Chapter 4 Communication

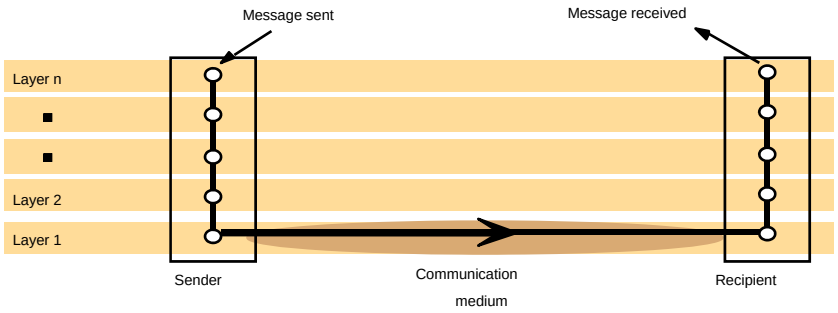
1

Network Communication Protocols

- Interprocess communication in distributed systems is based on **message passing** offered by the underlying network
- All communications between processes use standard rules that govern the format, contents, and meaning of messages sent and received. These rules are formalized in **protocols**.

2

Layering of Protocols

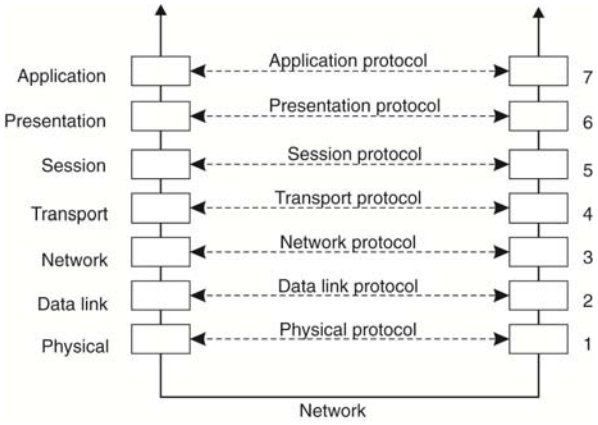


- Protocols are organized into layers
- A protocol is implemented by a pair of software modules located in the sending and receiving computers

The OSI Model

- The **Open Systems Interconnection model** (**OSI**) is a conceptual model that standardizes the internal functions of a communication system
  - Developed by the International Standards Organization (ISO) in 1980s
- The OSI model defines 7 layers, gives them standard names, and specifies the functionality of each layer
- The protocols developed as part of the OSI model were never widely used. However, the model itself is quite useful for understanding computer networks

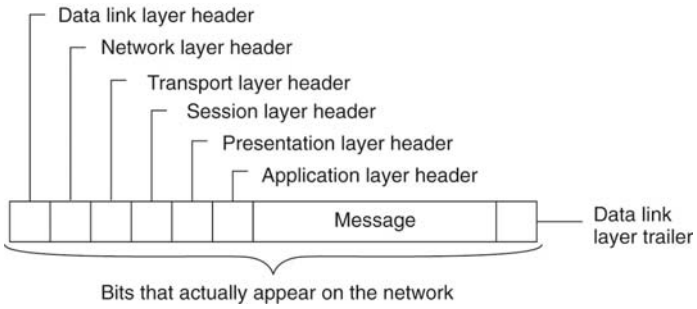
The OSI Model



5

Data Flow Between Layers

- At sending side, layer N accepts packet from layer N+1 and encapsulates it in a layer N packet before passing it to layer N-1
- At receiving side, layer N accepts packet from layer N-1 and removes the layer N header before passing it to layer N+1



A typical message as it appears on the network.

6

Functionality of OSI Layers (1)

- **Application layer** protocols are designed to meet the communication requirements of the specific applications
  - Examples: HTTP for Web, SMTP for Email
- **Presentation layer** protocols transmit data in a network representation that is independent of the representations used in individual computers, which may differ. Encryption is performed in this layer, if required.
  - Examples: TLS, CORBA common data representation
- **Session layer** establishes, manages, and terminates the connections between local and remote applications. It also performs failure detection and automatic recovery
  - Examples: SIP

7

Functionality of OSI Layers (2)

- **Transport layer** delivers application messages from source process to destination process
  - Examples: TCP, UDP
- **Network layer** transfers data packets from source host to destination host. Routing is performed at this layer.
  - Examples: IP, ATM virtual circuits
- **Data link layer** is responsible for transmission of packets between nodes that are directly connected by a physical link
  - Examples: Ethernet MAC, Wi-Fi MAC, PPP
- **Physical layer** is responsible for transmitting raw bits over a physical link
  - Examples: Ethernet baseband signaling, ISDN

8

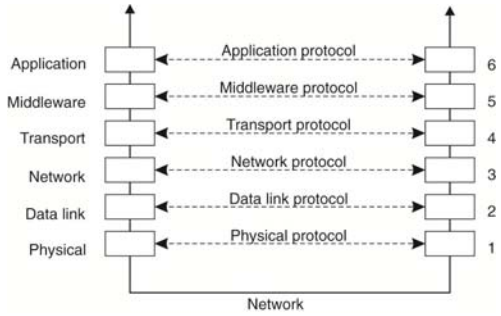
The Internet Architecture

- The protocols used in the Internet are organized into five layers
  - **Application layer**
  - **Transport layer**
  - **Network layer**
  - **Data link layer**
  - **Physical layer**
- The presentation layer is integrated with the application layer
- The session layer is integrated with the transport layer

9

Middleware Layer

- Middleware provides common services that can be used by many different applications; such services are implemented by middleware protocols
- Example middleware protocols
  - Protocols supporting high-level communication services, e.g., RPC, RMI, multicast
  - Security protocols, e.g., authentication protocols, authorization protocols
  - Distributed commit protocols for transactions
  - Distributed locking protocols for mutual exclusion



An adapted reference model for networked communication

10

Types of Communication

- Middleware can offer different types of communication to applications
  - Transient vs. persistent
  - Synchronous vs. asynchronous

11

Transient vs. Persistent Communication

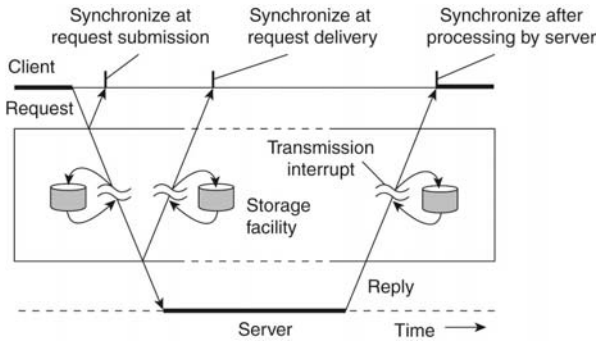
- **Persistent communication**: messages are stored by the communication middleware until receiver can accept it
  - Receiving application need not be executing when the message is submitted.
  - Example: Email
- **Transient communication**: a message will be discarded if the middleware cannot deliver it due to a transmission error or because the receiver is not executing
  - Example: TCP, UDP

12

Synchronous vs. Asynchronous Communication

- **Asynchronous communication**: sender continues execution immediately after submitting its message for transmission
  - The message is stored by the middleware upon submission
- **Synchronous communication**: sender is blocked until its request is known to be accepted, that is,
  - The middleware notifies acceptance of the message, *or*
  - The message has been delivered to the receiver, *or*
  - The receiver has processes the message & returned a response

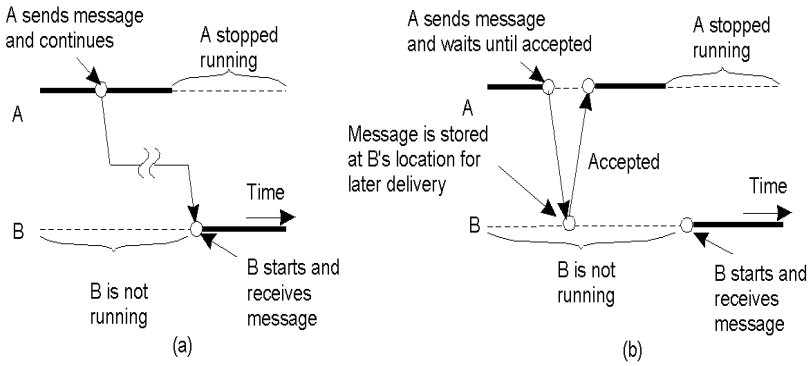
13



Three points where synchronization can take place.

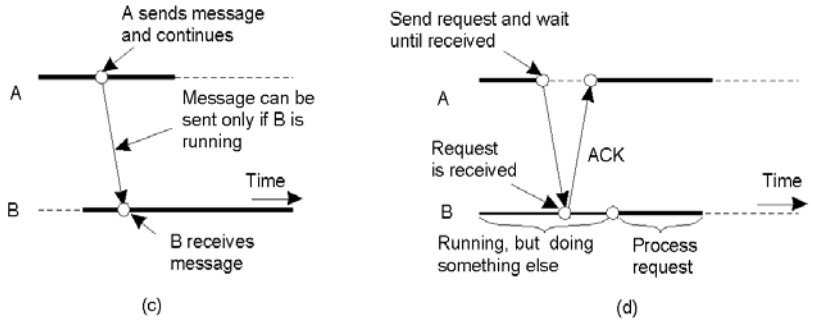
14

Persistence and Synchrony Combinations (1)



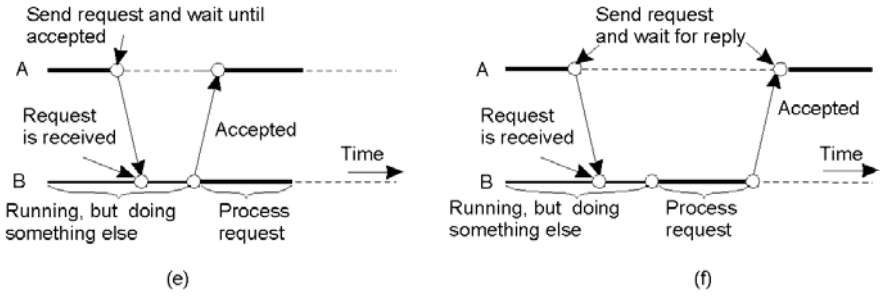
(a) Persistent asynchronous communication (e.g., email). (b) Persistent synchronous communication

Persistence and Synchrony Combinations (2)



(c) Transient asynchronous communication (e.g., TCP, UDP). (d) Receipt-based transient synchronous communication.

Persistence and Synchrony Combinations (3)



(e) Delivery-based transient synchronous communication (e.g., asynchronous RPC). (f) Response-based transient synchronous communication (e.g., RPC).