Chapter 4

Reference Models and Protocols

Outline

- Network Protocols
- Layered Models
  - The OSI Model
  - The TCP/IP Model
- Comparing OSI Model with TCP/IP Model
- Overview & functions of each layer
- Encapsulation and Decapsulation Process

Network Protocols

- In order for data packets to travel from a source to a destination on a network, it is important that all the devices on the network speak the same language or protocol.
- A data communications protocol is a set of rules or agreements that determines the data format, and how transmission of data occurs.
- A network protocol is a set of standards that make communication on a network more efficient.

Network Protocols

- **Network protocols** are formal standards and policies made up of rules, procedures and formats that defines communication between two or more devices over a network or the Internet
- Protocols are Rules that specify:
  - How the messages are sent
  - How they are directed through the network, and
  - How they are interpreted at the destination devices

Example of Network Protocols

- TCP/IP (Transmission Control Protocol/Internet Protocol) suite
- ARP (Address Resolution Protocol)
- DHCP (Dynamic Host Configuration Protocol)
- DNS (Domain Name System)
- FTP (File Transfer Protocol)
- HTTP (Hyper Text Transfer Protocol)
- HTTPS (Hypertext Transfer Protocol Secure)
- ICMP (Internet Control Message Protocol)
- IGMP (Internet Group Management Protocol)
- IMAP4 (Internet Message Access Protocol version 4)
- NTP (Network Time Protocol)
- SNMP2/3 (Simple Network Management Protocol version 2 or 3)
- SSH (Secure Socket Shell)
- POP3 (Post Office Protocol version 3)
- RTP (Real-time Transport Protocol
- SIP (Session Initiation Protocol)
- TFTP (Trivial File Transfer Protocol)
- TLS (Transport Layer Security)
- UDP (User Datagram Protocol)
- SMTP

Reference Models

- A reference model (Layered Model) is a conceptual blueprint of how communications should take place.
- It addresses all the **processes** required for effective data communication and **divides** these processes into logical groupings called layers.
- When a communication system is designed in this manner, it’s known as layered architecture.

Advantage of Reference Models

- It divides the network communication **process** into **smaller** and simpler components, thus aiding component development, design, and troubleshooting.
- It encourages industry standardization by defining what functions occur at each layer of the model.
- It allows various types of network hardware and software to communicate.
- It prevents changes in one layer from affecting other layers, so it does not hamper development.

Types of Reference Models

- OSI Reference model
    - Open Systems Interconnection
- TCP/IP Reference Model
    - Transfer Control Protocol/Internet Protocol

Open System Interconnection (OSI) Model

- OSI model has been developed by ISO – 'International Organization of Standardization', in the year 1974.
- It is a 7 **layer** architecture with each **layer** having specific functionality to perform
- The OSI isn't a physical model. Rather, it's a set of guidelines that application developers can use to create and implement **applications** that run on a network.

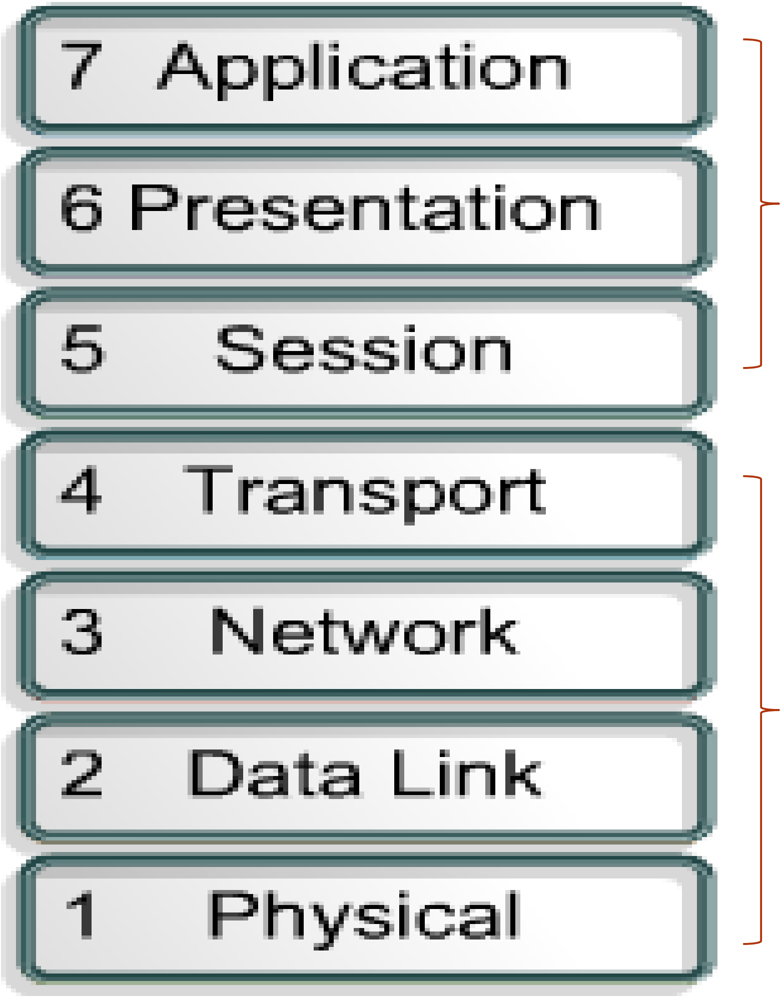
Open System Interconnection (OSI) Model

- OSI model also provides a framework for creating and implementing networking standards, devices, and internetworking schemes.
- The **OSI model** defines a networking framework to implement protocols in **layers**, with control passed from one **layer** to the next

The OSI Model

- The OSI has seven different layers, divided into two groups.
- The top three layers define how the **applications** within the end stations will communicate with each other and with users.
- The bottom four layers define how data is transmitted end-to-end.

Layers of the OSI Model



7 Application

6 Presentation

5 Session

4 Transport

3 Network

2 Data Link

1 Physical

Top 3 layers

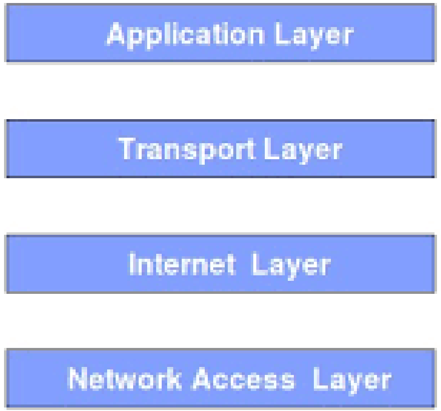
Bottom 4 layers

The TCP/IP Model

- The U.S. Department of Defense (DoD) created the TCP/IP reference model, because it wanted to design a network that could survive under any conditions, including a nuclear war.
- In a world connected by different types of communication media such as copper wires, microwaves, optical fibers and satellite links, the DoD wanted transmission of packets every time and under any conditions. This very difficult design problem brought about the creation of the TCP/IP model.

The TCP/IP Model

- The DoD model is basically a condensed version of the OSI model
- It's composed of four, instead of seven, layers:
  - Application layer
  - Transport layer
  - Internet layer
  - Network Access layer



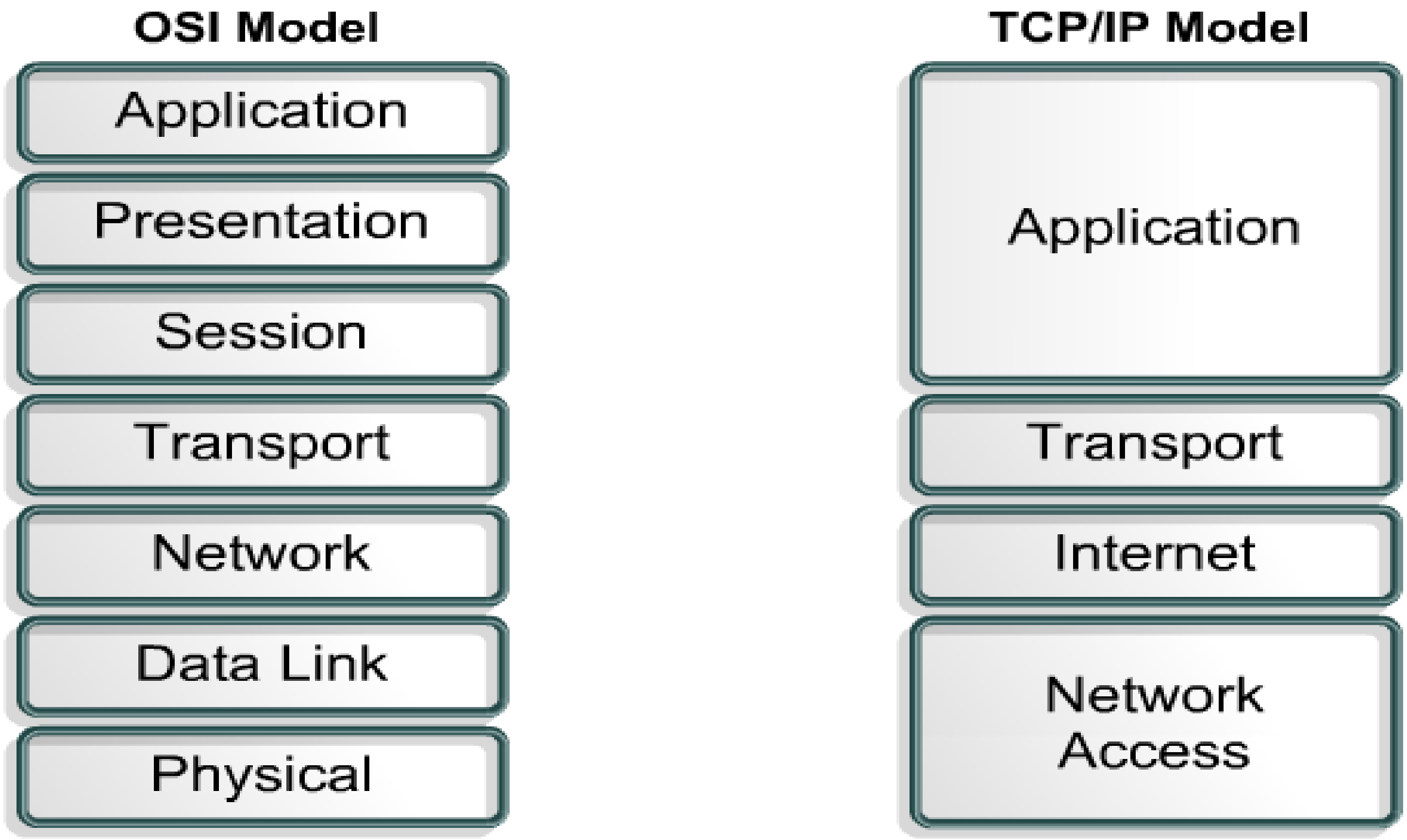
Application Layer

Transport Layer

Internet Layer

Network Access Layer

OSI vs TCP/IP Model



OSI Model

Application

Presentation

Session

Transport

Network

Data Link

Physical

TCP/IP Model

Application

Transport

Internet

Network Access

TCP/IP vs OSI

Similarities include:

- Both have layers.
- Both have application layers, though they include very different services.
- Both have comparable transport layers.
- Both models need to be known by networking professionals.
- Both assume packets are switched.

TCP/IP vs OSI

Differences include:

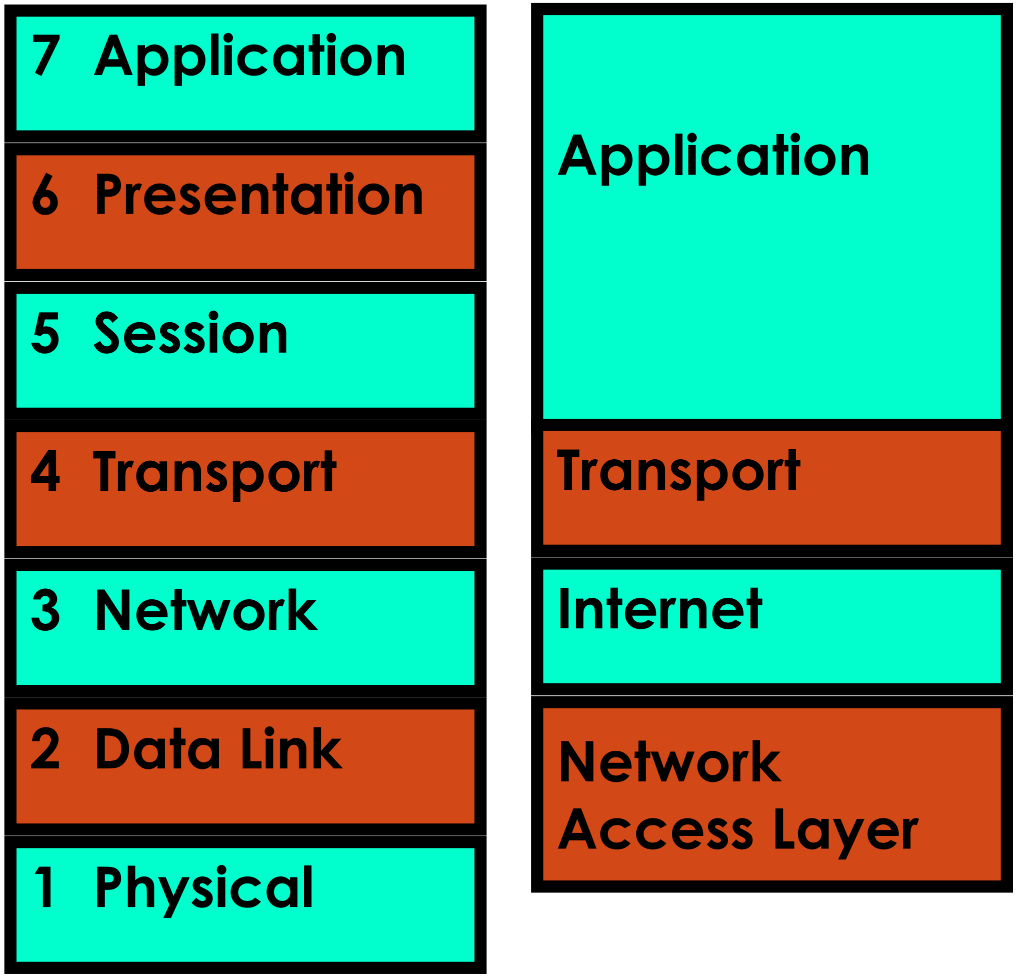
- TCP/IP combines the **presentation** and **session layer** issues into its application layer.
- TCP/IP combines the OSI **data link and physical layers** into the **network access layer**.
- TCP/IP appears simpler because it has fewer layers.
- TCP/IP protocols are the standards around which the **Internet** developed, so the TCP/IP model gains credibility just because of its protocols.

TCP/IP vs OSI

Although TCP/IP protocols are the standards with which the Internet has grown, the OSI model is useful for the following reasons:

- It is a generic standard.
- It has more details, which make it more helpful for teaching and learning, and for troubleshooting.
- Networking professionals differ in their opinions on which model to use. Due to the nature of the industry it is necessary to become familiar with both.
- The OSI model will be used to describe TCP/IP protocols.

Two Models: Side-By-Side



7 Application

6 Presentation

5 Session

4 Transport

3 Network

2 Data Link

1 Physical

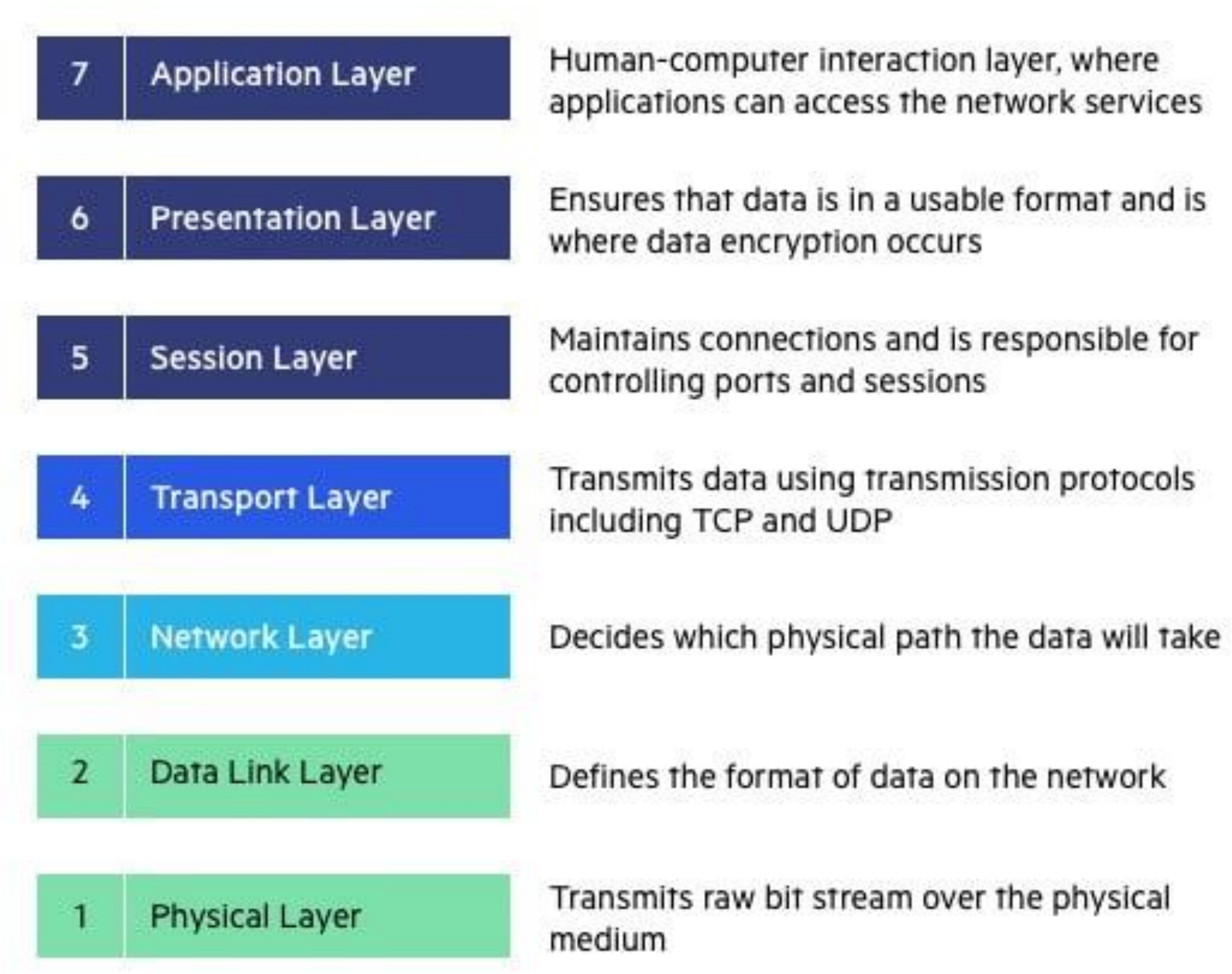
Application

Transport

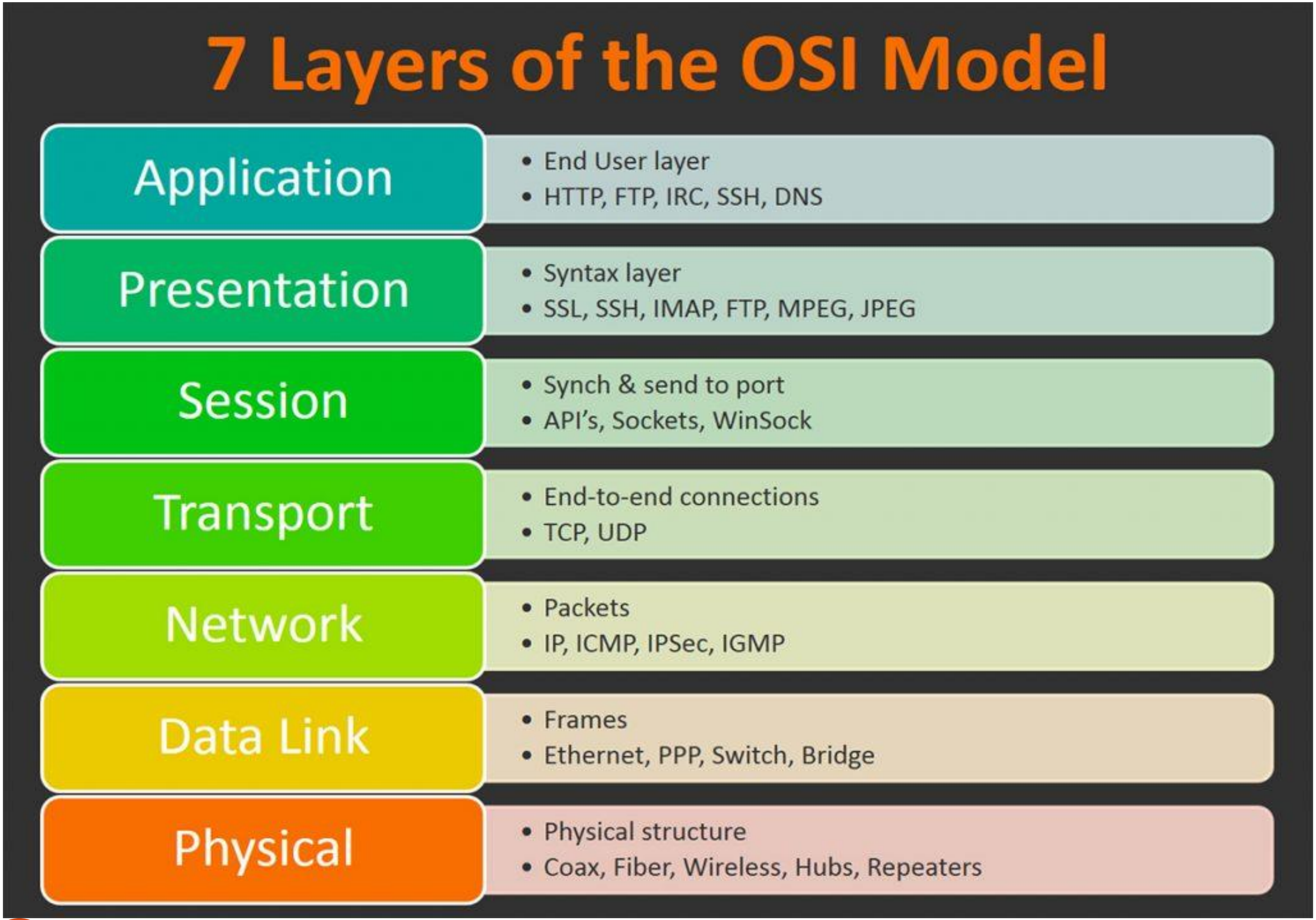
Internet

Network Access Layer

Functions of each layers



7	Application Layer	Human-computer interaction layer, where applications can access the network services
6	Presentation Layer	Ensures that data is in a usable format and is where data encryption occurs
5	Session Layer	Maintains connections and is responsible for controlling ports and sessions
4	Transport Layer	Transmits data using transmission protocols including TCP and UDP
3	Network Layer	Decides which physical path the data will take
2	Data Link Layer	Defines the format of data on the network
1	Physical Layer	Transmits raw bit stream over the physical medium



7 Layers of the OSI Model

Application	• End User layer • HTTP, FTP, IRC, SSH, DNS
Presentation	• Syntax layer • SSL, SSH, IMAP, FTP, MPEG, JPEG
Session	• Synch & send to port • API's, Sockets, WinSock
Transport	• End-to-end connections • TCP, UDP
Network	• Packets • IP, ICMP, IPSec, IGMP
Data Link	• Frames • Ethernet, PPP, Switch, Bridge
Physical	• Physical structure • Coax, Fiber, Wireless, Hubs, Repeaters

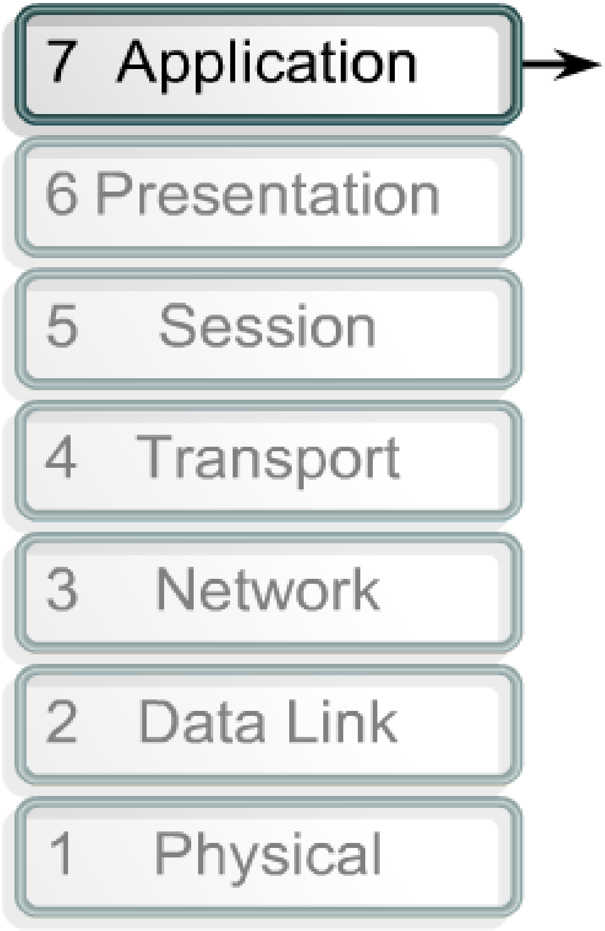
Application Layer (Layer 7)

- The Application layer of the OSI model marks the spot where **users** actually communicate to the computer.
- This layer only comes into play when it's apparent that **access to the network** is going to be needed
- It consists of protocols that focus on process-to-process communication across an IP network and provides a firm communication interface and end-user services.

Application Layer

- The application layer as the **user interface** responsible for displaying received information to the user.
- The application layer is used in both of the standard models: (TCP/IP) and the OSI models.
- The applications layer defines interface to user processes for communication and data transfer in network

Application Layer



7 Application

6 Presentation

5 Session

4 Transport

3 Network

2 Data Link

1 Physical

Network Processes to Applications

- Provides network services to application processes (such as electronic mail, file transfer, and terminal emulation)

Application Layer Protocols

Protocols available at the Application layer are:

- Hypertext Transfer Protocol (HTTP)
- Domain Name System (DNS)
- Simple Mail Transfer Protocol (SMTP)
- Post Office Protocol (POP)
- Telnet
- Dynamic Host Configuration Protocol
- File Transfer Protocol (FTP)

WWW services and HTTP (Hyper Text Transfer Protocol)

- When a web address (URL) is typed into a web browser, the web browser establishes a connection to the web server using the HTTP protocol.
- URL (Uniform Resource Locator) and URIs (Uniform Resource Identifier) are the names most people associate with web addresses. (http://www.google.com/resources.html)

Cont’d

- Web browsers are the client applications our computers use to connect to the WWW and access resources stored on a web server.
- As with most server processes, the web server runs as a background service and makes different types of files available.
- Web clients make connections to the web server and request the desired resources.
- The server replies with the resources and, upon receipt, the browser interprets the data and presents it to the user.

Cont'd

- Browsers can interpret and present many data types, such as plain text or Hypertext Markup Language (HTML, the language in which web pages are constructed). Example: user types http://www.google.com/resources.html

First, the browser interprets the three parts of the URL:

1) HTTP (the protocol or scheme)
2) www.google.com (the server name)
3) resource.html (the specific file name requested).

Cont'd

- The browser then checks with a DNS server to convert www.google.com <http://www.google.com > into a numeric address, which it uses to connect to the server.
- Using the HTTP protocol requirements, the browser sends a GET request to the server and asks for the file resource.html.
- The server in turn sends the HTML code for this web page to the browser. Finally, the browser deciphers the HTML code and formats the page for the browser window.

DNS (Domain Name System)

- In data networks each device has a unique IP address in order to communicate with devices on the data network. (198.132.219.25)
- Difficult to remember each and every IP address, hence domain names were used as a solution (www.google.com)
- As networks grew larger it became difficult to maintain or resolve the domain names and IP addresses manually, hence a system was formulated.

Cont'd

- The Domain Name System (DNS) was created for domain name to address resolution for these networks.

- DNS uses a **distributed set of servers** to resolve the names associated with these numbered addresses (IP Addresses).

- The DNS protocol defines an automated service that matches resource names with the required numeric network address.

FTP (File Transfer Protocol)

- FTP was developed to allow for file transfers between a client and a server.
- An FTP client is an application that runs on a computer that is used to **push** and **pull** files from the FTP server.
- The file transfer can happen in either direction.
- The client can download (pull) a file from the server or, the client can upload (push) a file to the server.

DHCP
(Dynamic Host Configuration Protocol)

- The Dynamic Host Confirmation Protocol (DHCP) service enables devices on a network to obtain IP addresses and other information from a DHCP server.
- The DHCP service automates the assignment of:
  - IP addresses
  - Subnet masks
  - Default Gateway and other IP networking parameters.

Cont'd

- The DHCP server is contacted and an address requested.

- The DHCP server chooses an address from a configured range of addresses called a **pool** and assigns ("leases") it to the client for a set of periods.

- On a larger local networks, or where the user population (number of computers) changes frequently, DHCP is preferred.

Presentation Layer (Layer 6)

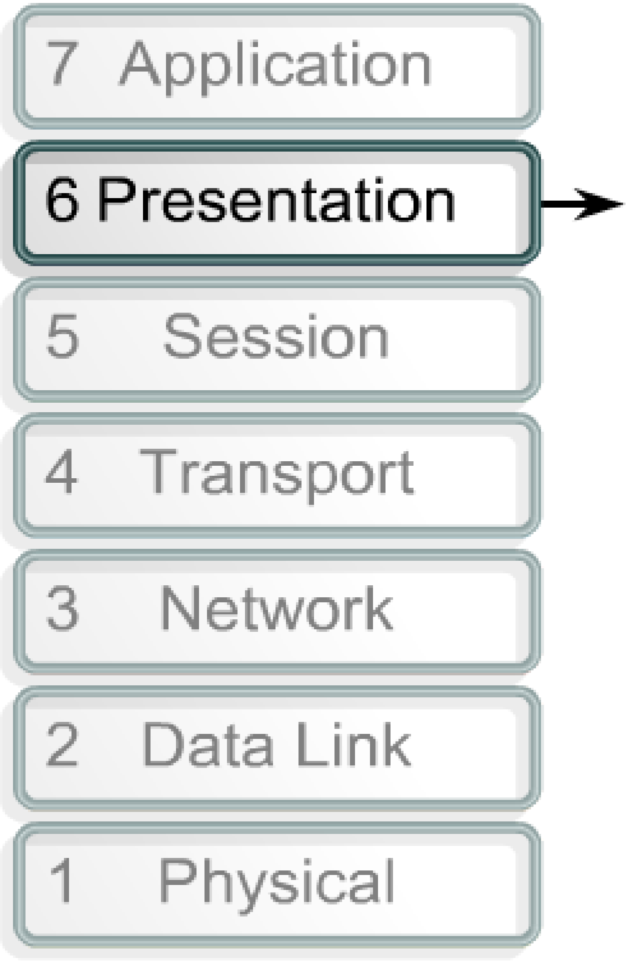
The Presentation layer gets its name from its purpose:

- It presents data to the Application layer
- It is responsible for data translation and code formatting.
- It is sometimes called the syntax layer
- Tasks like data compression, decompression, encryption, and decryption are associated with this layer.
- This layer is essentially a translator and provides coding and conversion functions.

Presentation Layer

- A successful data-transfer technique is to adapt the data into a standard format before transmission.
- Computers are configured to receive this generically formatted data and then convert the data back into its native format for actual reading.
- The presentation layer is the layer the **operating systems** interacts with.

Presentation Layer



7 Application

6 Presentation

5 Session

4 Transport

3 Network

2 Data Link

1 Physical

Data Representation

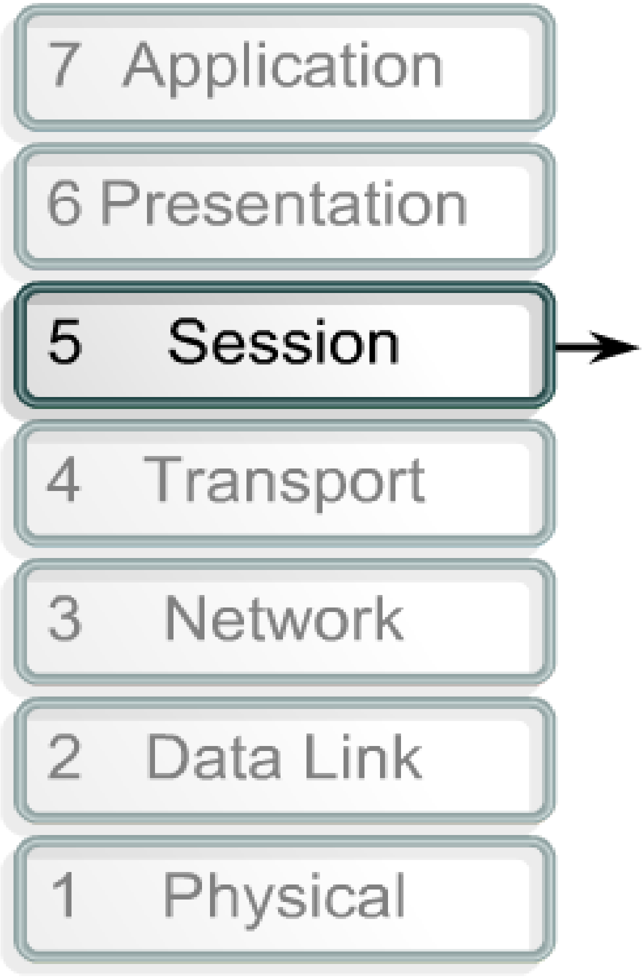
- Ensure data is readable by receiving system
- Format of data
- Data structures
- Negotiates data transfer syntax for application layer

- Masks the differences of data formats between dissimilar systems
- Specifies architecture-independent data transfer format
- Encodes and decodes data;
- Encrypts and decrypts data;
- Compresses and decompresses data

Session Layer (Layer 5)

- The Session layer is responsible for setting up, managing, and then tearing down sessions between the sending and receiving entities.
- This layer also provides **dialogue control** between multiple computers, or nodes.
- The session layer controls the connections between multiple computers.
- The session layer tracks the dialogs between computers, which are also called sessions.

Session Layer



7 Application

6 Presentation

5 Session

4 Transport

3 Network

2 Data Link

1 Physical

Interhost Communication

- Establishes, manages, and terminates sessions between applications

- Manages user sessions and dialogues
- Controls establishment and termination of logic links between users
- Reports upper layer errors

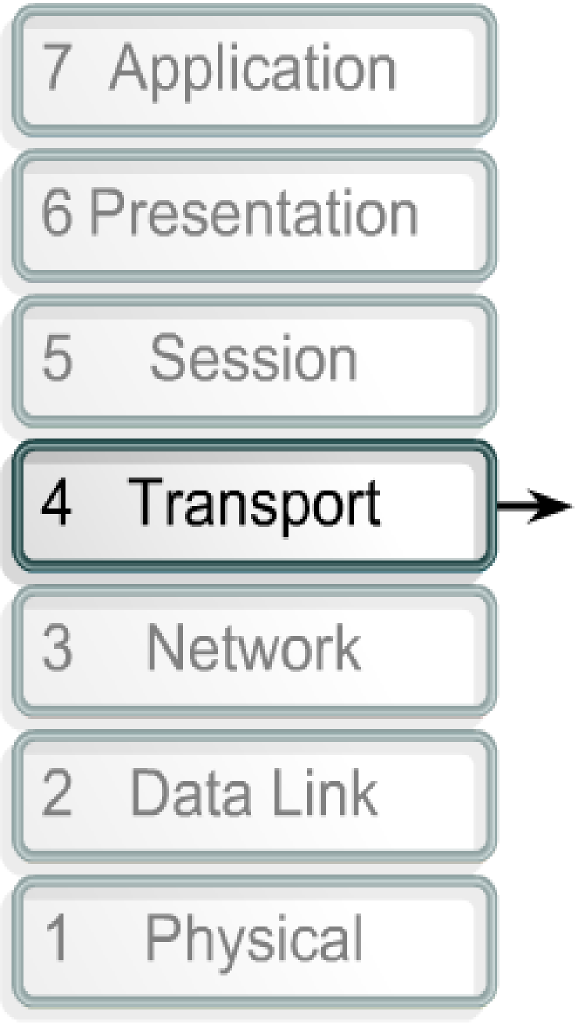
Transport Layer (Layer 4)

- The Transport layer **segments** and **reassembles** data into a data stream.
- Services located in the Transport layer both **segment** and **reassemble** data from upper-layer applications and ***unite*** it onto the same data stream.
- This layer provides **end-to-end** data transport services and can establish a **logical connection** between the sending host and destination host on an internetwork.

Transport Layer

- Transport layer, ***transports*** and **regulates** the flow of information from the source to the destination, **reliably** and **accurately**.
- Manages **end-to-end** message delivery in network
- Provides reliable and sequential packet delivery through error recovery and flow control mechanisms (TCP)
- Provides connectionless oriented packet delivery (UDP)

Transport Layer



End-to-end Connections

- Concerned with transportation issues between hosts
- Data transport reliability
- Establish, maintain, terminate virtual circuits
- Fault detection and recovery information flow control

Transport Layer

The Transport layer services can be:

- Connection-oriented (reliable).
- Connectionless-oriented (unreliable)

Connection –Oriented (Reliable)

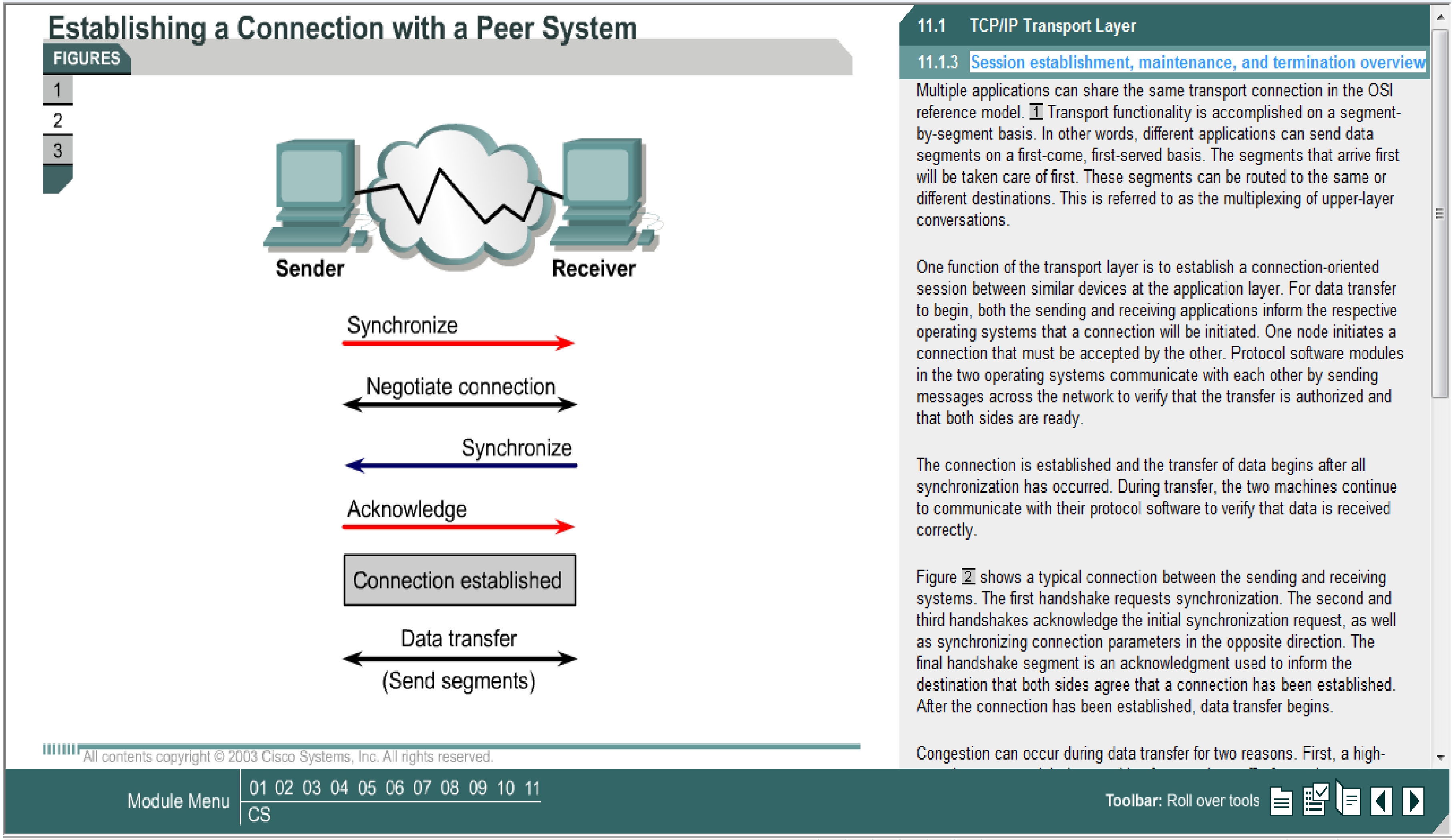
A service is considered connection-oriented if it has the following characteristics:

- A virtual circuit is set up
  - (e.g. three-way handshake)
- It uses Sequencing (sequence number)
- It uses Acknowledgments
- It uses Flow Control

Three-way handshake

- In **reliable** transport operation , a device that wants to transmit sets up a connection-oriented communication with a remote device by creating a Session.
- The transmitting (sender) device ***first*** establishes a connection-oriented session with its peer system (receiver), which is called a **call setup**, or a three-way handshake.
- Once the connection is established, data is then transferred; when finished, a call termination takes place to tear down the virtual circuit (VC).

Session establishment, maintenance, and termination



Multiple applications can share the same transport connection in the OSI reference model. 1 Transport functionality is accomplished on a segment-by-segment basis. In other words, different applications can send data segments on a first-come, first-served basis. The segments that arrive first will be taken care of first. These segments can be routed to the same or different destinations. This is referred to as the multiplexing of upper-layer conversations.

One function of the transport layer is to establish a connection-oriented session between similar devices at the application layer. For data transfer to begin, both the sending and receiving applications inform the respective operating systems that a connection will be initiated. One node initiates a connection that must be accepted by the other. Protocol software modules in the two operating systems communicate with each other by sending messages across the network to verify that the transfer is authorized and that both sides are ready.

The connection is established and the transfer of data begins after all synchronization has occurred. During transfer, the two machines continue to communicate with their protocol software to verify that data is received correctly.

Figure 2 shows a typical connection between the sending and receiving systems. The first handshake requests synchronization. The second and third handshakes acknowledge the initial synchronization request, as well as synchronizing connection parameters in the opposite direction. The final handshake segment is an acknowledgment used to inform the destination that both sides agree that a connection has been established. After the connection has been established, data transfer begins.

Congestion can occur during data transfer for two reasons. First, a high-

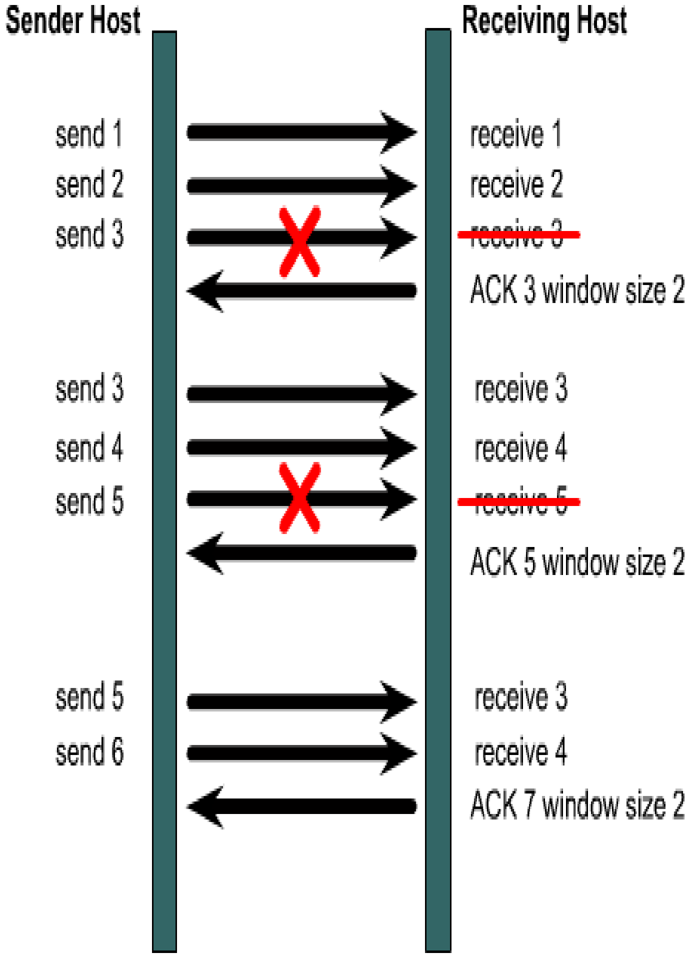
Acknowledgment

- **Reliable data delivery** guarantees that the data won't be duplicated or lost. This is achieved through something called *positive acknowledgment with retransmission (PAR).*
- This technique requires a receiving machine to communicate with the transmitting source by sending an ***acknowledgment*** message back to the sender when it receives data.

Acknowledgment

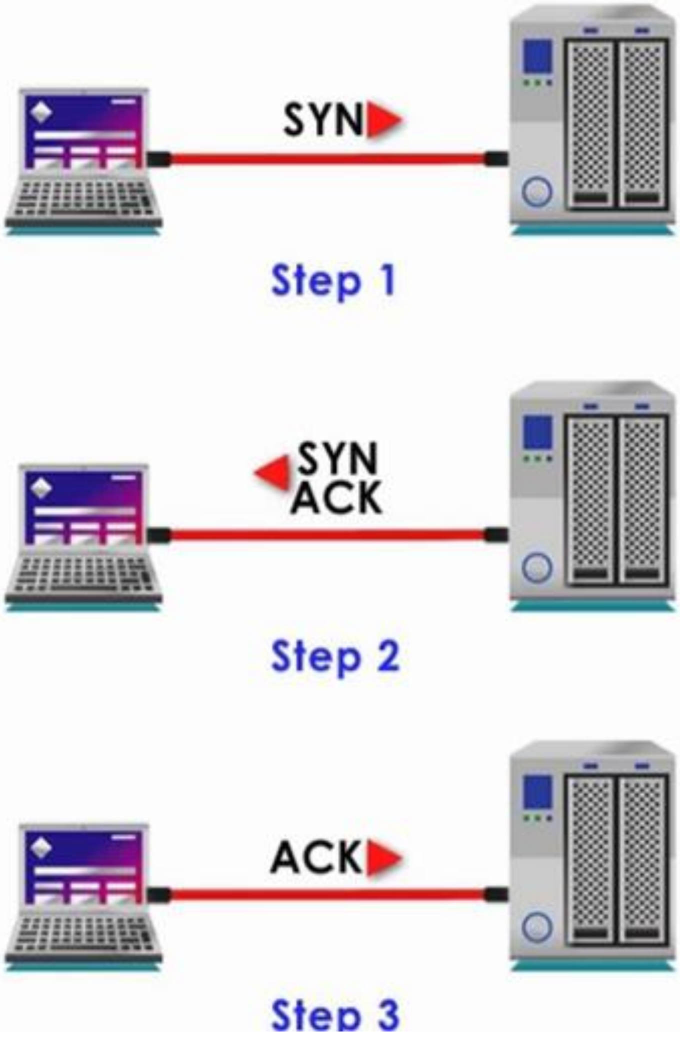
- The sender documents each *segment* it sends and *waits* for this acknowledgment before sending the next segment.
- When it sends a segment, the sender starts a *timer* and retransmits if it expires before an acknowledgment is returned from the receiving end.
- A **three-way handshake** is a method used in a TCP/IP network to create a *connection* between a sender and receiver/server.
- It is a **three**-step method that requires both the client and server to exchange SYN and ACK (**acknowledgment**) packets before actual data communication begins.

Acknowledgement

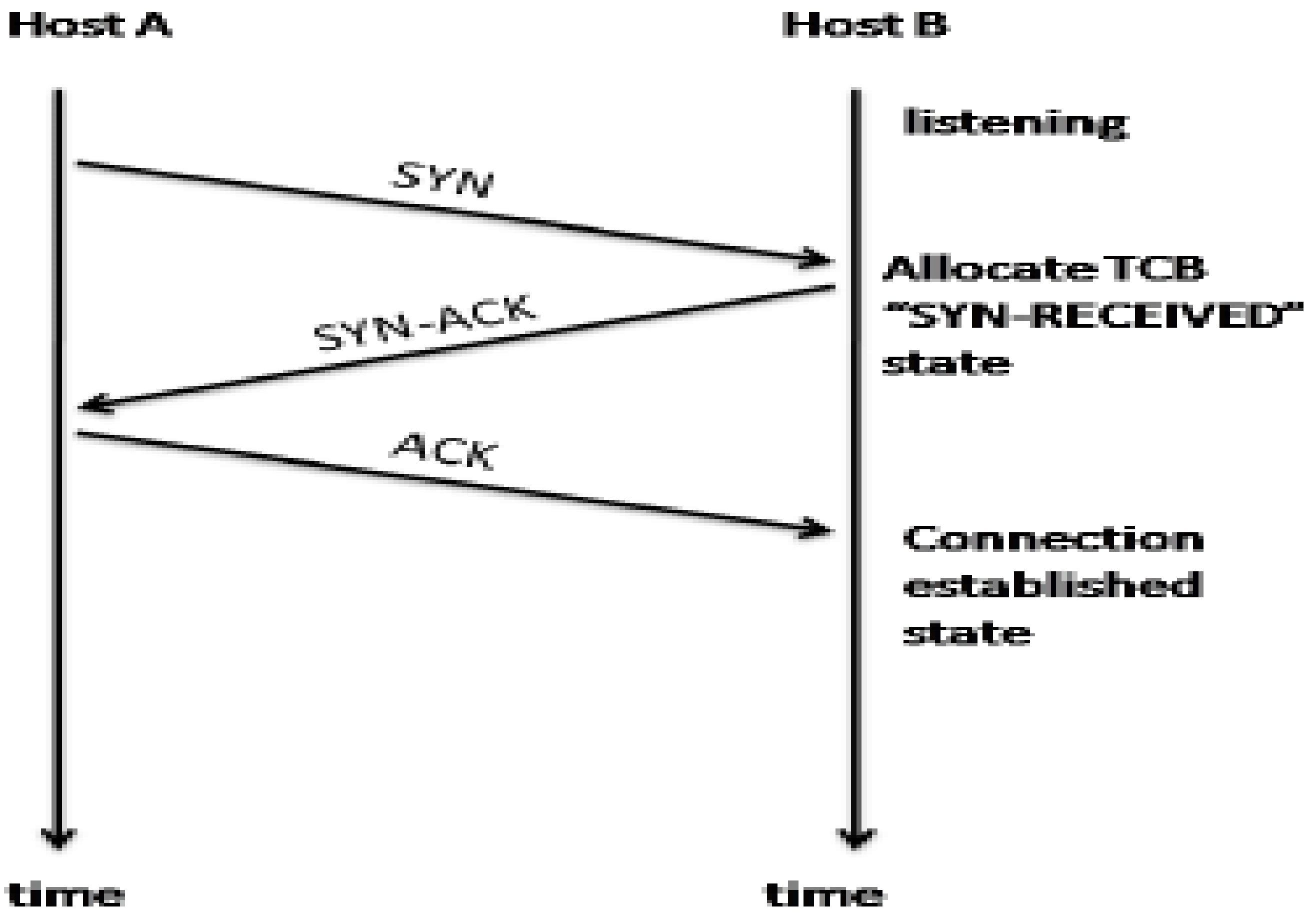


- With a window size of three, the source device can send three packets to the destination.
- The source device must then wait for an acknowledgment.
- If the destination receives the three packets, it sends an acknowledgment to the source device, which can now transmit three more packets.
- If the destination does not receive the three packets, because of overflowing buffers, it does not send an acknowledgment.
- Because the source does not receive an acknowledgment, it knows that the packets should be retransmitted, and that the transmission rate should be slowed.

Three-way handshake to start a session



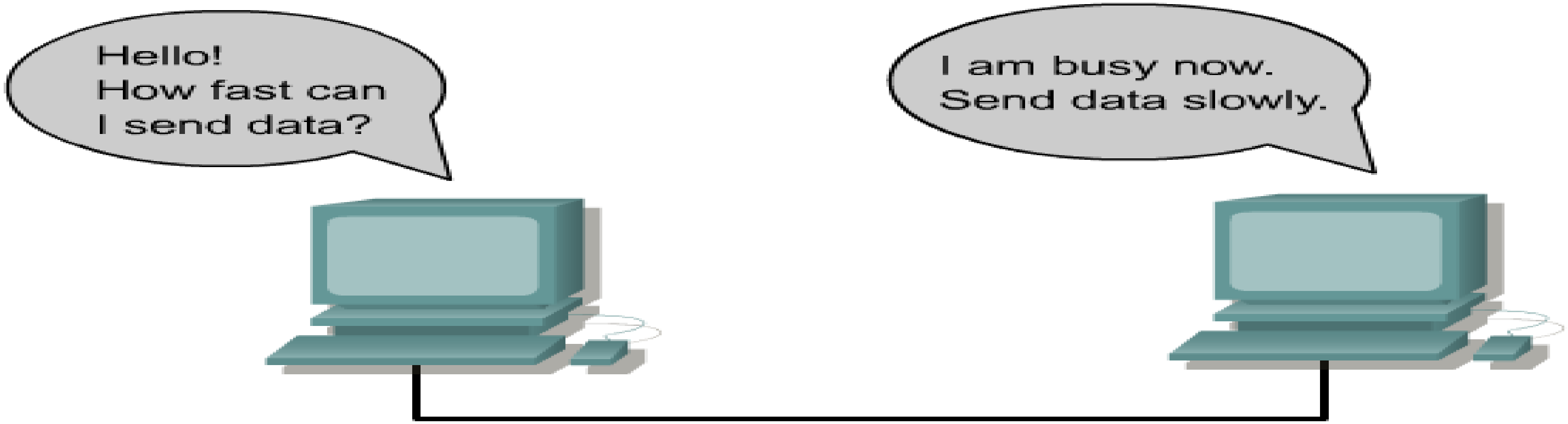
Three-way handshake



Flow Control

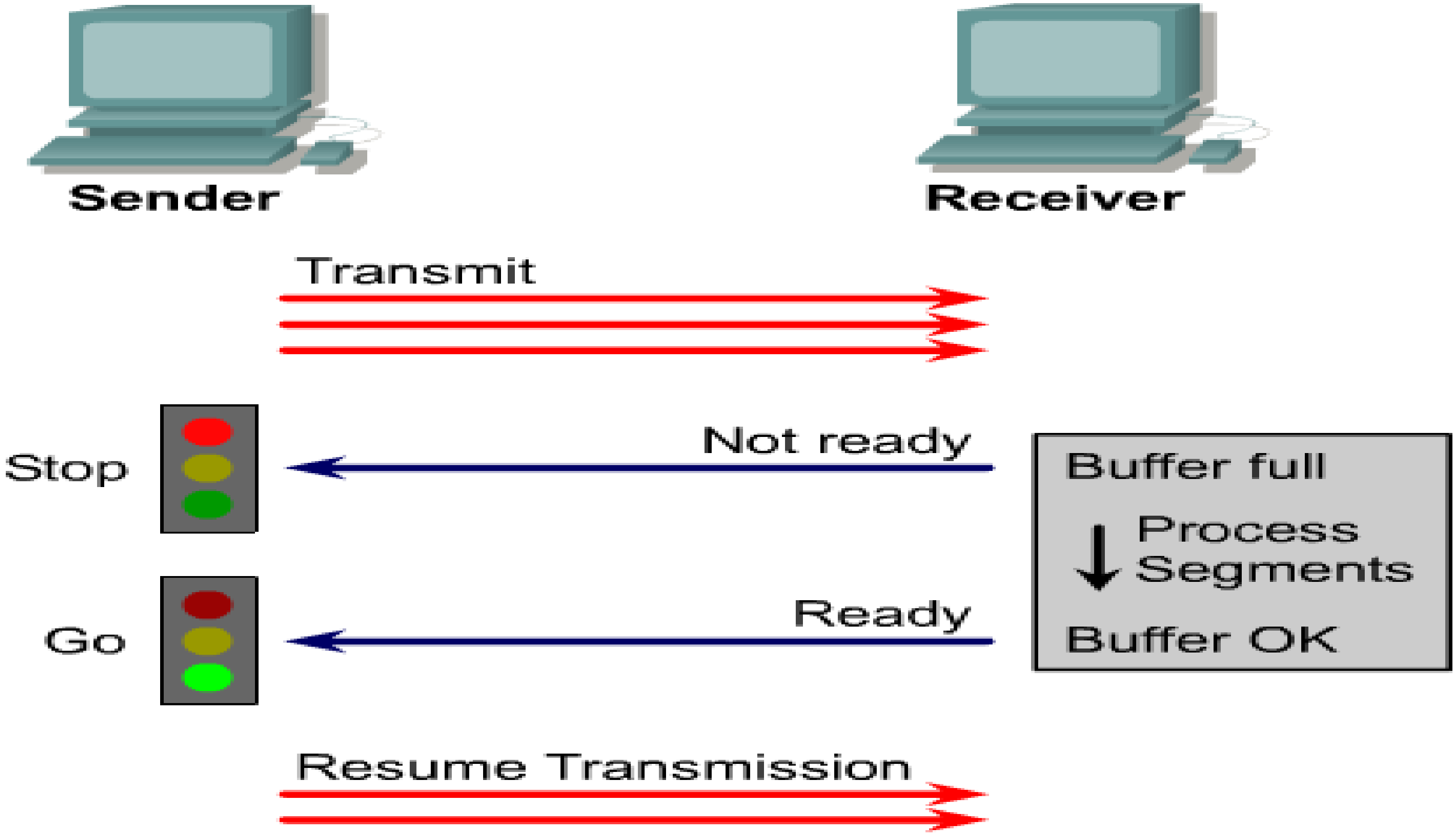
- Flow control prevents a sender on one side of the connection from overflowing the buffers in the receiver—an event that can result in data lost.
- A **buffer** is a temporary area for data storage.
- As the transport layer sends data segments, it tries to ensure that data is not lost.
- A receiver that is unable to process data as quickly as it arrives could be a cause of data loss. The receiving host is then forced to discard it.

Flow control



- Flow control avoids the problem of a transmitting host overflowing the buffers in the receiving host.
- TCP provides the mechanism for flow control by allowing the sending and receiving host to communicate. The two hosts then establish a data-transfer *rate that is agreeable to both*.

Flow Control



Transport Layer Protocols

- TCP: Transfer Control Protocol
  - (Connection Oriented/Reliable)
- UDP: User Datagram Protocol
  - (Connectionless Oriented /unreliable)

TCP (Transfer Control Protocol)

- TCP is responsible for breaking messages into segments, reassembling them at the destination station, resending anything that is not received, and reassembling messages from the segments.
- TCP acknowledges that data is successfully received and guarantees the data is reassembled in the correct order.
- TCP is a connection-oriented protocol that computers use to communicate over the internet.
- It is one of the main protocols in TCP/IP networks. TCP provides error-checking and guarantees delivery of data and that packets will be delivered in the order they were sent.

UDP (User Datagram Protocol)

- UDP is the connectionless transport protocol in the TCP/IP protocol stack.
- UDP is a simple protocol that exchanges datagrams, without acknowledgments or guaranteed delivery.
- UDP doesn't establish connections as TCP does, so UDP does not perform this 3-way handshake and for this reason, it is referred to as an unreliable protocol.
- But that doesn't mean UDP can't transfer data, it just doesn't negotiate how the connection will work, UDP just transmits and hopes for the best.

Network Layer (Layer 3)

- It manages device addressing (IP Addressing)
- Tracks the location of devices on the network and determines the best way to move data, which means that the Network layer must transport traffic between devices that aren’t locally attached.
- Routers (layer 3 devices) are specified at the Network layer and provide the *routing* services within an internetwork.

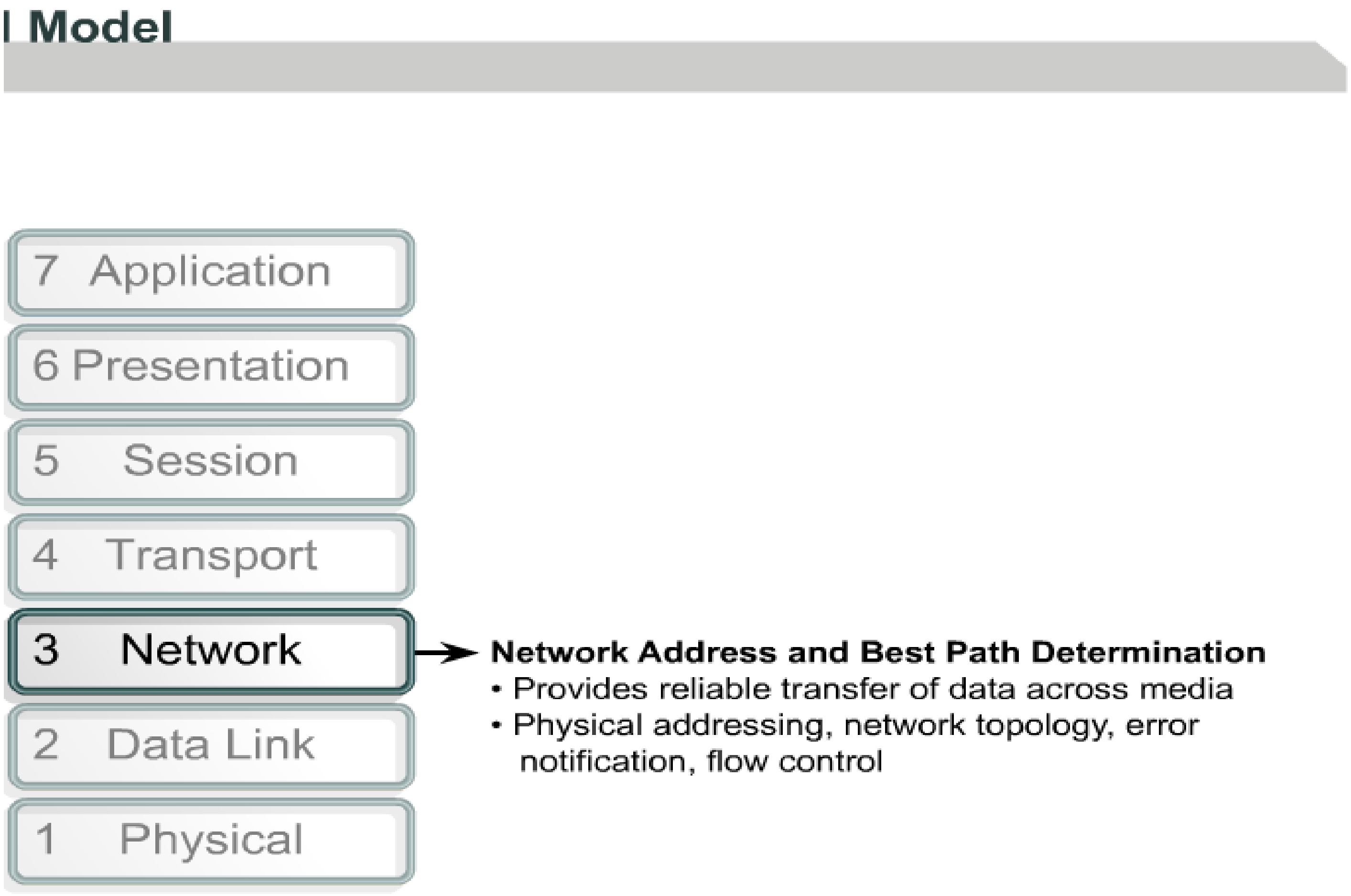
Network Layer

- Determines how data are transferred between network devices
- Routes packets according to unique network device addresses
- Provides flow and congestion control to prevent network resource depletion

Network Layer Protocol

- The network layer accepts and delivers packets for the network.
- Internet Protocol (IP): The Internet protocol suite is the conceptual model and set of communications protocols used in the Internet and similar computer networks

Network Layer



l Model

7 Application

6 Presentation

5 Session

4 Transport

3 Network

2 Data Link

1 Physical

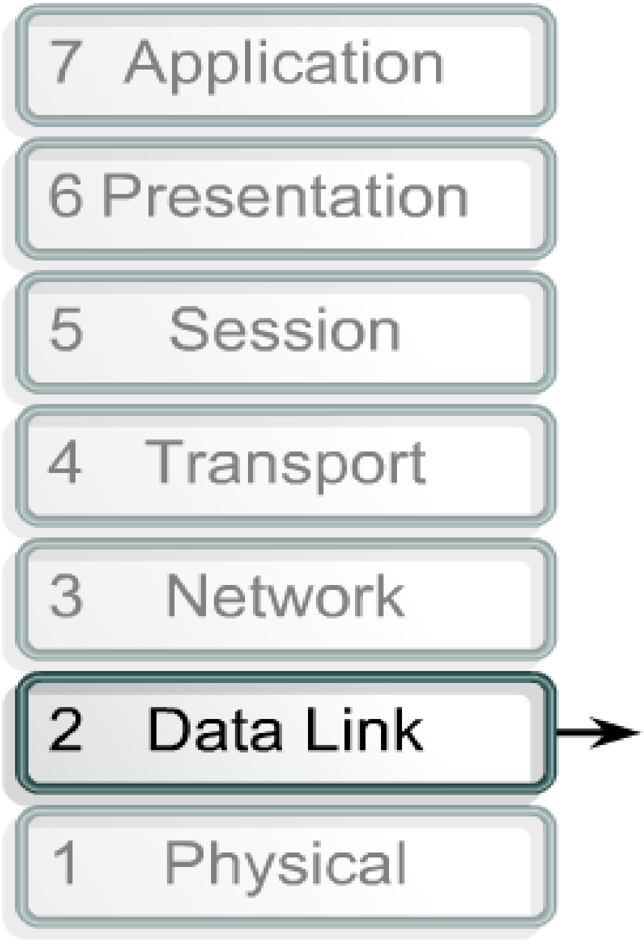
Network Address and Best Path Determination

- Provides reliable transfer of data across media
- Physical addressing, network topology, error notification, flow control

Data link Layer- Layer 2

- The data link layer handles the *moving of data into and out of a physical link in a network*.
- The data link layer defines procedures for operating the communication links
- The protocol data unit (PDU) on the data link layer is a Frames
- It manages physical addressing (MAC: Media Access Control) address
- Physical address (MAC) is a globally unique ID for your device and is burnt in the NIC
- MAC address is useful for local communication

Data Link layer



7 Application

6 Presentation

5 Session

4 Transport

3 Network

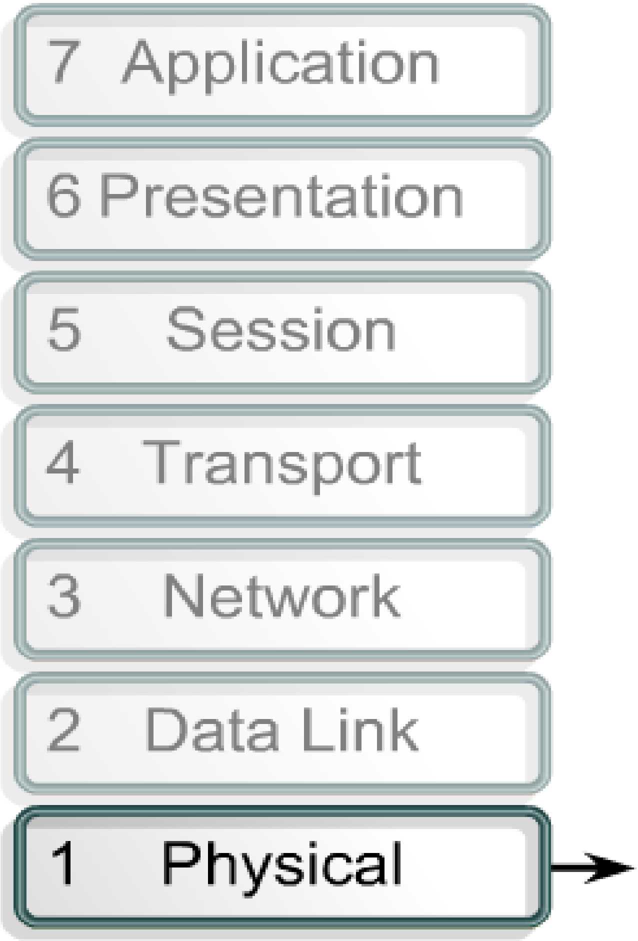
2 Data Link

1 Physical

Direct Link Control, Access to Media

- Provides reliable transfer of data across media
- Physical addressing, network topology, error notification, flow control

Physical Layer



7 Application

6 Presentation

5 Session

4 Transport

3 Network

2 Data Link

1 Physical

Binary Transmission

- Wires, connectors, voltages, data rates

- Defines physical means of sending data over network devices
- Interfaces between network medium and devices
- Defines optical, electrical and mechanical characteristics

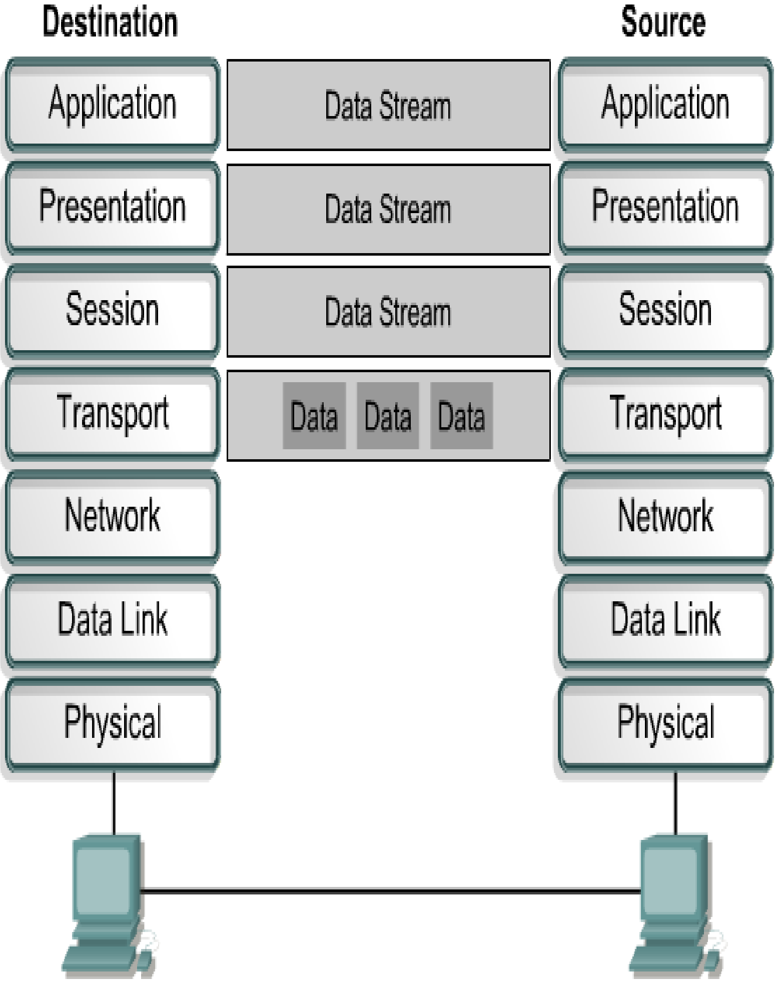
Encapsulation and Decapsulation process

- **Encapsulation** is the process of moving data from the upper layer to the lower layer and each layer includes a packet of information, called a header, with the actual data.
- **Decapsulation:** data moves from the bottom layer to the top layers and removes the packet information (header)

Encapsulation process

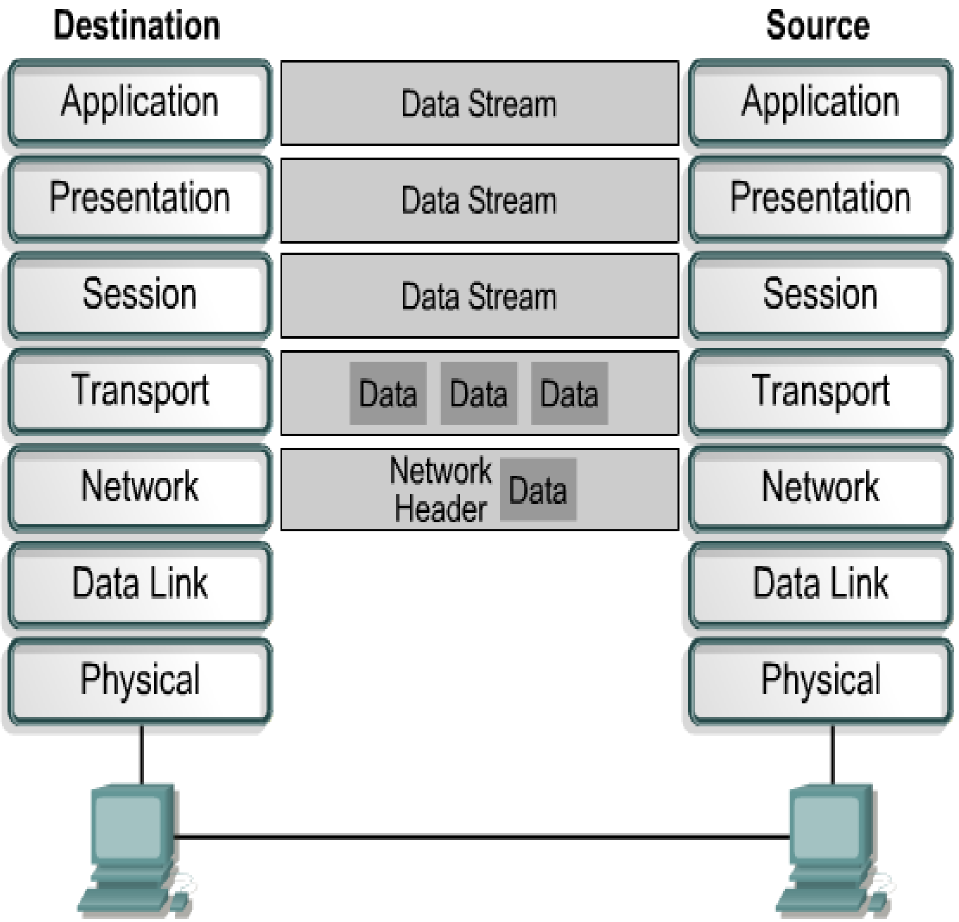
- All communications on a network originate at a source, and are sent to a destination.
- The information sent on a network is referred to as data or data packets.
- If one computer (host A) wants to send data to another computer (host B), the data must first be packaged through a process called encapsulation.
- Encapsulation is the process of taking data from one protocol and translating it into another protocol, so the data can continue across a network
- The protocol data unit (PDU) on each layer is different

Top three layer



- **Build the data.**
  As a user sends an e-mail message, its alphanumeric characters are converted to data that can travel across the internetwork.
- **Package the data for end-to-end transport.**
  The data is packaged for internetwork transport.
- By using segments, the transport function ensures that the message hosts at both ends of the e-mail system can reliably communicate.

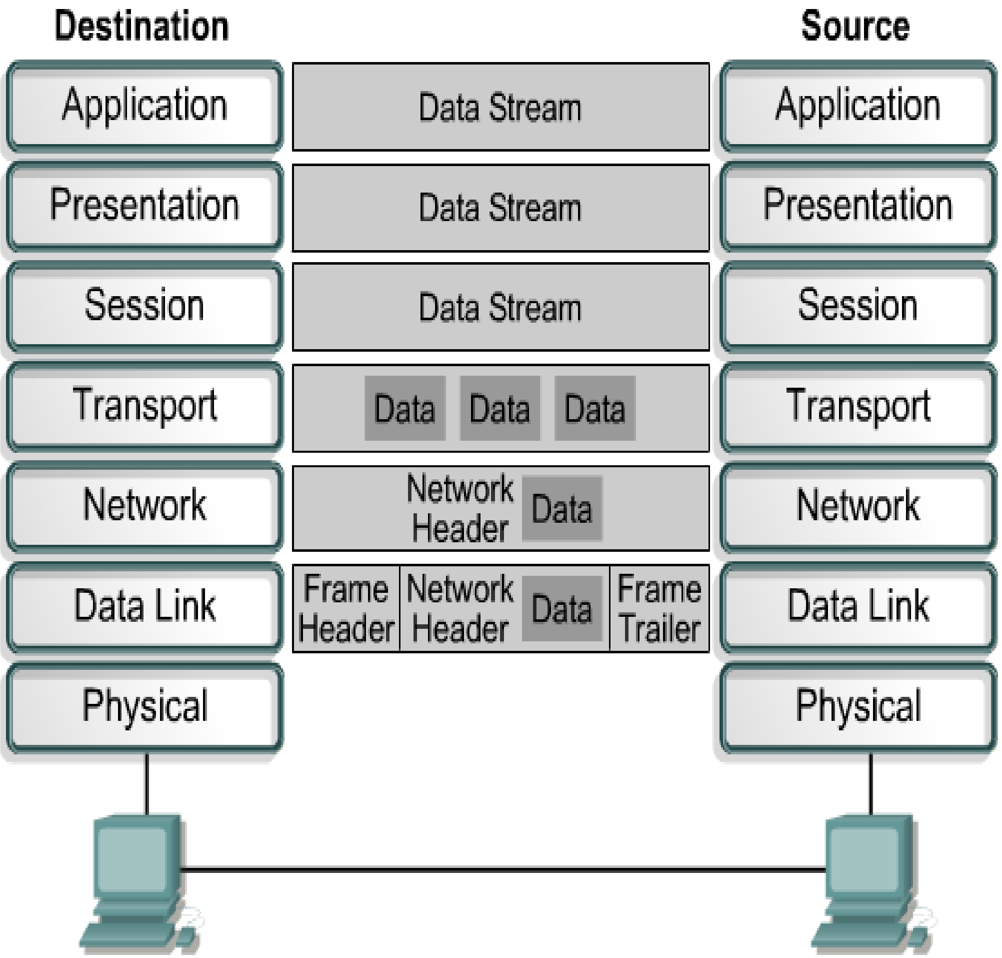
Network Layer



Add the network IP address to the header.

The data is put into a packet or datagram that contains a packet header with source and destination logical addresses. These addresses help network devices send the packets across the network along a chosen path.

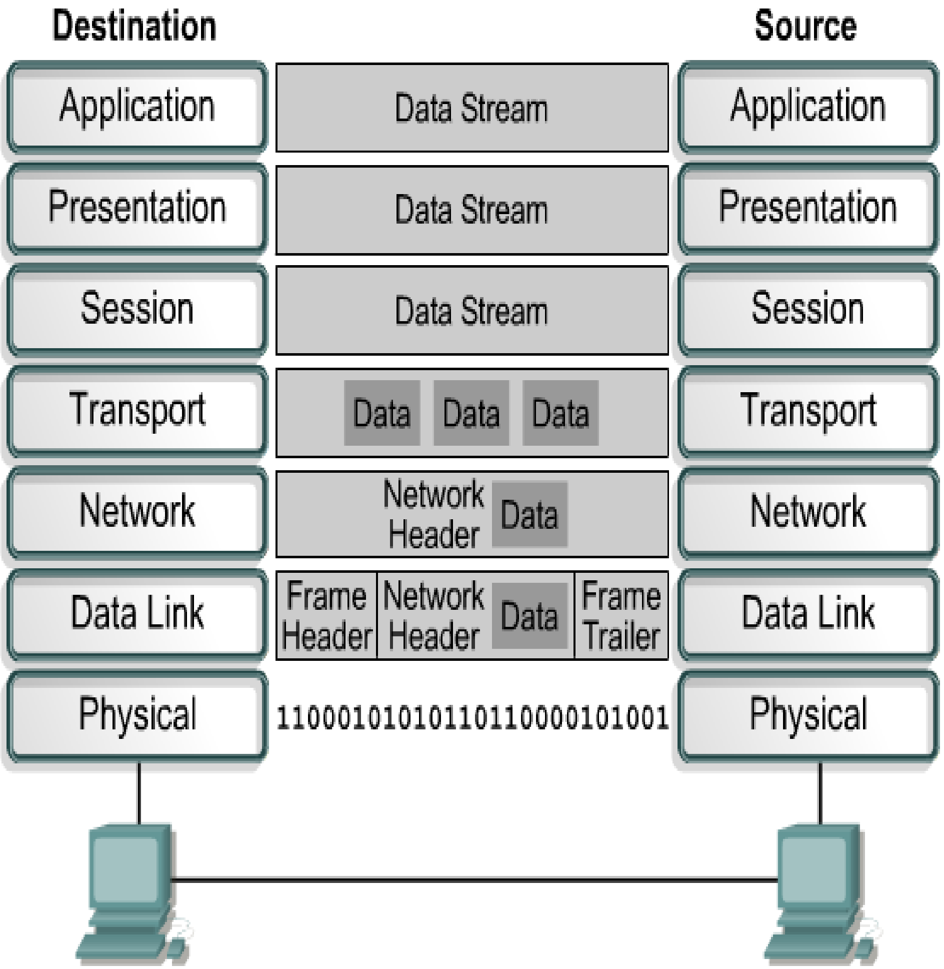
Data Link Layer



Add the data link layer header and trailer.

Each network device must put the packet into a frame. The frame allows connection to the next directly-connected network device on the link. Each device in the chosen network path requires framing in order for it to connect to the next device.

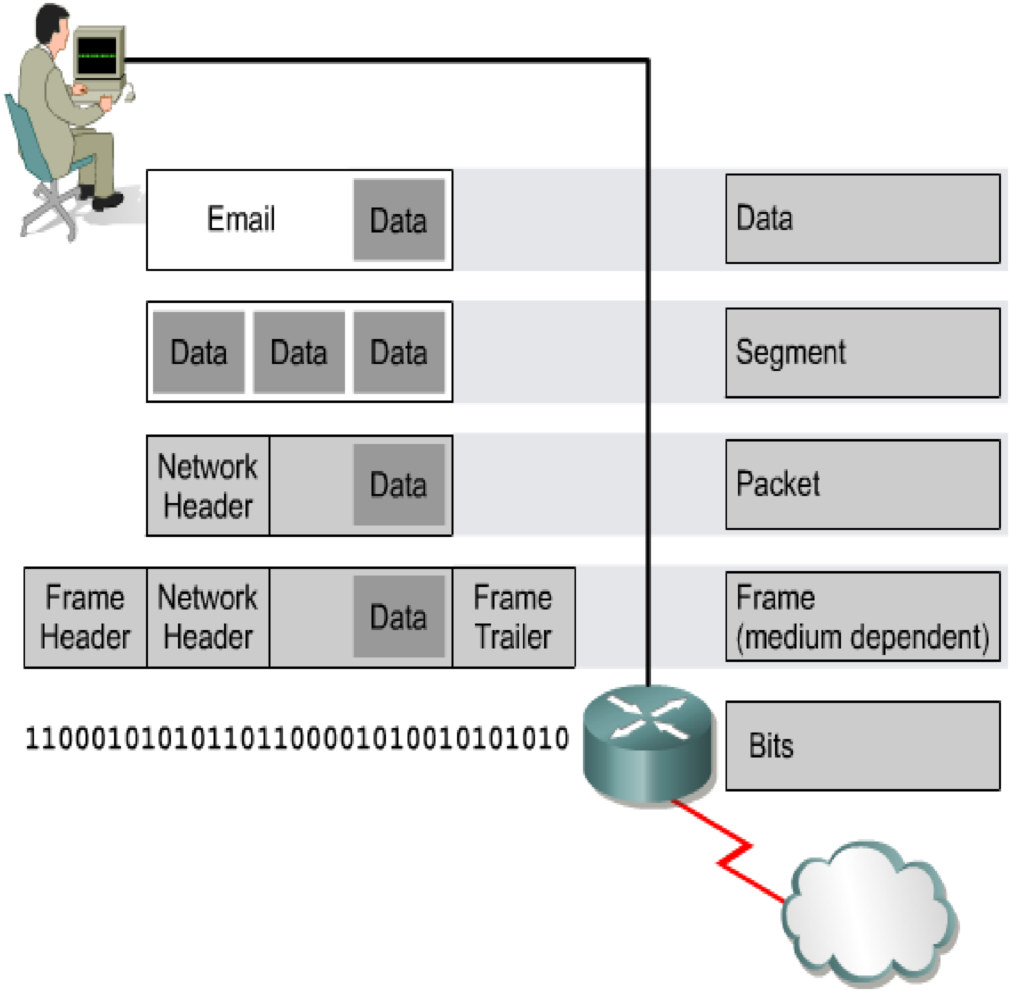
Physical Layer



Convert to bits for transmission.

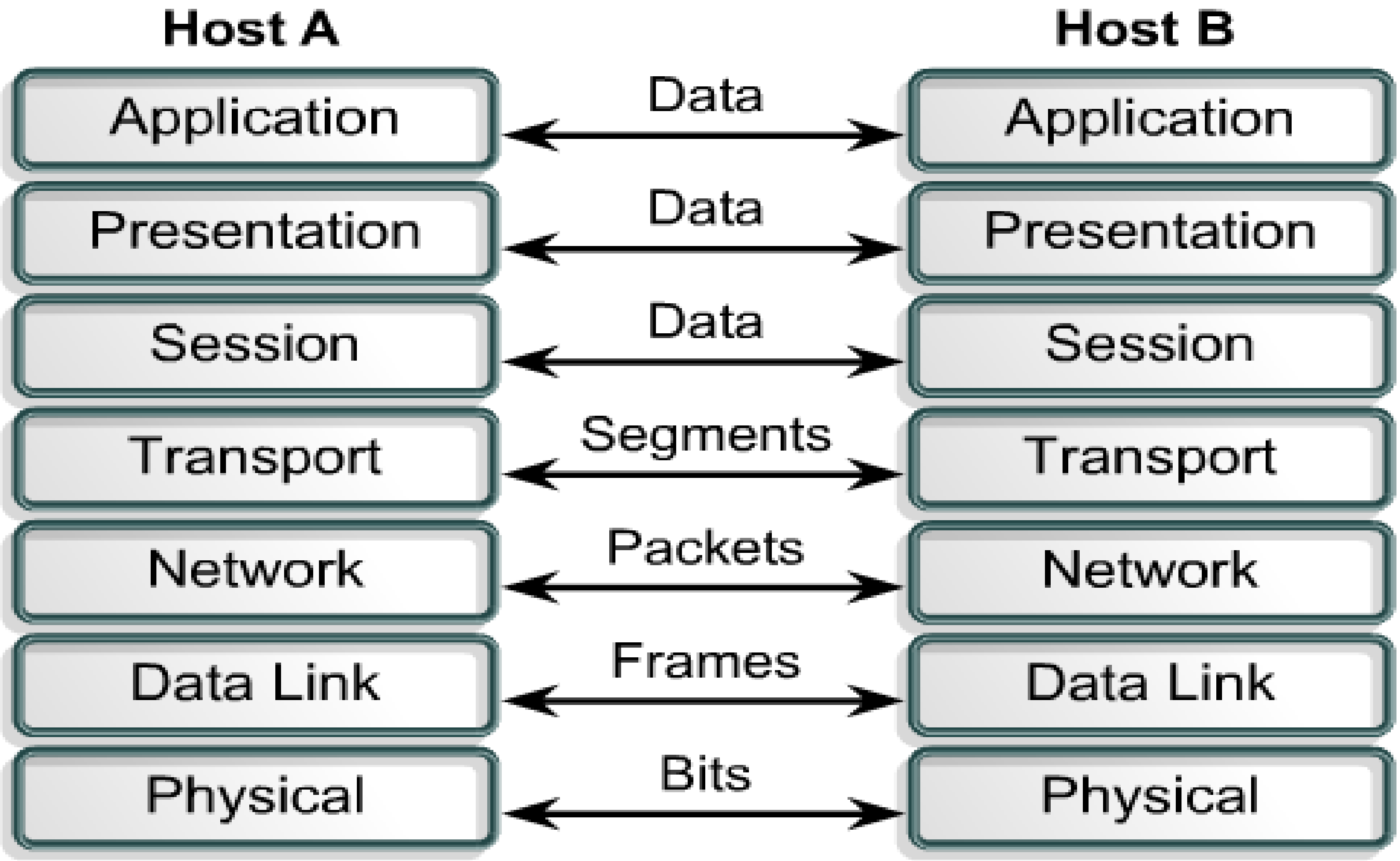
The frame must be converted into a pattern of 1s and 0s (bits) for transmission on the medium. A clocking function enables the devices to distinguish these bits as they travel across the medium. The medium on the physical internetwork can vary along the path used. For example, the e-mail message can originate on a LAN, cross a campus backbone, and go out a WAN link until it reaches its destination on another remote LAN.

Data Encapsulation Example



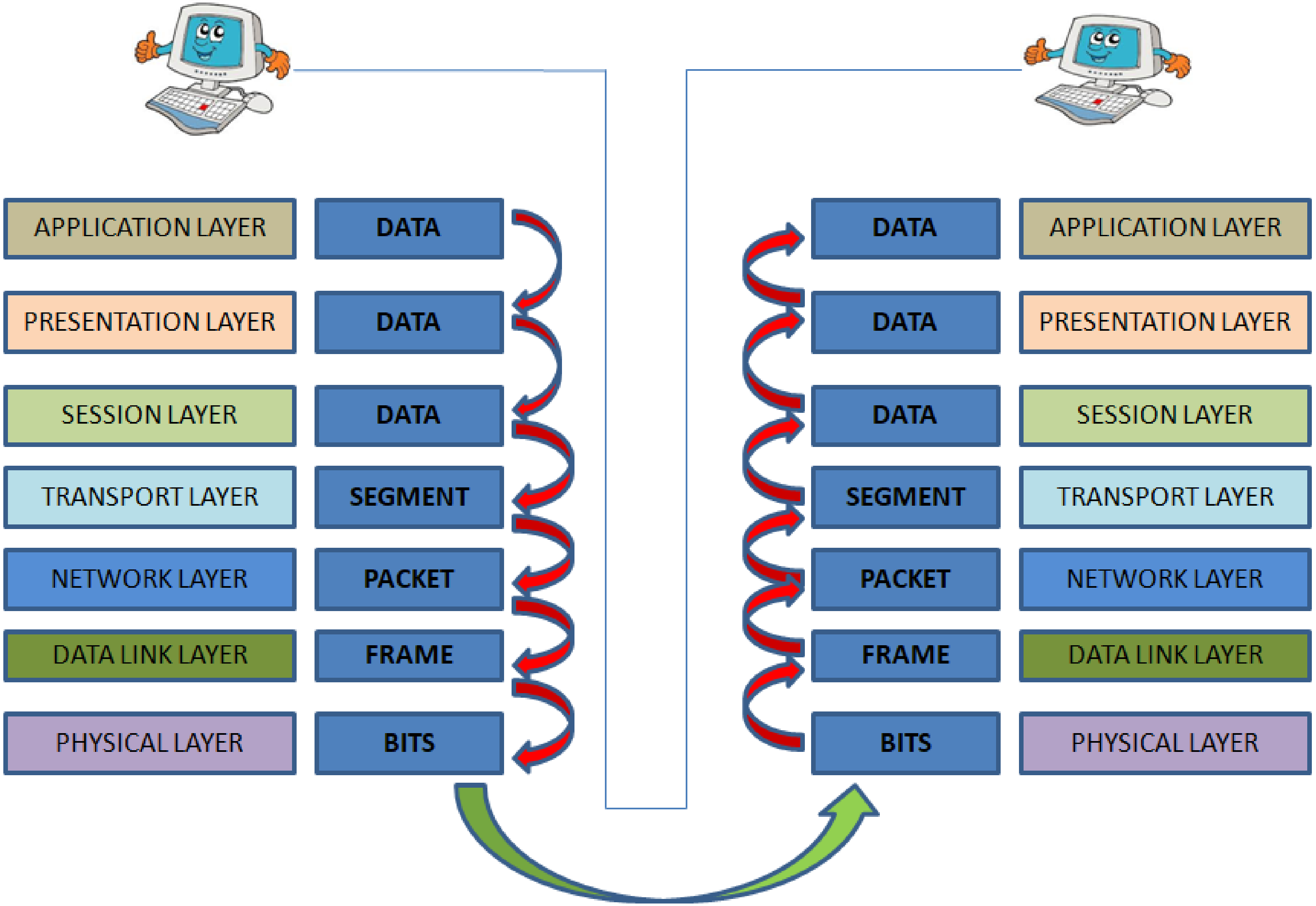
Once the packet has been sent to the destination, the protocols undo the construction of the packet that was done on the source side. This is done in reverse order. The protocols for each layer on the destination return the information to its original form, so the application can properly read the data.

PDU at Each Layer



Host A		Host B
Application	Data	Application
Presentation	Data	Presentation
Session	Data	Session
Transport	Segments	Transport
Network	Packets	Network
Data Link	Frames	Data Link
Physical	Bits	Physical

DATA ENCAPSULATION AND DECAPSULATION



APPLICATION LAYER | DATA | DATA | APPLICATION LAYER

PRESENTATION LAYER | DATA | DATA | PRESENTATION LAYER

SESSION LAYER | DATA | DATA | SESSION LAYER

TRANSPORT LAYER | SEGMENT | SEGMENT | TRANSPORT LAYER

NETWORK LAYER | PACKET | PACKET | NETWORK LAYER

DATA LINK LAYER | FRAME | FRAME | DATA LINK LAYER

PHYSICAL LAYER | BITS | BITS | PHYSICAL LAYER