# agement Aspects for OSI Transport Layer over a Satellite Channel.

Renzo Beltrame
Enrico Gregori

Report C82-22

Consiglio Nazionale delle Ricerche

CNUCE

PISA

# Management Aspects for OSI Transport Layer over a Satellite Channel.

Renzo Beltrame
Enrico Gregori

Report C82-22

Istituto CNUCE
PISA
June 1982

TABLE OF CONTENTS

Abstract

This paper describes the management aspects we found necessary for the simulation of a layer 4 of OSI architecture over a satellite channel.
No specialized management entities has been introduced, so as the management functions here described are to be considered generally distributed. A particolar type of connection has been also studied, that is the one-point to multi-end-points connection. The solution which has been found, particurarly for what addressing are concerned both minimize the differences with the transport protocol for point to point connections and allows us to use the TPDU format stated by ISO for point to point connections.

## 1.0 INTRODUCTION

In this paper we are presenting the management aspects we found necessary for the simulation on processor - by means of discrete simulation techniques - of a layer 4 ISO/OSI architecture. (see ref. R.4). The scheme is as follows.

A complete simulation is made of all connection opening and connection termination protocols of layer 4 in a point-to-point and one-point-to-multi-end-point connection-oriented system.

The underlying layers are simulated as common channels with a great propagation delay, that is, a typical satellite channel.

The overlying layers are simulated as end users with the only functions necessary for the opening and termination of connections.

As far as management is concerned, we have, in this case, a particularly interesting situation. Studies are made for problems of management pertinent to a certain layer as an isolated layer, for problems deriving from the underlying layers and for problems deriving from the connection of this layer to an end layer.

In designing the management we have not introduced an interconnection system different from that provided for the data, and this independently from the fact whether the entities are in the same system or not. The exchange of information between one entity and the other takes place only via service access points provided for data transfer. Problems of priority are demanded to the service access point context.

Besides, every entity is given the greatest functional autonomy, and has the greatest number of possible management functions.

Within the limits of managerial functions needed for the opening and termination of connections, the choice of this kind of architecture has proved to be perfectly efficient.

Other problems demanding a centralized management call for the introduction in the architecture of particular entities specialized in the functions of management but not in the interconnection logic. Historically, this choice reveals a formulation -AFNOR- (see ref. R.1) of the entity pertaining to an intermediate layer - the transport layer in our case - , which is an abstract machine having the architecture illustrated in Fig. 1.

In the case of end layer, the architecture of the abstract machine becomes that illustrated in Fig. 2.

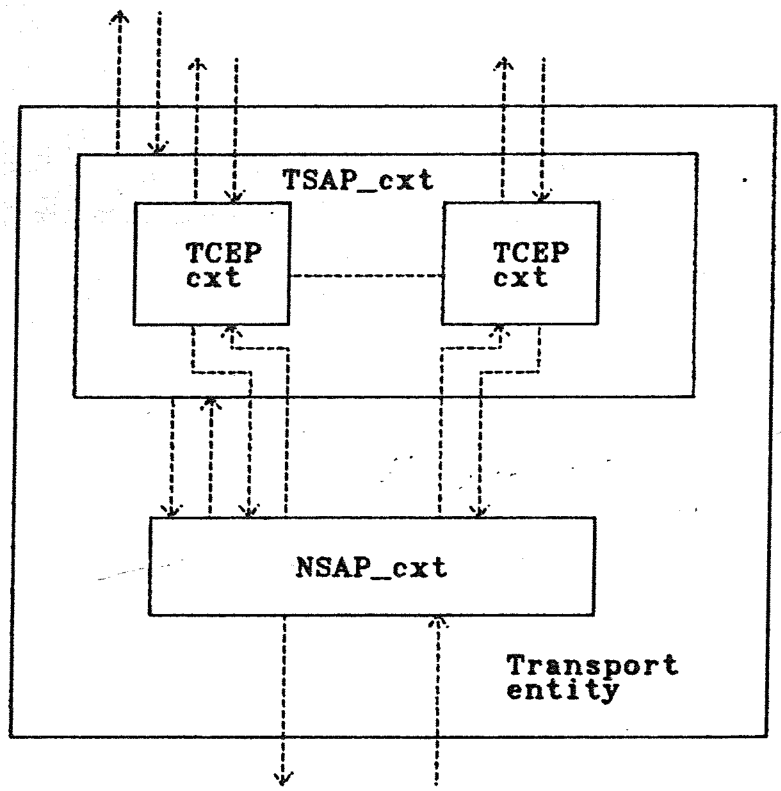



Fig. 1 - Transport entity architecture

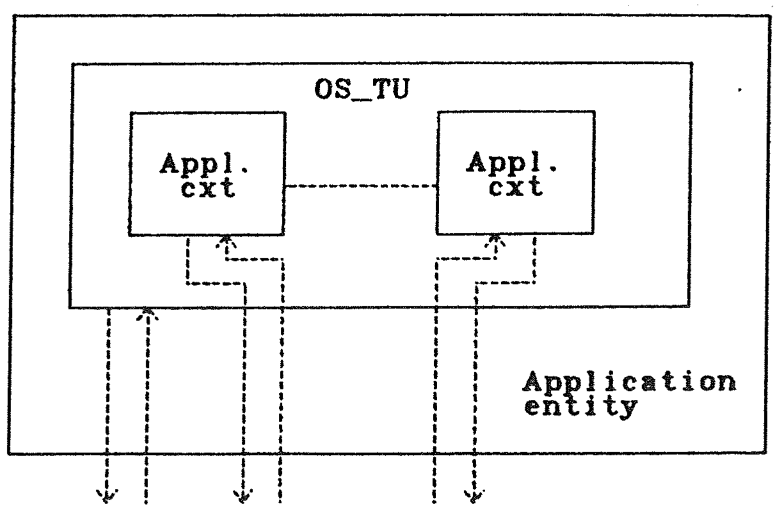



Fig. 2 - Application entity architecture

## 2.0 UTILITY APPLICATION LAYER

The transport entity is connected, in the overlying layer, to an entity bearing the characteristics typical to an application entity. No session functions nor presentation functions are introduced as these are not essential to the simulation. In coherence with the choice of the architecture, the entity comprises the management of the individual applications. These applications are activated and disactivated dynamically at the opening and termination of the respective connection.

Each application entity is provided with the maximum number of applications that can be activated, which we will call Transport Users (TU) and a manager of such (OS_TU) activated during the starting phase of the system.

It is decided that an application can be connected only to one end point connection.

## 2.1 MANAGEMENT OF THE APPLICATIONS

In order to carry out its functions, the manager needs the following information:

- data concerning the applications (name, state, resource, etc.) (see table 1);
- data concerning the other managers of the same layer (name, address) (see table 1).

The manager is connected to the manager of the underlying entity via the respective service access point during activation of the system.

The manager of the applications has two sets of procedures, one for the initialization (INITIALIZATION, READ_NAMES), and one for the activation and disactivation of the local applications (TU_EXIST, TU_START).

### 2.1.1 Activation of an Application

(Procedure TU_START)

When an activation is requested, the manager checks the existence of the application having the requested name (procedure TU_EXIST) and its availability (TU_STATE table).

If the answer is positive, the application requested is activated and the table of state is updated.

The transport address is then given (procedure GIVE_OS_TU_ADDR). The OS_TUs are provided with an identifier having a global significance within the layer, while the name of the applications is local to the entity.

| TSAP_addr / OS_TU name | |
|---|---|
| name1 | nn |
| | |
| nameN | mm |

Addressing table

| name | state | reference |
|---|---|---|
| appl1 | active | ptrX |
| | | |
| applK | free | nil |

Applications management table

**Table 1 - Application manager tables**

## 3.0 MANAGEMENT OF THE TRANSPORT ENTITY

The Transport Service Access Point Context (TSAP_CXT) has the task of managing the transport connections (opening, termination and respective assignment and release of the resources).
Protocols and related functions are instead performed by the Transpor Connection End Point Context (TCEP_cxt).

### 3.1 ADDRESSING

An identifer (Connection reference) having a significance local to the entity is assigned to every Transport Connection. The assignment algorithm is different for point-to-point connections and for one-point to multi-end-points connections.

In both cases, for the assignment of the connection references, a list structure and two pointers are provided, with one pointer at the first free reference and the other at the last.

The identifiers made available after the termination of a connection are re-usable. To avoid ambiguity, the released identifier can be re-used only if a certain interval of time has passed after the confirmation of termination to assure that within the network, no other packets relative to this connection exists (frozen reference principles). Besides, the identifier is queued up with the last one of the queue available so as to minimize statistically the frozen time.

#### 3.1.1 One-point to multi-end-point connection

Only one-point to multi-end-point centralized connections are considered. In such connections, only the side that has opened the connection sends TPDU data, while the others can only send control TPDUs. The solution found for addressing minimizes the modifications both to the semantics and to the syntax of the protocol.

The type of one-point to multi-end-points connection, introducing a one-point to multi-end-point connection in one direction and a multi-end-point to one-point in the other, is such that the destination reference used by the multi-end-points is equal for all multi-end-points and equal to the reference source of the master.

Since the data are always sent by the master to all the multi-end-points in a non-selective way, the master can use one sole destination reference.

This possibility enables the master to keep the format of the packet unchanged and, for simplicity, he uses the same number for source and destination reference in the protocol.

On the other hand, it is advisable, for performance reasons, that the addresees are distinguishable when they send control information. A second hierarchical level is thus introduced, so that the one-point to multi-point connection is provided with an identifier (source-reference=destination reference) which characterizes globally the system and with an inter-group-reference which characterizes the end points.

In the protocol each slave uses as source reference its inter-group references. To avoid ambiguity, the identifier of the connection (1st level) must have a global significance within the layer.

The re-usage of the identifiers made available after the termination of a connection is completely analogous to that of the point to point connection.

## 3.2 INFORMATION CHARACTERIZING THE SINGLE TRANSPORT CONNECTIONS

For each transport connection, a data structure is defined (TAB_TCEP_CXT) which provides the manager with all information relative to the connection (see table 2.1).

This structure is used by the manager to carry out the termination procedure and to release the resources of a TCEP_cxt whichever is its state.

## 3.3 CREATION OF A TRANSPORT CONNECTION CONTEXT (TCEP_CXT)

A TCEP_cxt can be created by the arrival of a request from the overlying entity (procedure CONN_START) or by the arrival of a request (CR) sent by a peer entity through the underlying level (procedure CONN_COMPLETE).

In the first case, control is done on the accessibility of the addressee (procedure PATH_AVAIL), on the possibility of assigning a TCEP_id (procedure TCEP_AVAIL), and on the availability of the resources asked for (procedure RESOURCE_AVAIL). If the results are positive, the path (procedure PATH_ASSIGN) and the identifier of the end point of the connection, (procedure TCEP_ASSIGN) are assigned, and the connection end point contexts and the respective data structure (TAB_TCEP_CXT) are created.

In the second case, it has first to be assured that the CR is not a duplicate (procedure CONN_EXIST). The manager possesses a data structure where all source references with which every transport entity has sent requests for connection opening are recorded (see table 2.2).

When the CR is not duplicate, it is checked: if there is any TCEP_ID available, if the resources requested are available, and if the manager of the upper layer is able to activate the transport user requested (procedure TU_START pertaining to OS_TU). If the results are positive, the path -see table 2.1-
(procedure PATH_ASSIGN) and the identifier -TCEP_ID (procedure TCEP_ASSIGN) are assigned and the connection end point context and the respective data structure created and linked (see table 2.2).

## 3.4 RESOURCE RELEASE

The resources are logically subdivided according to the possibility of releasing them at different times. We follows, in this way, the strategy of releasing a resource as soon as possible.

Two other procedures have been provided, a PURGE procedure which ends by force the transport connections and a reconfiguration procedure which ends by force the transport connections connected to a node which has become unreachable.

## 3.5 NSM_CXT

As far as management is concerned, here we have the information relative to the capacity of connection of the

| TCEP_id \ TSAP_addr | nn | | mm |
|---|---|---|---|
| 1 | T | | F |
| | | | |
| Nmax | F | | F |

Path information table

| TCEP_id \ Tab_TCEP_cxt ptr | |
|---|---|
| 1 | ptr1 |
| | |
| Nmax | nil |

| Tab_TCEP_cxt |
|---|
| source_ref |
| dest_ref |
| buff_ptrs |
| ..... |
| buff_len |
| ......... |
| ......... |

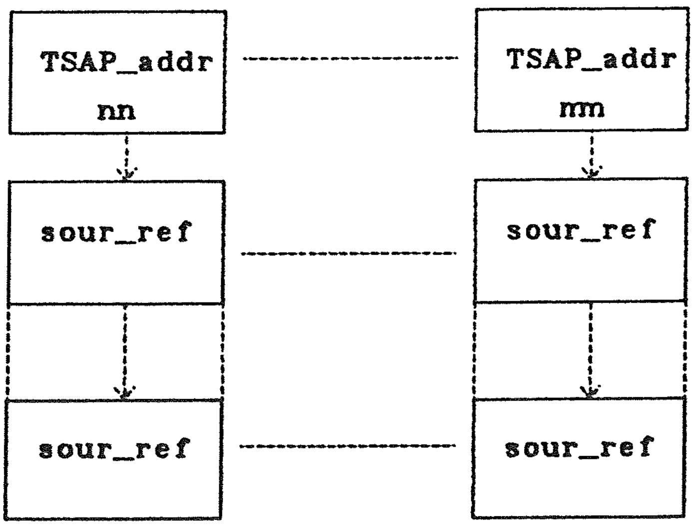
Reference table of blocks data
for TCEP_cxt management

Table 2.1 - Transport manager tables (part 1)

common channel. The NSM_CXT is notified about every variation in this capacity and will start the subsequent reconfiguration and purge procedures.

| TCEP_id \ TCEP_cxt ptr | |
|---|---|
| 1 | ptr1 |
| | |
| Nmax | nil |

TCEP_cxt pointers table




Open connections information structure

**Table 2.2 - Transport manager tables (part 2)**

<u>REFERENCES</u>

R.1 AFNOR,"Application of guideline to a proposed transport protocol", ISO/TC 97/SC 16 N-311, April 1980.

R.2 "Data processing - Open System Interconnection - Basic Reference Model ", ISO/TC 97/ SC 16 N-537, May 1981.

R.3 "Draft Transport protocol specification", ISO/TC 97/SC 16 N-698, May 1981.

R.4 R. BELTRAME, E. GREGORI, "Simulazione di un Transport Layer secondo la classe 4 della proposta di standard ISO - Apertura e chiusura delle connessioni", CNUCE C 81-4, Giugno 1981.

R.5 R. BELTRAME, "Simulazione di un canale comune a divisione di tempo via satellite come supporto ad un Transport Layer OSI compatibile", CNUCE C 81-6, Giugno 1981.