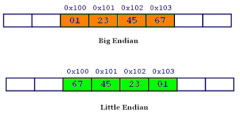
04. Five Layer Protocol Model

The ISO Reference Model - Layer by layer

- The **Application** Layer: Contains a variety of protocols commonly used on the Internet.
  - HTTP (HyperText Transfer Protocol) is the underlying protocol for the World Wide Web.
  - Other protocols are FTP (File Transfer Protocol), E-mail, ...

- The **Presentation** Layer: Concerned with the syntax and semantics of the information transmitted.
  - Facilitates communication between big-endian computers and little-endian computers.



Big-endian vs Little-endian

- The **Session** Layer: Facilitates the use of sessions between end stations. During a session, the user and the computer system engage in a dialogue.
  - The session layer establishes and maintains dialogues.
  - Determines the type of control to be used (two-way simultaneous communication, two-way alternate communication, one-way communication).
  - Determines re-synchronization of the dialogue after a crash.

- The **Transport** Layer: This is a key layer, it is a true end-to-end layer.
  - Its function is to isolate the applications from the underlying network hardware technology.
  - It splits the source data into manageable chunks and passes them to the network layer.

    - It uses the network as a reliable ‘deliverer’ of data.

- The **Network** Layer: Concerned with controlling the operation of the sub-network (subnet)
    - Deals with the routing of packets from the source station towards the destination station across sub-nets.
    - It handles the different sub-net addressing formats.
    - Essentially it is responsible for interconnecting heterogeneous networks.

- The **Data Link** Layer: Concerned with getting data across an individual link.
    - Transforms a raw transmission facility into a data communications channel that appears free of transmission errors.
    - Breaks up the data into data frames. Deals with flow control, controlling access to a shared channel, ...

- The **Physical** Layer: Concerned with transmitting raw bits over a communication channel. Must ensure that when a binary 1 is sent it is received as such by the receiver.
    - Deals with voltage levels used, bit duration, ...
    - Design issues deal with mechanical, electrical, and timing interfaces, and the physical transmission medium.

The TCP/IP Reference Model

The TCP/IP Reference Model was developed after the protocols (TCP and IP).

The design goals were:

- Ability to survive the loss of subnet hardware.
- Ability to handle multiple types of data including files and real-time speech.

These requirements led to the adoption of a connectionless packet-switching network within the internet layer.

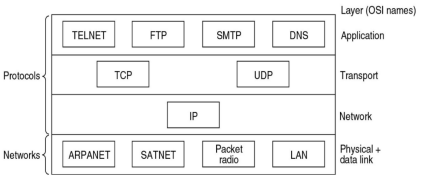
The TCP/IP Reference Model - Layer by layer

- The **Internet** Layer: This layer is key. It facilitates hosts injecting packets into any network and ensures correct routing of packets to the destination station.

- The **Transport** Layer: Facilitates end-to-end communication between the source and destination hosts. Two end-to-end transport protocols have been defined:
  - TCP (Transmission Control Protocol): Reliable, connection-oriented protocol that allows a byte to stream originating on one machine to be delivered without error to any other machine in the internet.
  - UDP (User Datagram Protocol): Unreliable, connectionless protocol for applications that provide their own sequencing and flow control functionality.

- The **Application** Layer: Contains all of the higher-level protocols including FTP, E-mail, DNS (Domain Name System) and HTTP.

- The **Host-to-Network** or **Network Interface** Layer: Meant to deal with hosts connecting to the network (similar to ISO OSI Data Link Layer).



- The **Physical** Layer: Similar to the Physical Layer of the ISO OSI model.

These lower two layers are not well defined within the TCP/IP reference model.

Comparison of OSI and TCP/IP Reference Models

Both models use the concept of a stack of independent protocols.

Both provide an end-to-end, network-independent transport service to applications.

Differences:

	OSI model	TCP/IP model
Number of layers	7 layers	5 layers
Services vs Interfaces/Protocols	defines what each layer does using service definitions	did not originally distinguish between service, interface and protocol. Hindered switching-out protocols to facilitate technological change
Timing	developed before the protocols were created	the protocols came first

Design Issues for Layered Software

- Addressing - each layer needs to be able to identify senders and receivers. Some form of addressing is required.
- Error control - receiver must be able to tell the sender which messages have been correctly received and which have not.
- Sequencing - the protocol software on the receiver must be able to resequence incoming messages.
- Flow control - the receiver must be able to control the flow of information from the sender.