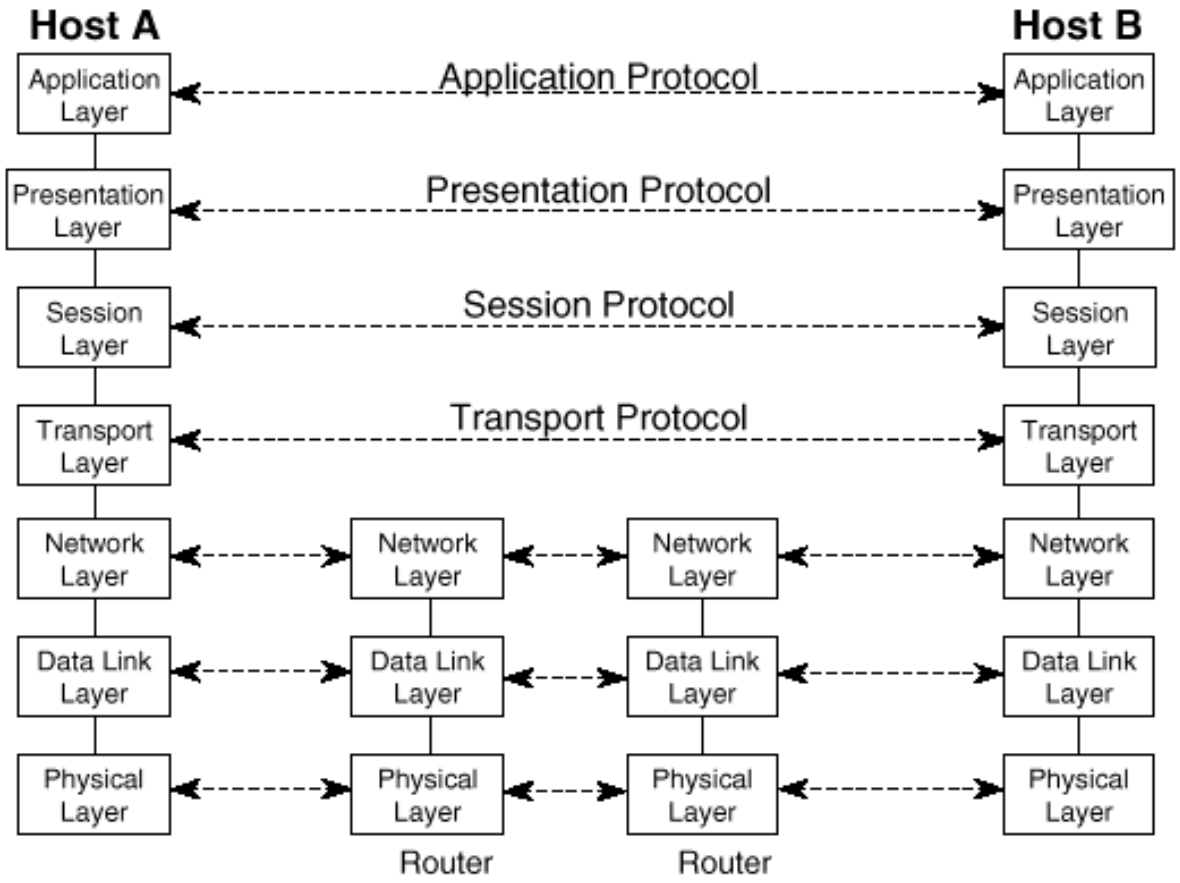
ISO Layering Architecture

Srinidhi Varadarajan

ISO Layering



ISO Layering Design

- **A layer should be created where a different level of abstraction is needed**
- **Each layer should perform a well- defined function**
- **The layer boundaries should be chosen to minimize information flow across the interfaces**
- **The number of layers should be large enough that distinct functions need not be thrown together in the same layer out of necessity, and small enough that the architecture does not become unwieldy**
- **Abstraction used to hide complexity**
  - **Need not know all the details of underlying support**
  - **Implies layering**

Physical Layer

- **Functions:**
  - **Transmission of a raw bit stream**
  - **Forms the physical interface between devices**
- **Issues:**
  - **Which modulation technique (bits to pulse)?**
  - **How long will a bit last?**
  - **Bit- serial or parallel transmission?**
  - **Half- or Full- duplex transmission?**
  - **How many pins does the network connector have?**
  - **How is a connection set up or torn down?**

Data Link Layer

- **Functions:**
  - **Provides reliable transfer of information between two adjacent nodes**
  - **Creates frames, or packets, from bits and vice versa**
  - **Provides frame- level error control**
  - **Provides flow control**
- **In summary, the data link layer provides the network layer with what appears to be an error- free link for packets**

Network Layer

- **Functions:**
  - **Responsible for routing decisions**
    - **Dynamic routing**
    - **Fixed routing**
- **Performs congestion control**
  - **In the Internet model, the network layer does not perform congestion control. Congestion control at the network layer is a current area of research**

Transport Layer

- **Functions:**
  - **Hide the details of the network from the session layer**
  - **Example: If we want replace a point- to- point link with a satellite link, this change should not affect the behavior of the upper layers**
  - **Provides reliable end- to- end communication**
  - **Perform end- to- end flow control**
  - **Perform packet retransmission when packets are lost by the network**
- **In the Internet model, the transport layer also offers congestion control.**

Session Layer

- **Functions:**
  - **May perform synchronization between several communicating applications**
  - **Groups several user- level connections into a single “session”**

Presentation Layer

- **Functions:**
  - **Performs specific functions that are requested regularly by applications**
  - **Examples:**
    - **Encryption**
    - **ASCII to Unicode, Unicode to ASCII**
    - **LSB- first representations to MSB- first representations**

Application Layer

- **Application layer protocols are application- dependent**
- **Implements communication between two applications of the same type**
- **Examples:**
  - **FTP**
  - **HTTP**
  - **SMTP (email)**

ISO Layering: Problems

- **Seven layers not widely accepted**
- **Standardized before implemented**
- **Top three layers fuzzy**
- **Internet or TCP/ IP layering widespread**

TCP/IP Layering

- **A simplified model**
- **Network Layer**
  - **Hosts drop packets into this layer, layer routes towards destination- only promise- try my best**
- **The transport layer**
  - **Reliable/unreliable byte-oriented stream**

Internet Design Principles

- **Scale**
  - **Protocols should work in networks of all sizes and distances**
- **Incremental deployment**
  - **New protocols need to be deployed gradually**
- **Heterogeneity**
  - **Different technologies, autonomous organizations**
- **End- to- end argument**
  - **Networking functions should be delegated to the edges; application knows best**

End-to-End Argument

- **Saltzer, Reed and Clark [1984]**
- **End- to- end arguments in system design**
- **Main idea**
  - **A function can only be completely and correctly implemented with the knowledge and help of the applications standing at the communication end points**
- **Complexity at the edges as opposed to the core**

End-to-End Argument: Advantages

- **Simple functions implemented in network**
- **Complex functions in the core tend to optimize the network for a given applications. E.g. reliability, encryption**
- **Hard to reengineer the network to some other application – telephony Vs data use**