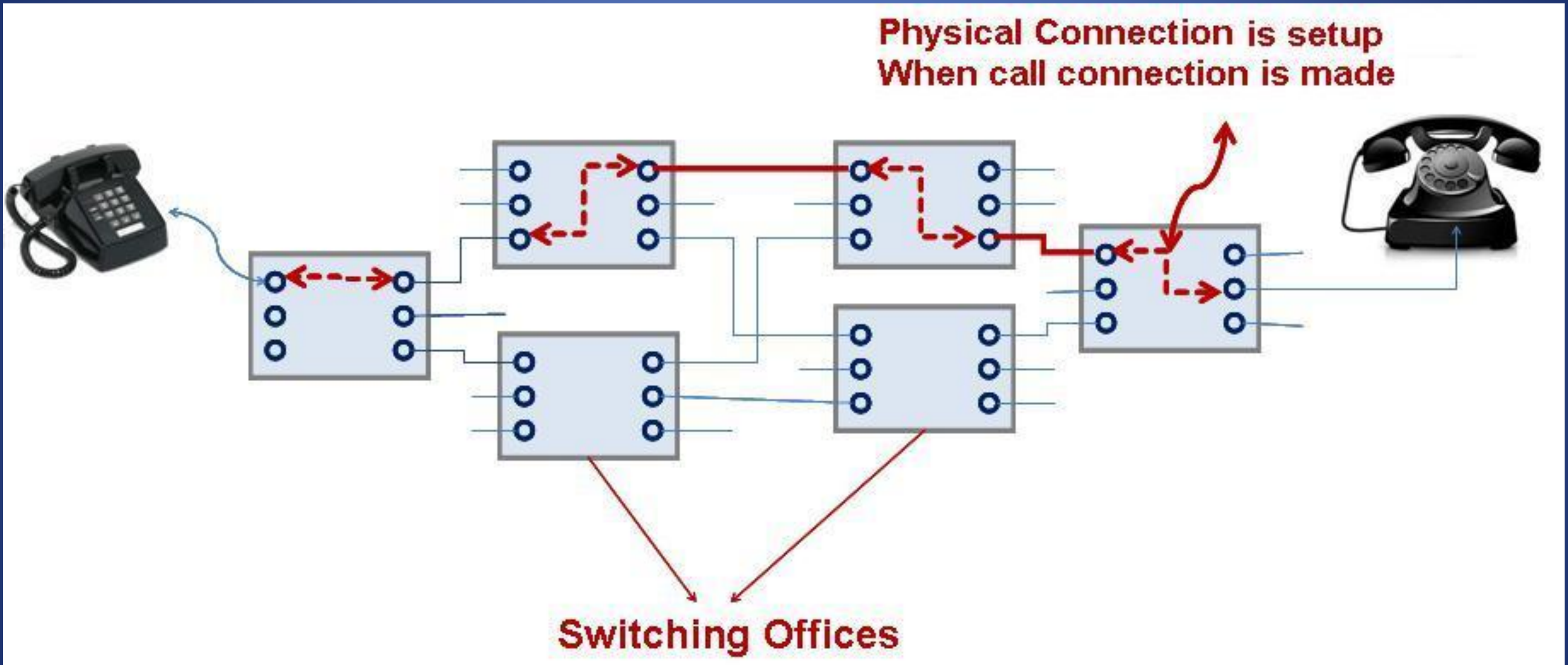
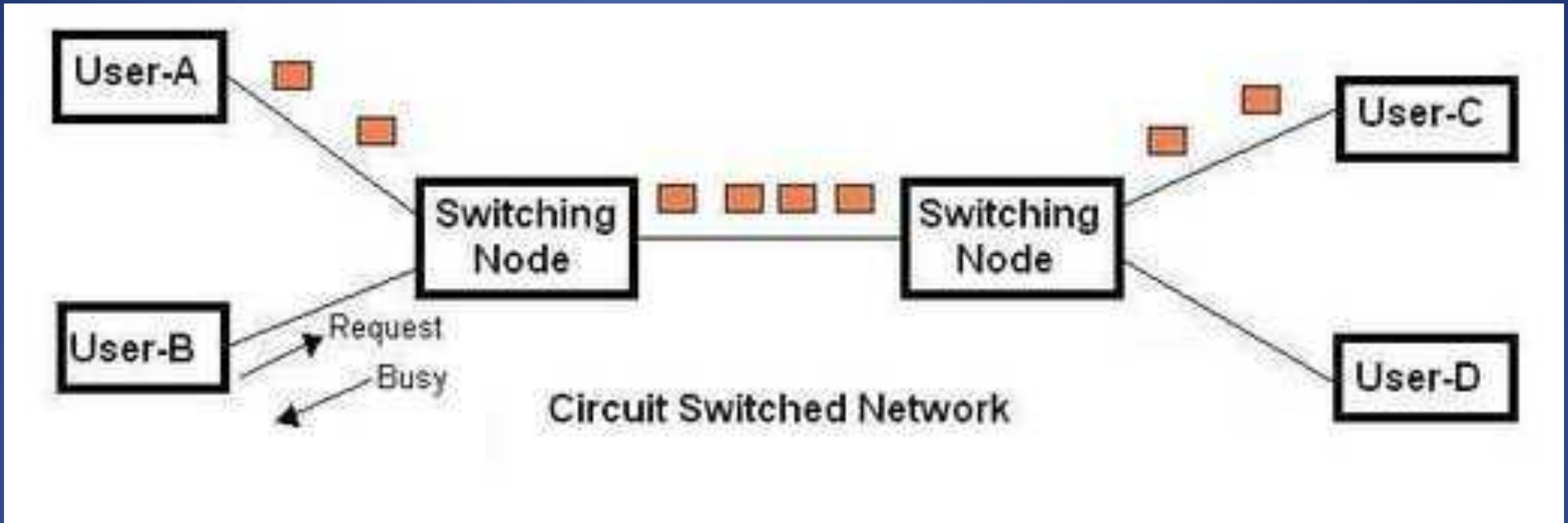
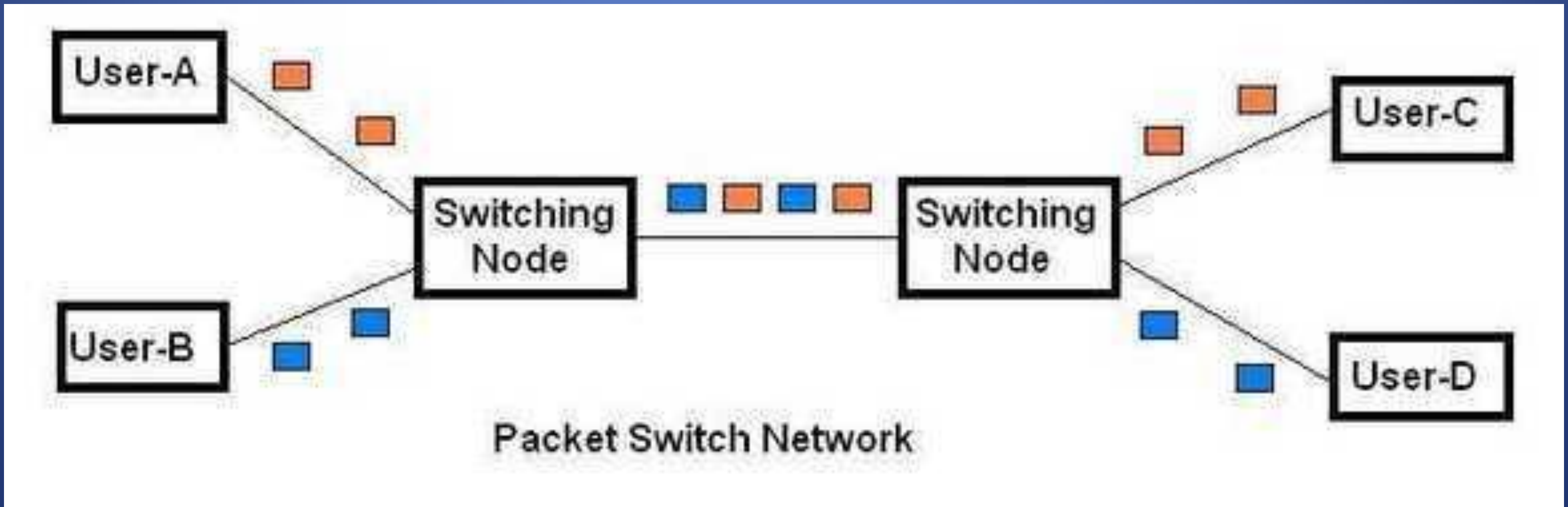
Unit 1 Introduction

1.1 Network Models

Circuit and Packet Switching



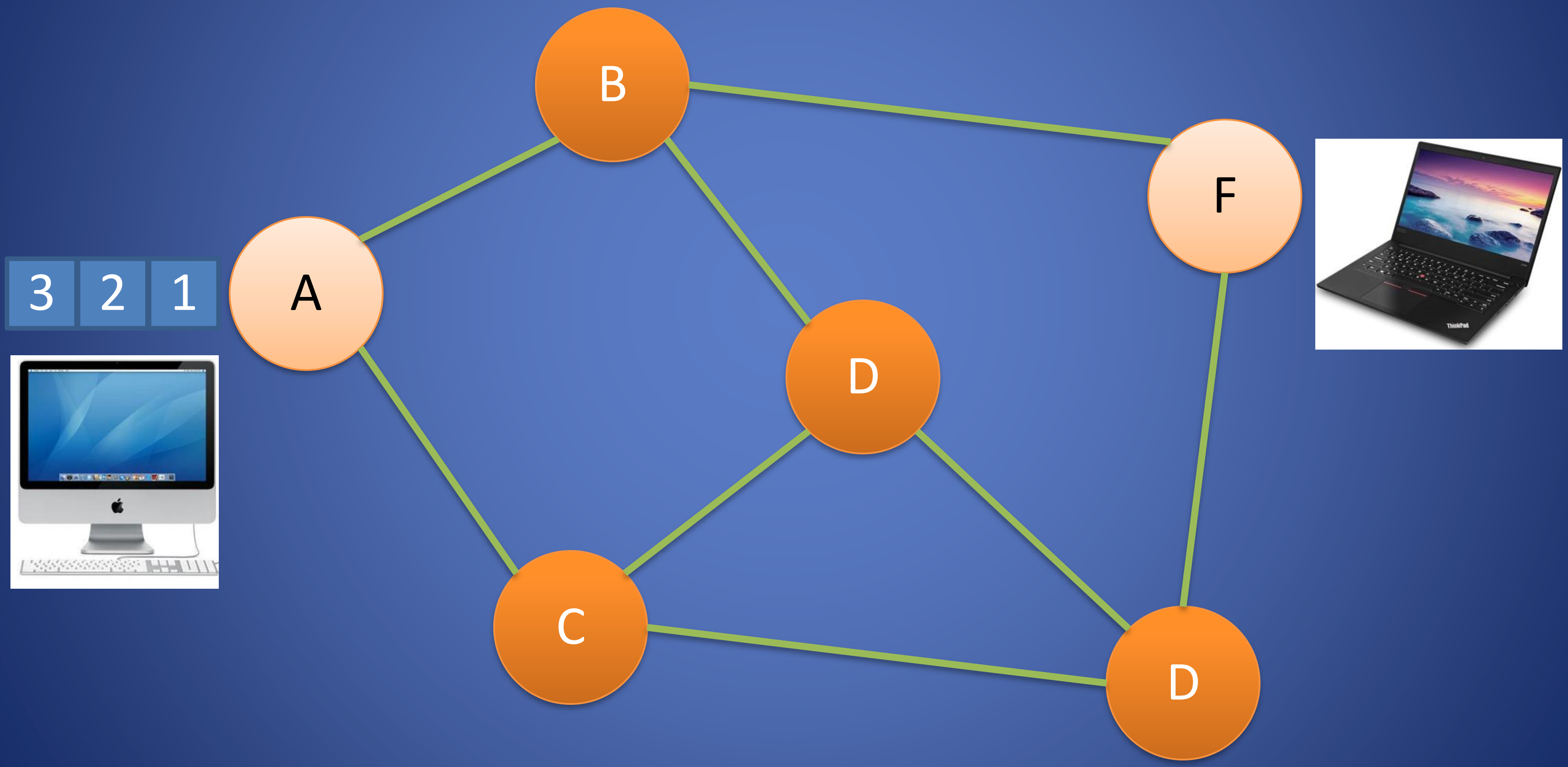

User-A
Switching Node
Switching Node
User-C
User-B
Request
Busy
Circuit Switched Network
User-D


User-A
Switching Node
Switching Node
User-C
User-B
User-D
Packet Switch Network

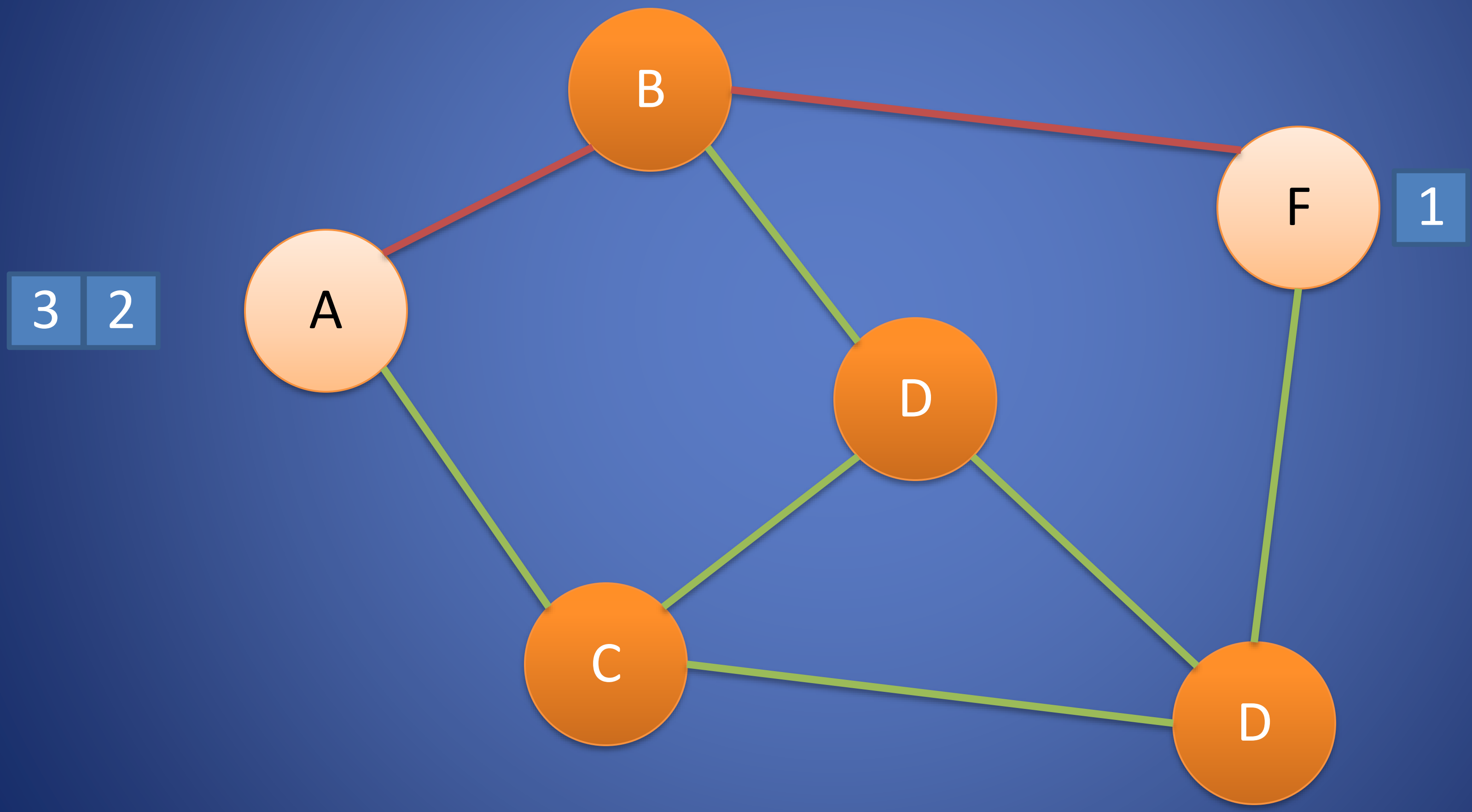
Circuit and Packet Switching

- Circuit switching
  - Legacy phone network
  - Single route through sequence of hardware devices established when two nodes start communication
  - Data sent along route
  - Route maintained until communication ends
- Packet switching
  - Internet
  - Data split into packets
  - Packets transported independently through network
  - Each packet handled on a best efforts basis
  - Packets may follow different routes

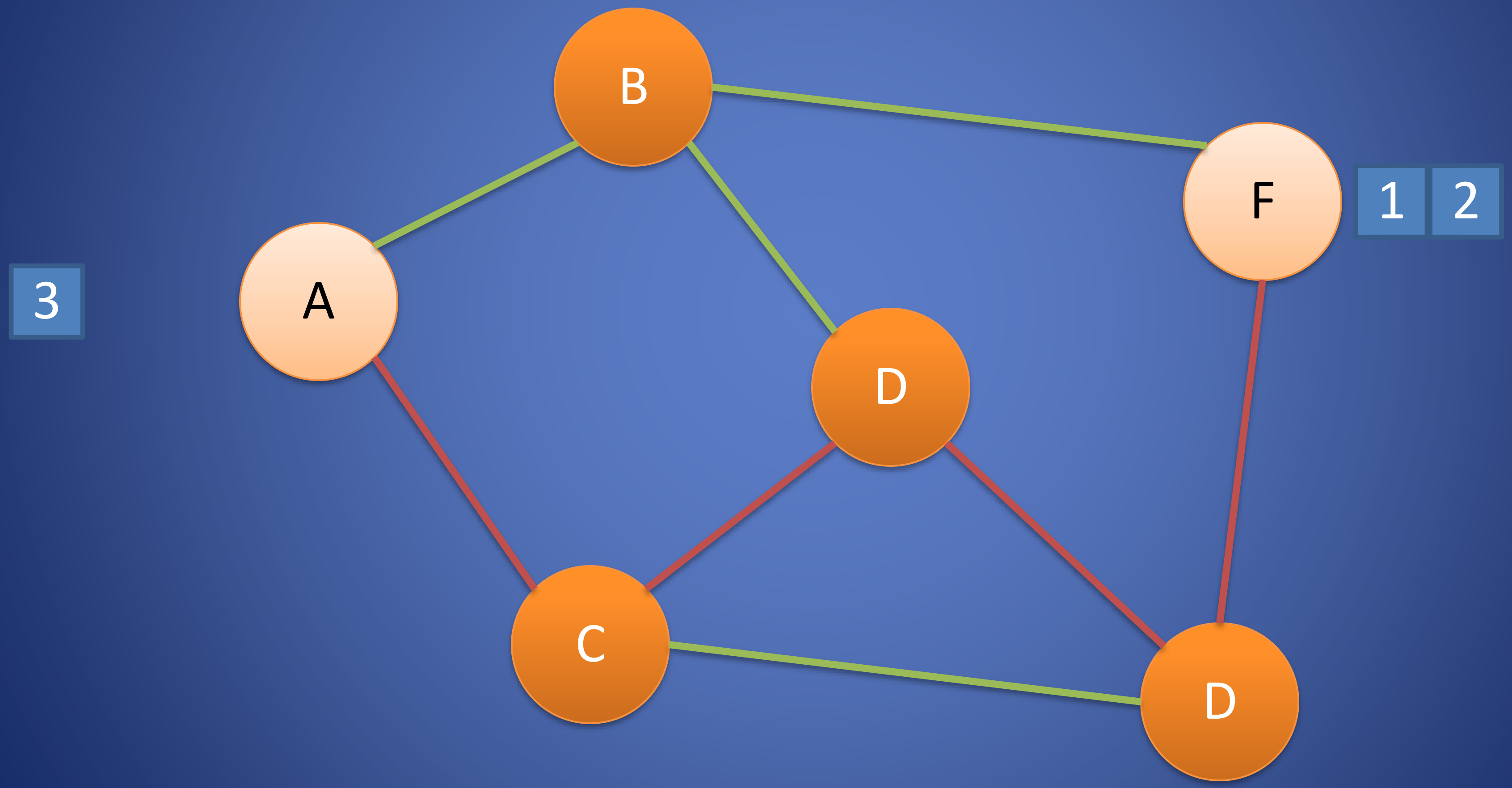
Packet Switching



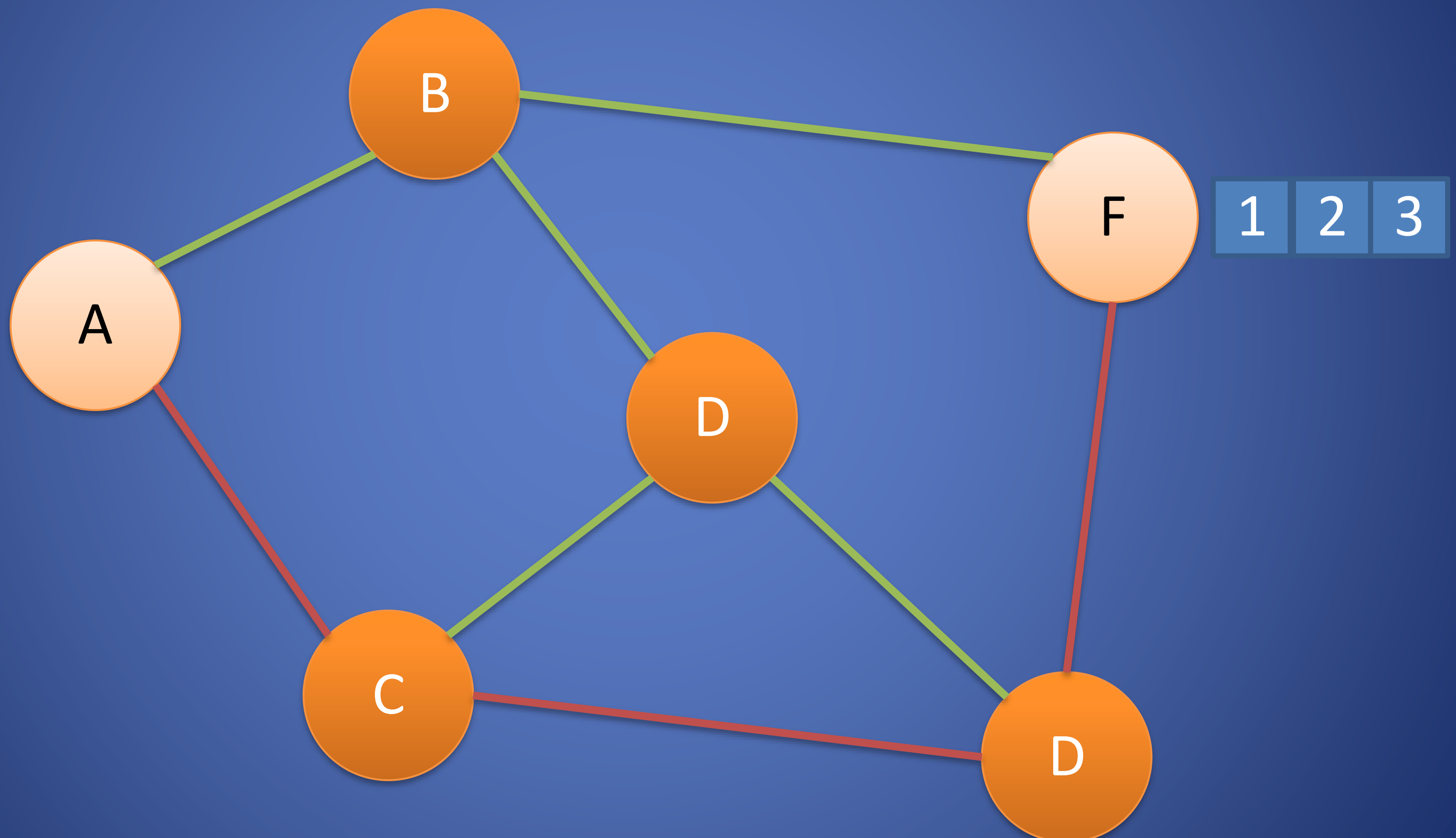
Packet Switching



Packet Switching



Packet Switching



Comparison table

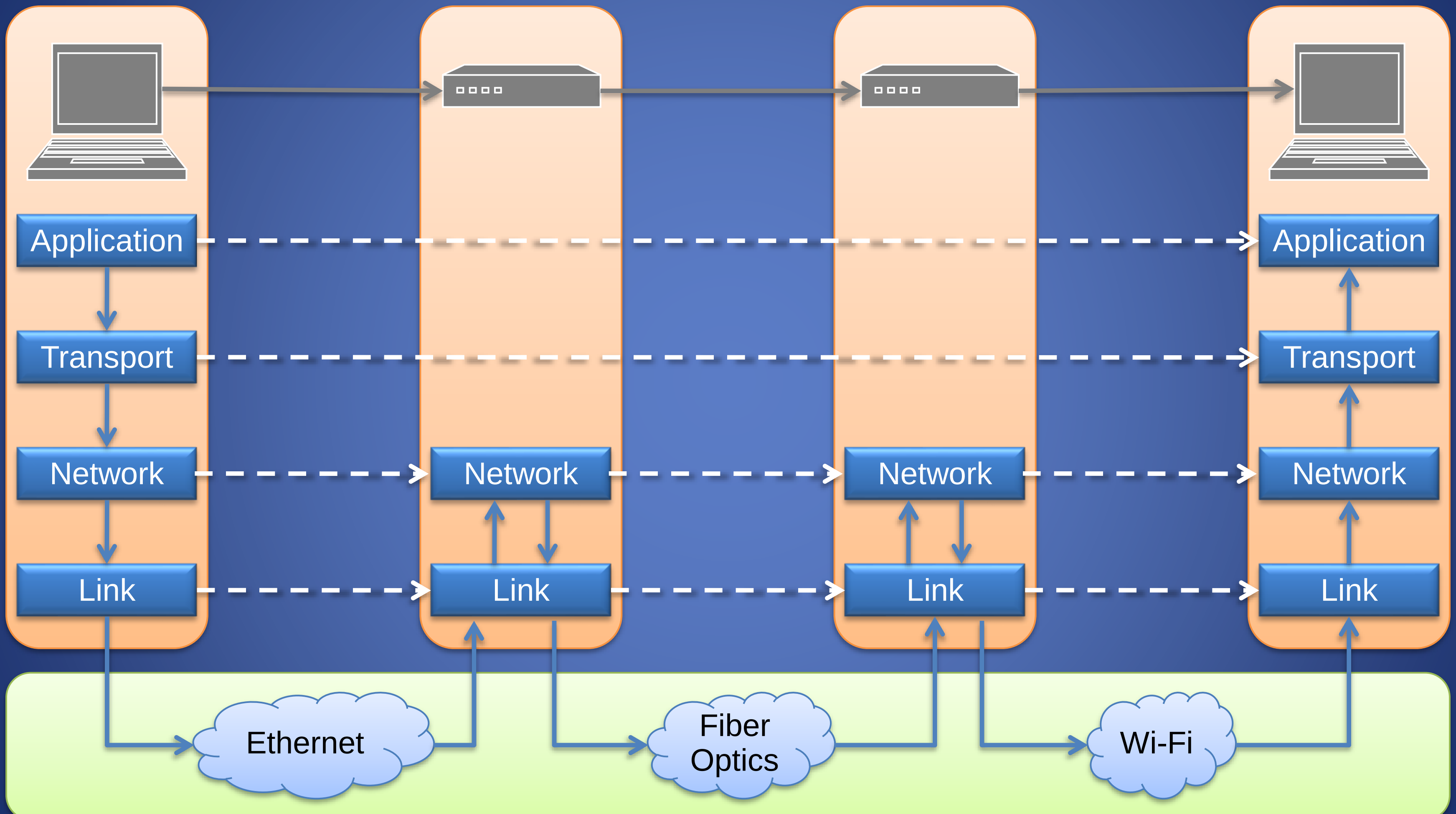
Circuit Switching	Packet Switching(Virtual Circuit type)	Packet Switching(Datagram type)
Dedicated path	No Dedicated path	No Dedicated path
Path is established for entire conversation	Route is established for entire conversation	Route is established for each packet
Call setup delay	call setup delay as well as packet transmission delay	packet transmission delay
Overload may block call setup	Overload may block call setup and increases packet delay	Overload increases packet delay
Fixed bandwidth	Dynamic bandwidth	Dynamic bandwidth
No overhead bits after call setup	overhead bits in each packet	overhead bits in each packet

Protocols

- A protocol defines the rules for communication between computers
- Protocols are broadly classified as connectionless and connection oriented
- Connectionless protocol
  - Sends data out as soon as there is enough data to be transmitted
  - E.g., user datagram protocol (UDP)
- Connection-oriented protocol
  - Provides a reliable connection stream between two nodes
  - Consists of set up, transmission, and tear down phases
  - Creates virtual circuit-switched network
  - E.g., transmission control protocol (TCP)

reliable

Internet Layers



Physical Layer

Encapsulation

- A packet typically consists of
  - Control information for addressing the packet: header and footer
  - Data: payload
- A network protocol N1 can use the services of another network protocol N2
  - A packet p1 of N1 is encapsulated into a packet p2 of N2
  - The payload of p2 is p1
  - The control information of p2 is derived from that of p1

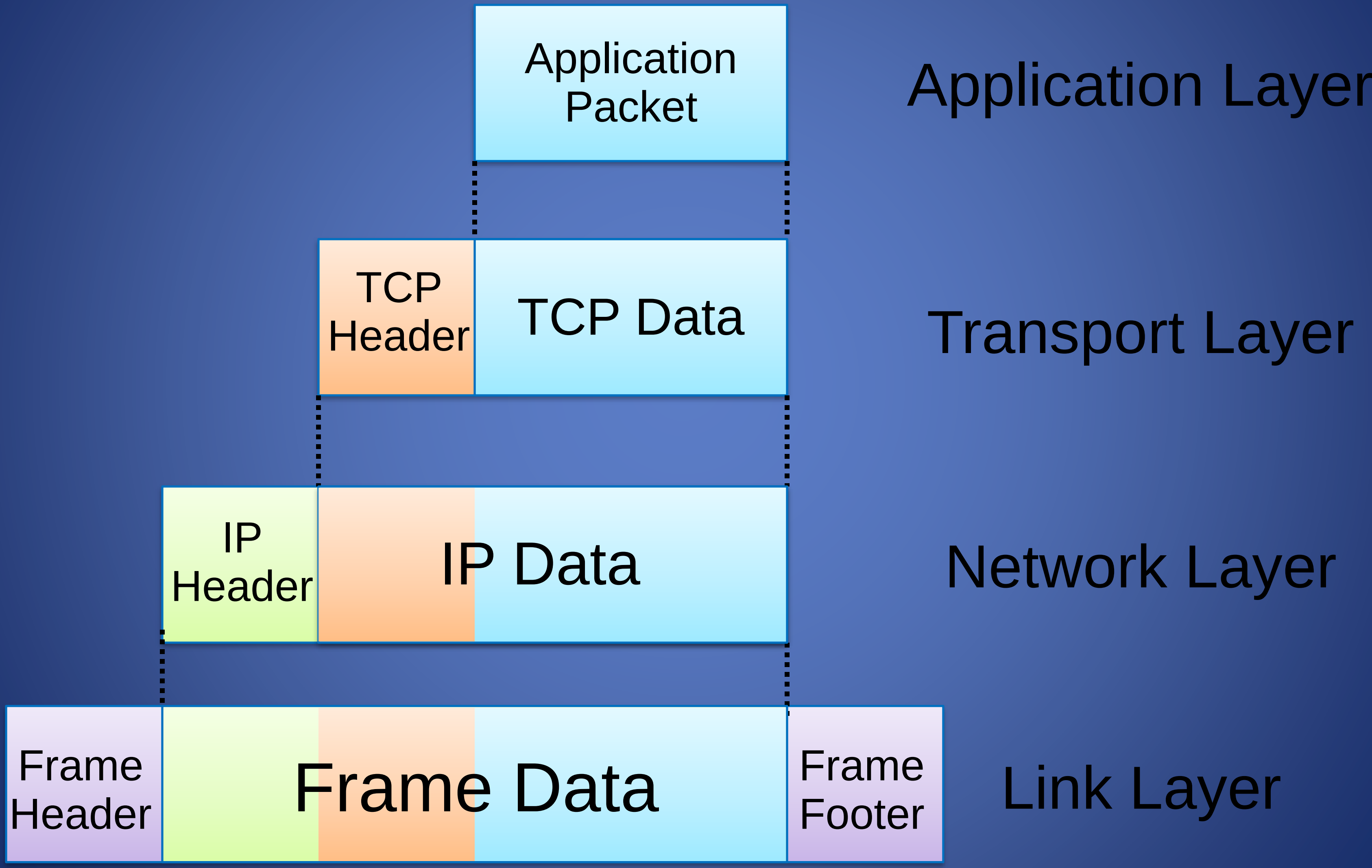
Network Layers

- Network models typically use a stack of layers
  - Higher layers use the services of lower layers via encapsulation
  - A layer can be implemented in hardware or software
  - The bottommost layer must be in hardware
- A network device may implement several layers
- A communication channel between two nodes is established for each layer
  - Actual channel at the bottom layer
  - Virtual channel at higher layers

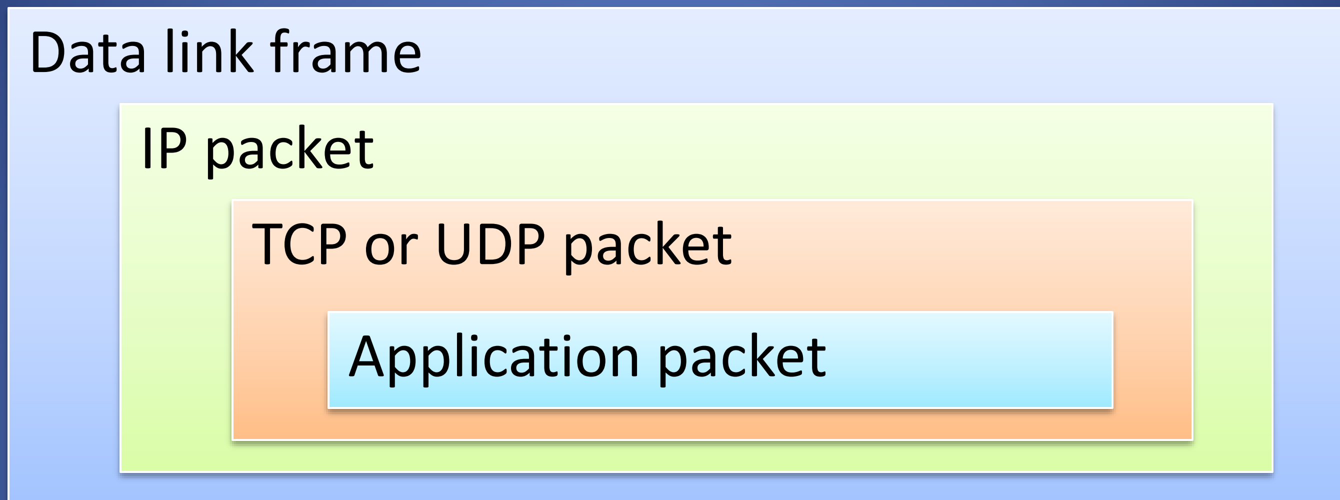
Intermediate Layers

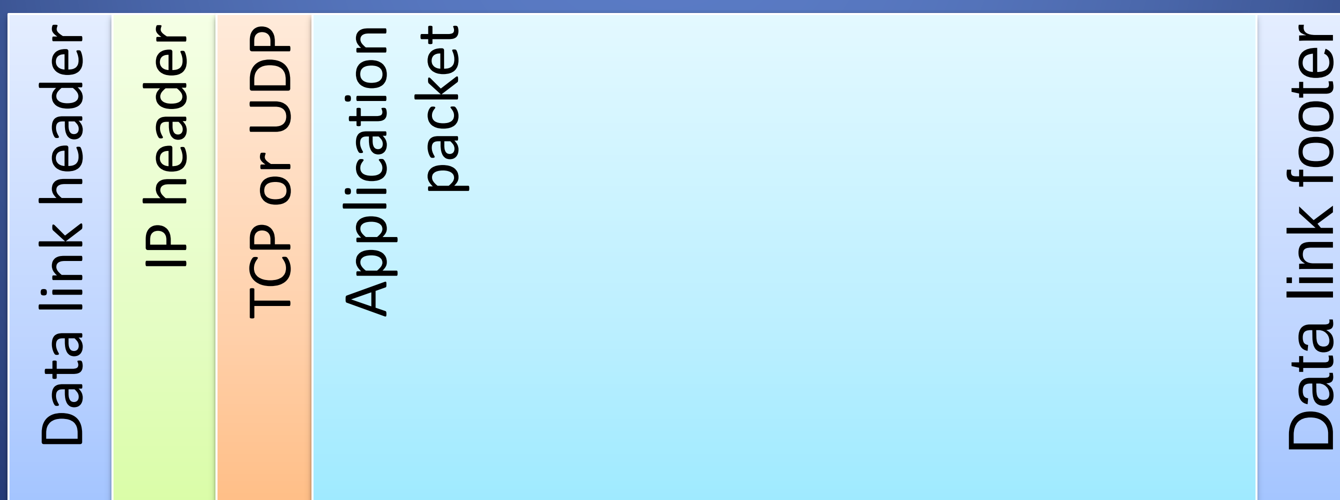
- Link layer
  - Local area network: Ethernet, WiFi, optical fiber
  - 48-bit media access control (MAC) addresses
  - Packets called frames
- Network layer
  - Internet-wide communication
  - Best efforts
  - 32-bit internet protocol (IP) addresses in IPv4
  - 128-bit IP addresses in IPv6
- Transport layer
  - 16-bit addresses (ports) for classes of applications
  - Connection-oriented transmission layer protocol (TCP)
  - Connectionless user datagram protocol (UDP)

Internet Packet Encapsulation

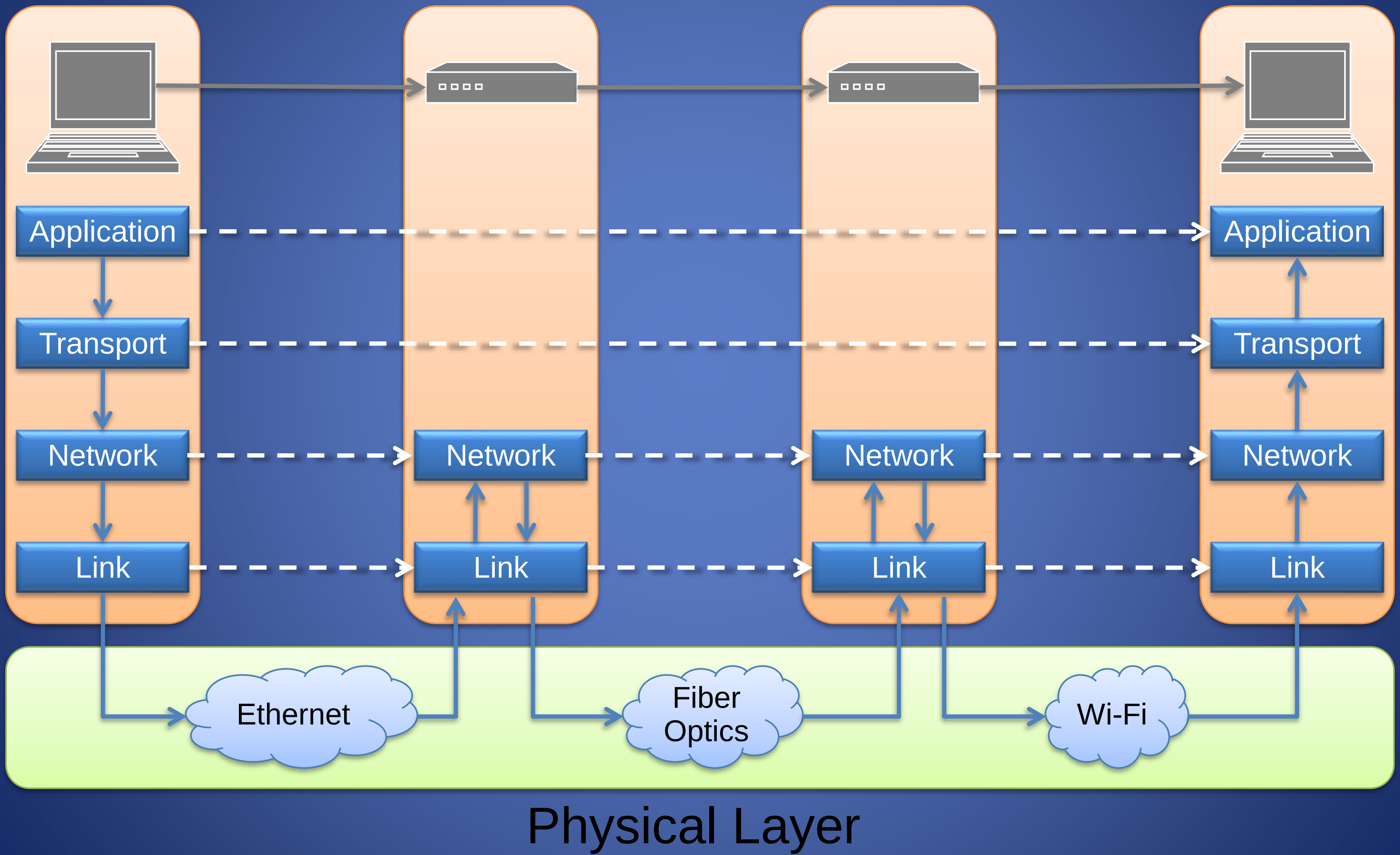


Internet Packet Encapsulation



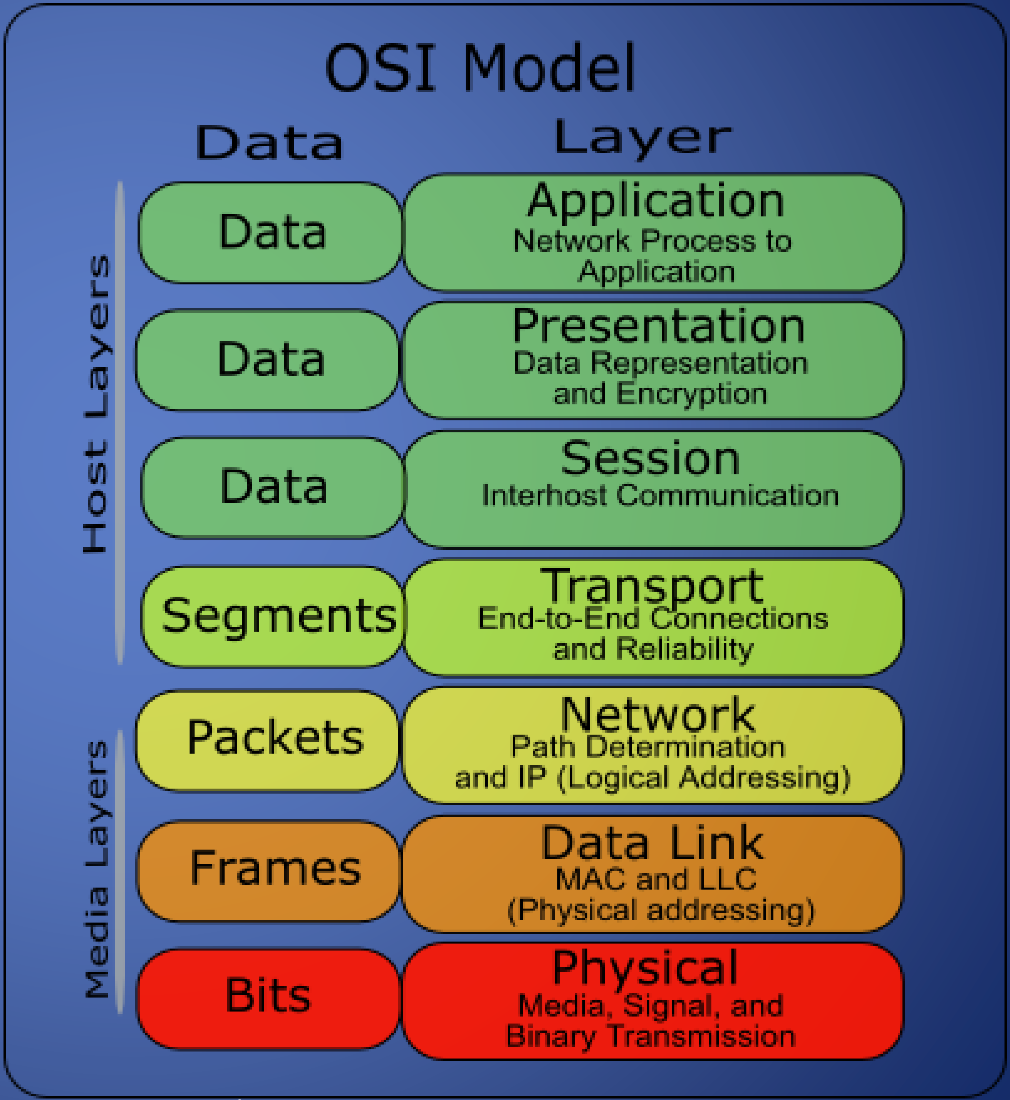


Internet Layers



The OSI Model

- The OSI (Open System Interconnect) Reference Model is a network model consisting of seven layers
- Created in 1983, OSI is promoted by the International Standard Organization (ISO)



Network Interfaces

- Network interface: device connecting a computer to a network
  - Ethernet card
  - WiFi adapter
- A computer may have multiple network interfaces
- Packets transmitted between network interfaces
- Most local area networks, (including Ethernet and WiFi) broadcast frames
- In regular mode, each network interface gets the frames intended for it
- Traffic sniffing can be accomplished by configuring the network interface to read all frames (promiscuous mode)

1.2 The Link Layer: Ethernet

Dealing with Collisions

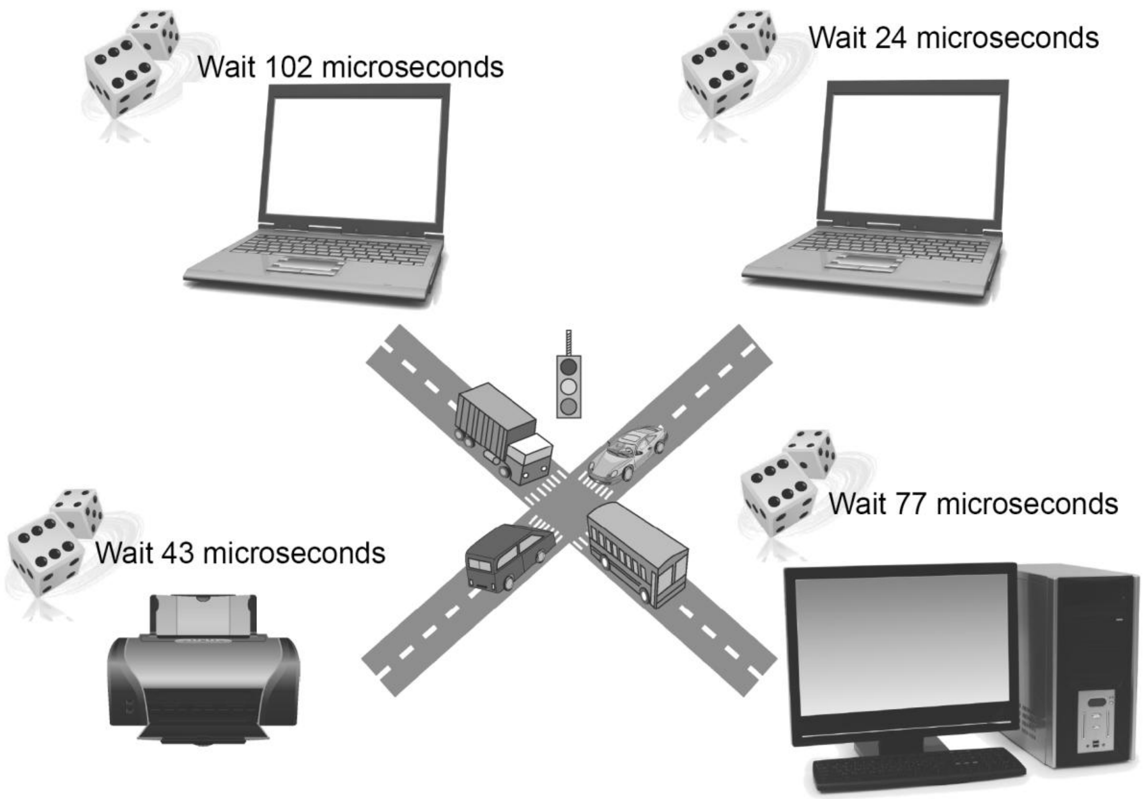
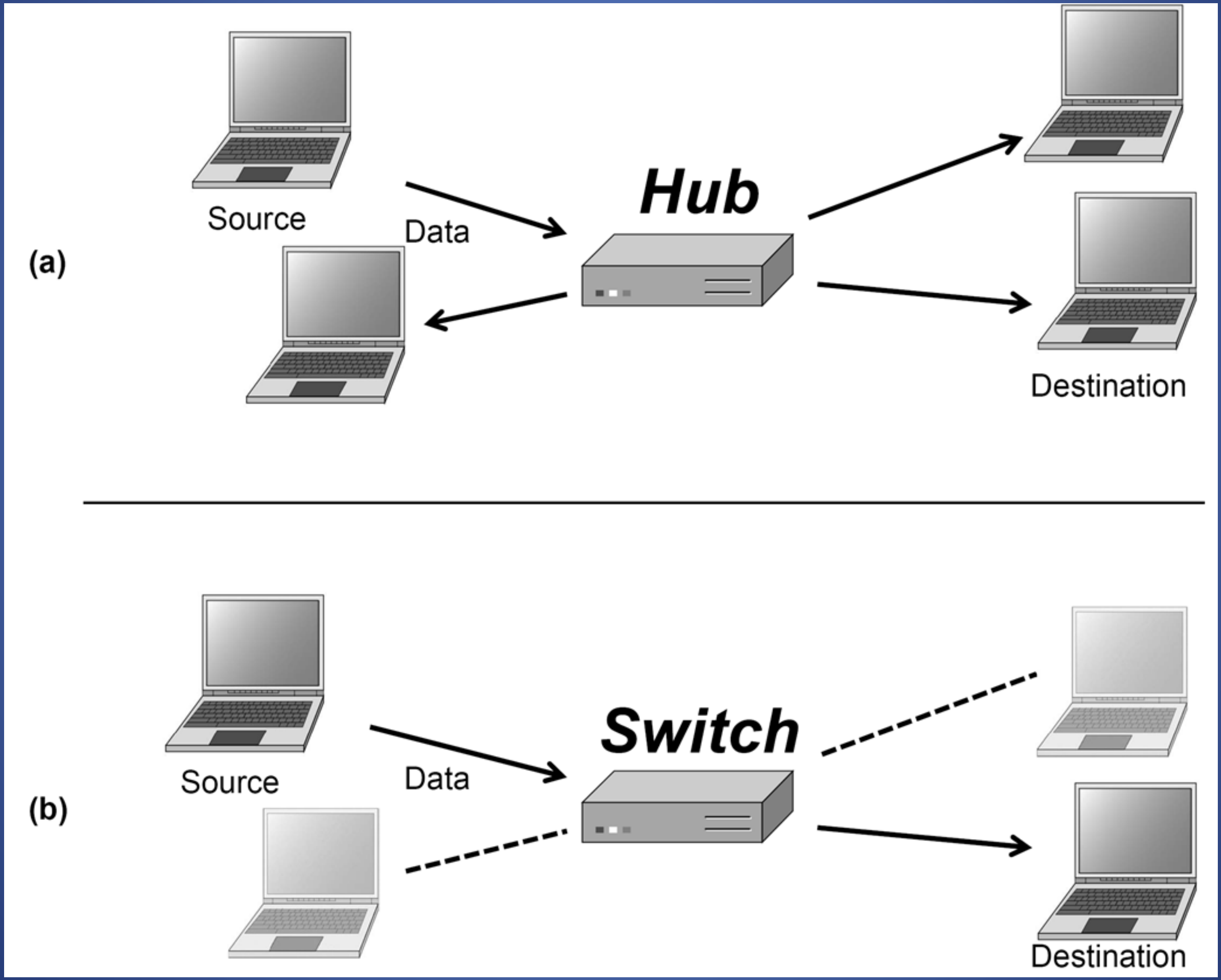


Figure 5.5: An Ethernet collision and how it is handled.

The Format of an Ethernet Frame

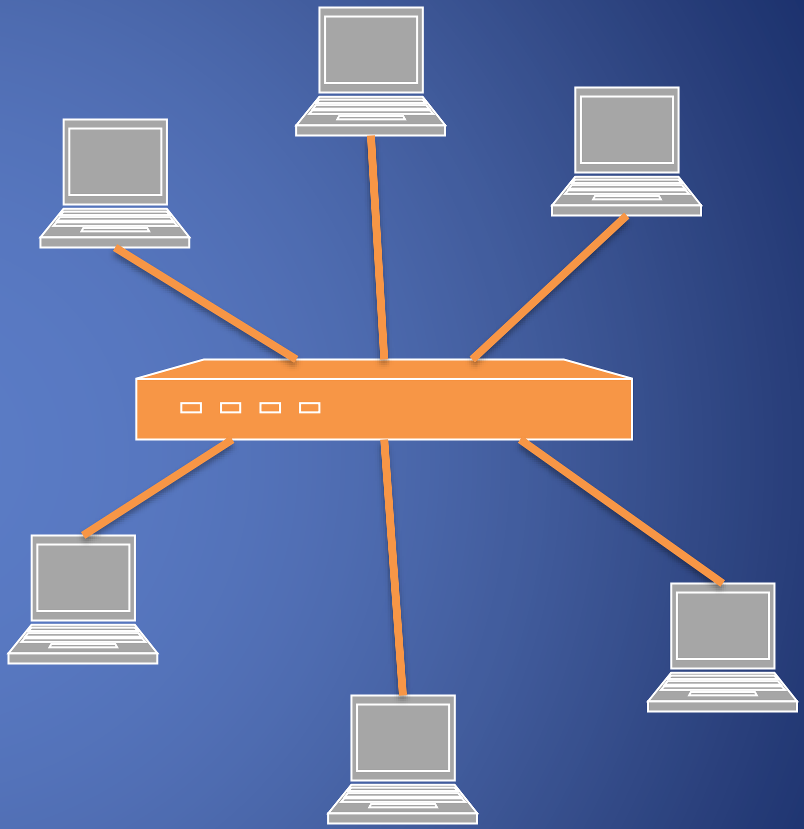
Bits	Field	
0 to 55	Preamble (7 bytes)	Header
56 to 63	Start-of-Frame delimiter (1 byte)	Header
64 to 111	MAC destination (6 bytes)	Header
112 to 159	MAC source (6 bytes)	Header
160 to 175	Ethertype/Length (2 bytes)	Header
176 to 543+	Payload (46-1500 bytes)	Payload
543+ to 575+	CRC-32 checksum (4 bytes)	Footer
575+ to 671+	Interframe gap (12 bytes)	Footer

Hubs and Switches



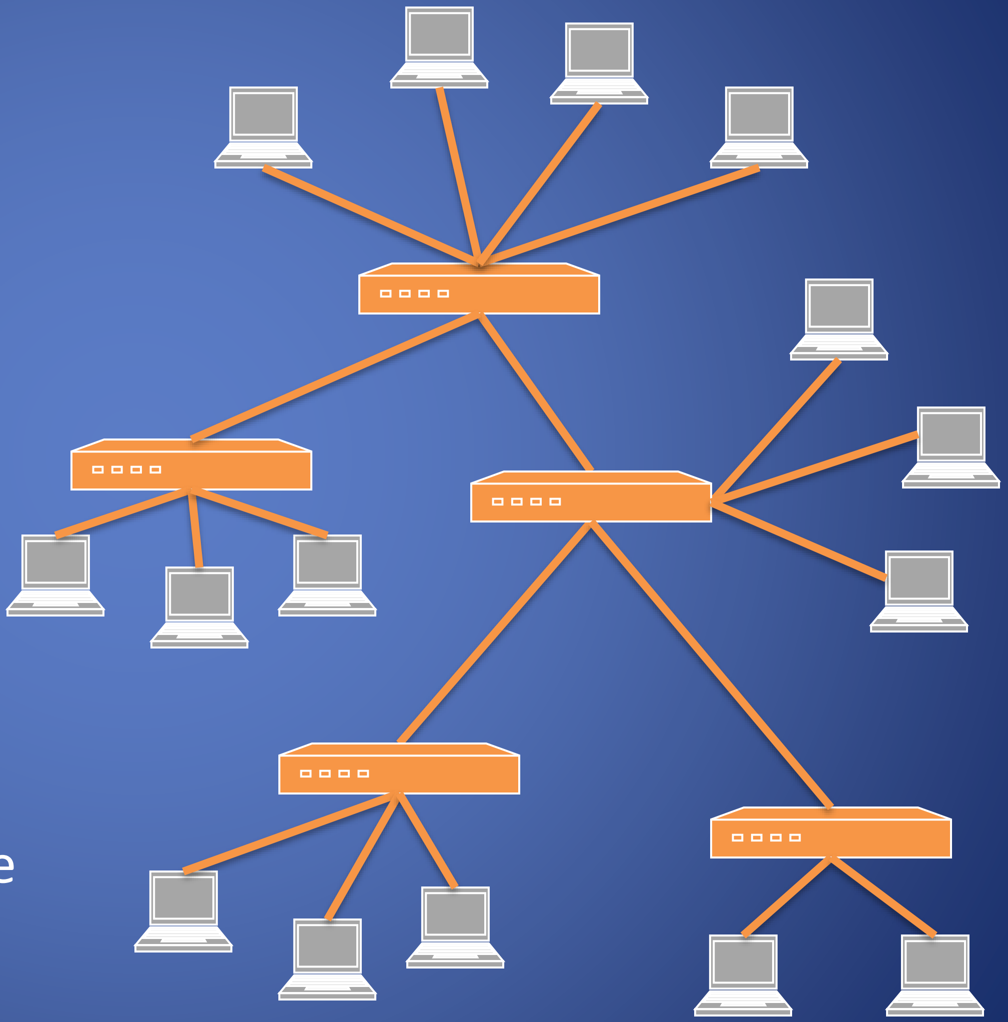
Switch

- A switch is a common network device
  - Operates at the link layer
  - Has multiple ports, each connected to a computer
- Operation of a switch
  - Learn the MAC address of each computer connected to it
  - Forward frames only to the destination computer



Combining Switches

- Switches can be arranged into a tree
- Each port learns the MAC addresses of the machines in the segment (subtree) connected to it
- Fragments to unknown MAC addresses are broadcast
- Frames to MAC addresses in the same segment as the sender are ignored



4.2 The Link Layer: MAC Addresses

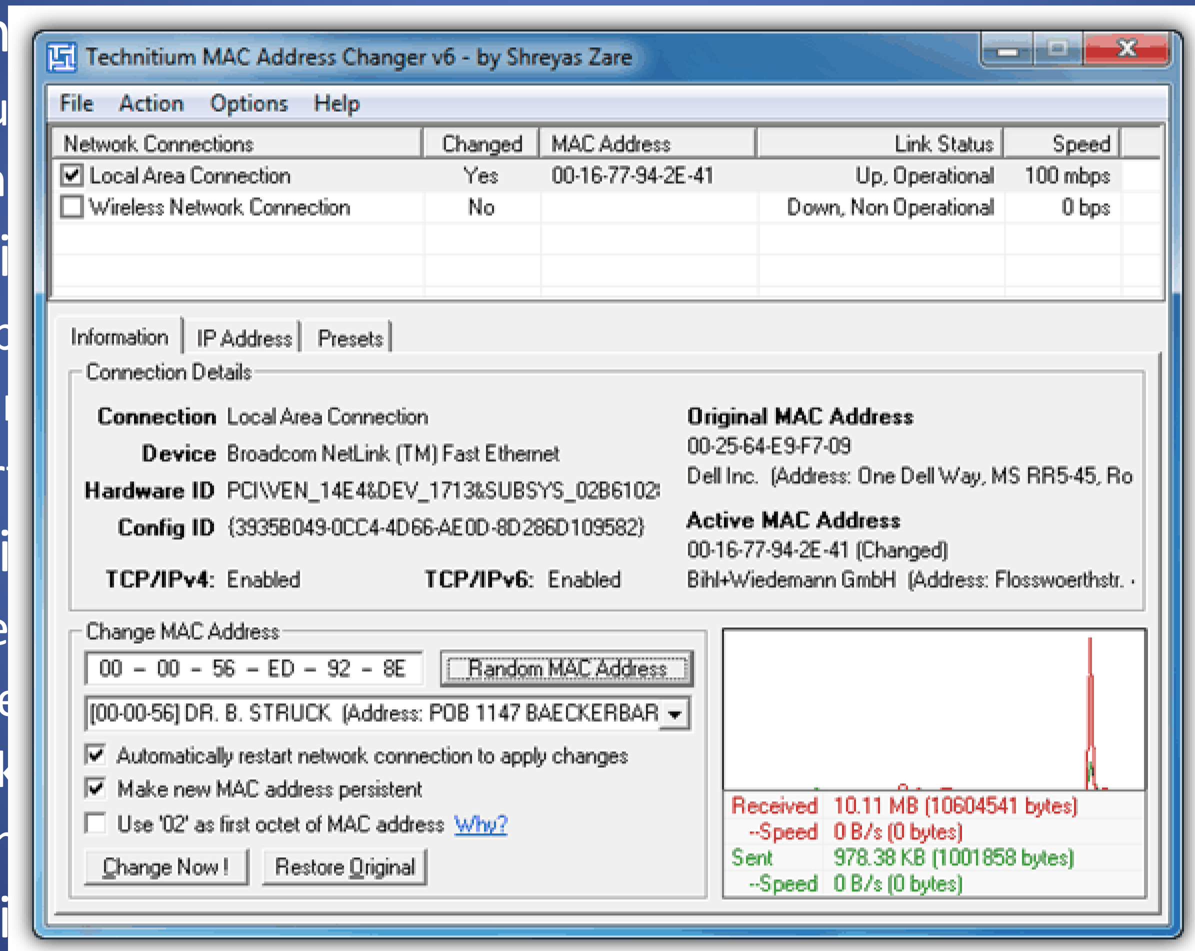
- Most network interfaces come with a predefined MAC address
- A MAC address is a 48-bit number usually represented in hex
  - E.g., 00-1A-92-D4-BF-86
- The first three octets of any MAC address are IEEE-assigned Organizationally Unique Identifiers
  - E.g., Cisco 00-1A-A1, D-Link 00-1B-11, ASUSTek 00-1A-92
- The next three can be assigned by organizations as they please, with uniqueness being the only constraint
- Organizations can utilize MAC addresses to identify computers on their network
- MAC address can be reconfigured by network interface driver software

MAC Address Filtering

- A switch can be configured to provide service only to machines with specific MAC addresses
- Allowed MAC addresses need to be registered with a network administrator
- A MAC spoofing attack impersonates another machine
  - Find out MAC address of target machine
  - Reconfigure MAC address of rogue machine
  - Turn off or unplug target machine
- Countermeasures
  - Block port of switch when machine is turned off or unplugged
  - Disable duplicate MAC addresses

Viewing and Changing MAC Addresses

- Viewin ... chine
  - Linu
  - Win
- Changi
  - Stop
  - Char ... ress>
  - Star
- Changi
  - Ope
  - Acce
  - Click
  - In th ... sired value
- Changi ... eges



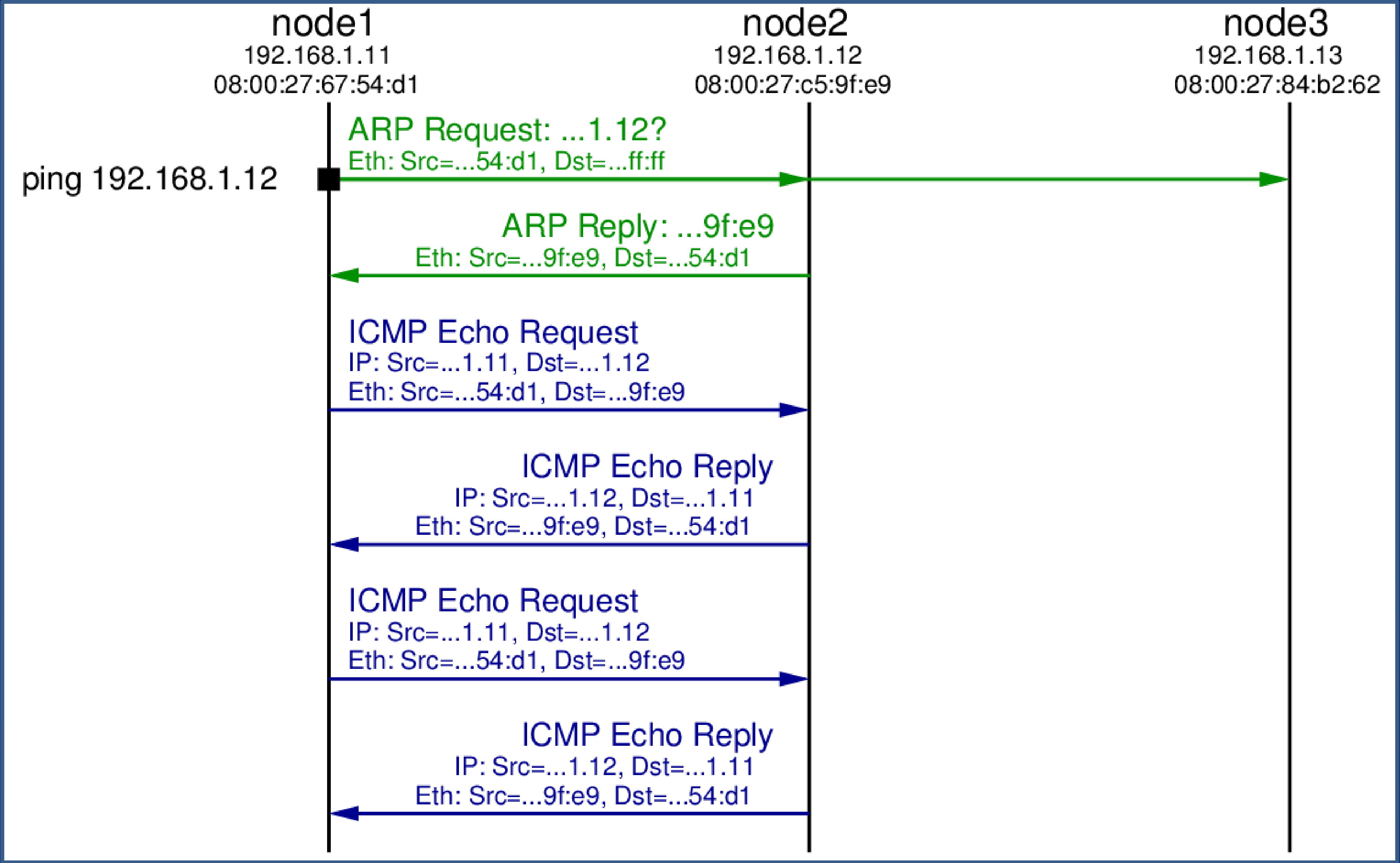
1.2 The Link Layer: ARP

- The address resolution protocol (ARP) connects the network layer to the data layer by converting IP addresses to MAC addresses
- ARP works by broadcasting requests and caching responses for future use
- The protocol begins with a computer broadcasting a message of the form

  who has <IP address1> tell <IP address2>
- When the machine with <IP address1> or an ARP server receives this message, its broadcasts the response

  <IP address1> is <MAC address>
- The requestor's IP address <IP address2> is contained in the link header
- The Linux and Windows command arp - a displays the ARP table

Internet Address	Physical Address	Type
128.148.31.1	00-00-0c-07-ac-00	dynamic
128.148.31.15	00-0c-76-b2-d7-1d	dynamic
128.148.31.71	00-0c-76-b2-d0-d2	dynamic
128.148.31.75	00-0c-76-b2-d7-1d	dynamic
128.148.31.102	00-22-0c-a3-e4-00	dynamic
128.148.31.137	00-1d-92-b6-f1-a9	dynamic

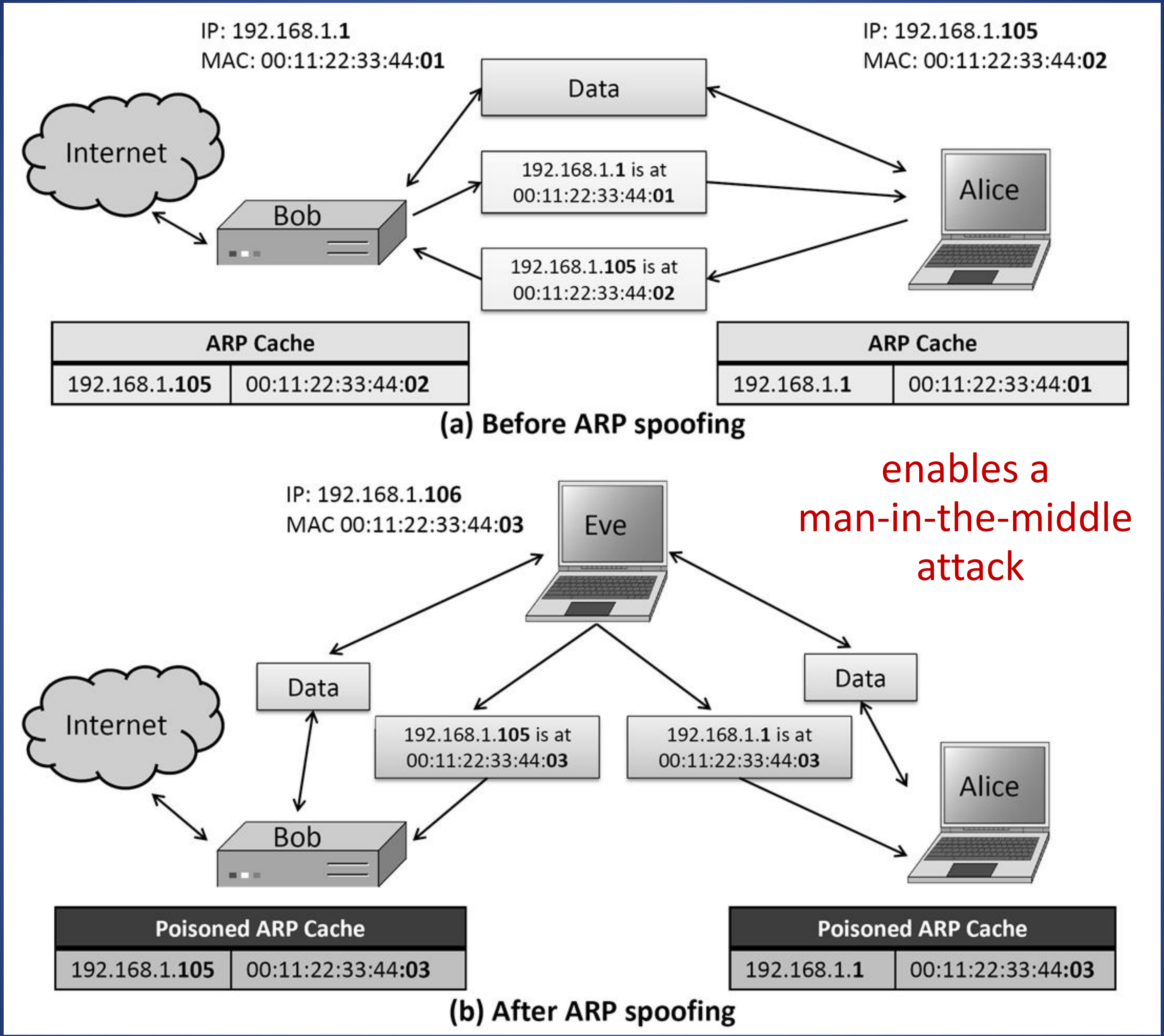

node1
192.168.1.11
08:00:27:67:54:d1
node2
192.168.1.12
08:00:27:c5:9f:e9
node3
192.168.1.13
08:00:27:84:b2:62
ping 192.168.1.12
ARP Request: ...1.12?
Eth: Src=...54:d1, Dst=...ff:ff
ARP Reply: ...9f:e9
Eth: Src=...9f:e9, Dst=...54:d1
ICMP Echo Request
IP: Src=...1.11, Dst=...1.12
Eth: Src=...54:d1, Dst=...9f:e9
ICMP Echo Reply
IP: Src=...1.12, Dst=...1.11
Eth: Src=...9f:e9, Dst=...54:d1
ICMP Echo Request
IP: Src=...1.11, Dst=...1.12
Eth: Src=...54:d1, Dst=...9f:e9
ICMP Echo Reply
IP: Src=...1.12, Dst=...1.11
Eth: Src=...9f:e9, Dst=...54:d1

ARP Spoofing

- The ARP table is updated whenever an ARP response is received
- Requests are not tracked
- ARP announcements are not authenticated
- Machines trust each other
- A rogue machine can spoof other machines

ARP Poisoning (ARP Spoofing)

- According to the standard, almost all ARP implementations are stateless
- An arp cache updates every time that it receives an arp reply… even if it did not send any arp request!
- It is possible to “poison” an arp cache by sending gratuitous arp replies
- Using static entries solves the problem but it is almost impossible to manage!



enables a man-in-the-middle attack