Protocol architectures

Gruppo Reti TLC
name.lastname@polito.it
http://www.telematica.polito.it/

Architectures and protocols

• CCITT definition
  – Communication: transfer of information according to pre-established conventions
• Communication requires cooperation
• One abstract description of the communication paradigm between two or more users requires the definition of a reference model
• At the maximum level of abstraction, the reference model specifies a network architecture

Protocol architectures

• A network architecture defines the objects used to describe
  – the communication process
  – the relation among such objects
  – the functionalities to support the communication
  – the structure of the functions
• Layered architectures are used because of
  – simple design
  – simple management
  – simple standardization
  – separation among functions
COMPUTER NETWORKS – Protocol architectures

Separation among functions: Internet

Open System Interconnection
OSI

Layered architectures

Pag. 2
OSI and Internet

OSI
- Application
- Presentation
- Session
- Transport
- Network
- Data link
- Physical

Internet Protocol Suite
- Telnet
- FTP
- SMTP
- SSH
- SNMP
- TCP
- UDP
- NFS
- XDR
- RPC
- ARP
- RARP
- ICMP
- Routing protocols

B - ISDN

Management plane
- Control plane
- User plane
- High layers
- High layers
- AAL
- ATM
- Physical

Protocols

- CCITT definition
  - formal definition of the procedures adopted to guarantee the communication between two or more objects on the same hierarchical level
- Protocol definition:
  - semantics
    - set of commands and answers
  - syntax
    - structure of commands and answers
  - timing
    - temporal sequence of commands and answers
Protocols

- Protocols are set of
  - semantic rules
  - algorithms
- syntactic rules
  - formats
- timing

ISO/OSI model

- (Open System Interconnection) defined in the following standards
  - ISO IS 7498
  - CCITT X.200
- The fundamental principles defined in the OSI model are universally accepted
  - this does not mean that all the protocol architectures conform to OSI model
Application process

Layers (or levels)

Entities

- active elements in a subsystem
- run the functions of the layer
- interact within the same layer
Layering

- Each layer (or level)
  - provides services to the higher layer
  - using
    - the services from the lower layer
    - its own functionalities
- Can be identified:
  - service provider
  - service user
  - SAP (Service Access Point)

Services

- The users at layer N and the (N+1) - entities cooperate and communicate using the (N) – service offered by the (N) – service provider

- A service can be:
  - connection-oriented (CO): a preliminary agreement (connection) is established between the network and the communication end-points, then the data is transferred and finally the connection is released
  - connectionless (CL): data is sent to the network without any preliminary agreement and is treated independently from each other
Each (N-1)-SAP is associated with at most one (N)-entity.
Protocols

System A

(N) - entity

(N+1) - protocol

(N) - service

(N) - layer

(N) - SAP

(N) - entity

System B

(N) - entity

(N+1) - protocol

(N) - service

(N) - layer

(N) - SAP

(N) - entity

Addressing

(N) - layer

(N) - entity

(N) - title

(N-1) - SAP

(N-1) - layer

(N-1) - entity

Functions for identification

• Address translation
  – (N) directory

(N) - entity

(N) - title

(N-1) - address
Functions for identification

- Address translation
  - \((N)\) - mapping

Possible mappings

Connections
Agreement among three possible elements

The three elements

(N) – service provider
(N+1) - entity
(N+1) - entity

Agreement

• In the case of information transfer without connection, it is sufficient an agreement among two elements

The two elements

(N) – service provider
(N+1) - entity
(N+1) - entity

Agreement

• In the case of information transfer with connection, it is necessary an agreement among the three elements

The three elements

(N+1) - entity
(N+1) - entity
(N) – service provider
Connections

- multiplexing of (N) – connections into one (N-1) - connection

Connections

- subdivision of one (N) – connection in many (N-1) - connections

PDU creation
PDU creation

- On data units, there exists the possibility of
  - segmentation
  - concatenation
- Segmentation can occur either by building many (N) - PDU from one (N) - SDU or by building many (N-1) - SDU from one (N) - PDU
- Similarly for the concatenation
Primitives

- set of interactions on an interface, occurring in different times and offering a service
- Example: service of mail transfer in the postal system
  - Deposit of the letter in the mailbox by the sender
  - Delivery of the letter into receiver's mailbox by the postman
- Similar to a procedure

Use of primitives

Acknowledged service
Use of primitives

Un-acknowledged service

Service provider

Service user

Request primitive

Service provider

Service user

Indication primitive

Use of primitives

Service started from the provider

Service user

Service user

Indication primitive

Use of primitives

To open a connection

(N+1) - entity

(N) - entity

(N) - service provider

(N) - CONNECT

REQUEST

CONFIRM

(N) - SAP

(N) - SAP

(N) - CONNECT

INDICATION

(N) - CONNECT

RESPONSE
To transfer data

Use of primitives

(N) - service provider
(N+1) - entity
(N+1) - entity
(N) - DATA
INDICATION
N - SAP
N - SAP

To close a connection started from the user

Use of primitives

(N) - service provider
(N+1) - entity
N - DISCONNECT
INDICATION
N - SAP
N - SAP

The seven OSI layers

application
presentation
session
transport
network
data link
physical

Application protocol
Presentation protocol
Session protocol
Transport protocol
Network protocol
Data link protocol
Physical layer protocol

transmission media

application
presentation
session
transport
network
data link
physical
**Systems**

- terminal system
- relay system

<table>
<thead>
<tr>
<th>TERMINAL S. A</th>
<th>RELAY SYSTEM</th>
<th>TERMINAL S. B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application</td>
<td>Application</td>
<td></td>
</tr>
<tr>
<td>Presentation</td>
<td>Presentation</td>
<td></td>
</tr>
<tr>
<td>Session</td>
<td>Session</td>
<td></td>
</tr>
<tr>
<td>Transport</td>
<td>Transport</td>
<td></td>
</tr>
<tr>
<td>Network</td>
<td>Data link</td>
<td>Data link</td>
</tr>
<tr>
<td>Physical</td>
<td>Physical</td>
<td>Physical</td>
</tr>
<tr>
<td></td>
<td></td>
<td>transmission media</td>
</tr>
</tbody>
</table>

**Public networks**

<table>
<thead>
<tr>
<th></th>
<th>user protocols</th>
<th>user protocols</th>
<th>transfer layers</th>
<th>transfer layers</th>
<th>transfer layers</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>terminal protocols</td>
<td>network protocols</td>
<td>access node</td>
<td>transit node</td>
<td>access network</td>
</tr>
</tbody>
</table>

**Layer 1: physical**

- Physical layer:
  - provides the mechanical, physical, functional and procedural means, to activate, maintain and disable the physical connections
  - allows to transfer binary digits exchanged among the data link entities
  - data units are bits or symbols
  - defines transmission codes, connectors, voltage levels, etc.
Layer 2: data link

• Data link layer
  – provides the functional and procedural means to transfer data units among network entities
  – handle malfunctions and failures at physical level
  – main functions:
    • error detection and error correction for the transmission
    • flow control
    • data unit delimitation

Layer 3: network

• Network layer
  – provides the means to setup, maintain and close the connections among the entities at transport level
  – provides the functional and procedural means to exchange the information among entities at transport level
  – main functions
    • routing
    • flow control and congestion control
    • pricing

Layer 4: transport

• Transport layer
  – provides the connections at transport level to the entities at session level
  – compensates the possible lack of quality of service in the connections at network level
  – optimizes the use of the network layer
  – main functions
    • error control
    • control of sequence
    • flow control
Layer 4: transport
- lower layer with end-to-end meaning
- provides multiplexing and subdivision of the connections
- allows the fragmentation of messages in packets and their reassembly

Layer 5: session
- Session layer
  - provides one session connection to the entities at presentation layer
  - organizes the communication among entities at presentation level
  - provides the structure and synchronize the data exchange to allow suspending, recovering and terminating
  - masks the interruptions at service level

Layer 6: presentation
- Presentation layer
  - solves the compatibility issues regarding the data formats
  - solves the issues of data syntax translation
  - may provide services of data encryption
Layer 7: application

- Application layer
  - provides the application processes with the means to access the OSI environment

- Examples of service
  - file transfer - FTAM
  - virtual terminal - VT
  - e-mail - X.400

Example

- Trivial network

Example

- Assume that one (4) – entity must communicate with one remote (4) – entity
- We will follow, step-by-step:
  - primitives
  - SDUs
  - PDUs
• layer 3 service with connection
• layer 2 service connectionless
• layer 1 service connectionless
Example

N-CONNECT.request(H, A, ...)

Example

N-PDU(H, A, VCid', call request, ...)

Example

Routing

A, H → A, Z
Example

Mapping

Example

DL-DATA.request(D, B, DL-SDU, ...)

Example

DL-PDU(D, B, DL-SDU, DL-PCI)
Example

Mapping

PH-DATA.request(E, C, symbol)

Example

Example
Example
N-PDU(H, A, Vcid', call request, ...)

Example
Routing
A, H → Z, H

Example
Mapping
Z, H → F, I
Example

N-PDU(H, A, VCld', incoming call, ...)

Example

DL-DATA.request(I, F, DL-SDU, ...)

Example

DL-PDU(I, F, DL-SDU, DL-PCI)
Example

Mapping

F, I → G, J

Example

PH-DATA.request(J, G, symbol)

Example

01100111
Example

N-PDU(H, A, VClid", incoming call, ...)

Example

N-CONNECT.indication(H, A, CEP.id", ...

Example

N-CONNECT.response(CEP.id", ...)
Example

N-PDU(A, H, VClid', call connected, ...)

Example

N-CONNECT.confirm(A, H, CEP.id'', ...)

Example

T-PDU(T-PCI, T-SDU)
Example

N-DATA.request(CEP.id', N-SDU, ...)

Example

N-DATA.indication(CEP.id', N-SDU, ...)

Example

T-PDU(T-PCI, T-SDU)
Example

N-DISCONNECT.request(CEP.id)

Example

N-DISCONNECT.indication(CEP.id)