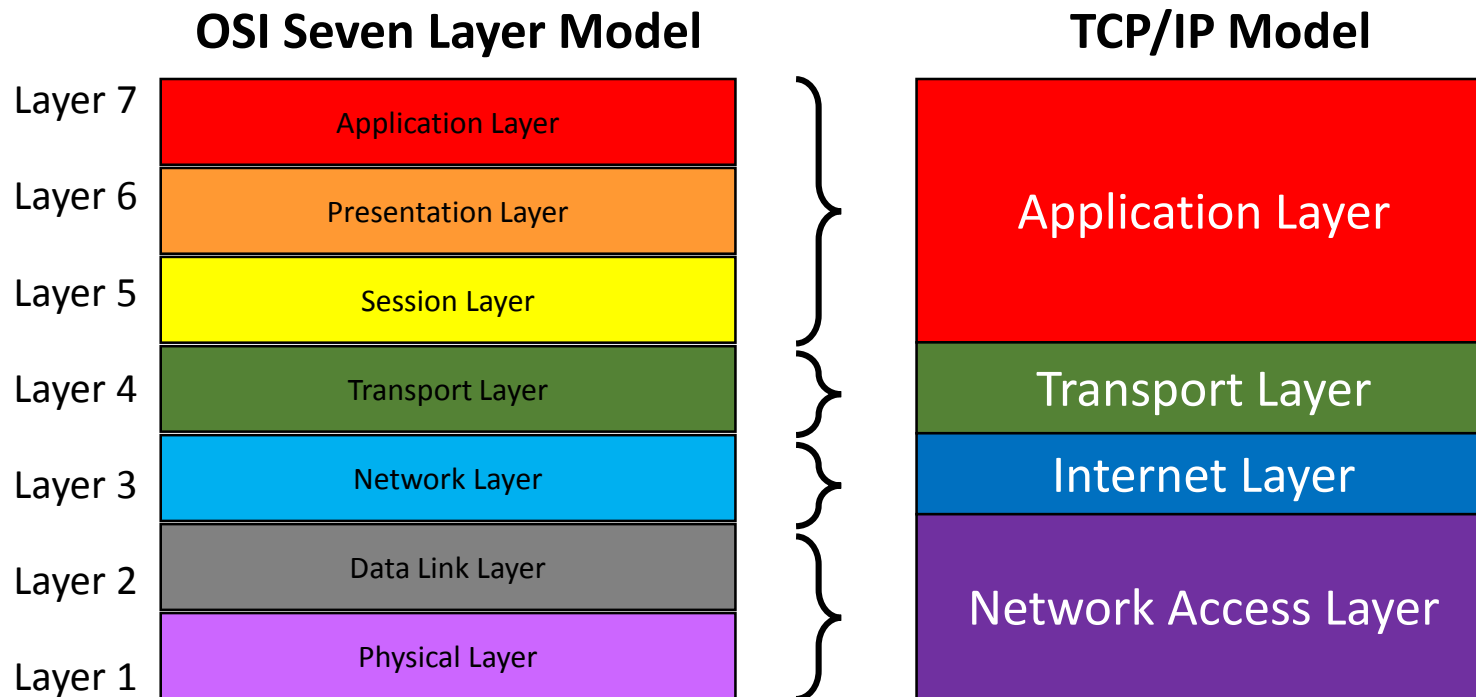
# TCP/IP Model (4)

- The TCP/IP Model also uses the same concept of encapsulation and de-encapsulation of data as the data goes through the layers.



# Mapping between OSI Seven Layer Model and TCP/IP Model

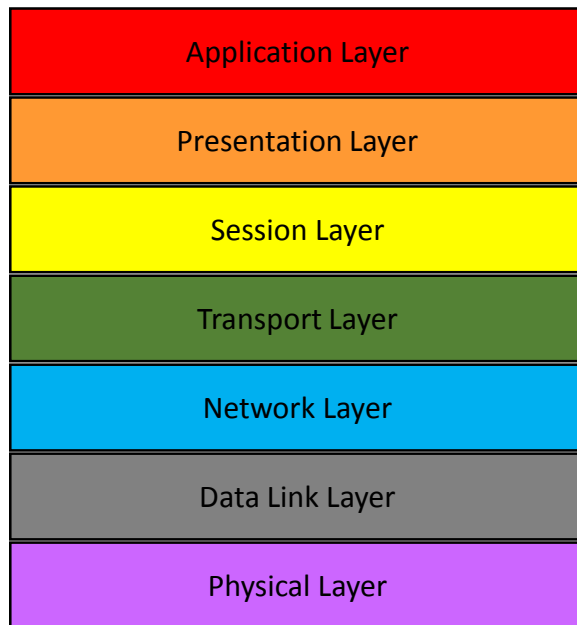
- The TCP/IP Model merges the top three layers of the OSI Seven Layer Model into the “Application Layer” in TCP/IP Model.
- The TCP/IP Model also merges the bottom two layers of the OSI Seven Layer Model into the “Network Access Layer” in TCP/IP Model.



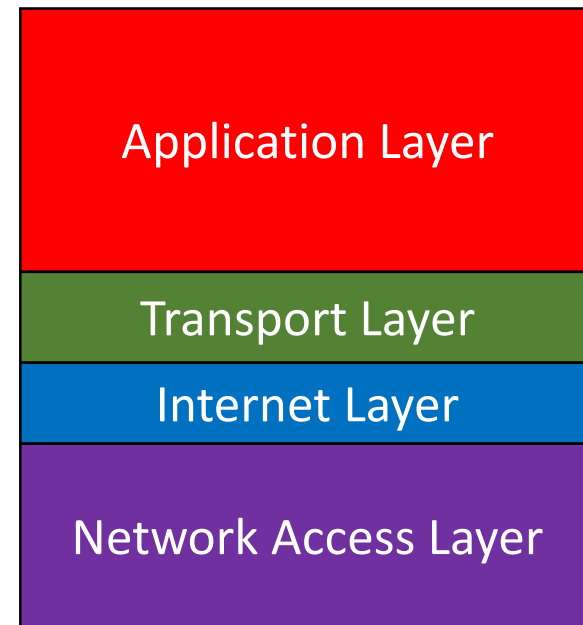
# Which Model to Use?

- Network engineers need to know both models.
- In this module, we will use both models.

**OSI Seven Layer Model**



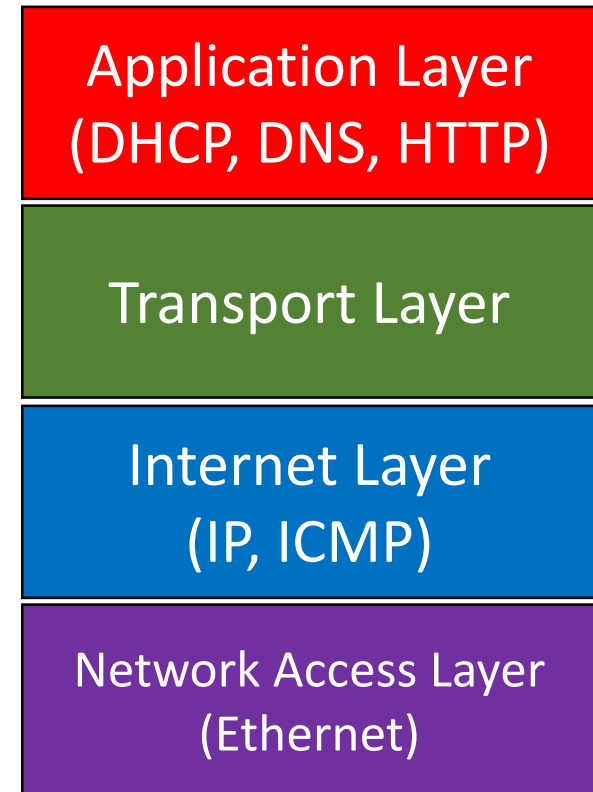
**TCP/IP Model**





# TCP/IP Protocols

- You have already come across some of the TCP/IP protocols.
- Example:
  - DHCP (Application Layer)
  - DNS (Application Layer)
  - HTTP for web service (Application Layer)
  - IP (Internet Layer)
  - ICMP for pinging (Internet Layer)
  - Ethernet (Network Access Layer)
- In this chapter, you will learn the Transport Layer.



# Transport Layer

- The Transport Layer establishes a **data channel** for an application to achieve **end-to-end data exchange**.
- Besides providing end-to-end data exchange, the Transport Layer is also responsible for:
  - Application Addressing
  - Segmentation of Data
  - Error Control
  - Flow Control
  - Congestion Control
- In this module, we will only look at “Application Addressing”. The rest will be covered in year-2 module.

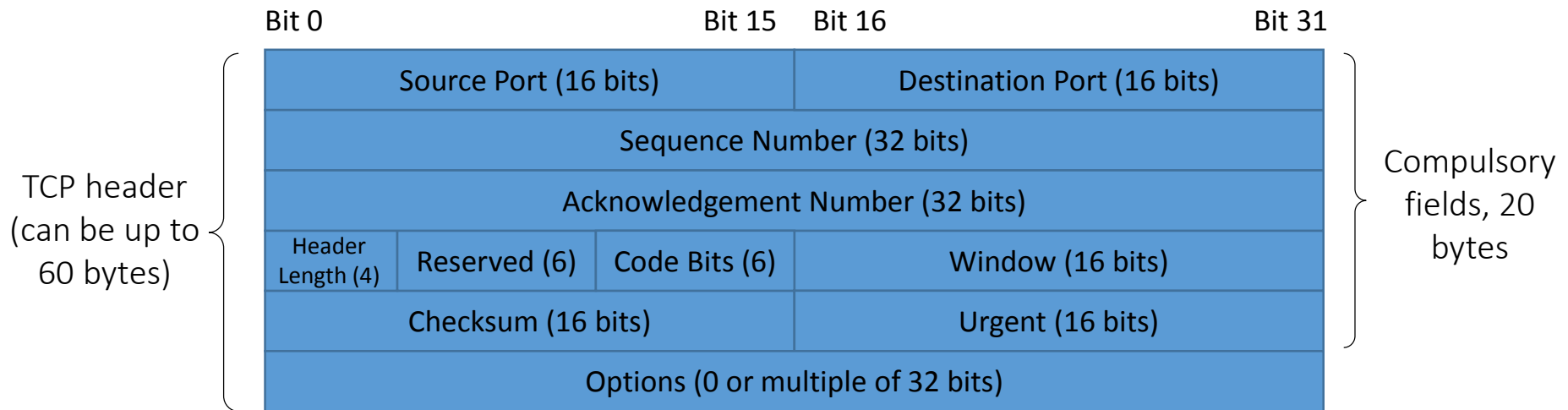
# Transport Layer Protocols

- Two main protocols in TCP/IP's Transport Layer are:
  - **Transmission Control Protocol (TCP)**
    - Connection-oriented
  - **User Datagram Protocol (UDP)**
    - Connectionless

# Transmission Control Protocol (TCP)

- TCP is for connection-oriented communications.
- Connection establishment
  - TCP establishes connection between the hosts before communication can take place.
- Flow-control
  - TCP provides flow-control to allow the hosts to adjust the speed of communication between the hosts.
- Reliable transmission
  - TCP provides a mechanism to keep track of the data transmission between the hosts.
  - Receiving host **sends acknowledgement** to sending host upon receiving incoming data successfully.
  - Unsuccessful transmission can be detected, and **re-transmission** will be carried out to resend the data.

# TCP Header – For Information



- *Minimum 20 bytes long.*
  - *“Options” can be as long as 40 bytes (i.e. TCP header can be up to 60 bytes).*
- *There are ten compulsory fields in TCP header.*
  - *Source Port, Destination Port, Sequence Number, Acknowledgement Number, Header Length, Reserved, Window, Checksum and Urgent.*

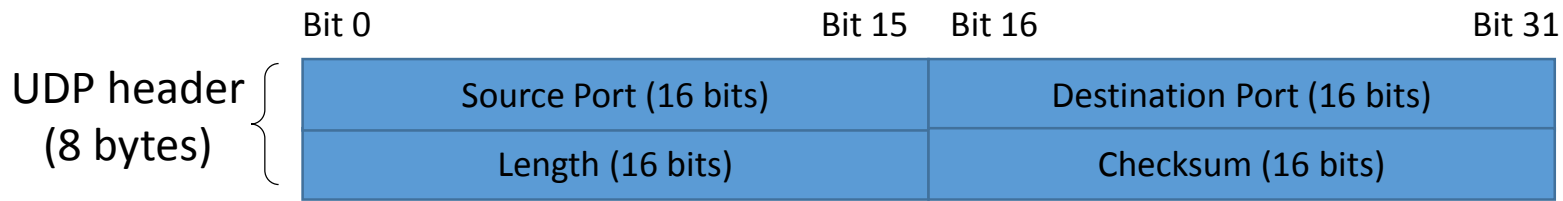
## *TCP Header – For Information*

- *Feature of TCP, and header field(s) involved:*
  - *Application Addressing (“channel for end-to-end data exchange”)*
    - *Source Port*
    - *Destination Port*
  - *Reliable transmission*
    - *Sequence Number*
    - *Acknowledgement Number*
  - *Flow-control*
    - *Window*

# User Datagram Protocol (UDP)

- UDP is for connectionless communications.
- No connection establishment
  - The sending host sends out data without establishing the connection first.
  - Assume that there is a path to reach the receiving host.
- No flow-control.
- Unreliable transmission
  - The sending host does not expect (and does not wait) for acknowledgement from the receiving host.
  - No detection of failed transmission, and hence no re-transmission is possible.

## *UDP Header – For Information*



- *8 bytes long (much shorter than TCP).*
- *UDP header consists of 4 fields only.*
  - *Source Port, Destination Port, Length and Checksum.*
- *No “Sequence Number” and “Acknowledgement Number”, hence unreliable transmission.*
- *No “Window” field, hence no flow-control is possible.*



# Comparison between TCP and UDP

- TCP is **reliable but slow**.
  - The sending host needs to wait for acknowledgement from the receiving host before sending out more data.
  - Analogous to sending “registered posts”.
- UDP is **unreliable but fast**.
  - The sending host simply sends out data as it desires.
  - Analogous to sending normal posts.
- TCP is less efficient than UDP because TCP has a much longer header (variable, from 20 to 60 bytes) than UDP (only 8 bytes).

# Which to use? TCP or UDP?

- Whether to use TCP or UDP as the Transport Layer protocol depends on the application.
- If “reliability” is important, use TCP.
  - Examples: file transfer, web-browsing
- If “speed” is critical, use UDP.
  - Examples: voice/video streaming

# Internet Sockets (1)

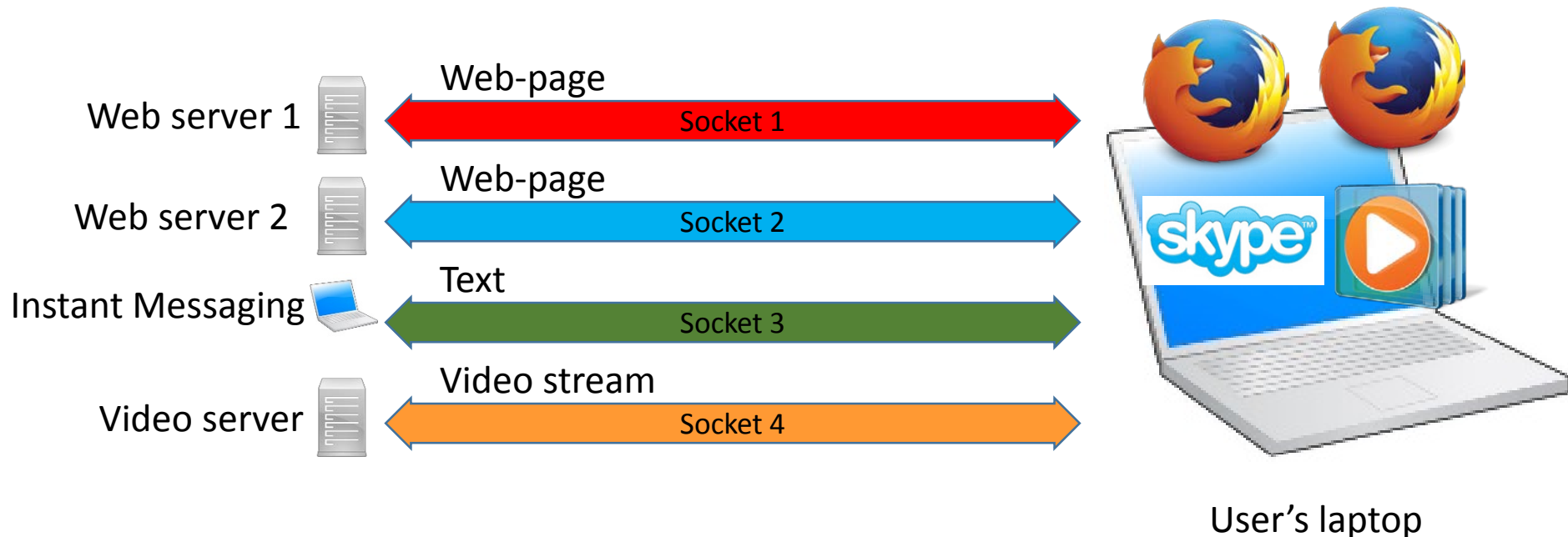
- To provide “Application Addressing”, the Transport Layer uses the concept of the “port”.
- Both the TCP and UDP headers come with two fields related to “port”:
  - Source Port
  - Destination Port
- The Transport Layer uses port numbers (16 bits, hence from 0 to 65535) to identify sending and receiving application end-points on a host.

# Internet Sockets (2)

- Each side of a Transport Layer connection has a port number reserved by the sending or receiving application.
  - Sending application uses “Source Port”.
  - Receiving application uses “Destination Port”.
- An internet socket is the combination of 4 elements:
  - source host address
  - source port
  - destination host address
  - destination port

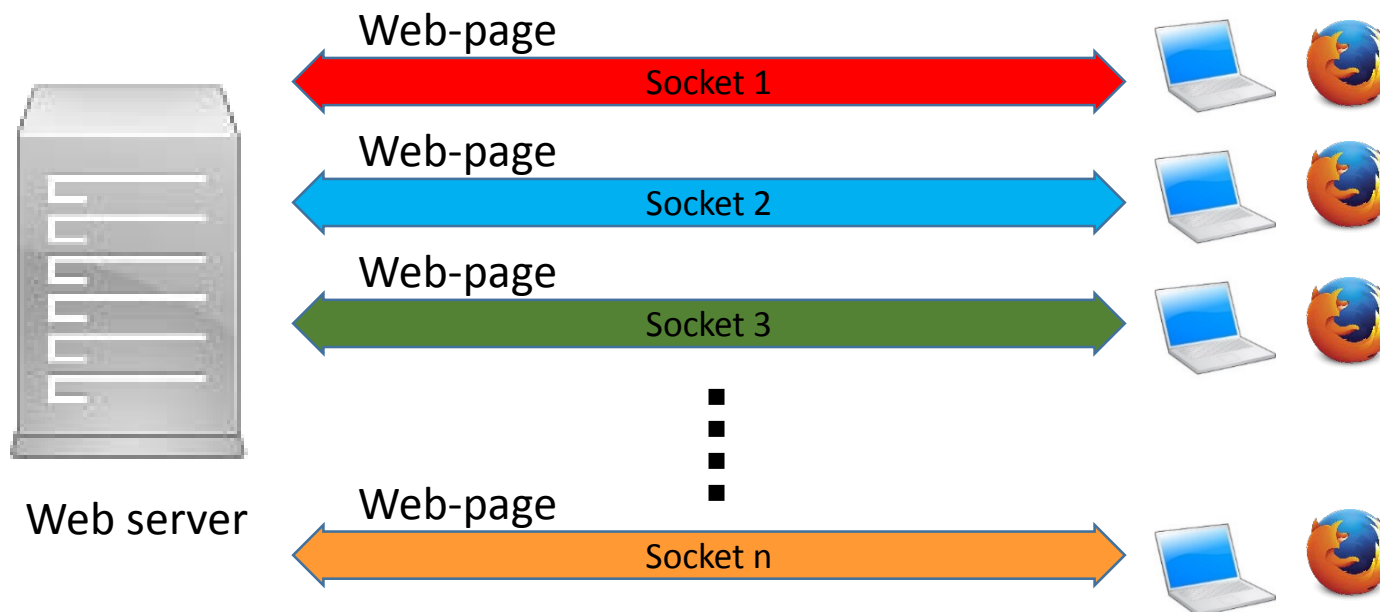
# Internet Sockets (3)

- Arriving data packets are identified as belonging to a specific Transport Layer connection by its internet socket.



# Internet Sockets (4)

- Not only that a client (laptop) may have many sockets, a server can have many sockets to provide several clients with several services simultaneously.
  - Clients initiate simultaneous internet sockets with the server.



# Port Number (1)

- The Transport Layer provides separate sockets using different **port numbers**.
- Port numbers are **16 bits** long.
- $2^{16} = 65536$  different port numbers (0 to 65535).
- Some port numbers are reserved for commonly used applications.
- Examples:
  - Web service : Port 80
  - File Transfer (FTP) : Ports 20 and 21
  - Telnet : Port 23
  - Email (SMTP): 25

# Port Number (2)

- Three basic categories of Port Numbers
  - Well-known Ports
  - Registered Ports
  - Dynamic, Private or Ephemeral Ports



Port Number

# Well-known Ports

- Range: 0 To 1023.
- Assigned by the IANA.
- Used by system processes that provide widely used network services.
- Run on servers and passively listen for connections.
- Examples

Application	Port(s)
HTTP	80
FTP	20 and 21
SMTP	25
TELNET	23
SSH	22
SSL	443

Port Number

# Registered Ports

- Range: 1024 to 49151.
- Assigned by IANA for specific service upon application by a requesting entity.
- Typically used by end user applications as ephemeral (short-lived, temporary) source ports when contacting servers.

Port Number

# Dynamic/Private/Ephemeral Ports

- Range: 49152 to 65535.
- Contains dynamic or private ports that cannot be registered with IANA.
- Used for private, or customised services or temporary purposes and for automatic allocation of ephemeral ports.

# Source & Destination Port Numbers (1)

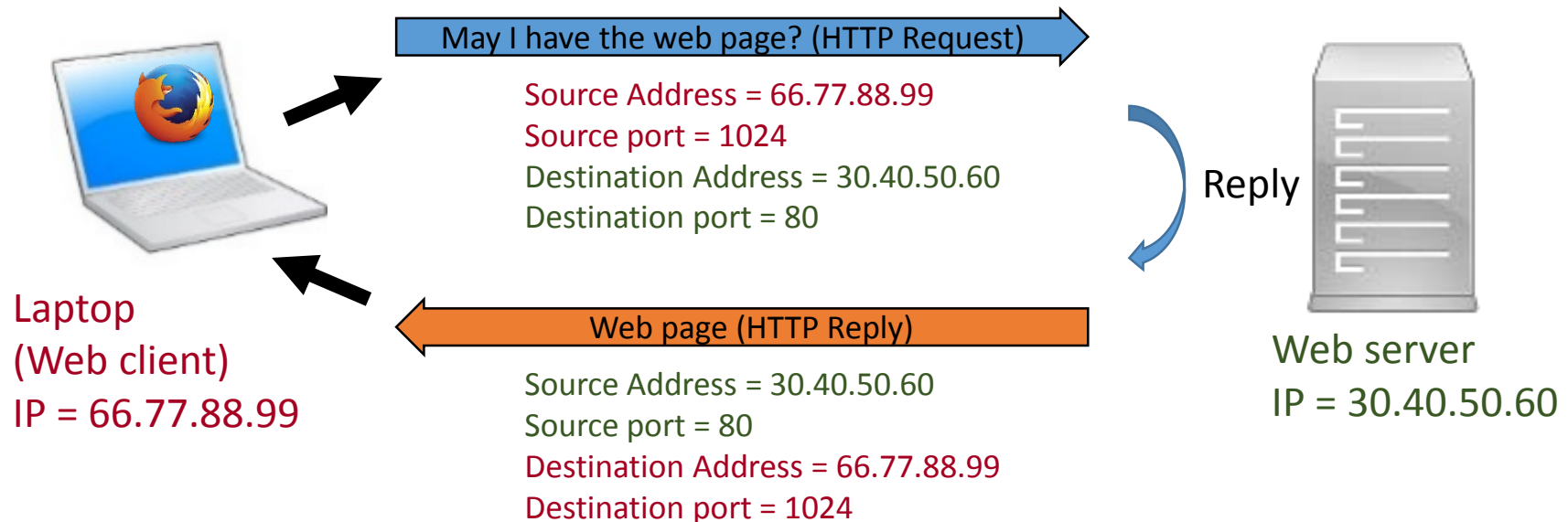
- Both Source Port and Destination Port fields are available in TCP and UDP headers.
- Source Port
  - The port number of the process that produces the data segment on the source host.
  - Normally this is an ephemeral (short-lived,  $\geq 1024$ ) port number for a request sent by a client to a server.
- Destination Port
  - The port number of the process that is the intended recipient of the data segment on the destination host.
  - Usually this is a “well-known” or “registered” port number for a server to “listen” to.

# Source & Destination Port Numbers (2)

- When a sending host sends a data segment to the receiving host,
  - the sending process's port number becomes the Source Port number in the Transport Layer header.
  - the sending host will use the Destination Port number in the Transport Layer header to indicate the recipient process at the receiving host.
- When the receiving host replies to the sending host, the Source port number and Destination Port number are swapped.

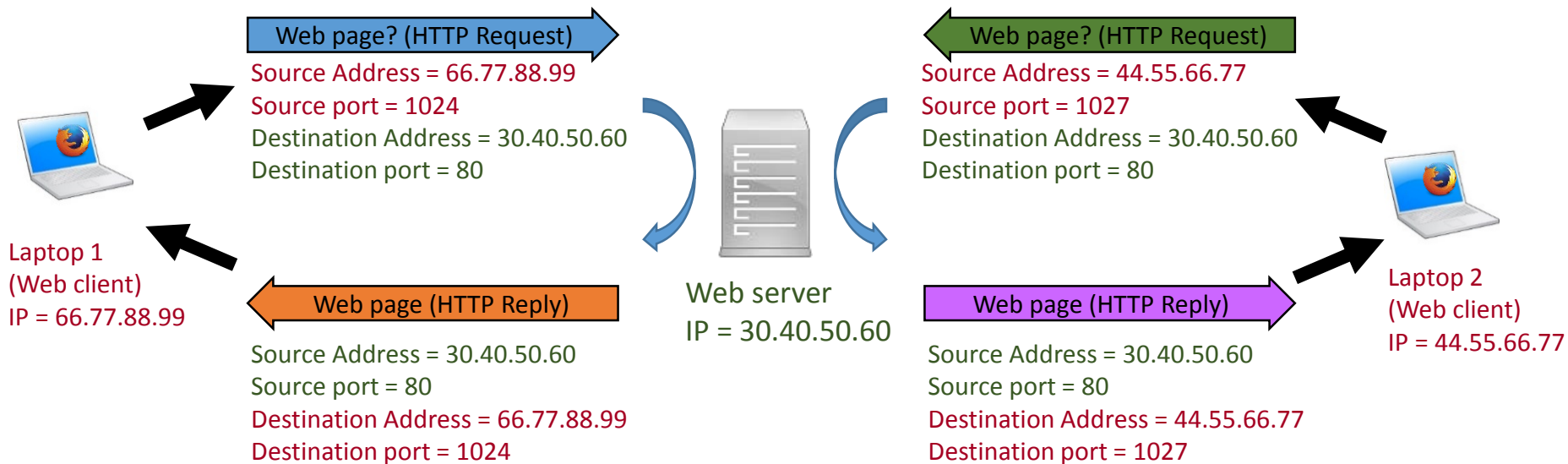
# Source & Destination Port Numbers (3)

- Example:
  - A laptop (sending host) sends a request to a web server (receiving host) for the web page.
    - Note: Well-known port for web service is Port 80.



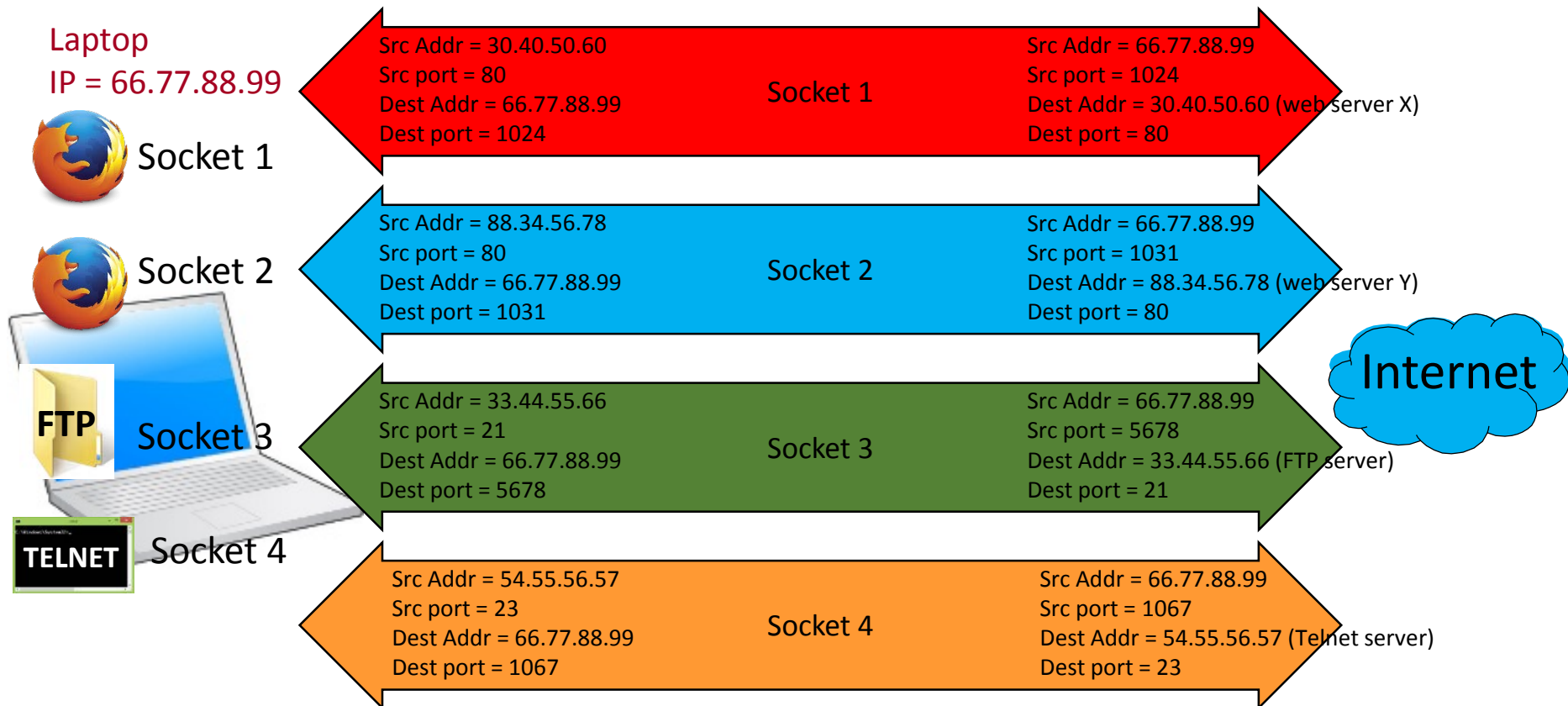
# Source & Destination Port Numbers (4)

- Example:
  - Two laptops (sending hosts) send request to a web server (receiving host) for the web page.



# Source & Destination Port Numbers (5)

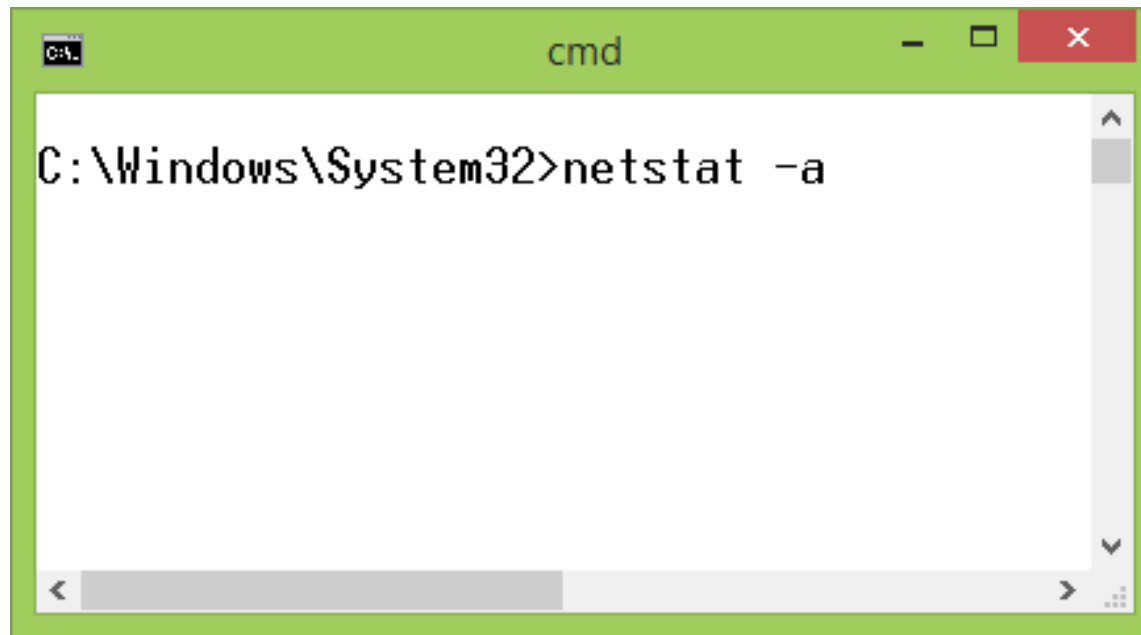
- Example:
  - A laptop (sending host) is used for surfing two web pages, connected to a FTP server, and a TELNET session to a remote server.





# What TCP & UDP Ports are Opened on Your Computer?

- You can use the command `netstat -a` on Windows computers to check what TCP and UDP ports are opened on the computers.



A screenshot of a Windows Command Prompt window. The window has a green title bar with the text "cmd" and standard window controls (minimize, maximize, close). The command prompt shows the directory `C:\Windows\System32` and the command `netstat -a` has been entered. The command prompt has a scroll bar on the right and a horizontal scroll bar at the bottom.

```
C:\Windows\System32>netstat -a
```

# Firewalls & Port Numbers

- It is important to know what port numbers are used by an application.
- If the firewall blocks the port number needed by an application, the IP packet for that application cannot get through the firewall.
  - Result: Application will not work.
- You also want to block unused TCP and UDP port numbers in your firewall to better protect your computer.

# Questions & Answers

