HO CHI MINH CITY UNIVERSITY OF SCIENCE
FACULTY OF ELECTRONICS AND TELECOMMUNICATIONS
DEPARTMENT OF TELECOMMUNICATIONS AND NETWORKS

COURSE

COMPUTER NETWORKS

Chapter 01

OVERVIEW OF NETWORK

Reference: Peter L Dordal, "An Introduction to Computer Networks," Jul 26, 2019

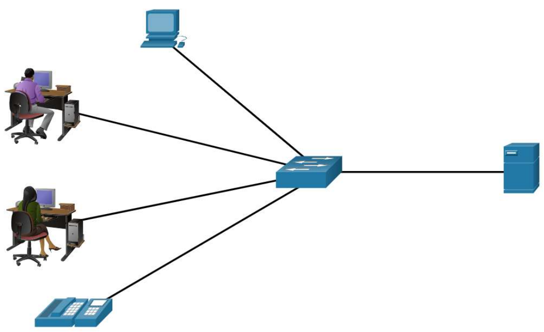
Lecturer: Nguyen Viet Ha, Ph.D. Email: nvha@fetel.hcmus.edu.vn

1 Layers

1. Layers

❖**Application:** software you use.

❖**LANs (Local Area Networks):** "physical" networks that provide the connection between machines within a home, school or corporation.



❖**IP (Internet Protocol):** provides an abstraction for connecting multiple LANs (e.g., into the Internet).

❖**TCP (Transport Control Protocol):** deals with transport and connections and actually sending user data.

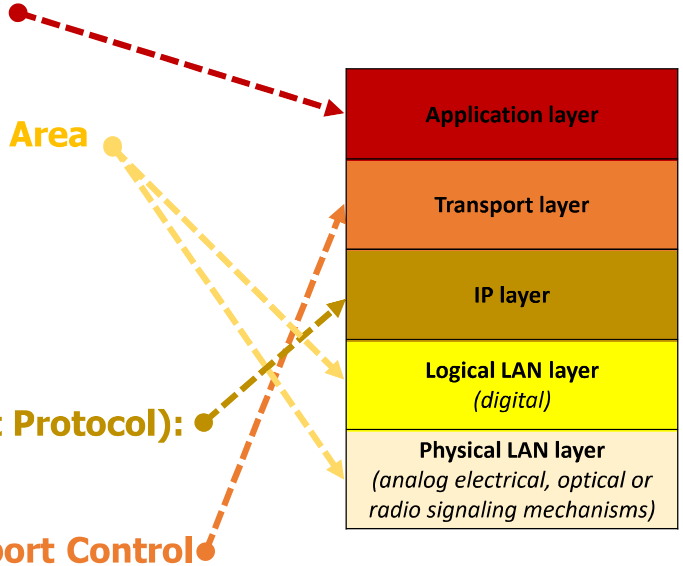
1. Layers

❖**Application:**

❖**LANs (Local Area Networks):**

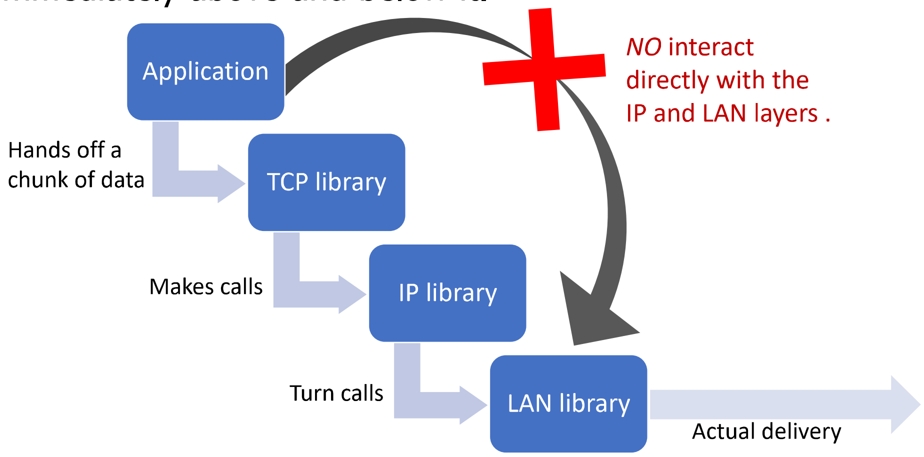
❖**IP (Internet Protocol):**

❖**TCP (Transport Control Protocol):**



1. Layers

- **A layer:** corresponds to the idea of a programming interface or library, with the understanding that a given layer communicates directly only with the two layers immediately above and below it.



1. Layers

These are the benefits of using a layered model:

- Assist in protocol design because protocols that operate at a specific layer have defined information that they act upon and a defined interface to the layers above and below
- Foster competition because products from different vendors can work together
- Prevent technology or capability changes in one layer from affecting other layers above and below
- Provide a common language to describe networking functions and capabilities

2 Packet

2. Packet

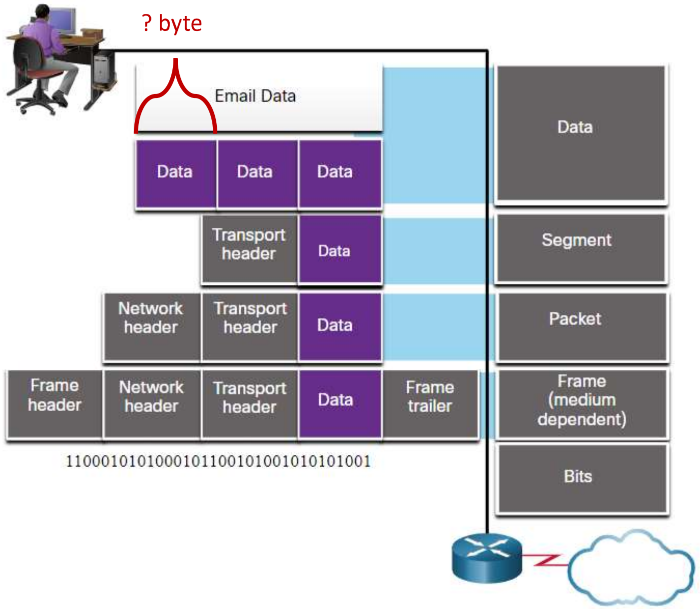
- **Packets:** modest-sized buffers of data, transmitted as a unit through some shared set of links.
- **Header:** containing delivery information, e.g., destination address.

header	data

header1	header2	data

- Almost all networking today is packet-based (vs. circuit-switched).

2. Packet



2. Packet

100 Kg

500 Kg

1 Ton

4 Tons

[2] Google image search. [Online]

2. Packet

❖**The maximum packet size.**
- ➢Ethernet packets: 1500 bytes of data.
- ➢TCP/IP packets: (originally) 512 bytes of data,
- ➢Token Ring packets: up to 4 kB of data.
- ➢ATM (Asynchronous Transfer Mode) packets 48 bytes of data.

❖**Header size (original):**
- ➢Ethernet headers: 14 bytes.
- ➢IP headers: 20 bytes.
- ➢TCP headers: 20 bytes.

3 Data Rate, Throughput, Bandwidth, and Delay

3. Data Rate, Throughput, Bandwidth, and Delay

❖**Data rate** (or **bandwidth**): the rate at which bits are transmitted.

❖**Throughput:** overall effective transmission rate, taking into account things like transmission overhead, protocol inefficiencies and perhaps even competing traffic.

❖**Goodput:** "application-layer throughput" - the amount of usable data delivered to the receiving application.

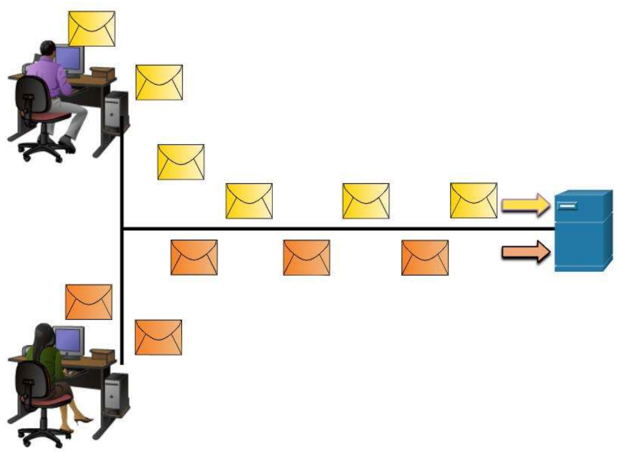
❖kilobits per second (kbps), megabits per second (Mbps)

3. Data Rate, Throughput, Bandwidth, and Delay

Bandwidth	>	**Throughput**	>	**Goodput**
Theoretical		Transmission overhead e.g., header		Application data only
Bit by bit without delay		Protocol inefficiencies e.g., synchronous, retransmission		
		Competing traffic e.g., network collision		

3. Data Rate, Throughput, Bandwidth, and Delay

❖**Packets** are the key to supporting **shared transmission lines**; that is, they support the **multiplexing** of multiple communications channels over a single cable.



3. Data Rate, Throughput, Bandwidth, and Delay

❖When a router or switch receives a packet, it decides to what next node to forward it.

➢Need reads in the packet:

- ○ **Store-and-forward**: reads in the entire packet before forward a packet.
- ○ **Cut-through**: read header and forward a packet before it has fully arrived.
- ○ Need queue (buffer).

❖**forwarding delay** equal to the time needed to read the entire/a part of packet.

3. Data Rate, Throughput, Bandwidth, and Delay

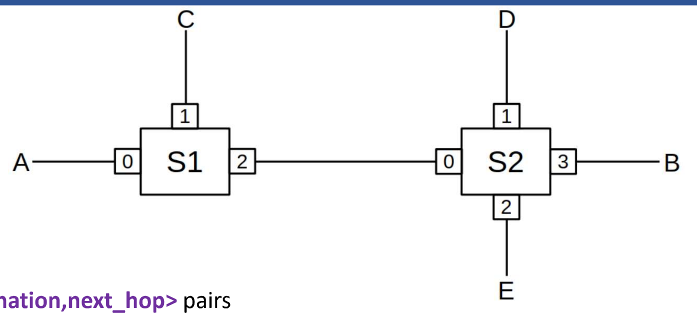
- **Bandwidth delay:** a per-link delay.
  - E.g., sending 1000 *Bytes* at 20 *Bytes/millisecond* will take 50 *ms*.
- **Propagation delay:** on-link-transmission delay.
  - E.g., sending a packet on a 5000 *km* cable with a propagation speed of 200 *m/µsec* (= 200 *km/ms*, about **2/3 the speed of light**), the first bit will not arrive at the destination until 25 *ms* later.
- **Store-and-forward delay**: equal to the sum of the bandwidth delays out of each router along the path.
- **Queuing delay**: waiting time in line at busy routers.
  - At bad moments this can exceed 1 *sec*, though that is rare.
  - Generally it is less than 10 *ms* and often is less than 1 *ms*.
  - Queuing delay is the only delay component amenable to reduction through careful engineering.

4 Datagram Forwarding

4. Datagram Forwarding

- Header will contain the address of the destination and perhaps other delivery information.
- Internal nodes of the network called **routers** or **switches** will forward the packet to the correct destination.
- **Forwarding table: <destination,next_hop>** pairs.
  - The "destination" entries do not have to correspond exactly with the packet destination addresses.
  - For IP routing, the table "destination" entries will correspond to **prefixes** of IP addresses.
    => Savings in space.
- *How the forwarding table is built -> Later (other chapters)*

4. Datagram Forwarding



<destination,next_hop> pairs

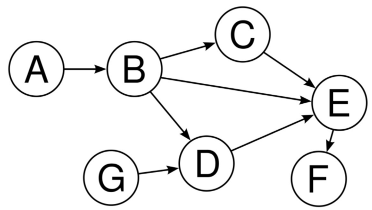
S1	
Destination	Next hop
A	
B	
C	
D	
E	

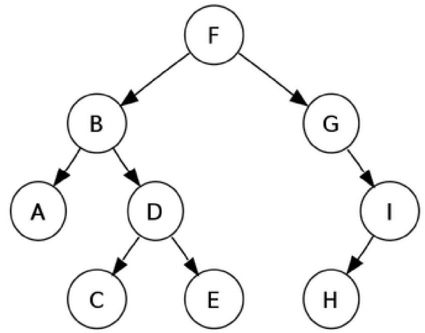
S1	
Destination	Next hop
A	
B	
C	
D	
E	

5 Topology

5. Topology (network analysis)

❖Network graph is **acyclic**, or is a **tree**

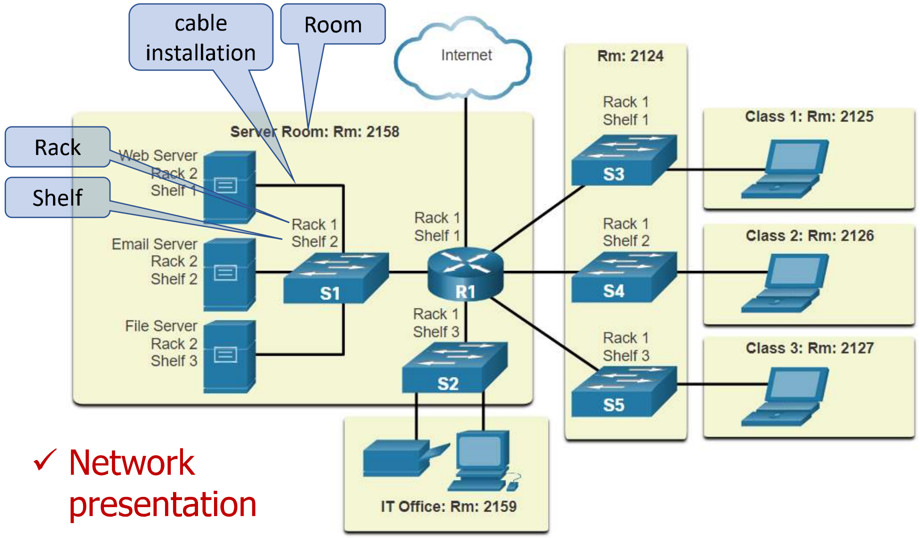




✓ How data transmit in the network

5. Topology (network design)

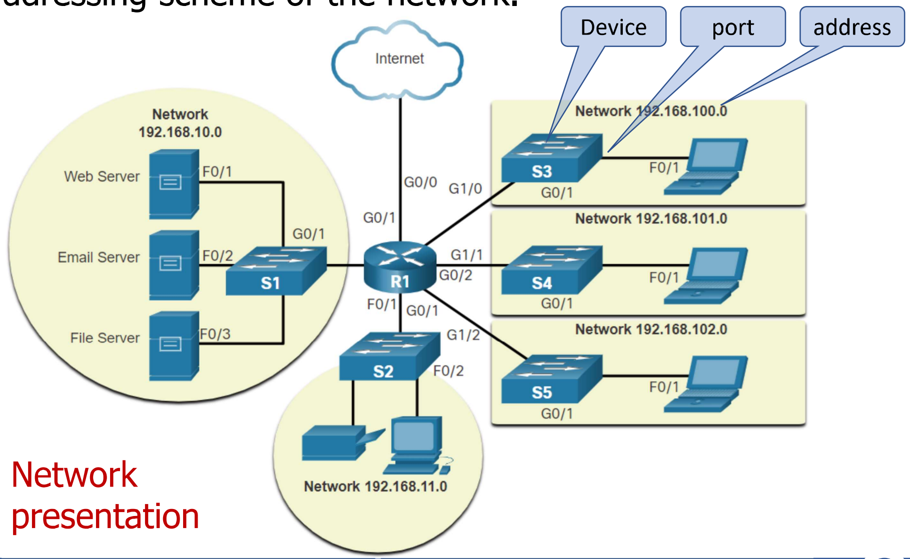
❖**Physical Topology Diagrams:** physical location of intermediary devices and cable installation.



✓ Network presentation

5. Topology (network design)

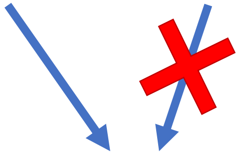
❖**Logical Topology Diagrams:** devices, ports, and the addressing scheme of the network.



✓ Network presentation

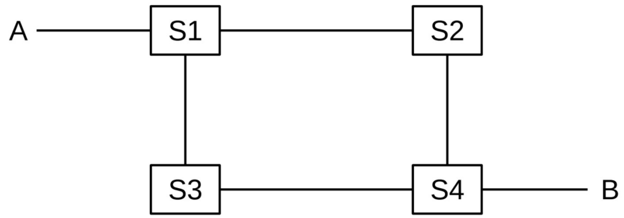
5. Topology

❖**Loop** vs. **Loop-free** network.



any broken link will result in partitioning the network into two pieces that cannot communicate.

REDUNDANCY



6 Routing Loops

6. Routing Loops

❖A potential drawback to **datagram forwarding** is the possibility of a **routing loop** (cause some packets to **circulate endlessly**).

- ➢ Consume a large majority of the bandwidth.

❖Routing loops can **also occur** in networks where the underlying **link topology is loop-free**.

- ➢E.g., misconfiguration.

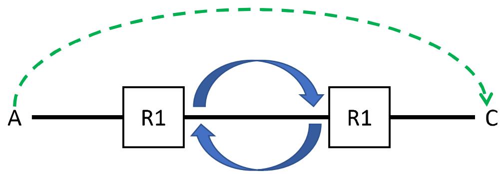
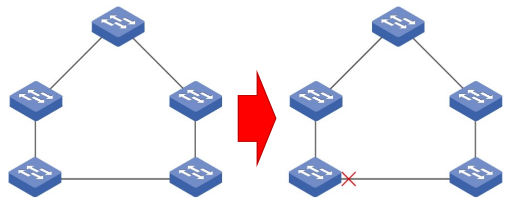


*Illustration of **LINEAR** routing loop.*

6. Routing Loops

❖Some solutions (mechanism or algorithm):

- ➢**Ethernet:**
  - ○ **Nonlinear routing loop:** disallowing loops in the underlying network topology. E.g., Spanning Tree algorithm (later).



  - ○ **Linear routing loop:** not having switches forward a packet back out the interface by which it arrived. (default)

6. Routing Loops

❖Some solutions (mechanism or algorithm):

➢**IP:**

- Using **"Time to Live" (TTL)** field in the IP header.
  - Set by the sender (e.g., 64 internal, 128 external).
  - Decremented by 1 at each router.
  - A packet is discarded if its TTL reaches 0.

Infinite (endlessly) loop → finite loop

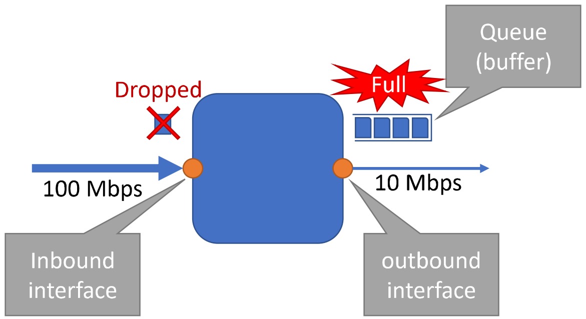
Consume a large majority of the bandwidth.

7 Congestion

7. Congestion

❖Packets arriving faster than they can be sent out.

➢Inbound interface has a higher bandwidth than the outbound interface.



7. Congestion

➢On the Internet, most packet losses are due to congestion.

- This is not because congestion is especially bad (though it can be, at times), but rather that other types of losses (e.g., due to packet corruption) are insignificant by comparison.

? On the Internet, collision loss is good or not

8

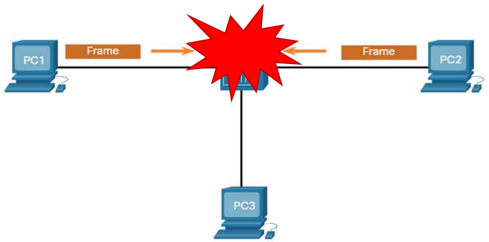
LANs and Ethernet

8. LANs and Ethernet

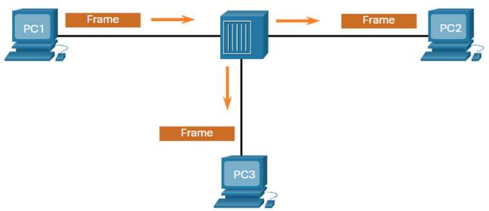
- LAN (**local-area network**): a system consisting of:
  - Physical links that are, ultimately, serial lines.
  - Common interfacing hardware connecting the hosts to the links.
  - Protocols to make everything work together.
- The most common type of (wired) LAN is Ethernet.
  - 10 *Mbps*, 100 *Mbps*, 1000 *Mbps* (1 *Gbps*).
- The most common type of (wireless) LAN is Wi-Fi.
  - 802.11a/b/g/ac/ax.

8. LANs and Ethernet

- Many early Ethernet installations were **unswitched**.
  - **Collision**: two stations could then transmit at the same time.
    - Reduce throughput



  - Every packet is received by every host:
    - Security threat, e.g., password sniffers.



8. LANs and Ethernet

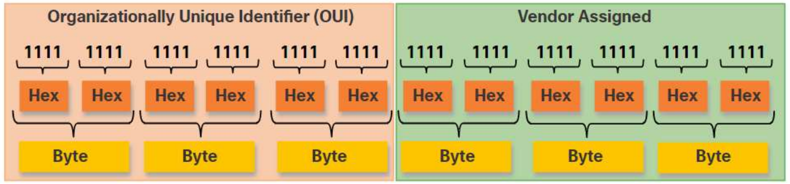
- Almost Ethernets today are **fully switched**.
  - No collision.
  - Each packet is delivered only to the host to which it is addressed.
  - Prevents host-based eavesdropping.
  - However: Queuing issue.
    - So seldom fill up that they are almost invisible.

8. LANs and Ethernet

❖**Ethernet address:**

➢Card's **physical** address or **hardware** address or **MAC** (Media Access Control) address.

- Burned into the card's ROM.
- Six bytes long
  - The **first three bytes** of the physical address have been assigned to the manufacturer.
  - The **subsequent three bytes** are a serial number assigned by that manufacturer.



8. LANs and Ethernet

❖**Ethernet address:**

➢Card's **physical** address or **hardware** address or **MAC** (Media Access Control) address.

- **Broadcast address. (FF-FF-FF-FF-FF-FF)**
  - If a switch receives a broadcast packet on one port, it forwards the packet out every other port.
    - i.e., allows host A to contact host B when A does not yet know B's physical address.
- **Unicast address.**
  - Traffic addressed to a particular host.

8. LANs and Ethernet

❖**Ethernet address:**

➢Switches must know where all active destination addresses in the LAN are located; traditional Ethernet switches do this by a passive **learning** algorithm.

MAC Address Table	
Port	**MAC Address**

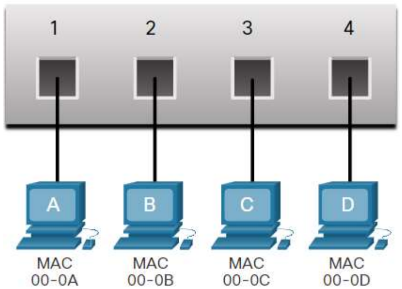


Fig. 16a. Switch Learning and Forwarding [1].

8. LANs and Ethernet

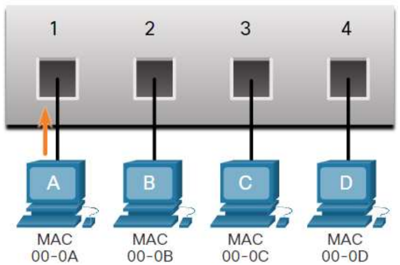
❖**Ethernet address:**

➢Switches must know where all active destination addresses in the LAN are located; traditional Ethernet switches do this by a passive **learning** algorithm.

MAC Address Table	
Port	**MAC Address**
1	00-0A

Switch Learning and Forwarding

1. LEARNING



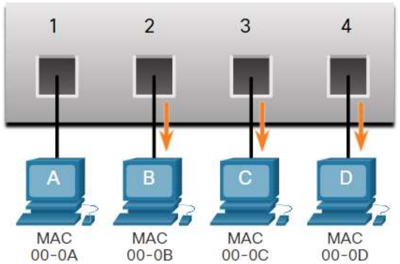
❖Ethernet address:

➢Switches must know where all active destination addresses in the LAN are located; traditional Ethernet switches do this by a passive **learning** algorithm.

MAC Address Table	
Port	**MAC Address**
1	00-0A

Switch Learning and Forwarding

2. FLOODING



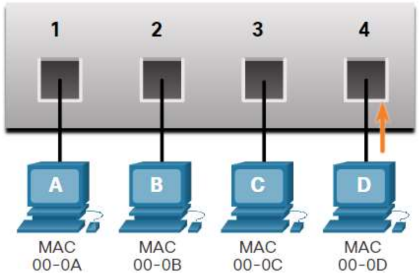
❖Ethernet address:

➢Switches must know where all active destination addresses in the LAN are located; traditional Ethernet switches do this by a passive **learning** algorithm.

MAC Address Table	
Port	**MAC Address**
1	00-0A
4	00-0D

Switch Learning and Forwarding

1. LEARNING



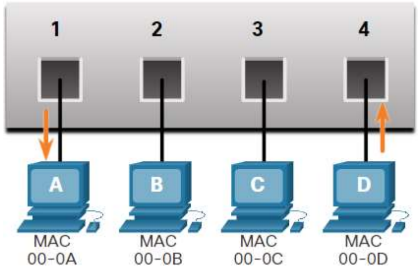
❖Ethernet address:

➢Switches must know where all active destination addresses in the LAN are located; traditional Ethernet switches do this by a passive **learning** algorithm.

MAC Address Table	
Port	**MAC Address**
1	00-0A
4	00-0D

Switch Learning and Forwarding

3. FORWARDING



9 IP - Internet Protocol

❖**Network address:**

➢**IP addresses** is the primary means of enabling devices to locate one another and establish end-to-end communication on the internet.

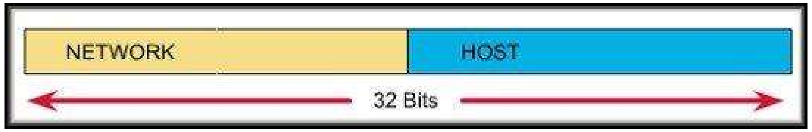
➢Unlike Ethernet addresses, are administratively assigned.

➢IP version 4 (IPv4), IP version 6 (IPv6).

❖**Network address:**

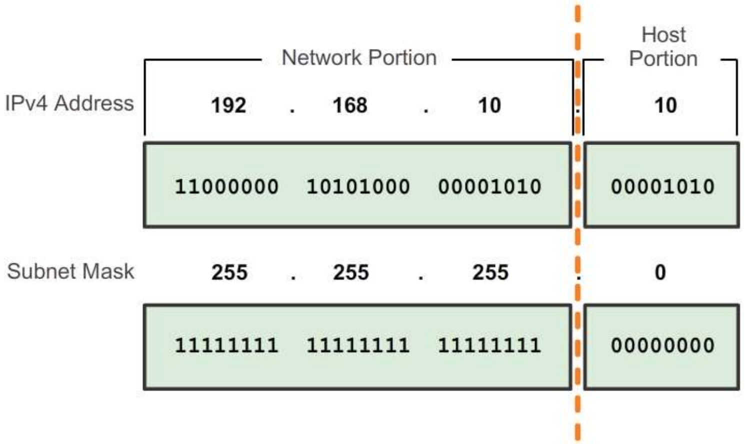
➢An IP Version 4 address has two parts:

- Network number
- Host number



- The **network portion** of the address **IS THE SAME** for all hosts on the network.
- Each device **IS IDENTIFIED** by a unique **host portion**.

❖To define the **network** and **host portions** of an address, a devices use a separate 32-bit pattern called a **subnet mask**



? Check your IP

❖The IP layer does not maintain information about endpoint-to-endpoint connections, and simply forwards packets.

➢**Called CONNECTIONLESS**

➢FYI: Connectionless vs. **CONNECTION-ORIENTED** (in TCP layer).

❖The most common form of IP packet loss is router queue overflows, representing **network congestion**.

9. IP - Internet Protocol

❖**IP Forwarding**

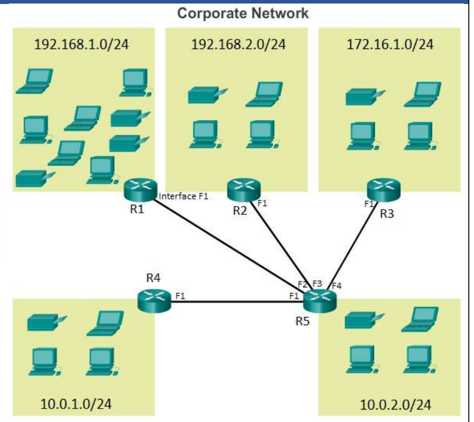
➢IP routers use datagram forwarding.

○ But the "destination" values listed in the forwarding tables are network prefixes (network address).

- Don't care Host number.
- E.g., 192.168.10.0/24

9. IP - Internet Protocol

❖**IP Forwarding**



R1	
Network address	Int

R2	
Network address	Int

R3	
Network address	Int

R4	
Network address	Int

R5	
Network address	Int

10 Transport

10. Transport

❖**IP issues**:

➢"Best-effort" (connectionless) mechanism, which means packets can and do get lost sometimes.

➢Data can arrive out of order.

➢IP only supports sending to a specific host

○ No method to access multiple applications in one host (e.g., Email and web).

10. Transport

❖Solution: **Transport layer**

➢E.g., Transmission Control Protocol (TCP)

- **reliability**: TCP numbers each packet and keeps track of which are lost and retransmits them after a timeout. It holds early-arriving out-of-order packets for delivery at the correct time. Every arriving data packet is acknowledged by the receiver; timeout and retransmission occurs when an acknowledgment packet isn't received by the sender within a given time.
- **connection-orientation**: Once a TCP connection is made, an application sends data simply by writing to that connection. No further application-level addressing is needed. TCP connections are managed by the operating-system kernel, not by the application.

10. Transport

❖Solution: **Transport layer**

➢E.g., Transmission Control Protocol (TCP)

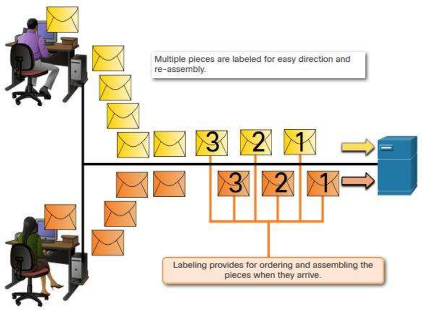
- **stream-orientation**: An application using TCP can write 1 byte at a time, or 100 kB at a time; TCP will buffer and/or divide up the data into appropriately sized packets. (Segmentation)
  - Segmenting messages has two primary benefits:
    - **Increases speed** - Large amounts of data can be sent over the network without tying up a communications link.
    - **Increases efficiency** - Only segments which fail to reach the destination need to be retransmitted, not the entire data stream.

10. Transport

❖Solution: **Transport layer**

➢E.g., Transmission Control Protocol (TCP)

- **stream-orientation**: An application using TCP can write 1 byte at a time, or 100 kB at a time; TCP will buffer and/or divide up the data into appropriately sized packets. (Segmentation)
  - Sequencing messages is the process of numbering the segments so that the message may be reassembled at the destination.



10. Transport

❖Solution: **Transport layer**

➢E.g., Transmission Control Protocol (TCP)

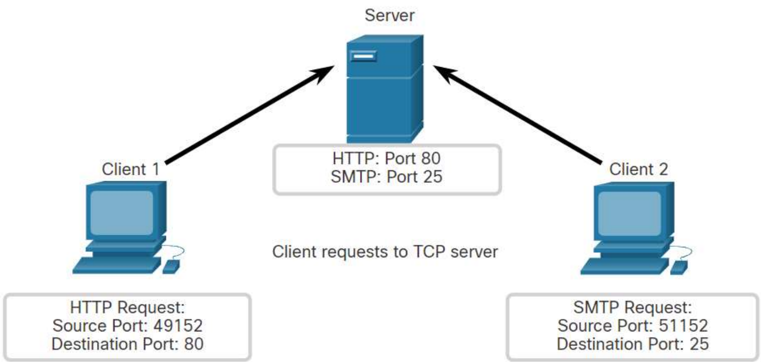
- **port numbers**: these provide a way to specify the receiving application for the data, and also to identify the sending application.
- **throughput management**: TCP attempts to maximize throughput, while at the same time not contributing unnecessarily to network **congestion.**

10. Transport

❖**socket addresses:**

➢TCP endpoints are of the form <host,port>

- **Servers** *listen* for connections to sockets they have opened.
- **Client** *initiates* a connection to a server.



10. Transport

❖**User Datagram Protocol (UDP):**

➢Like TCP, provides port numbers to support delivery to multiple endpoints within the receiving host, in effect to a specific process on the host.

➢Also use socket addresses.

➢**NO**:

- Connection setup,
- Lost-packet detection,
- Automatic timeout/retransmission
- Segmentation/Reassemble (application must manage its own packetization)
- Ordered Delivery.
- **BUT** data transmission can get started faster.

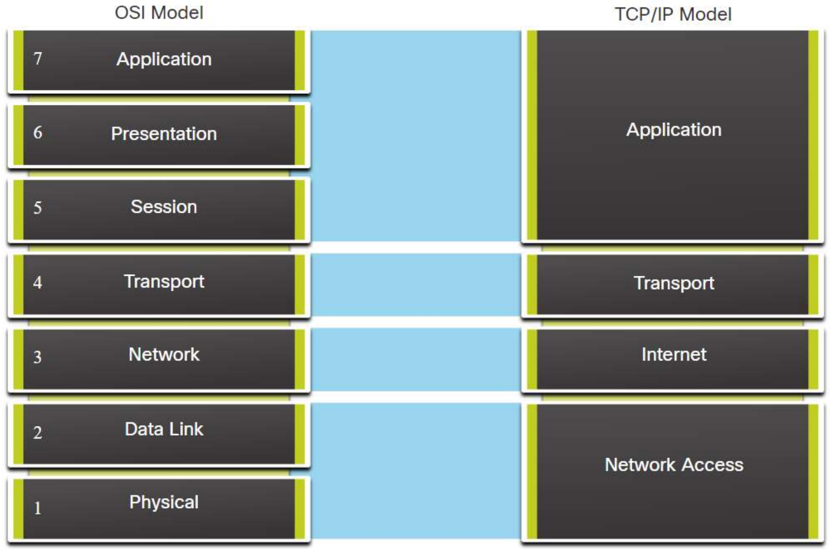
11 Network Protocol Suites

11. Network Protocol Suites

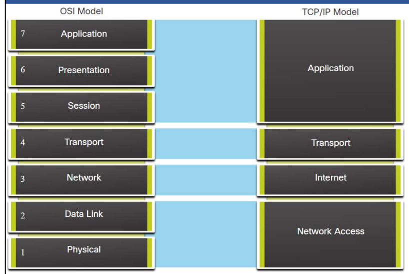
❖**Internet Protocol Suite or TCP/IP** - This is the most common and relevant protocol suite used today. The TCP/IP protocol suite is an open standard protocol suite maintained by the Internet Engineering Task Force (IETF).

❖**Open Systems Interconnection (OSI) protocols** - This is a family of protocols developed jointly in 1977 by the International Organization for Standardization (ISO) and the International Telecommunications Union (ITU). The OSI protocol also included a seven-layer model called the OSI reference model. The OSI reference model categorizes the functions of its protocols. Today OSI is mainly known for its layered model. The OSI protocols have largely been replaced by TCP/IP.

11. Network Protocol Suites

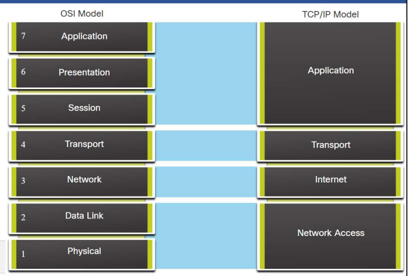


11. Network Protocol Suites



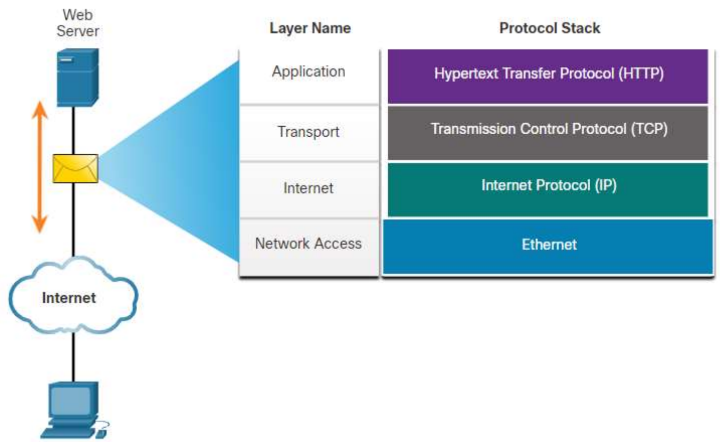
TCP/IP Model Layer	Description
4 – Application	Represents data to the user, plus encoding and dialog control.
3 – Transport	Supports communication between various devices across diverse networks.
2 – Internet	Determines the best path through the network.
1 – Network Access	Controls the hardware devices and media that make up the network.

11. Network Protocol Suites

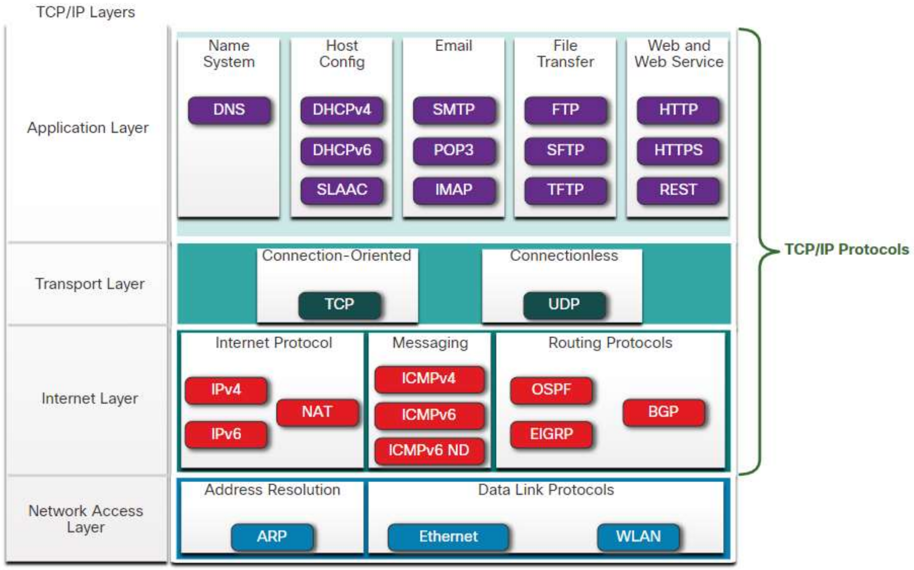


OSI Model Layer	Description
7 - Application	The application layer contains protocols used for process-to-process communications.
6 - Presentation	The presentation layer provides for common representation of the data transferred between application layer services.
5 - Session	The session layer provides services to the presentation layer to organize its dialogue and to manage data exchange.
4 - Transport	The transport layer defines services to segment, transfer, and reassemble the data for individual communications between the end devices.
3 - Network	The network layer provides services to exchange the individual pieces of data over the network between identified end devices.
2 - Data Link	The data link layer protocols describe methods for exchanging data frames between devices over a common media
1 - Physical	The physical layer protocols describe the mechanical, electrical, functional, and procedural means to activate, maintain, and de-activate physical connections for a bit transmission to and from a network device.

11. Network Protocol Suites

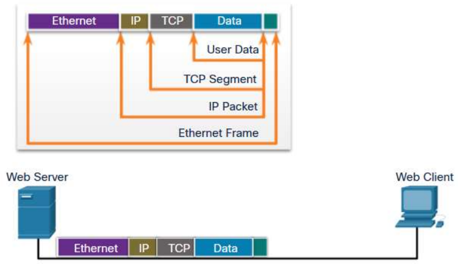


11. Network Protocol Suites

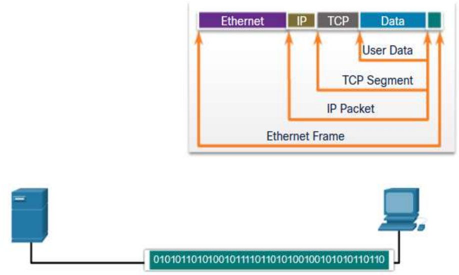


11. Network Protocol Suites

- A web server encapsulating and sending a web page to a client.



- A client de-encapsulating the web page for the web browser



QA

Lecturer: Nguyen Viet Ha, Ph.D.
Ho Chi Minh City University of Science
Faculty of Electronics and Communications
Department of Telecommunication and Networks
Email: nvha@fetel.hcmus.edu.vn

❖ Explain the "stateless forwarding." What is the opposite of "stateless forwarding"? Give some examples.