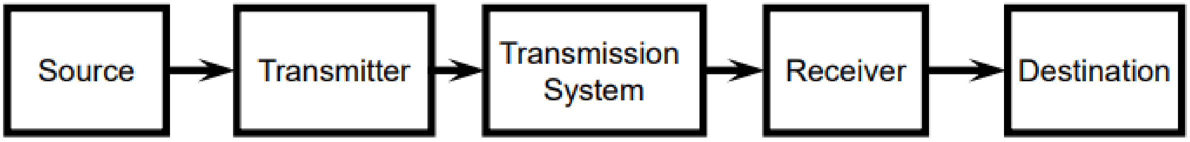
02. Data Communications. Protocol Model

Protocol Hierarchies

Computer networks are generally comprised of numerous pieces of hardware and software.

The Five-component Communications Model is a previous attempt to model how these components interconnect and interact with each other.



Five-component Communications Model

Whilst this horizontal model is useful when looking at hardware components, the reality is it does not explain everything.

From previous discussions on **Data Link Control**, **LANs technologies**, **Internetworking**, etc much of the functionality required to provide communications between host machines is not necessarily provided on individual hardware components.

For example: the transmission / reception of MAC frames requires sending and receiving binary 1s and 0s on a variety of transmission media using a variety of access techniques.

This mixture of functionality is provided on **NICs** (one per LAN technology) and in software elsewhere on the host.

A better model is required to explain these concepts and techniques.

To simplify network design most networking technologies are organized as **layers** of protocol software. The purpose of each layer is to provide one component of the overall solution to the Communications Problem such as frame reception / transmission etc.

These layers of software can exist on separate hardware components and / or hosts or is some instances multiple layers can exist on a single hardware platform and / or hosts.

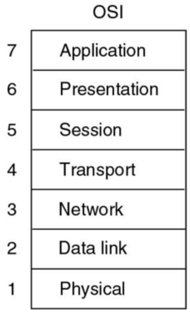
This approach is used for network protocol software. Each layer of software provides one part of the solution such as frame transmission / reception or transmitting one / zeros onto a wire / wireless transmission medium, etc.

With networking software, the layers of protocol software are organised into a **Stack**.

Each layer is said to offer **services** to higher layers and to use the services offered by the lower layers.

Each layer has a specific function

- Physical: reads 1s and 0s and stores them
- Data Link: interprets 1s and 0s previously read



	OSI
7	Application
6	Presentation
5	Session
4	Transport
3	Network
2	Data link
1	Physical

Terminology

- **Protocol:** set of rules that apply to software. Each protocol is for a particular application (different frames). They exist within the data link layer.

- **Protocol / Protocol Entity:** software module that performs one sub-task such as frame transmission / reception, etc.
- **Protocol suites / stacks:** a number of such software modules that work together to perform all of the tasks required to enable two hosts communicate.
- **Protocol Architectures / Model:** model by which we can categorize or view the different protocol layers.

- **Transport Layer Protocol:** TCP
- **Network Layer Protocol:** IP
- **Data Link Layer Protocol:** Ethernet (802.3)
- The application is where the data comes from

IP (Internet Package): IP allows communication between two hosts that are not physically connected to the same local area network (LAN). It is a layer of protocol network and uses routers (connectivity) and IP datagrams (structure).

IP addresses are able to connect local area networks, which exist in the data link layer.

The framing structure are the IP datagrams.

IP: connectionless protocol, there is no guaranty that several packets will travel the same path towards the destination.

MAC Addresses: to communicate between hosts in the same LAN network.

3 key components for communication protocols:

- Connectivity - being able to connect physically
- Unique framing structure
- Unique address

Router: main goal is to get rid of package as soon as possible, make routing decisions. Routing decisions are based on routing map and IP destination addresses.

IP Encapsulation: to move packages they must be put inside a frame associated with the network between the routers.

An **IP address** is 32 bits long. **IP addresses** do not change in packages (source and destination IP addresses are stored). Only **MAC addresses** are updated in the packages.

(EXAMPLE ROUTING iPad)

The framing structure of the **Transport** Layer are **segments**.

The Data field of a package (TCP field) starts with a TCP Header.

- MAC addresses belong on a frame
- IP addresses belong on a packet
- Ports belong on a segment

The OSI ISO Reference (7-Layer) Model

The **OSI** model deals with connecting **open** systems, systems that are open for communication with other systems regardless of who built them or how they were manufactured.

The principles that were applied to arrive at the seven layers are as follows:

- Each layer was created when a different **level of abstraction** was required.
- Each layer performs one well-defined function with each function chosen carefully to facilitate the development of **standardized protocols**.
- The number of layers chosen was sufficient enough to ensure that distinct functions were not lumped together to become unwieldy.

The layers of software work together in unison to provide overall solutions to the problems of communicating between host computers across complex networks.

Information flows up-down the stack, across each of the interfaces between the layers:

- These boundaries were carefully defined to minimize information flow across the interfaces.

Layer by Layer

- The **Physical** Layer: concerned with transmitting raw bits over a communication channel.
    - Its purpose it to ensure that the transmission of binary 1s and 0s adheres to what is appropriate for the transmission medium and what is expected by the receiver.

    - Primarily it deals with matters such as: What voltage levels are used? What frequencies are used? How long does it take to transmit a bit (bit duration)?, ...
    - There are certain issues to address such as the mechanical and electrical design of the plugs (RJ-45, BNC, ...) and sockets, timing interfaces, the physical transmission medium, ..

- The **Data Link** Layer: concerned with the successful transmission of data across an individual link.
    - Its purpose is to transform a raw transmission facility into a data communications channel that appears free of transmission errors.
    - Primarily it deals with matters such as the transmission / reception of data frames.
    - There are certain design issues to address such as the creation of localised unique addressing, the creation of a unique framing structure, flow control, error control, controlling access to a shared channel, ...

- The **Network** Layer: concerned with the successful delivery of data between hosts across complex network infrastructures such as a large internet.
    - Its purpose is to control the operation of the sub-networks to achieve host-to-host delivery.
    - Primarily it deals with matters such as the routing of packets from the source station towards the destination station across sub-nets, ...
    - There are certain design issues to address such as the creation of a globally unique address space, the creation of a globally unique framing structure.
    - Essentially this layer is responsible for managing data communications across interconnected heterogeneous networks.

- The **Transport** Layer: key layer, it is a true end-to-end layer responsible for the reliable delivery of data between processes on end-host machines (process-to-process delivery).

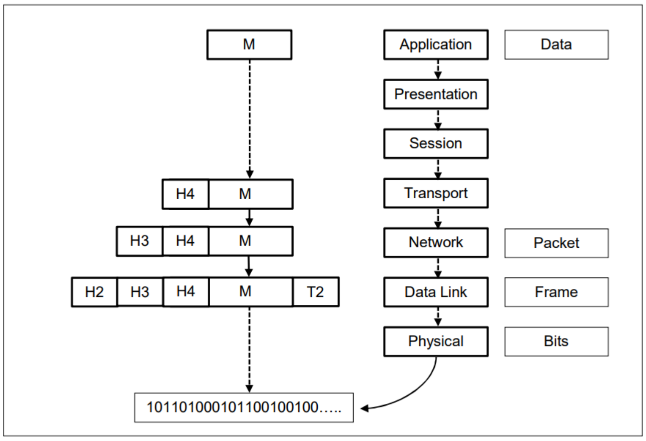
- Its purpose is to provide a reliable data transport service to applications.
- Primarily it deals with interfacing with applications for the purpose of exchanging data between applications across a network.
- There are certain design issues to address such as multiplexing data streams from / to remote applications, data loss, network latency, ...

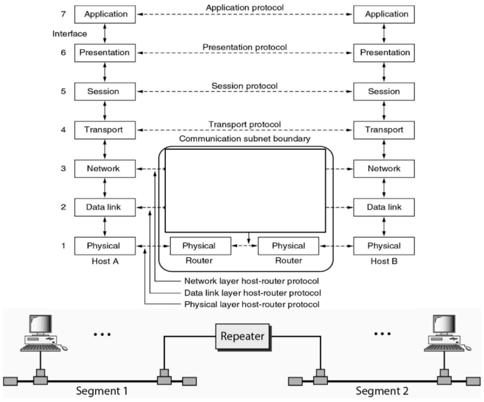
Data Flows

The layers work together to provide the complete functionality required to facilitate communications between two remotely connected host machines regardless of how they connect to the internet.

Note: sometimes it is necessary to break up the data to meet some size restrictions associated with some lower layer protocol software. (Recall sue of **Fragmentation** at the Network Layer).

Some of the layers are also implemented on intermediary networking devices such as Routers. This is because the functionality associated with these layers is needed for a particular purpose such as frame reception / transmission, packet routing, ...





How the layers are implemented on the end-hosts and the intermediary devices