Protocol Architecture

Layered Protocol Architectures
OSI Reference Model
TCP/IP Protocol Stack

Need for Protocols

- The task of exchanging information between devices
  - requires a high degree of cooperation between the involved parties
  - can be quite complex

- Protocols are a set of rules and conventions. By enforcing that communicating parties adhere to a common protocol, communication is made possible.

- The complexity of the communication task is reduced by dividing it into subtasks:
  - Each subtask is implemented independently.
  - Each subtask provides a service to another subtask.
Example: Subtasks of Communications

- Example: The purchasing director of the Italian company “Vendetta”, located in Milan, Italy, wants to ask the Sales Director of the US company “Crash”, with headquarters in Mobile, AL, about the price of the Ultimo 6000 Supercomputer which is produced by Crash.

- When we divide the described communication task into subtasks we see that:
  - Separate entities in a company perform certain subtasks
  - Company entities provide services to other entities
  - An entity responsible for a certain subtask performs the task by following a protocol
Network Architecture

- **Protocol**: A set of rules and conventions used for communication of entities in different systems
  - **System**: Object that contains several entities (e.g., the company).
  - **Entity**: Anything capable of sending or receiving information (e.g., the secretary in a company).

- A Network Architecture is a structured set of protocols that implement the exchange of information between computers.

Layered Network Architecture

- In a **Layered Network Architecture**, the services are grouped in a hierarchy of layers.
  - An entity of layer N uses only services of layer N-1.
  - An entity of layer N provides services only to layer N+1.

- Example: Network Architecture

```
not layered
A ─ B ─ C ─ D ─ E

layered
A ─ B
  ─ C
   ─ D
   ─ E
```

© Jörg Liebeherr, 1998,1999

CS457
Layered Communications

• Each entity of a system is assigned to a layer

• An entity of a particular layer can only communicate with:
  1. adjacent layer entities via Service Interfaces
     above - to provide service
     below - to receive services
  2. peer layer entity using a common protocol (Peer Protocol)

Layered Communications

• A communication layer is completely defined by
  (a) The peer protocol between peer entities at the same layer
  (b) The service interface used to offer/provide services between adjacent layers

• Note: When talking about two adjacent layers,
  (a) the higher layer is a service user, and
  (b) the lower layer is a service provider
Layered Communications

- Layer interfaces define physical data flow.
- Peer protocols describe exchange of logical messages between peer layer entities.

Service Access Points

- A service user accesses services of the service provider at Service Access Points (SAPs).
- A SAP has an address that uniquely identifies where the service can be accessed.
Exchange of Data

- Assume a layer-N entity at A wants to send data to a layer-N peer entity to B.
- The unit of data send between peer entities is called a Protocol Data Unit (PDU).
- For now, let us think of a PDU as a single packet.

What actually happens: Layer N passes the PDU to one of A’s SAPs at layer N-1.
- The layer N-1 entity (at A) then constructs its own PDU which it sends to the layer N-1 entity at B.
- Note: PDU at layer N-1 = Header + PDU at layer N

Exchange of Data

When passed to the SAP, the PDU is called a Service Data Unit (SDU).
(Layer-N PDU = Layer-N-1 SDU)
Service Primitives

• Communication between adjacent layers is done via function calls. The functions are called service primitives

• Almost all communication is done with only four types of service primitives:
  - REQUEST: entity wants service provider to do work
  - INDICATION: service provider informs entity about an event
  - RESPONSE: entity wants to respond to an event
  - CONFIRM: response to an earlier request has come back

Assume the name of a service is called “X”.

N+1 Layer Entity \( \xrightarrow{X. \text{Request}} \) N Layer Entity

N Layer Entity \( \xrightarrow{X. \text{Response}} \) N+1 Layer Entity

N+1 Layer Entity \( \xrightarrow{X. \text{Indication}} \) N Layer Entity

N Layer Entity \( \xrightarrow{X. \text{Confirm}} \) N+1 Layer Entity

N+1 Layer Protocol
**Service Primitives**

Recall: A layer N+1 entity sees the lower layers only as a service provider.

![Diagram of Service Primitives](image)

**Example: Sending a Letter**

- Bob sends a letter to Alice

![Diagram of Example: Sending a Letter](image)
Putting the Example into our Context

- Identify the entities?
- Identify layers?
- What is the service?
- Who is a service user?
- Who is a service provider?
- What are the SAPs?
- What are the PDUs?
- Describe the peer protocol between Bob and Alice?

- Which service primitives are invoked?

- Bob, Alice, Postman
- Layer 2: Bob, Alice
  Layer 1: Postman
- Deliver_Letter (“L”)
- Bob and Alice
- Postman
- Mailboxes
- Letter (at Layer 2)
- The protocol is actually complicated: When Bob sends a letter to Alice, the letter must identify Alice as the recipient. The letter must be written in a language that Alice can read. Bob must be able to write, and Alice must be able to read. Both need to know where their respective mailboxes are, and they must be able to operate a mailbox, etc.

- Letter is dropped off by Bob (L.Request);
  Letter is delivered to Alice’s mailbox (L.Indicate)

(Un-)Acknowledged Service

- The example showed only two service primitives: L.Request, L.Indicate
- A service which uses these two primitives is called unconfirmed service

- If Bob asks for a “certificate of delivery” we would need: L.Request, L.Indicate, L.Response, L.Confirm
- The resulting service is called acknowledged service
Protocol Architectures

• There are only few protocol architectures that are relevant today:
  – OSI Reference Model
    • Defined as a big effort in the 1970’s by ISO to specify a comprehensive set of protocols for networking.
    • The effort failed, in that the defined protocols are not widely used. However, the concepts and terminology defined in the OSI model are the lingua franca of many networkers
  – TCP/IP Protocols Suite
    • The Internet protocol architecture is not the result of a design effort, but has evolved over several decades
  – ATM Protocol Stack
    • An example that protocols can be designed by a committee. Future relevance will depend on the success of ATM

OSI Reference Model

• In 1977 the International Standardization Organization (ISO) developed a model for a layered network architecture
• This effort was completed in 1983 and is known as the Open Systems Interconnection (OSI) Reference Model
• The OSI model defines seven layers:
  Layer 7: Application Layer
  Layer 6: Presentation Layer
  Layer 5: Session Layer
  Layer 4: Transport Layer
  Layer 3: Network Layer
  Layer 2: Data Link Layer
  Layer 1: Physical Layer
  (Layer 0: Interconnection Media)
OSI Layers

OSI Layers and Encapsulation
A Tour of the OSI Layers

- **Physical Layer (Layer 1):**
  - **Service:** Transmission of a raw bit stream over a communication channel
  - **Functions:** Conversion of bits into electrical or optical signals
  - **Examples:** X.21, RS-232-C

- **Data Link Layer (Layer 2):**
  - **Service:** Reliable transfer of frames over a link
  - **Functions:** Synchronization, error Control, flow control
  - **Examples:** HDLC, CCITT LAP-D
A Tour of the OSI Layers

• **Network Layer (Layer 3):**
  *Service:* Moves packets inside the network.
  *Functions:* Routing, Addressing, Switching, Congestion Control.
  *Examples:* IP, X.25, CLNP.

• **Transport Layer (Layer 4):**
  *Service:* Controls delivery of data between hosts.
  *Functions:* Connection establishment/management/termination, Error Control, Flow Control, Multiplexing.
  *Examples:* TCP, UDP, ISO TP0 - TP4.

• **Session Layer (Layer 5):**
  *Service:* Support the dialog between cooperating application programs
  *Functions:* Session establishment/management/termination, Synchronization, Recovery
  *Examples:* ISO session protocol, RPC

• **Presentation Layer (Layer 6):**
  *Service:* Provides freedom from compatibility problems
  *Functions:* Virtual device support, syntax conversion, encryption
  *Examples:* ISO presentation protocol

• **Application Layer (Layer 7):**
  *Service:* Provides network access to application programs
  *Functions:* Everything is application specific
  *Examples:* File Transfer, Electronic Mail
TCP/IP Protocol Suite

- The TCP/IP protocol suite was first defined in 1974
- The TCP/IP protocol suite is the protocol architecture of the Internet
- The TCP/IP suite has four layers: Application, Transport, Internet, and Network Interface Layer

Example: File Transfer
Encapsulation in the TCP/IP Suite

- As data is moving down the protocol stack, each protocol is adding layer-specific control information.

TCP/IP Protocol Suite

- The complete TCP/IP protocol suite contains many protocols. The following graph is far from complete.
Comparison of OSI Model and TCP/IP Suite

<table>
<thead>
<tr>
<th>OSI</th>
<th>TCP/IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application</td>
<td>Application</td>
</tr>
<tr>
<td>Presentation</td>
<td>Transport</td>
</tr>
<tr>
<td>Session</td>
<td>Internetwork</td>
</tr>
<tr>
<td>Transport</td>
<td>Network Access</td>
</tr>
<tr>
<td>Network</td>
<td>Physical</td>
</tr>
<tr>
<td>Data Link</td>
<td></td>
</tr>
<tr>
<td>Physical</td>
<td></td>
</tr>
</tbody>
</table>

The B-ISDN ATM Reference Model

- ATM technology has its own protocol architecture

Control Plane | User Plane
---|---
Upper Layer | Upper Layer
ATM Adaptation Layer (AAL) | End-to-end layer
ATM Layer | Transfer of Cells
Physical Layer | Same as in OSI
Layers of ATM

- The ATM Layer is responsible for the transport of 53 cells across an ATM network.
- The ATM Layer can provide a variety of services for cells from an ATM virtual connection:
  - **Constant Bit Rate (CBR)**
    - guarantees a fixed capacity, similar to circuit switching
    - guarantees a maximum delay for cells
  - **Variable Bit Rate (VBR)**
    - guarantees an average throughput
    - can guarantee maximum delay
  - **Available Bit Rate (ABR)**
    - guarantees “fairness” with respect to other traffic
  - **Unspecified Bit Rate (UBR)**
    - service is on a “best effort” basis
ATM Adaptation Layer (AAL)

- AAL provides services which are between upper layers and ATM layers.
- An important service is the segmentation and reassembly of upper layer data.

AAL Service Classification

- AAL has 4 different protocols: AAL 1, AAL 2, AAL 3/4, AAL 5
- Each protocol provides a different service

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Class A</th>
<th>Class B</th>
<th>Class C</th>
<th>Class D</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timing information in data?</td>
<td>Required</td>
<td>Not required</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bit rate of data</td>
<td>Constant</td>
<td>Variable</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mode of connection</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AAL protocol to be used</td>
<td>AAL 1</td>
<td>AAL 2</td>
<td>AAL 3/4</td>
<td>AAL 5</td>
</tr>
</tbody>
</table>
ATM Services and AAL Protocols

CBR  rt-VBR  nrt-VBR  ABR  UBR

AAL 1  AAL 2  AAL 3/4  AAL 5