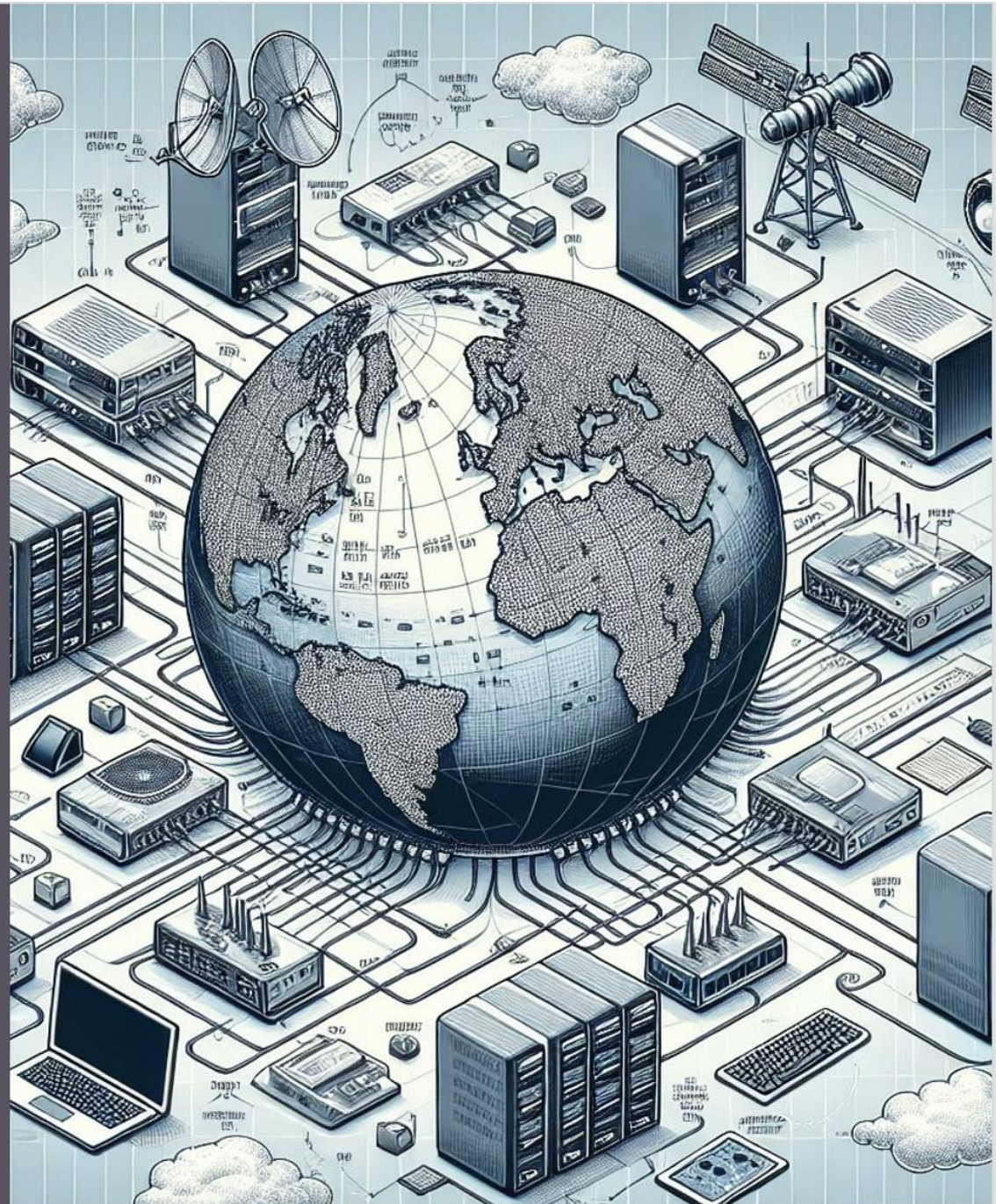
UAB COLLEGE OF ARTS AND SCIENCES
The University of Alabama at Birmingham
Department of Computer Science

CS 334/534 NETWORKING

Dr. Ragib Hasan

Lecture 1.3:
Network Model Overview



Lecture goals

- Introduce OSI and TCP models
- Discuss each layer of OSI model
- Learn about layer Encapsulation

Real-life Analogy

Bob send a letter to Alice

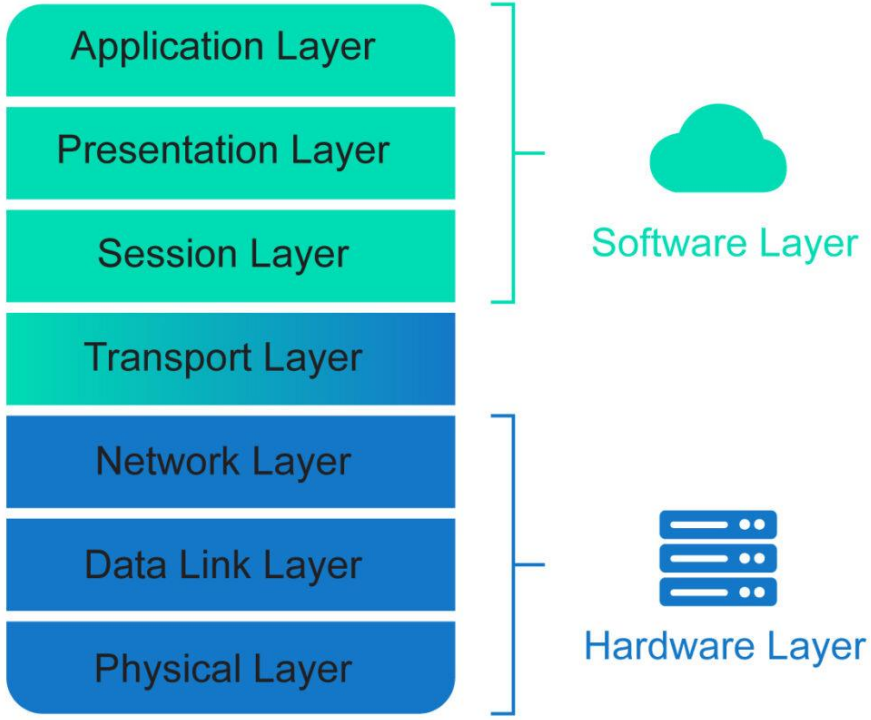
- Bob wrote an invitation letter.
- Format it nicely.
- Confirm Alice address.
- Put it into envelopes.
- Address it.
- Hand it to the post office for routing.
- The postal service delivers it to Alice.

What is Network Model?

- A network model is **a set of guidelines and standards** that defines how data is transmitted and received over a network.
- It is also known as a **networking architecture** or a **networking blueprint**.
- It provides a common framework for network devices and software to communicate with each other.

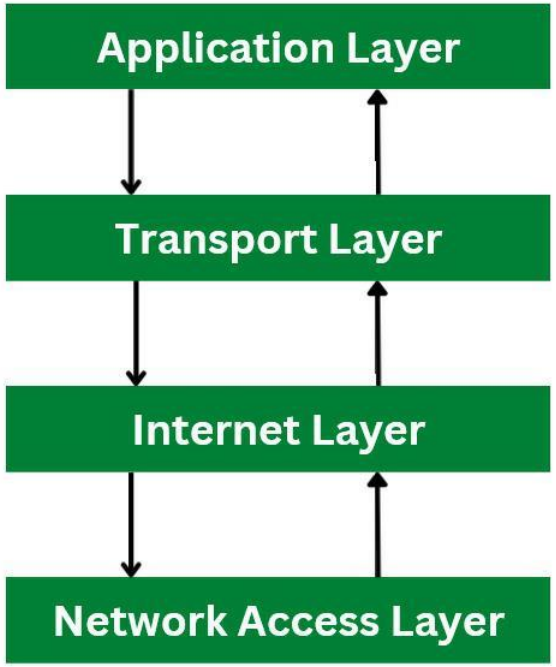
OSI Model

- Open System Interconnection (OSI) Model.
- A conceptual framework in the for establishing connections between the systems.
- The computation in the OSI Model is done through seven different layers.
- It ensure that different technologies can work together **seamlessly** while **simplifying the complexities** of communication systems.

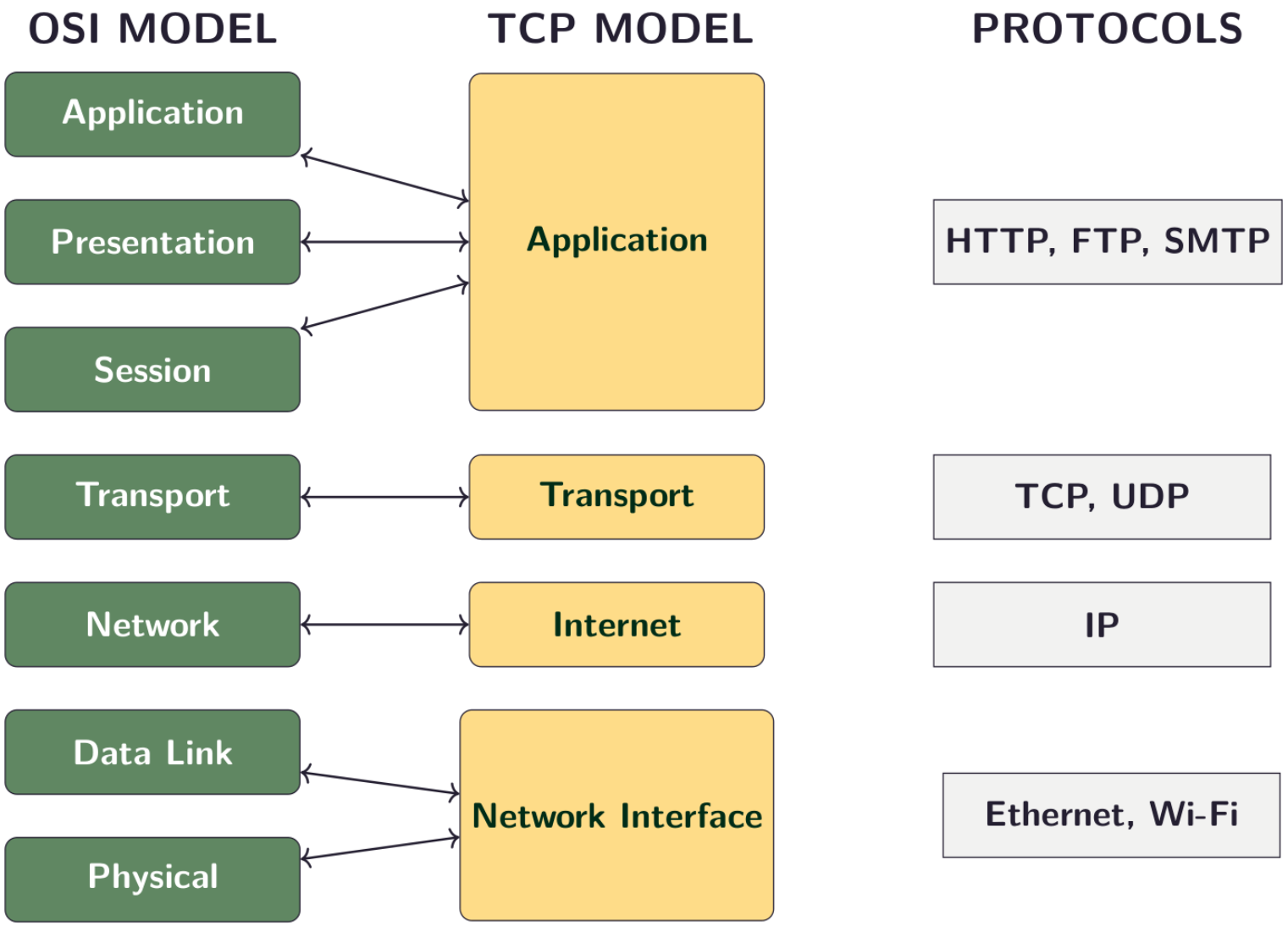


TCP Model

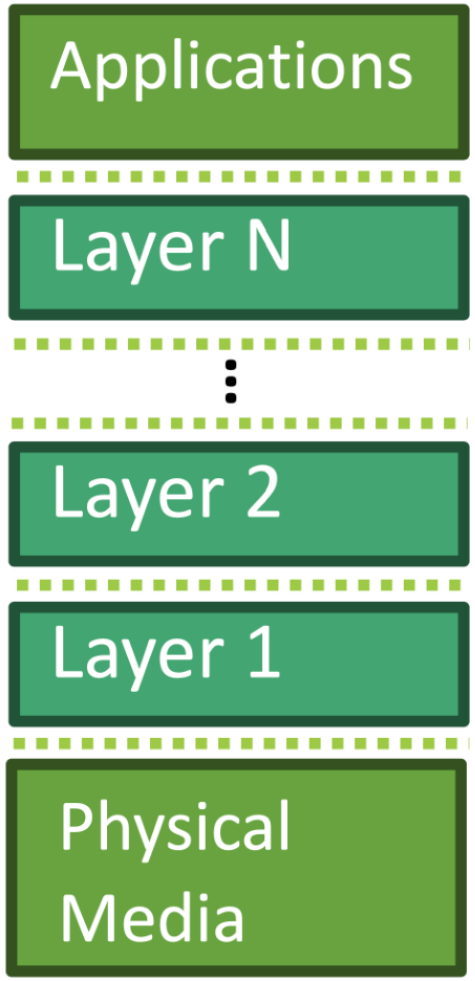
- Transmission Control Protocol/Internet Protocol (TCP/IP).
- A conceptual framework to describe how **data is transmitted** over the internet or any other network.
- It consists of **four layers**, which correspond to the functions of the OSI model's seven layers but is more simple.
- Widely used in modern networking and serves as the foundation of the internet.



Network Models

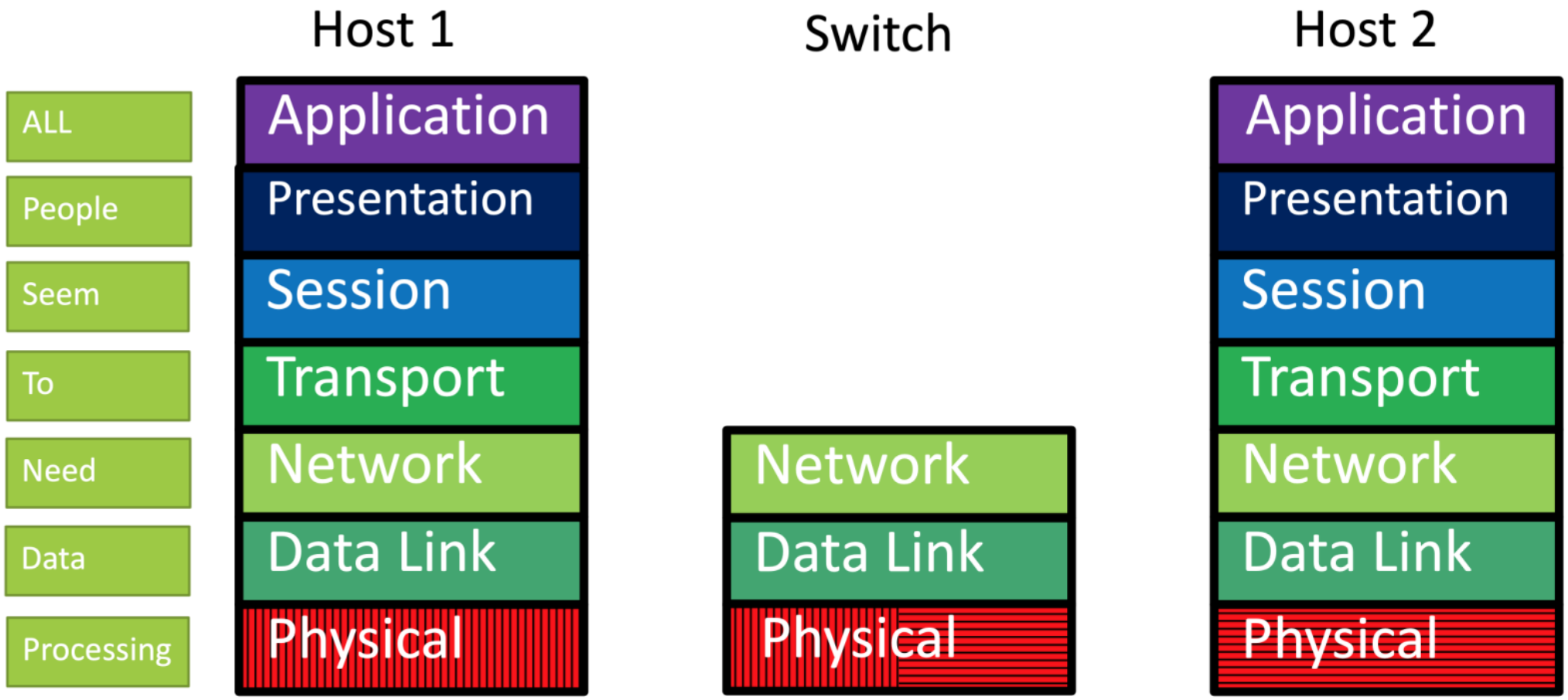


Layered Network Stack

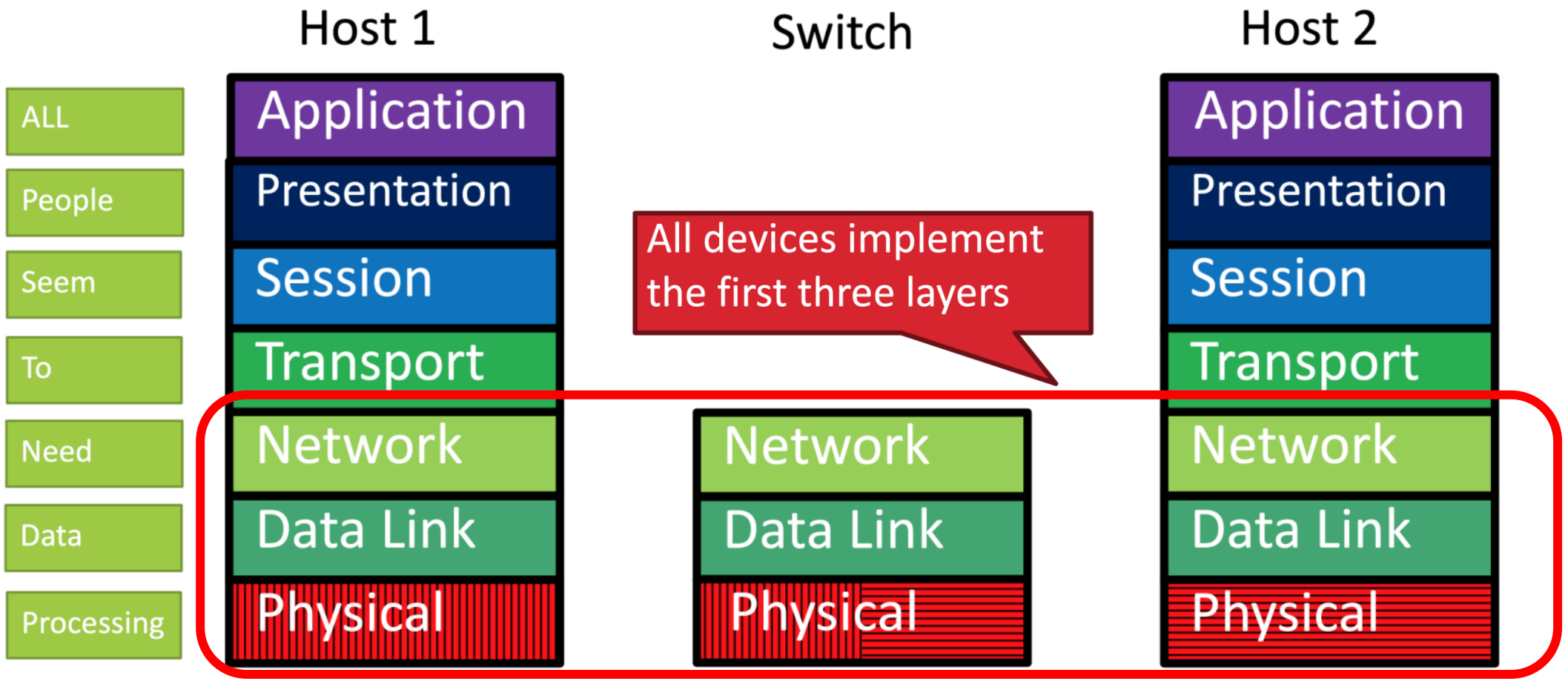


- **Modularity**
  - Instead of specifying how things should be implemented, the layered model focuses on **organizing functionality** into independent modules.
- **Encapsulation**
  - Each layer **adds its own information** (e.g., headers) to the data as it passes down through the layers
- **Flexibility**
  - Reuse the code across the network. Modules within a layer can be **replaced or improved** without requiring changes to other layers.
- **Tradeoffs**
  - Layers **cannot access details** from other layers. Each layer **adds small delay**, which might hurt performance.

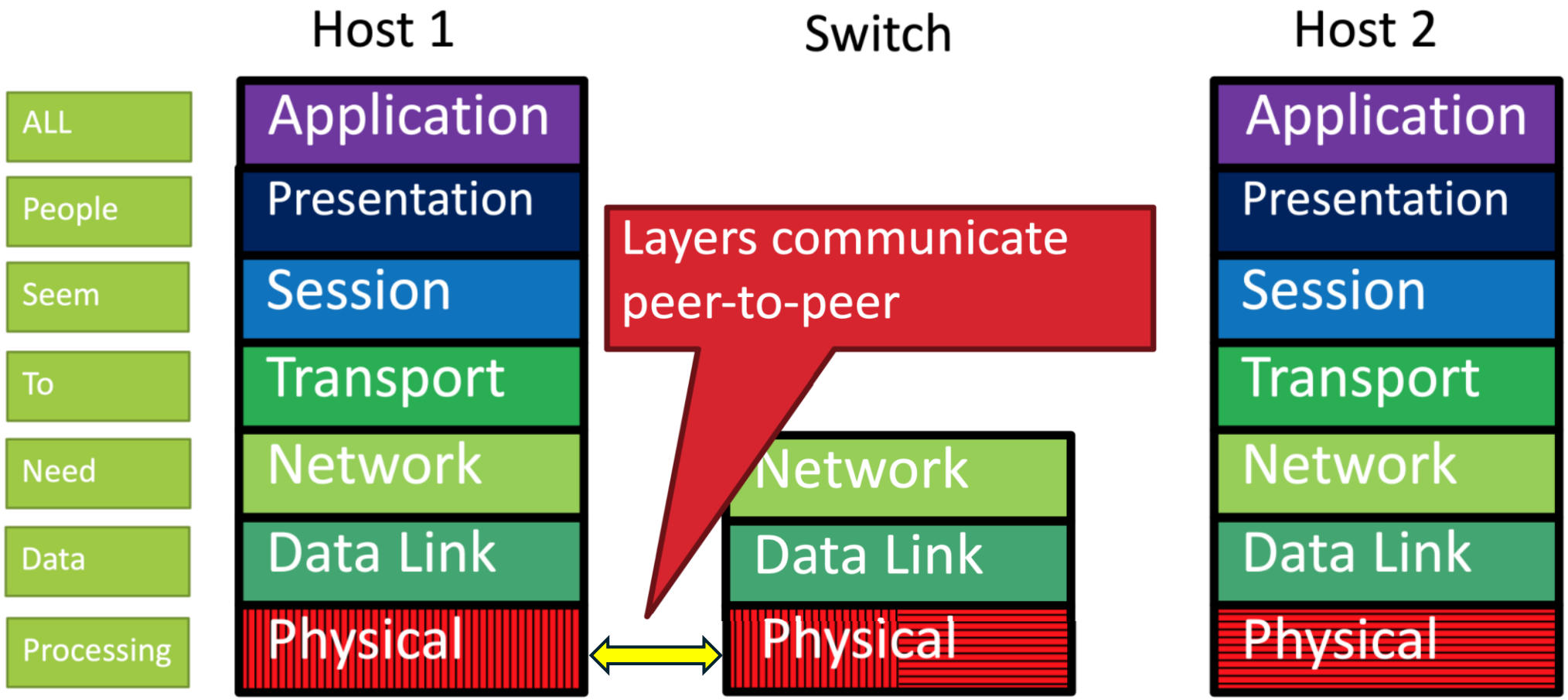
The OSI Model



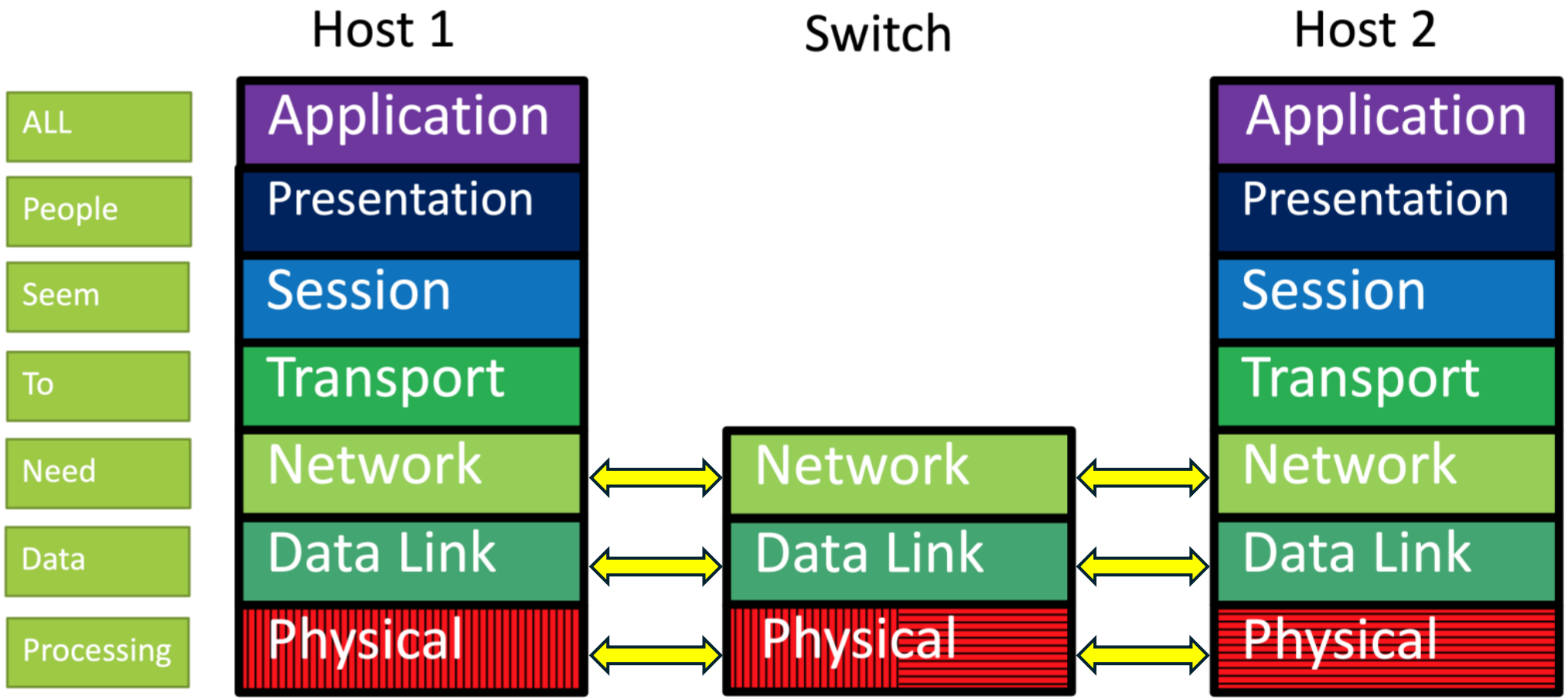
The OSI Model



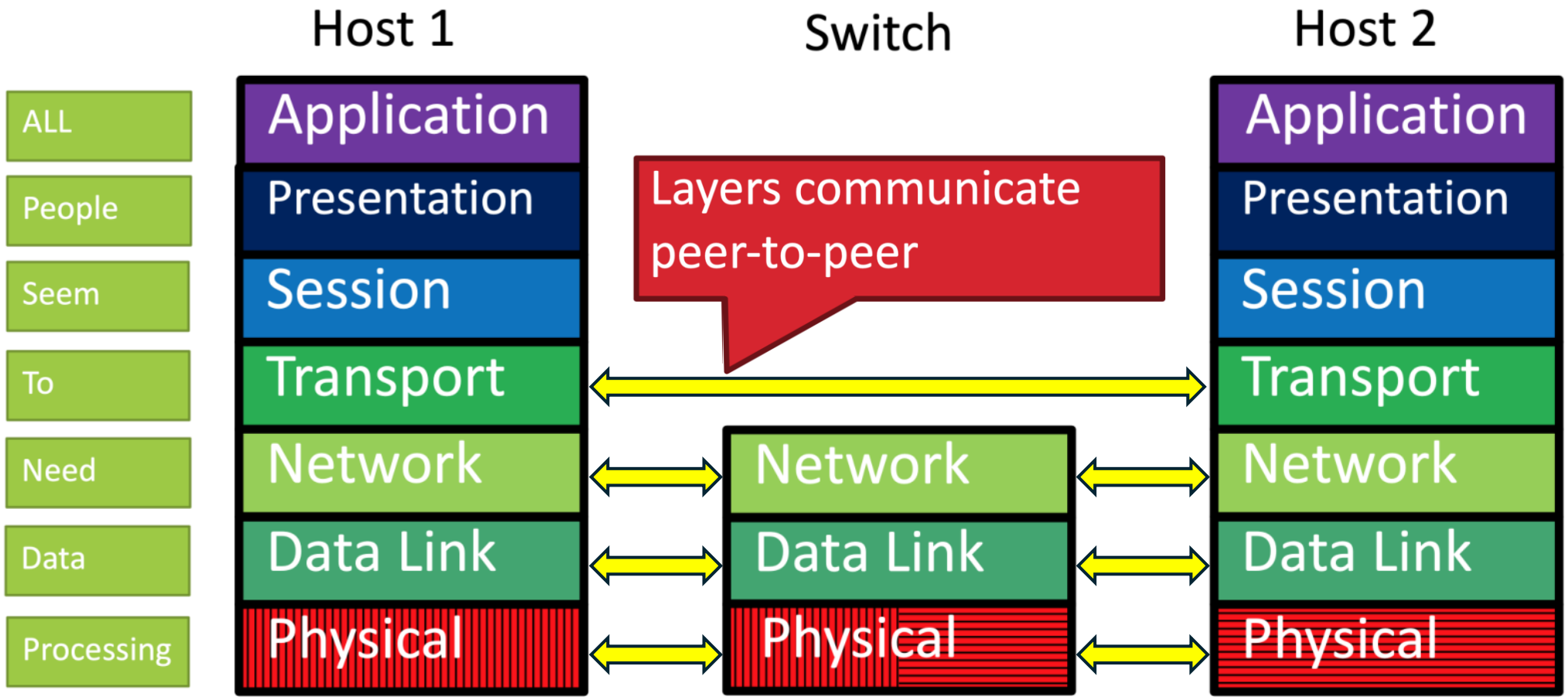
The OSI Model



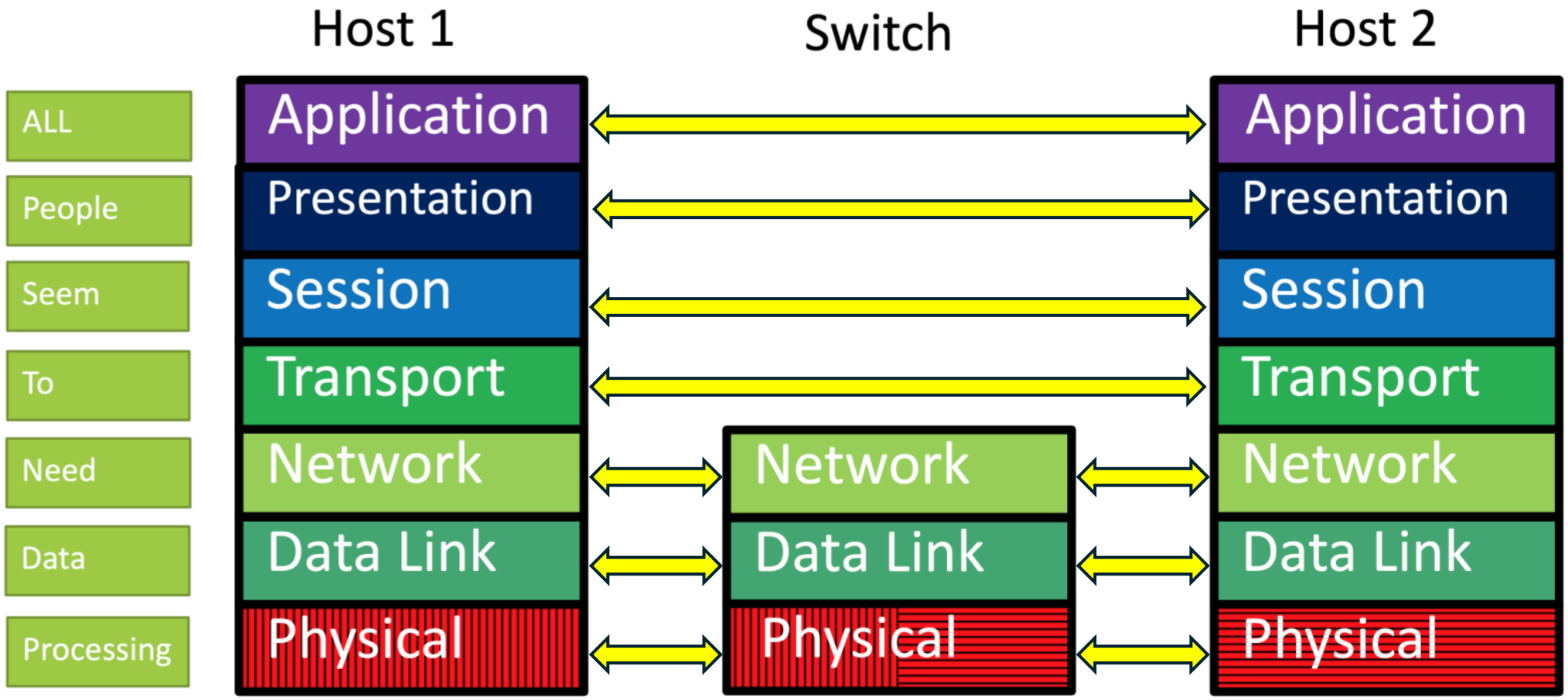
The OSI Model



The OSI Model



The OSI Model



Thoughts?

Service:

– What does this layer do?

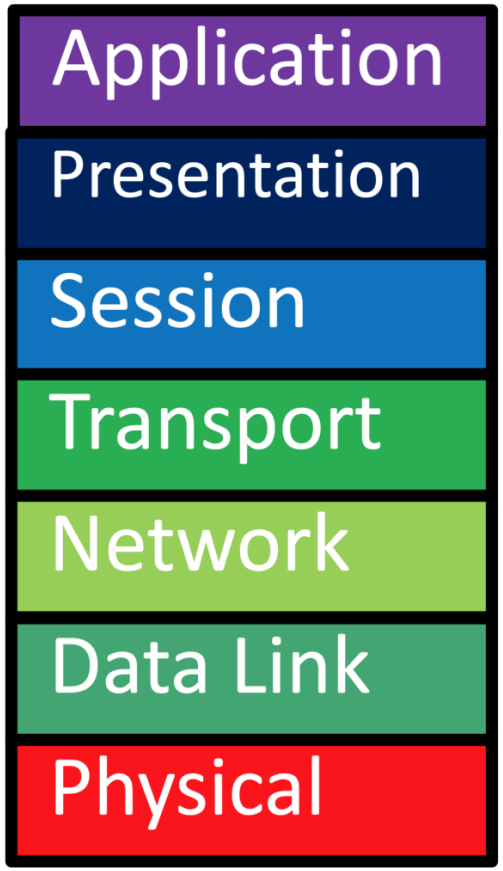
Interface:

– How do we access the layer?

Protocols:

– How is this layer implemented?

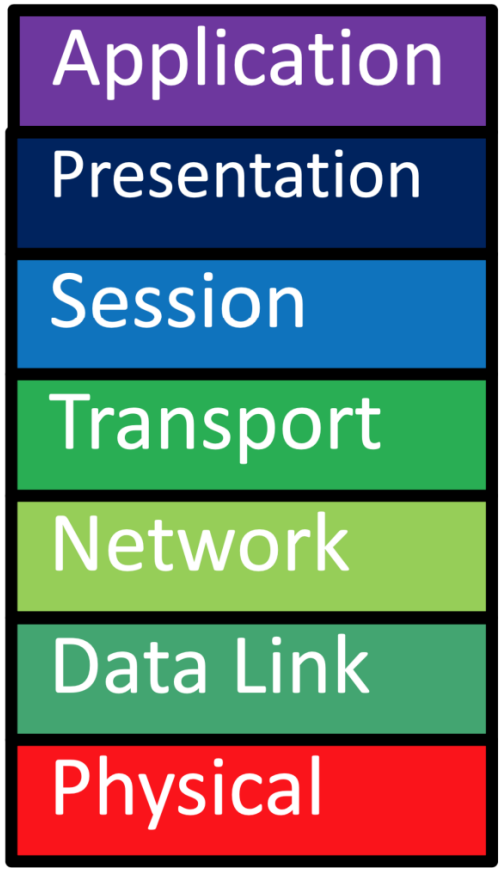
Physical Layer



- **Service**
  - Move information between two systems connected by a “Physical link”
- **Interface**
  - Specifies how to send one bit.
  - A USB interface specifies how a bit is sent over a USB cable.
- **Protocols**
  - Encoding scheme for one bit, voltage level, Timing of Signals

Example: Coaxial cable, Fiber optics, radio frequency transmitters.

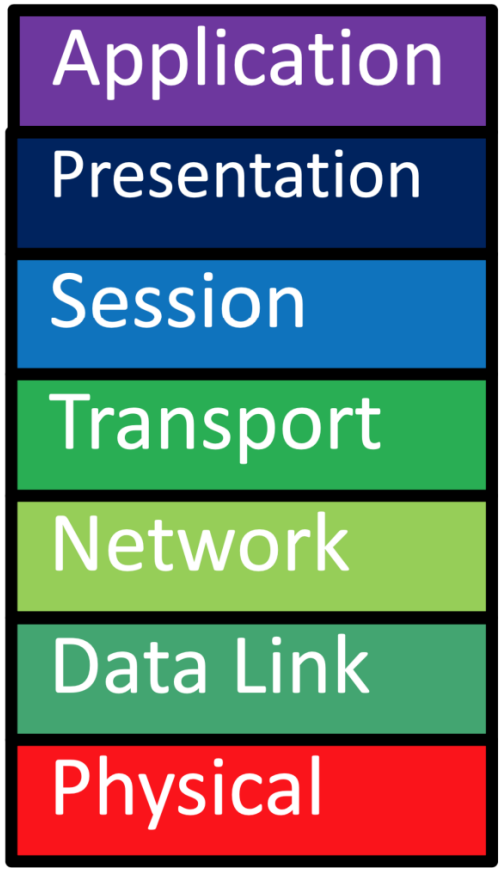
Data Link Layer



- **Service**
  - Data framing: boundaries between packets.
  - Media access control (MAC).
  - Per-hop reliability and flow control.
- **Interface**
  - Sends one frame between two hosts connected to the same media.
- **Protocols**
  - Physical addressing (MAC address)

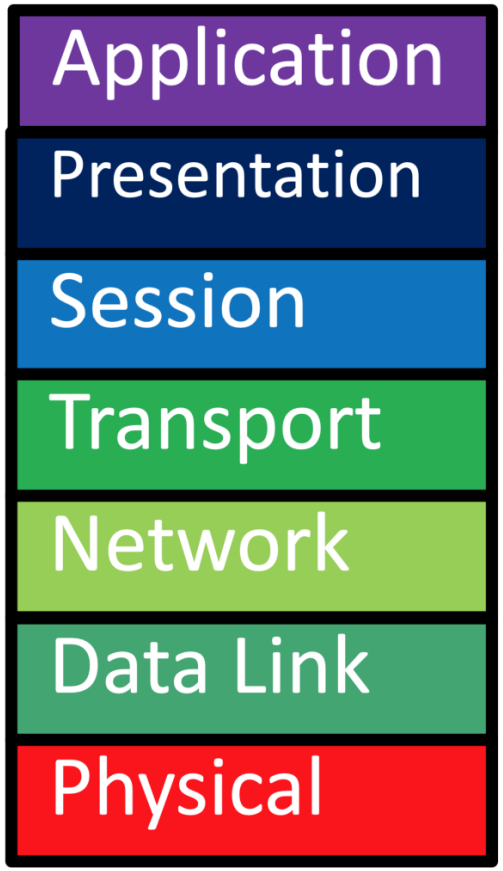
Example: Ethernet, WiFi, etc.

Network Layer



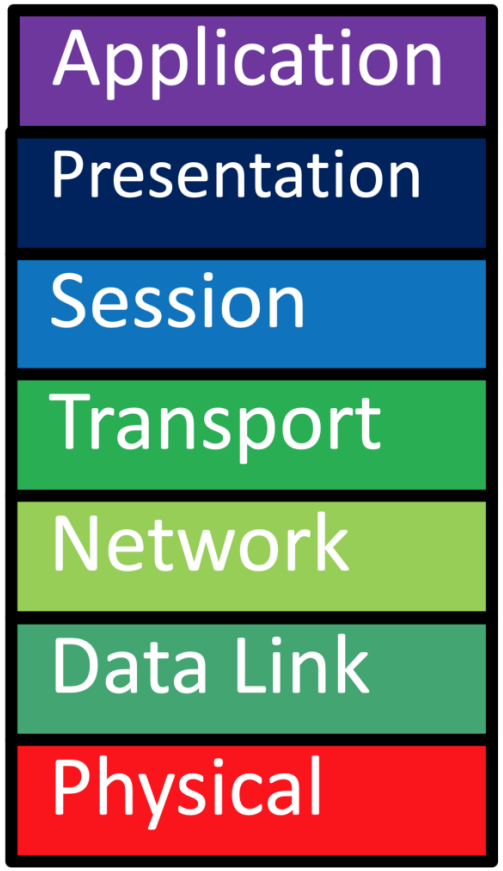
- **Service**
  - Deliver packets across the network
  - Handle fragmentation/reassembly
  - Packet scheduling
  - Buffer management
- **Interface**
  - Sends one packet to specific destination
- **Protocols**
  - Define globally unique addresses
  - Maintain routing tables
  - IPv4, IPv6

Transport Layer



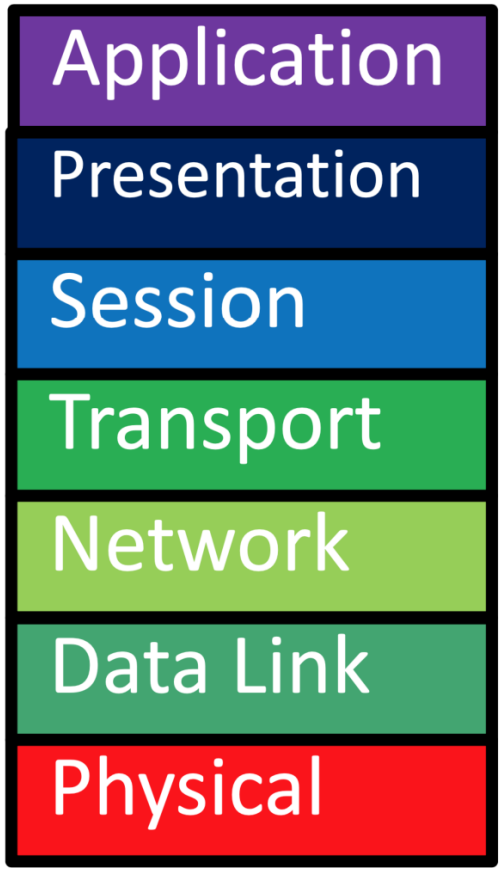
- **Service**
  - Multiplexing/Demultiplexing
  - Congestion control
  - Reliable, in-order delivery
- **Interface**
  - Sends a **message segments** to destination
- **Protocols**
  - Port numbers
  - Reliability/error correction
  - Flow control information
  - TCP, UDP

Session Layer



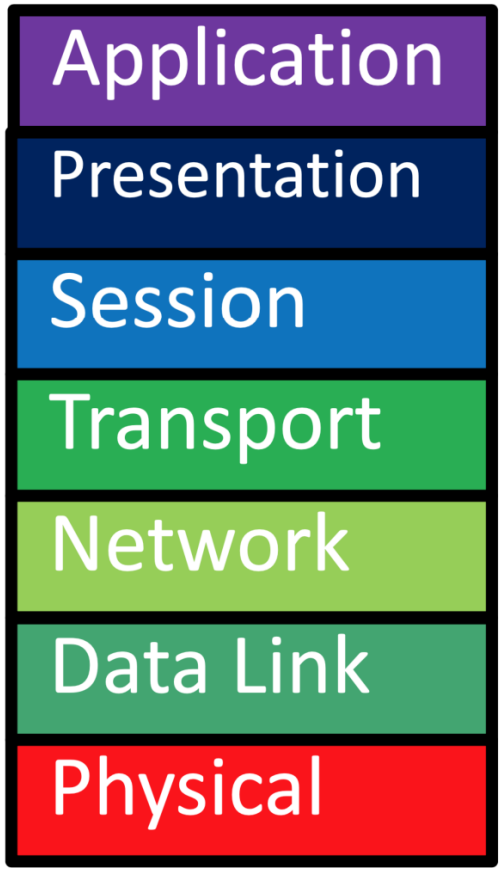
- **Service**
  - Access management
  - Synchronization
- **Interface**
  - It depends……
- **Protocols**
  - Token management
  - Insert checkpoints

Presentation Layer



- **Service**
  - Convert data between different representations
  - E.g., ASCII to UNICODE
- **Interface**
  - It depends……
- **Protocols**
  - Define data formats
  - Apply transformation rules

Application Layer



- **Service**
  - Whatever we want!
- **Interface**
  - Whatever we want!
- **Protocols**
  - Whatever we want!

Example: When we use our smartphone, we are interacting with the application layer through apps that communicate with lower layers of the OSI model.

Encapsulation