ECMA

EUROPEAN COMPUTER MANUFACTURERS ASSOCIATION

MAINTENANCE AT THE INTERFACE BETWEEN DATA PROCESSING EQUIPMENT AND PRIVATE SWITCHING NETWORK

TR/34

June 1986

Free copies of this document are available from ECMA,
European Computer Manufacturers Association
114 Rue du Rhône – 1204 Geneva (Switzerland)

ECMA

EUROPEAN COMPUTER MANUFACTURERS ASSOCIATION

MAINTENANCE AT THE INTERFACE BETWEEN DATA PROCESSING EQUIPMENT AND PRIVATE SWITCHING NETWORK

TR/34

June 1986

BRIEF HISTORY

This Technical Report considers maintenance aspects of the Data Processing Equipment (DPE) interface to a Private Switching Network (PSN). It is intended as guidelines to the developers of the relevant standards. The principle areas to be considered are:

- Assistance with testing at the boundary by defining domains and boundaries
- Requirements for maintenance
- Architectural framework for maintenance applications
- Basic mechanisms involved in remote maintenance communication
- Guidelines for the development of maintenance procedures
- Guidelines for the development of protocol elements

It assumes the ISDN concept as developed by CCITT and described by Standards ECMA-102, ECMA-103, ECMA-104, ECMA-105 and ECMA-106. It has the following objectives:

- to state the ECMA policy with regard to the application of CCITT Recommendations and ECMA Standards relevant for the maintenance at the interfaces concerned;
- to serve as a reference document and to fulfill a tutorial role for ECMA to guide the development of standards for the maintenance at the interfaces concerned.

The purpose of this Technical Report is not to define completely new approaches or solutions in the maintenance field, but rather to complement existing international standards and support standardization efforts. ECMA's major concern is that its standards be compatible with CCITT and CEPT Recommendations describing maintenance at the user-to-ISDN interface. Where possible and appropriate, compatibility with standards and recommendations of other bodies, such as ISO and IEC, has been maintained; where discrepancies have been noted between such specifications, a preferred solution is recommended.

Special facilities required for installation, commissioning, load testing or type approval and network management are beyond the scope of this Technical Report. However, they have been kept in mind so that the mechanisms developed do not preclude further enhancement for such applications.

Adopted as an ECMA Technical Report at the General Assembly of June 26, 1986.

TABLE OF CONTENTS

1.

1. GENERAL

The ECMA ISDN maintenance concepts take into consideration basic ISDN features, such as:

- Open communication via the S reference points.
- Portability of DPEs between S reference points, from PSN to PSN and from ISDN to ISDN.

The ECMA ISDN maintenance concepts are based on defining a user's domain of responsibility for equipment beyond the ISDN's reference point (as seen from the ISDN or from the PSN).

The PSN's S reference point has been chosen, in accordance with CCITT Rec. I.412, as an adequate point for description of the interworking protocols in Layers 1 through 3, and for demarcation between the PSN and terminal suppliers' domains of responsibility, see Figure 1.

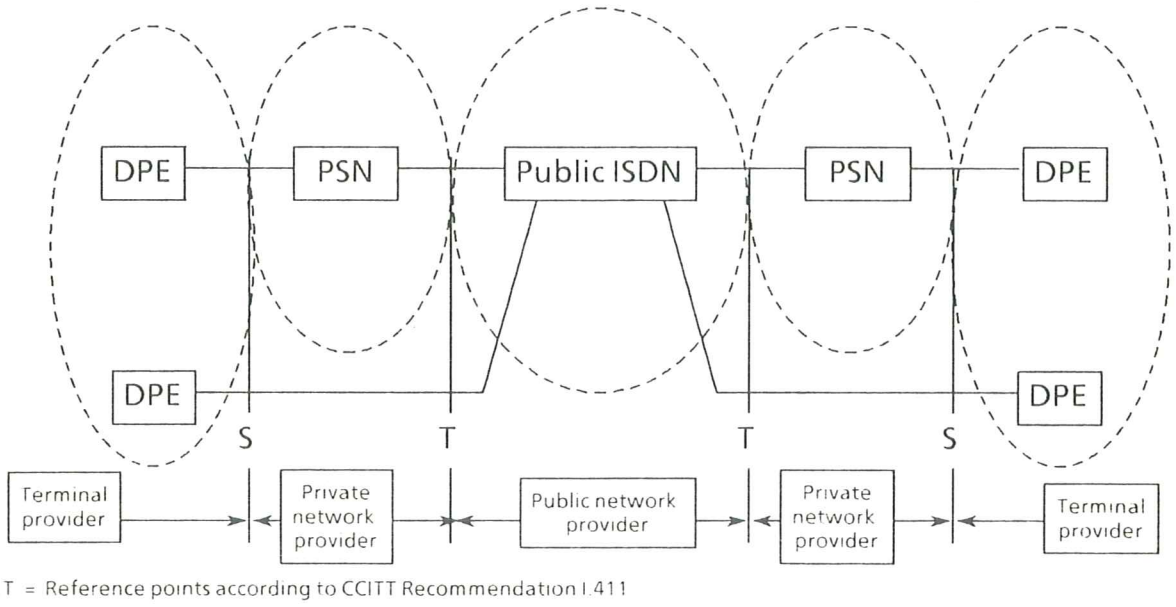


Figure 1: ECMA ISDN Maintenance Domains

The ECMA ISDN maintenance concepts define three basic domains of maintenance responsibility:

- a domain for the public network provider,
- a domain for the private network provider,
- a domain for the terminal provider.

The ECMA ISDN maintenance concept assumes that every domain provider should be capable of determining, whether a supposed fault lies in his or in a foreign domain. This should be possi-

ble locally and remotely, i.e. across intervening networks, between any maintenance entities. It will, however, be subject to appropriate access supervision mechanisms which enable each domain to protect against unauthorized access.

The ECMA ISDN maintenance concepts allow comprehensive maintenance, e.g. for a customer to maintain the totality of his own equipment (PSNs and terminals) from a maintenance centre.

2. SCOPE

The scope of this Technical Report extends to the S_0 and S_2 interfaces between DPEs and PSNs, as described in Technical Report ECMA TR/24. Only testing and maintenance of the interfaces' Physical Layers and D-channel protocol functions are considered here. Other DPE and PSN functions are excluded, i.e. they are considered subjects of separate maintenance procedures. No assumption is made concerning the internal structure of each end of a connection. For reasons of symmetry, the two ends of a connection will only be differentiated by the part they play in maintenance operations, i.e. requesting or executing.

Emphasis is given to maintain (or at least diagnose) the functions in question not only locally but also remotely. Figure 2 shows the scenario for maintenance of a local interface while Figure 3 shows the scenario for maintenance at a remote interface.

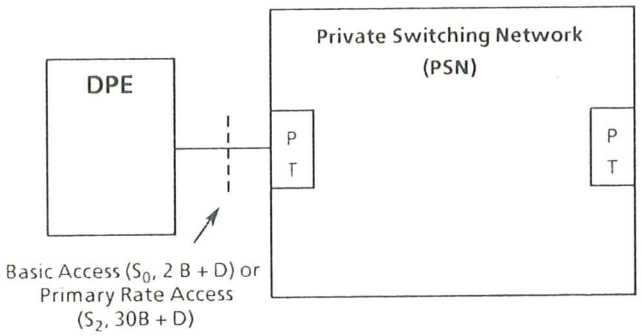


Figure 2 - Local Maintenance Scenario

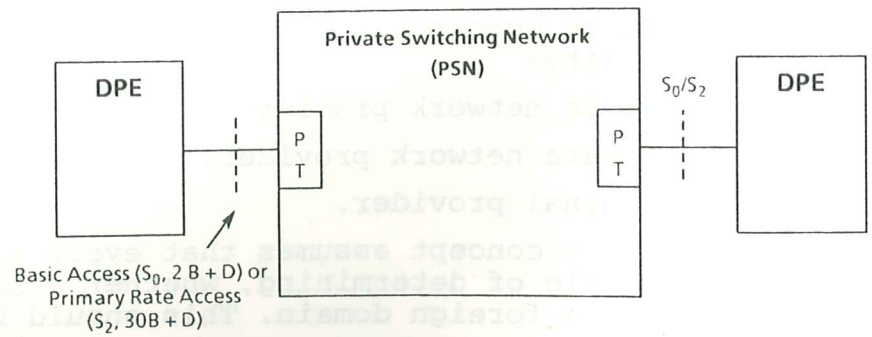


Figure 3 - Remote Maintenance Scenario

Maintenance of connections via intervening non-ISDN networks, i.e. involving Interworking Units (IWU), are beyond the scope of this Technical Report. However, they have been kept in mind to produce open ended solutions and not to preclude further studies on this topic.

3. REFERENCES

ECMA-102 : Rate Adaptation for the Support of Synchronous and Asynchronous Equipment Using the V-Series Type Interface on a Private Circuit Switching Network

ECMA-103 : Physical Layer at the Basic Access Interface between Data Processing Equipment and Private Circuit Switching Networks

ECMA-104 : Physical Layer at the Primary Rate Access Interface between Data Processing Equipment and Private Circuit Switching Networks

ECMA-105 : Data Link Layer Protocol for the D-Channel of the S-Interfaces between Data Processing Equipment and Private Circuit Switching Networks

ECMA-106 : Layer 3 Protocol for Signalling over the D-Channel at the S-Interfaces between Data Processing Equipment and Private Circuit Switching Networks

ECMA TR/24 : Interfaces between Data Processing Equipment and Private Automatic Branch Exchange, Circuit Switching Application

ECMA TR/ : OSI Management Framework (in preparation)

CCITT Rec. G.704 : Functional characteristics of interfaces associated with network nodes

CCITT Rec. G.732 : Characteristics of primary rate PCM multiplex equipment operating at 2048 Kbit/s

CCITT Rec. G.803 : Maintenance of digital networks

CCITT Rec. I.411 : ISDN user-network interfaces - reference configurations

CCITT Rec. M.20 : Maintenance philosophy for analogue, digital and mixed networks

CCITT Rec. V.54 : Loop test devices for modems

CCITT Rec. X.21 : Annex A: Interface signalling state diagrams

CCITT Rec. X.150 : DTE and DCE test loops for public data networks

CCITT Rec. X.409 : Presentation transfer syntax and notation

CCITT Rec. X.410 : Remote operations and reliable transfer services

CEPT Rec. T/CS 54-08 : Digital subscriber line maintenance

IEEE 802.1 (Draft) : Systems Management

ISO/TC97/SC21 N 383 : Fault management requirements

ISO/TC97/SC21 N 391 : Basic management framework

4. DEFINITIONS

4.1 Alarm

A signal, indicating a request for maintenance intervention. It is generated by a System Management Application as a consequence of an event in the system when the system has been disturbed or is in imminent danger of being disturbed so that its total performance is impossible or in danger.

4.2 Data Processing Equipment (DPE)

Specific type of terminal equipment, exclusively or mainly used to process data (in contrast to a voice-only terminal).

4.3 Error

A malfunction in the operation of a system.

4.4 Event

The occurrence of a significant normal or abnormal condition.

4.5 Failure

A failure may be caused by a fault depending on its severeness and the (non)-existence of redundancy.

4.6 Fault

The mechanical or algorithmic cause of a malfunction.

4.7 Maintenance Transaction

The execution of a basic step of the maintenance procedure. Such step cannot be sub-divided any further.

4.8 Private Circuit Switching Network (PCSN)

A PCSN provides circuit switching functions, being operated by the user and located on his premises to cover the communications needs in his domain. Terminal equipment is connected to a PCSN at its S reference points.

4.9 Private Packet Switching Network (PPSN)

A PPSN provides packet switching functions, being operated by the user and located on his premises to cover the communications needs in his domain. Terminal equipment is connected to a PPSN at its S reference points.

4.10 Private Switching Network (PSN)

A PSN provides switching functions (circuit and/or packet switching). It is operated by the user and located on his premises to cover the communications needs in his domain. Terminal equipment is connected to a PSN at its S reference points.

The term Private Switching Network includes both, the private circuit switching network and the private packet switching network.

4.11 PSN Termination (PT)

The PT is the termination of a PSN at the S reference point.

4.12 R Reference Point

In the ISDN user-to-network reference configuration, the R reference point indicates the interface between terminal equipment, not complying with the standards applying to an S reference point interfaces, and a Terminal Adaptor (see also there). It serves for indirect connection of non-ISDN terminal equipment to a PSN (see CCITT Rec. I.411). The interfaces at the R reference point include CCITT V- and X-series interfaces.

4.13 S Reference Point

In the ISDN user-to-network reference configuration, the S reference point is defined between the DPE and the PSN termination (see CCITT Rec. I.411). ECMA has identified two interfaces at the S reference point, the S_0 interface which provides 2B- and one D-channel (basic access) and the S_2 interface which provides 30B- and one D-channel (primary rate access).

4.14 S_0 Interface

The basic access interface at the S reference point (see CCITT Rec. I.411) operating at a physical bit rate of 192 kbit/s. It provides access to two B-channels and one D-channel (2B + D). The S_0 interface forms one of the user access points to a PSN.

4.15 S_1 Interface

The primary rate access interface at the S reference point (see CCITT Rec. I.411) operating at a physical bit rate of 1544 kbit/s. It provides access to 23 B-channels and one D-channel (23B + D). The S_1 is not used in Europe and no further reference will be made to it in this Technical Report.

4.16 S_2 Interface

The primary rate access interface at the S reference point (see CCITT Rec. I.411) operating at a physical bit rate of 2048 kbit)s. It provides access to 30 B-channels and one D-channel (30B + D). The S_2 interface forms one of the user access points to a PSN.

4.17 Significance

A definition or a procedure may have different ranges of application. Internal significance applies to the internal structure or an entity (DPE, PSN) and is not subject to standardization. Local significance applies to peer entities on each side of an interface, global significance applies to end-to-end configurations.

4.18 Terminal Equipment (TE)

Any terminal (voice or data processing or combination of both) connected to a PSN at the S_0 or at the S_2 interface (these are called TE-1) or, via a terminal adaptor, at the R reference point (these are called TE-2).

4.19 Terminal Adaptor (TA)

The Terminal Adaptor (function) is required to adapt from an interface at the R reference point (e.g. according to the V.- or X.-Series) to the interface at the S reference point. The Terminal Adaptor function is an integral functional entity as part of the terminal, while a Terminal Adaptor is a separate physical unit connected between an interface at the R reference point and the S interface.

5. SCENARIOS

5.1 Reference Configuration

The reference configuration is depicted in Figure 4. The conditions at the bus configuration of a basic access leads to the requirement to differentiate between DPEs directly connected to an S_0 reference point (also called DPE-1) and DPEs connected at an R reference point (V.-Series or X.-Series interface) which need a Terminal Adaptor function (TA) for conversion to the PSN's S_0 interface (these are called DPE-2).

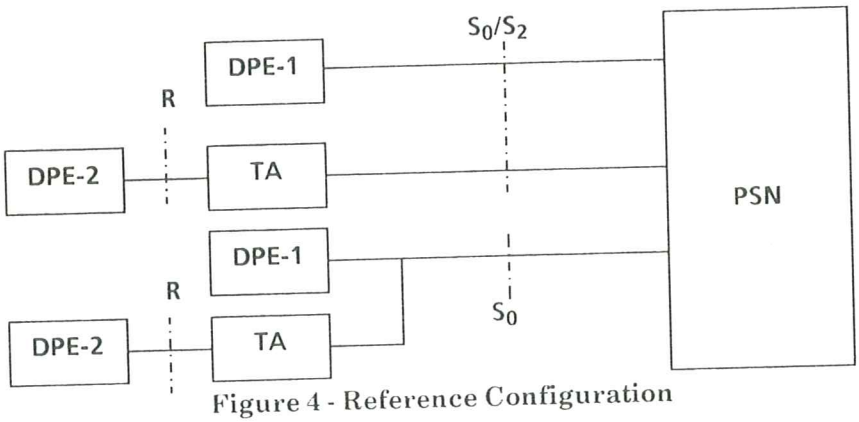


Figure 4 - Reference Configuration

The different configurations, point-to-point or multipoint, will also have an influence on the maintenance procedures.

Maintenance of the S_2 interface between a DPE-1 and the PSN or a TA and the PSN should be the same, although no multipoint configurations will occur.

Hereafter, the term DPE will be used as a generic term for either a DPE-1 or a combination of a DPE-2 and a TA.

5.2 Test Reference Points

A set of reference points may be conveniently used to identify and delimit the borders of different logical or physical domains to which faults can be isolated. CCITT Rec. V.54 and X.150 give a basic set of reference points. A reduced set of reference points (6 and 7 of CCITT Rec. V.54) is appropriate to the scope of this Technical Report, which concerns the S interfaces only. The reference points are shown in Figure 5. They should not be confused with the loop locations described in 5.3.3.

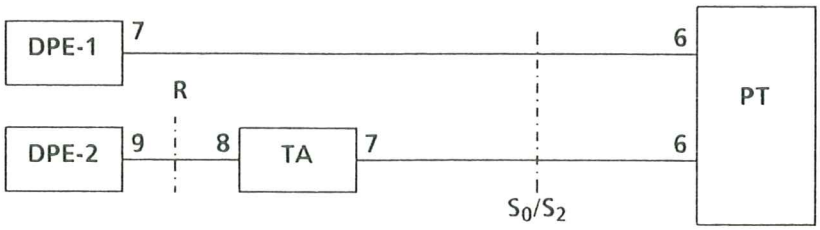


Figure 5 - Test Reference Points

NOTE 1

The interworking of maintenance functions between the R interface reference points 8 and 9 of CCITT Rec. V.54 (see Figure 5) and the S interfaces is not covered by this Technical Report, but is for further study.

5.3 Test Loops

The activation of test loops can provide an effective means to isolate faults. Careful consideration is needed to determine how many loops of what types and at what locations are sufficient and necessary for adequate localization of faults.

This section defines and names the types of loop and possible locations. Loops may be required within an established call or separately.

5.3.1 Loop Types

A number of loops are possible which can be used to reduce the scope of investigation and localize a fault. These loops may be grouped as follows.

5.3.1.1 Physical Loop

A physical loop is a Physical Layer mechanism which loops back all the information within a channel, a number of channels or a complete interface.

The use of the term "physical" is not related to any implementation since such a loop may be provided by means of active logic elements, metallic connections, etc. A physical loop may be partial, acting on the information of one or more channels, or it may act on the information of the whole interface.

5.3.1.2 Logical Loop

A logical loop acts selectively on certain information within a channel (for example in the D-channel). The use of a logical loop may result in some modification of the looped information.

5.3.1.3 Transparent Loop

"Transparent" may refer either to a physical loop or to a logical loop. In a transparent loop the information looped back may still be received by the equipment beyond the loop, see Figure 6a.

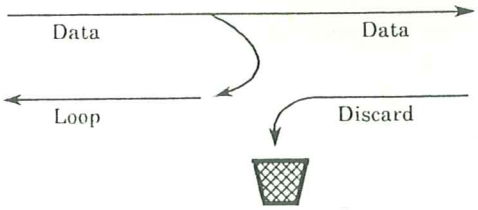


Figure 6a - Transparent Loop

5.3.1.4 Non-transparent Loop

"Non-transparent" may refer either to a physical loop or to a logical loop. In a non-transparent loop, some defined code could be sent to the equipment beyond the loop, as shown in Figure 6b.

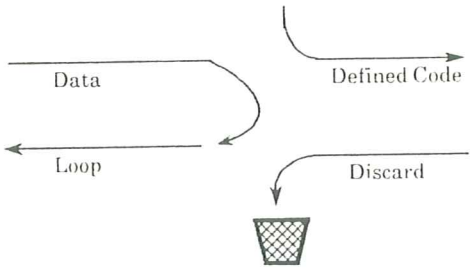


Figure 6b - Non-Transparent Loop

5.3.2 Functional Blocks

In order to determine possible loop positions it is necessary to break an interface down into its basic functional blocks, as shown in Figure 7.

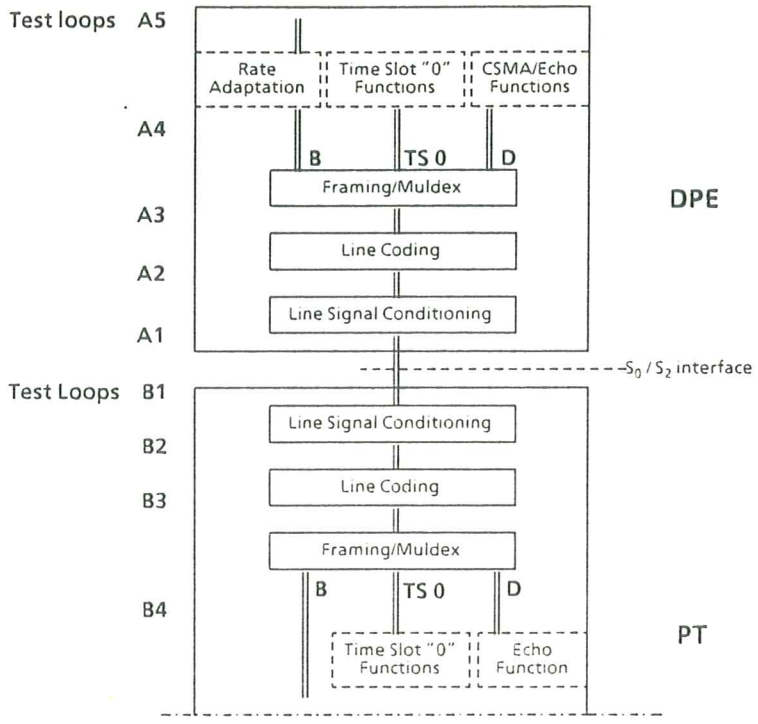


Figure 7 - Functional Blocks and Location of Test Loops

The blocks identified are appropriate for either the S_0 or the S_2 interface and for either the DPE or PSN side of the interface with some minor reservations, which are:

- On the PSN side of the interfaces (within a PT) the rate adaptation function is not required.
- The block on the D-channel labelled CSMA/ECHO will vary as follows:
  - . at the DPE side of an S_0 interface it carries out the CSMA contention function, see ECMA-103,
  - . at the PSN side of an S_0 interface it provides the D-channel echo function, see ECMA-103,
  - . at an S_2 interface its functions are null, see ECMA-104.
- The block labelled TS 0 is null in the case of the S_0 interface. In the case of the S_2 interface, it provides the time slot 0 functions as defined in ECMA-104.

5.3.3 Locations of Loops

The loops suggested by CCITT Rec. V.54 and X.150 were considered but found to be inappropriate for ISDN interfaces. Therefore, a new notation as shown in Figure 7 has been defined for loops at reference points 6 and 7 of Figure 5.

- The DPE side of the interface is defined as the A-side.
- The PSN side is defined as the B-side.

Loops are numbered from 1 on upwards as they move away from the S interface point.

The test loops to be provided are shown in Table 1.

Test Loop	Direction of reflected signal	S_0 Interface	S_2 Interface
A 5 (B channel only)	↓	Mandatory	Mandatory
A 4 (B channel only)	↓	Mandatory	Mandatory
A 3	↓	–	Optional
A 2	↓	–	–
A 1a	↓	Not applicable	Optional
A 1b	↑	Not applicable	Optional
B 1	↑	Not applicable	Recommended
B 2	↑	–	–
B 3	↑	Not applicable	Optional
B 4	↑	Optional	Optional

" – " = beyond the scope of this Technical Report

Table 1 - Provision of Test Loops

Loop A5 is mandatory when, in the case of bi-directional transmission, rate adaptation is present. It provides a loop back of data after the rate adaptation function on any channel or number of channels.

Loop A4 is mandatory. It applies to the S_0 and to the S_2 interface and provides loop back of any channel or number of channels towards the PSN. The loop shall preserve octet integrity. It shall reflect any bit pattern received at the earliest possible time.

Loop A3 is optional and applies to the S_2 interface only. When provided, it shall loop back any bit pattern received between its line coding functions (HDB3) and its framing/muldex functions. When loop A3 can be activated via the S_2 interface, provisions shall be taken to allow also its deactivation, e.g. by means of a timer.

Loop A1a and A1b are optional and apply to the S_2 interface only. When provided, they shall loop back any bit pattern received directly at the S_2 interface (i.e. the pulse shape, see CCITT Rec. G.703, is not regenerated; this should be taken into account when using this loop for measurement purposes). If Loop A1a can be activated via the S_2 interface, provisions shall be taken to allow also its deactivation, e.g. by means of a timer.

Loop B1 applies to the S_2 interface only. Whether or not Loop B1 is provided, depends on the implementation of the specific PSN. When provided, it will loop back any bit pattern received directly at the S_2 interface (i.e. the pulse shape, see CCITT Rec. G.703, is not regenerated; this should be taken into account when using this loop for measurement purposes). If activated via the S_2 interface, provisions shall be taken to allow also its deactivation, e.g. by means of a timer.

Loop B3 applies to the S_2 interface only. Whether or not Loop B3 is provided, depends on the implementation of the specific PSN. When provided, it will loop back any bit pattern received between its line coding functions (HDB3) and

its framing/muldex functions. If Loop B3 can be activated via the S_2 interface, provisions shall be taken to allow also its deactivation, e.g. by means of a timer.

Loop B4 applies to the S_0 and to the S_2 interface. Whether or not Loop B4 is provided, depends on the implementation of the specific PSN. When provided, it will loop back the signals of any B-channel or any of a number of B-channels towards the DPE. The loop shall preserve octet sequence integrity. It shall reflect any bit pattern received at the earliest possible time.

Loops A1 to A4 and B1 to B4 are physical loops while loop A5 is a logical loop.

It should be remembered that the establishment of a data link on the D-channel implicitly tests many of the functions illustrated.

6. ASSUMPTIONS AND REQUIREMENTS

6.1 Assumptions

This Technical Report addresses the maintenance and testing of DPE to PSN interfaces from its principal point of view. It does not address other functions of the PSN or of the DPE, as it is assumed that they will have their own test procedures and facilities.

Furthermore, it is assumed that the PSN will provide at least minimum facilities required for tests originating at the DPE and vice versa.

6.2 Requirements

The general requirements for maintenance of DPE or PSN interfaces are:

- Assisting with localization of faults shall at least be possible to one or the other side of the interface.
- Preventing unauthorized access to maintenance features shall be possible in order to avoid abuse.
- Maintenance procedures should be symmetrical although they may not always be used symmetrically.
- It shall be possible to abandon a routine test call to allow a normal call to proceed. However, it should be possible also to clear a call on exceptional maintenance procedures.
- The use of maintenance features within an established call shall be possible.
- Remote activation of tests shall be possible. The cooperation of the PSN may be necessary in this case.
- Communication capabilities shall be provided between corresponding systems for the exchange of management information.
- Some test procedures are more important than others, therefore a minimum set of mandatory procedures shall be defined.
- A response should be returned to any request from a remote system.

7. MANAGEMENT FRAMEWORK

Maintenance of interfaces is part of the general management process. It is intended that management of DPE to PSN interfaces should follow the principles of Open Systems Management.

System management in OSI environment is achieved through a set of application processes running on different open systems, which communicate with each other and play complementary roles in order to provide management activities. Communication rules are defined by system management protocols which are subject of the standards to be developed.

Three main reasons indicate the layered approach as the right one:

- According to OSI principles a maintenance activity is better located at the application layer. This does not preclude Layer Management Entities (LME) achieving locally significant activity.
- The need for remote maintenance (either maintenance of a remote interface by a DPE or global maintenance of a PSN interfaces by a Maintenance Centre Unit (MCU) identifies an addressing problem that should be provided by a Network Layer service independently of the maintenance activity itself.
- The need to execute maintenance activities in a stand-alone mode or as part of an established call indicates that basic call control procedures should be used as a basis on which to build on additional features.

7.1 General

Within a DPE, system management functions are controlled and performed by the System Management (SM). This entity receives, analyses and makes decisions out of the information generated locally in the Layer Management Entities (LMEs) or received from a remote SM. An Open System Management Application Process OSMAP will communicate with a remote OSMAP by use of an Open System Management Application Entity (OSMAE), see Figure 8 and the ECMA TR on OSI Management Framework.

The layer-specific management functions are provided for each layer by the respective Layer Management Entity (LME). The LME is a component of the layer which effects control of the communication functions of the layer provided by the Protocol Entity (PE). An LME is logically interfaced to the System Management by its Layer Management Interface (LMI).

This Technical Report particularly describes an OSMAE dedicated to maintenance activities. The OSMAE is interfaced:

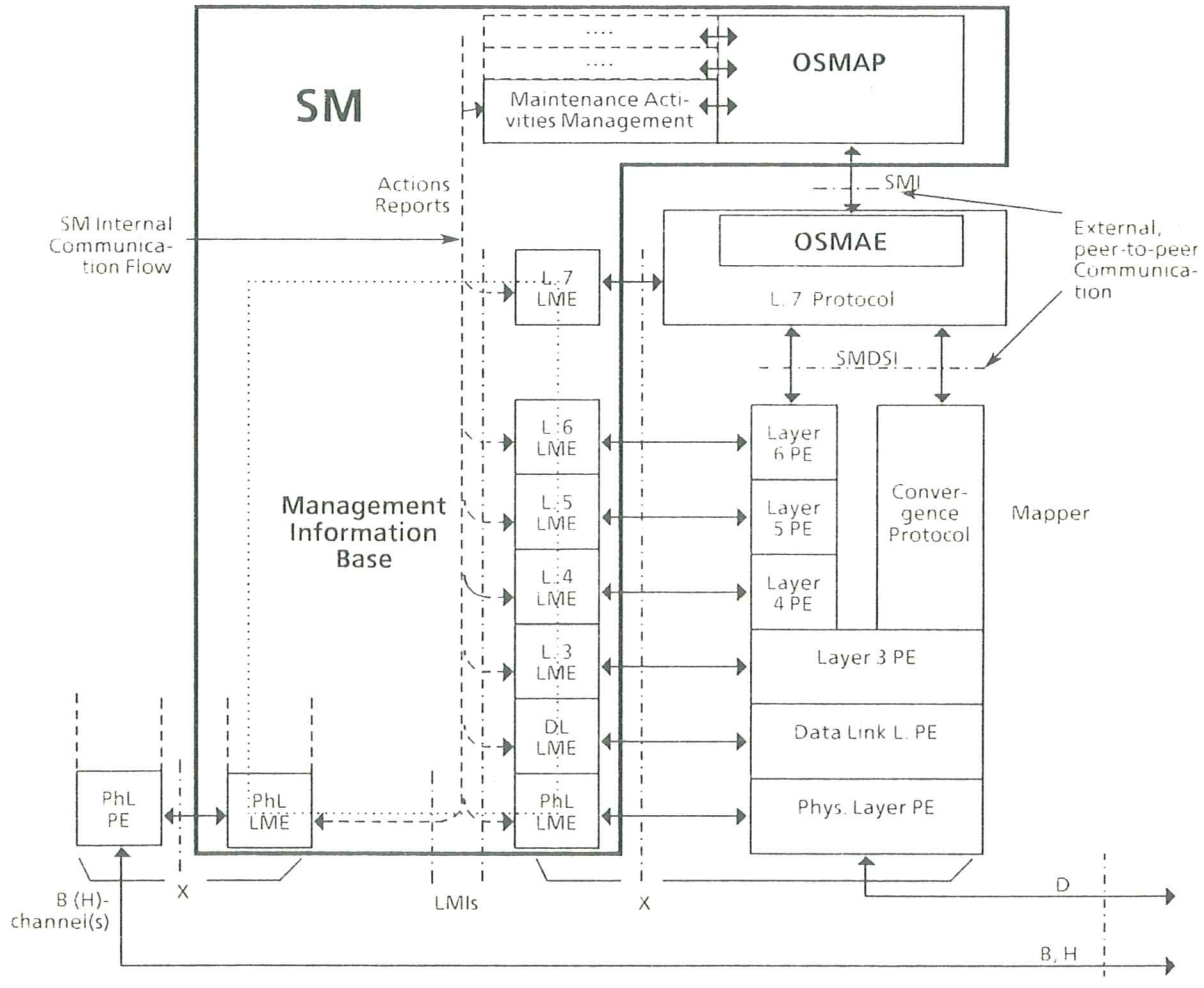
- to the communication channel via the System Management Data Service Interface (SMDSI), and
- to the System Management Entity via the System Management Interface (SMI).

Although only the SMDSI and the SMI services needed to be described for defining the maintenance protocol and Maintenance Protocol Data Units (MPDUs), a tutorial presentation of the services offered at the LMI is added to enable a better overview of the complete execution of a maintenance transaction. Whether or not Layers 4 to 6 are to be provided in an ISDN environment, is to be studied. A convergence protocol, mapping from the SMDSI directly to Layer 3, could be used instead. Its need and definition depends on the outcome of CCITT studies.

NOTE 2

In the context of describing the services at the LMI and SMI, the term "interface" is not used in the OSI sense. The LMI and SMI are to be understood as conceptual reference points where maintenance services are provided by the SM.

Figure 8 illustrates the major components of and the embedment of the OSMAE in the architecture described in this Section.



X = D, B or H-channel protocol stack with its layer internal interfaces, included in layer specifications

Figure 8 - Management Architecture for Maintenance Application

In the following subsections the services are described in an abstract way. The description does not imply any particular implementation nor exposed interface.

7.2 System Management Data Service Interface (SMDSI)

The SMDSI is the service interface for the exchange of protocol data units between the PEs and the OSMAE. This interface is used by the OSMAE

- to pass system management protocol data units (SM_PDUs) to signalling protocol entities for transmission to remote OSMAEs, and
- to receive incoming SM_PDUs from remote OSMAEs.

Protocol Data Unit (PDU) transfer services are required from the layer Protocol Entities by the Open System Management Application Entity (OSMAE).

Interactions: The following primitives are defined for the OSMAE to request service from the PEs:

- SM_DATA.request/SM_DATA.response
- SM_DATA.indication/SM_DATA.confirm

The SM_DATA.request and .response primitives are passed to a layer PE to request that a System Management Protocol Data Unit /SM_PDU) be sent. The SM_DATA.indication and SM_DATA.confirm primitives are passed from a layer PE to indicate the arrival of a SM_PDU from a remote OSMAE.

7.3 Layer Management Interface (LMI)

The LMI is the service interface between the LMEs and the System Management SM. It is used

- by the SM to make management requests of the particular layer, and
- by the layer to inform the SM when events occur.

The precise nature of the LMI depends on the layer being managed and is specified in detail in the layer specification. However, for tutorial purposes the set of generic primitives are shown in Table 2.

Primitive Name	Request	Confirm	Indication
LM_SET_VALUE	X	X	-
LM_COMPARE_AND_SET_VALUE*)	X	X	-
LM_READ_VALUE	X	X	-
LM_ACTION	X	X	-
LM_EVENT	-	-	X

*) = see text below

Table 2 - Primitive Types and Contents at the LMI

The services provided by the LMEs to the SM by means of these primitives are specified hereafter. Due to the internal significance of the LMIs, the definition of protocols related to these interfaces is beyond the scope of this Technical Report.

The LM_SET_VALUE.request primitive is passed from the SM to the LME to set a new value for a layer operational parameter. The LM-SET-VALUE.confirm primitive is passed from the LME to the SM to return the success or failure status for the associated request.

The LM_COMPARE_AND_SET_VALUE.request primitive is passed from the SM to the LME to verify that the current value of a layer operational parameter is equal to its expected value, and to set a new value for the corresponding layer operational parameter. The LM_COMPARE_AND_SET_VALUE.confirm primitive is passed from the LME to the SM to return the success or failure status for the associated request. The need for and the use of this primitive is to be studied.

The LM_READ_VALUE.request primitive is passed from the OSMAE to the LME to request the current value of a layer operational parameter. The LM_READ_VALUE.confirm primitive is passed from the LME to the SM to return the value of the requested layer operational parameter or a failure status for the associated request.

The LM_ACTION.request primitive is passed from the OSMAE to the LME to generate an action or state transition within a target layer. The LM_ACTION.confirm primitive is passed from the LME to the SM to return the success or failure statues for the associated request.

The LM_EVENT.indication primitive is passed from the LME to the SM to indicate that a significant event has occurred, e.g. a fault or the change of the value of a significant layer parameter.

7.4 System Management Interface (SMI)

The SMI is the service interface between the System Management and the Open System Maintenance Application Entity (OSMAE). The two entities exchange primitives in order to allow remote maintenance operations. This Section provides a set of generic primitives whose use in a maintenance context is described in Section 9. The generic service primitives are listed in Table 3.

Primitive Name	Request	Confirm	Indication	Response
SM_SET_ATTRIBUTE	X	X	X	X
SM_COMPARE_AND_SET_ATTRIBUTE	X	X	X	X
SM_READ_ATTRIBUTE	X	X	X	X
SM_ACTION_ATTRIBUTE	X	X	X	X
SM_M_REPORT	X	X	X	X

Table 3 - Primitive Types and Contents at the SMI

The SM_SET_ATTRIBUTE service provides the capability of requesting the setting of one or more management attributes in a peer system.

The SM_COMPARE_AND_SET_ATTRIBUTE service provides a Manager with the capability of requesting the comparing and conditional resetting of one or more management attributes by an LSM in a peer system. The need for and the use of this primitive are to be studied.

The SM_READ_ATTRIBUTE service provides the capability of requesting the reading of one or more management attributes in a peer system.

The SM_ACTION_ATTRIBUTE service provides the capability of requesting actions to be performed in a peer system. This service provides a mechanism for initiating actions, or causing state transitions within a layer of a remote system.

The SM_M_REPORT service element is used to transmit event information from a monitoring management application across the SMI. It is an optionally confirmed service element. The SM_M_REPORT.response and SM_M_REPORT.confirm primitives may or may not be issued depending on whether acknowledged service is used.

A set of parameters can be defined that will be used in relation with the above described primitives. The use of these parameters may be optional depending on the primitive. Examples are given in Table 4.

Parameter Name *Description*	**Primitives**									
	SET		COMPARE		READ		ACTION		M_REPORT	
	R/I	R/C	R/I	R/C	R/I	R/C	R/I	R/C	R/I	R/C
Originating Manager *The SM initiating the request*	O	O	O	O	O	O	O	O	R	
Destination Manager *The SM to which the message is sent*	R	R	R	R	R	R	R	R	R	R
Target Layer *The recipient Layer Management Entity*	R	R	R	R	R	R	R	R		
Access Control *Defined by the target SM or LME*	O		O		O		O			
Resource Identifier *Identifies the resource affected*	R	O	R	O	R	O	R	O	R	
Attribute List *Identifies attributes and their values*	R		R		R		R			
Status *Global status returned in the response*		R		R		R		R		R
Action Report List *Returned data or statuses of actions performed*		R		R		R		R		
Day and Time *Day / time of event occurence*									R	
Event Type *Event definition*									R	
Event Report *Additional information on the event*									O	
Report Acknowledgement *Acknowledgement is required*									O/R	O/R
Report Reference *Provided by the originator*									O/R	O/R

Legend: *R/I = Request / Indication Primitives*
R/C = Response / Confirm Primitives
O = Optional
R = Required
O/R = Optional, but required in R/C when provided in R/I

Table 4 - Examples of Parameters associated with Generic Primitives

7.5 Examples

The maintenance application processes have the overall control on the maintenance transactions. They will manage the resources under test and the end-to-end (peer-to-peer) dialogue required for the control of the transaction. The layers under test depend on the connection of the bearer service and the channel involved. Figure 9 shows as an example the use of the model for loop establishment.

The detection of a error and the transmission of an alarm to a remote entity would follow the path described in Figure 10.

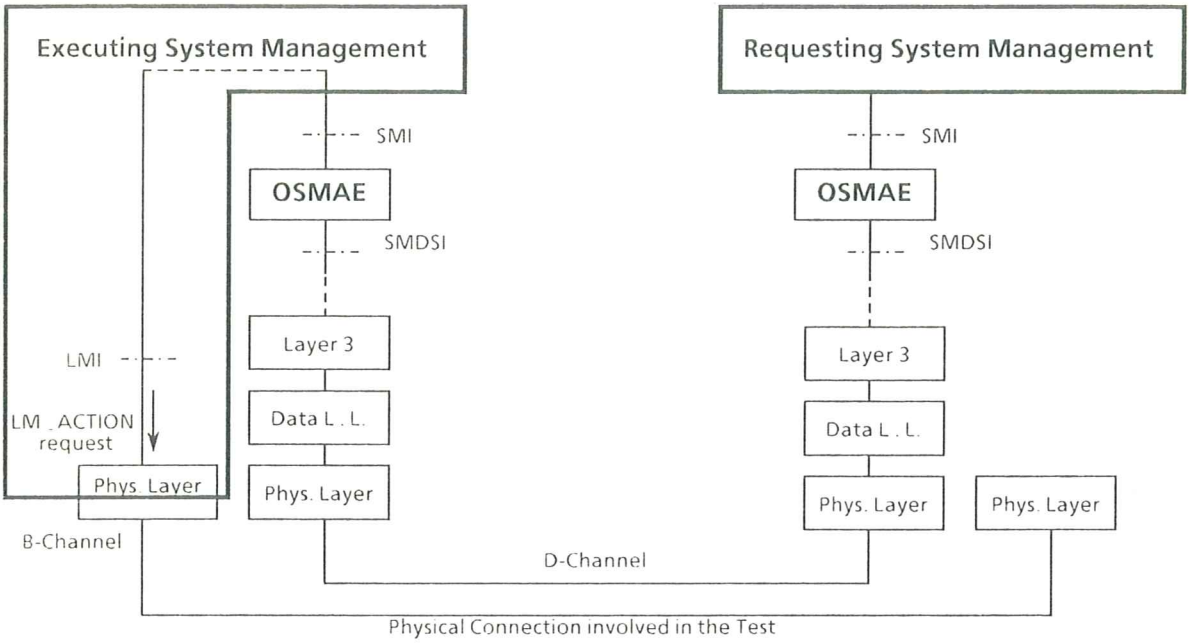


Figure 9 - Example for a Maintenance Transaction: Loop Establishment

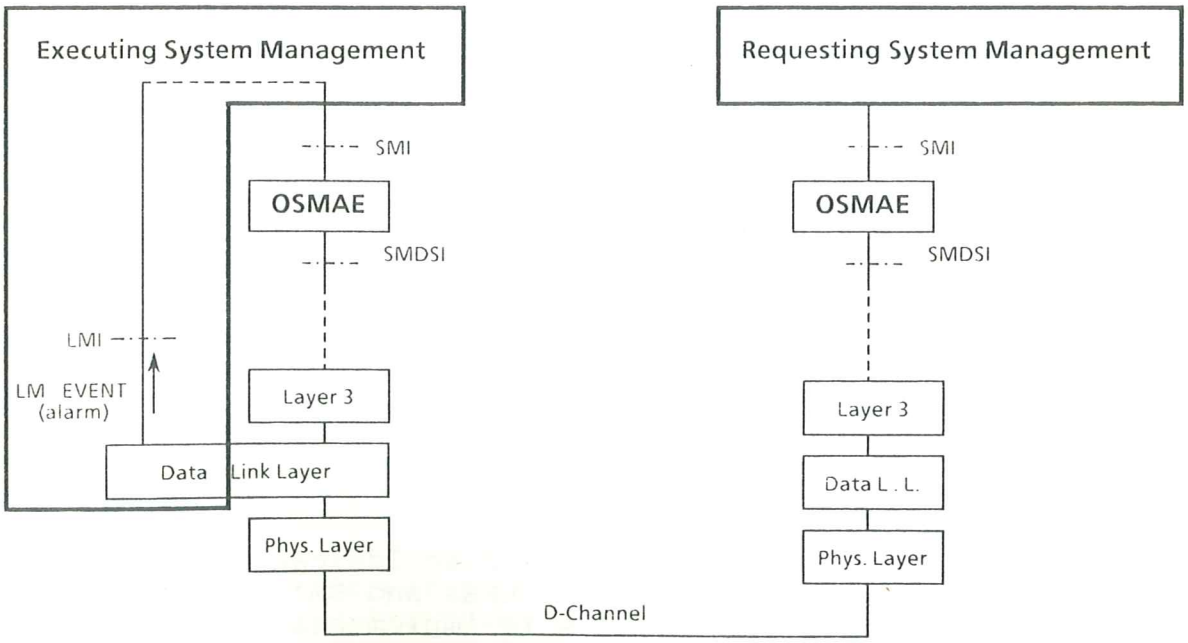


Figure 10 - Example for a Maintenance Transaction: Layer 2 Remote Alarm

8. ERROR PROCESSING

Error processing can be divided into four phases, briefly described in the following.

Detection of an error will normally be the result of continuous supervision or as a consequence of an on-demand or routine test.

Confirmation of an error consists of repetition, where possible, of the circumstances in which the error occurred or similar circumstances which result in the error being repeated.

Reporting of errors occurs either on detection or confirmation, depending on the nature of the fault.

Localization takes place when it is not yet clear in which domain the fault lies.

8.1 Error Detection

Errors can be detected in different ways:

- by an entity during normal operation, generating an error report (automatic supervision);
- while running confidence tests (on-demand tests and measurements or automatic routing tests);
- by thresholds being exceeded.

Elements of all three mechanisms may be involved in the maintenance of a single link.

Automatic error detection will be looked for by creating artificial activity on the D-channel, when there is no normal traffic, and by periodically or systematically comparing certain counters to predetermined error thresholds. The additional activity should not exceed certain limits to be defined.

Activation of test loops via maintenance procedures will also help in detecting errors.

8.1.1 Automatic Supervision

8.1.1.1 Principles of Supervision

Detection of errors may be done by monitoring the interface activity. The activity can be described by a set of counters and statuses that are monitored on a continuous basis or on a per call basis.

The supervision of the activity (or non-activity) will also make use of timers. These will create events (timeouts) on which counters will be updated as appropriate.

The statuses and counters have two purposes. They can be retrieved at any time for information and statistics gathering. They can also be periodically or systematically compared to preset thresholds as described in 8.1.1.2.

8.1.1.2 Error Counters and Thresholds

A number of counters and thresholds can be identified for each entity (i.e. the LMEs and SM). The actual choice of these elements is implementation dependent.

When the elements have to be monitored and/or managed from a remote entity, accurate descriptions are required in order to avoid any ambiguity. Vague definitions could lead to divergent values or interpretations by different vendors of the same element used in peer entities (e.g. the use of interpolation or extrapolation in the coupling of time-dependent counters may lead to significant differences).

Examples of counters can be found in Appendix B.

8.1.1.2.1 Error counter information element

The error counter information element provides a value which represents the number of occurrences of a particular error over a specified period of time.

Error counters can either "wrap" or "latch" when they reach their maximum value. If they latch, they require a mechanism to be reset; if they wrap, they need not have such mechanism. An error counter is identified by an entity identifier (e.g. the LME indication) and an error type; when the counter is read, information on its running time shall be associated.

Counts will not inherently be changed by the Physical Layer ACTIVATION/DEACTIVATION procedure.

8.1.1.2.2 Aggregate error information element

The aggregate error information element provides similar information to that of an error counter, except that it contains accumulated occurrences of a set of error types instead of just a single error type.

Therefore, the aggregate identification must include a list of possible error types. The aggregate value must also contain a list of which errors have actually occurred since the aggregate has been initialized. Due to this list keeping, there is a need to provide a mechanism to initialize the aggregate and the lists; furthermore, aggregates will need to "latch" at their maximum value rather than being allowed to "wrap" like simple error counters may do.

8.1.1.2.3 Alarm information element

The alarm information element will be used to indicate that an alarm condition has been entered or that restoration to normal operating conditions has taken place. An alarm is given as the result of a threshold being exceeded within a given, presettable period of time (e.g. threshold of the normal or the error rate).

A threshold is identified by its type and the type of the counter or aggregate that exceeded that threshold.

A mechanism to change the threshold is required.

8.1.2 Automatic Routine Tests

These tests will be based on a systematic or periodic exchange of information by peer entities. They are characterized by their low level of priority and are run as a background activity, see 8.5, in order not to cause any distur-

bances. Particular examples are the automatic loops on the D-channel. These loops will make use of spare time on the channels to create artificial activity.

The definition of the parameters (duration, frequency, message size, etc.) will be a trade-off between not disturbing the system and early error detection.

8.1.2.1 Physical Layer

The use of the D-Echo bit on the D-channel for the basic access is an already existing mechanism that will prove correct functioning at the Physical Layer of the DPE to PSN interface, see ECMA-103.

The primary rate access case is to be studied.

8.1.2.2 Data Link Layer of the D-channel

When there is no LAP established, data link establishment or the exchange of UI frames will provide an implicit test of the correct functioning of a significant part of the interfaces for both basic and primary rate access.

On an established data link, the status enquiry function of LAPD provides an already existing test mechanism, see ECMA-105.

8.1.2.3 Loops at higher Layers on the D-channel

The specific development of an automatic test mechanism as part of Layer 3 is not required. Rather, the periodic exchange of information between OSMAEs is recommended, which is, however, implementation dependent.

8.1.2.4 Loops on B-channels

Similarly, an OSMAE will have the capability to require the periodic establishment of short dummy calls involving B-channels. These will allow the test of a complete information chain including hardware and software. However, due to possible external constraints (e.g. tariffs) their use is not subject to standardization.

8.1.3 On-demand Routine Tests

These tests are characterized by the fact that they are not run in a systematic or periodic manner. They may have a high level of priority, see 9.5. They may involve the immobilization of resources.

On-demand routine tests could be based on the same procedures as are employed for automatic test routines.

8.2 Error Confirmation

Whenever alarms or suspect counters reveal possible malfunctions of the equipment, the first step will be to confirm the existence of an error. Two procedures may be involved: analysis of the existing counters (local and remote) and creation of additional activity on the suspected link in order to reproduce (when possible) the problem or refine and complement the available information. The counters already defined may be used for the first phase and the tools available for on-line

diagnostic can be used. Error confirmation could also be called a first fault localization process.

8.3 Error Reporting

8.3.1 Overview and General Characteristics

Layer Entities must provide management information elements on errors which they detect and report to their SM. As shown in Figure 11, a System Management can

i) receive error indications from entities within its own system;

ii) notify the error status to the system administration of its system;

iii) produce reports on detected errors to be sent to other SMs residing in remote systems.

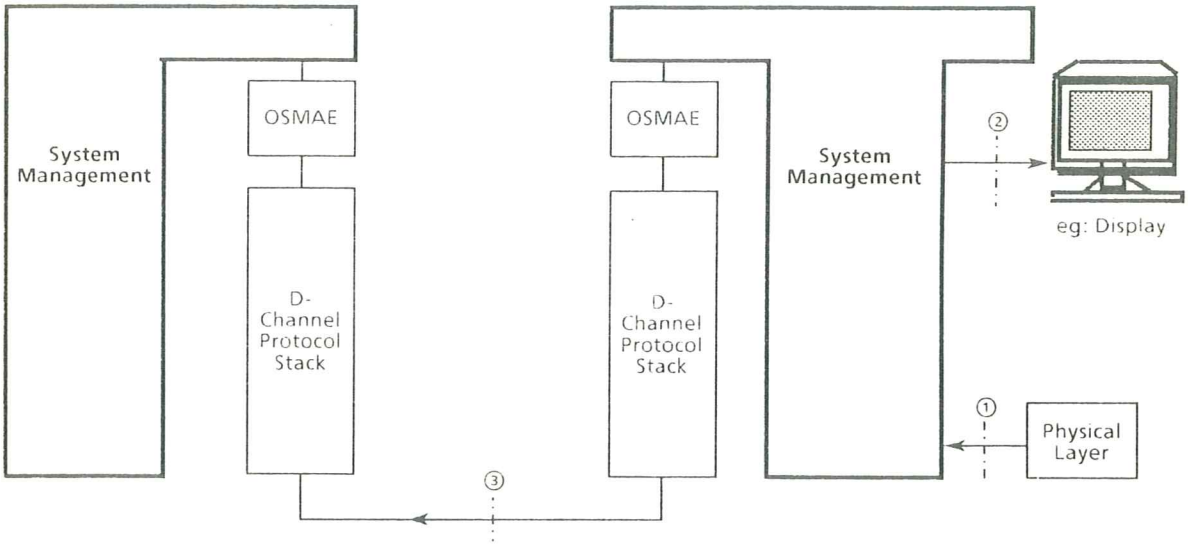


Figure 11 - Error Reporting (Example)

Only protocols for the support of item iii) are a matter for standardization. Information exchange at items i) and ii) is implementation dependent and, therefore, not a subject for standardization.

To achieve item iii) a DPE will exchange messages with its peer via the D-channel. These messages belong the the event report category described in 7.3.

8.3.2 Reporting

Several pieces of information are necessary to completely describe the occurrence of an error. The identity of the entity in which it was detected is needed to determine the responsible domain. The time of its occurrence identifies it against other errors of the same type. Further information may be required for other maintenance purposes.

Two mechanisms may be used to obtain counter and aggregate values. Either the system desiring the information may periodically poll the system keeping it, or thresholds may be set so that reports will be generated whenever a threshold value is reached or exceeded.

8.3.3 Alarms

Alarm reporting shall be based on event reporting mechanisms. It will be possible to report alarms or the restoration to normal operating conditions to remote SMs.

The possible repetition of alarms is under the responsibility of the System Management and is context dependent.

8.4 Fault Localization

Whenever an error has been detected (e.g. as a result of statistical analysis, on reaching a threshold...) the use of diagnostic procedures will assist in fault localization, i.e. in the determination of the smallest replaceable unit, so that the appropriate repair actions can be taken.

This can be done by analyzing symptoms (past events) or by using diagnostic procedures. These procedures will be based on any available test mechanism. They will allow a hierarchical approach in order to reduce the suspected domain.

The analysis of event logs in order to deduce the cause of an error from previous activities is called symptom-directed diagnostics. Artificial intelligence techniques may be used to assist in the analysis; however, this is beyond the scope of this Technical Report.

This document describes test procedures that will be used to diagnose a problem located at the DPE to PSN interface (up to and including TA functions when existing). Several types of procedures may be used depending on their availability:

- Basic call control procedures used as tests (always available).
- Test loops (some of them are always available, as described in 5.3).
- Self-tests (always optional).

The fault isolation is performed by - as far as necessary - execution of the following procedures:

i) Identification of an appropriate test domain.

ii) Selection of a test sequence.

iii) Execution of the test.

iv) Determination of whether the faulty part is in the domain tested.

v) Re-definition of the test domain and resumption at i).

The context of the error may be of interest; it should be maintained when possible and practical.

8.4.1 Call Control Procedures

Call control procedures can be used in a suspected error situation to regenerate the error or to recreate the conditions causing the error (load test, configuration test).

8.4.2 Test Loops

The test loops that will be used are described in 5.3. In order to adopt a hierarchical approach, the loops will be possible at different levels. Activation/deactivation of loops will be controlled by the requestor via the signalling functions of the D-channel. Test loops will be provided by both sides of the interface, and a minimum support of the tested entity, the executor, will be required.

Test messages will be transmitted by the requestor to the executor and the returned messages will be analyzed. Any test pattern may be used; the selection of these patterns is under the control of the executor.

A number of different possibilities exists:

- Testing the signalling functionality is possible by an exchange of messages at Layer 3 between peer entities. This will actually allow a logical loop on the D-channel as described in Figure 12.
- Testing the rate adaptation mechanism requires an A5 test loop as described in Figure 13. This logical loop will echo the information (synchronous or asynchronous) at a data rate lower than 64 kbit/s. In the case of universal terminal adaptors, the data rate may be changed as part of the test. It may also be the case that the data rate received is different from the data rate transmitted, e.g. when the transmit data rate is lower than or equal to the receive data rate, or when flow control is used.
- Test of the transmitter/receiver parts will be done by echoing the 64 kbit/s information on one B-channel by means of A4 test loops (see Figure 14).
- On the primary rate access A1 or A3 type loops will allow tests by exclusion, as indicated in Figure 15, by involving as little hardware as possible. This loop will echo all bits of a frame.

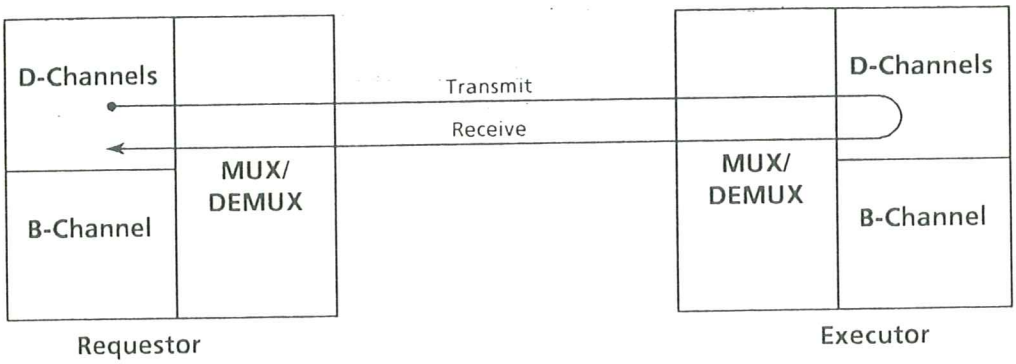


Figure 12: Logical Loop on the D-Channel

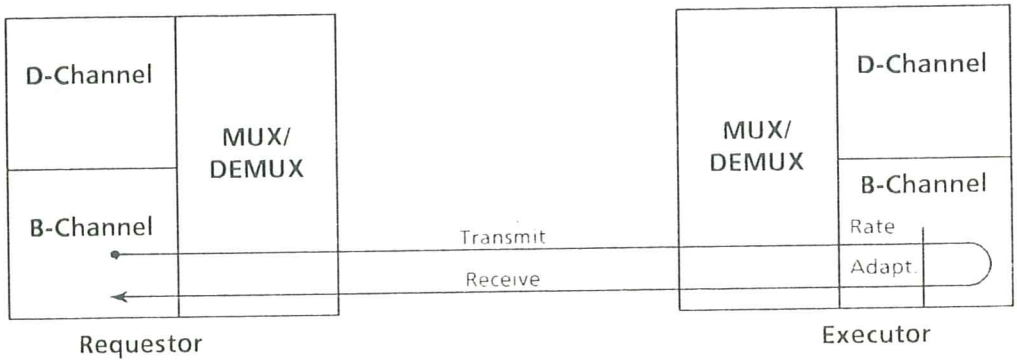


Figure 13: Logical Loop on one B-Channel after Rate Adaptation (Loop A5)

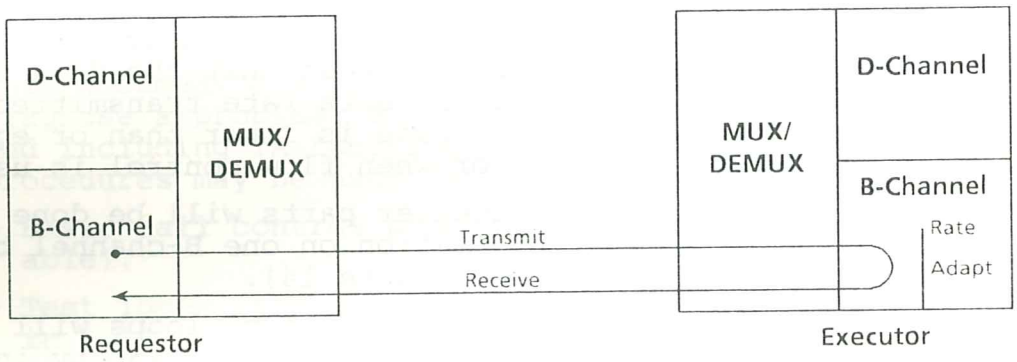


Figure 14: Loop on one B-Channel before Rate Adaptation (Loop A4)

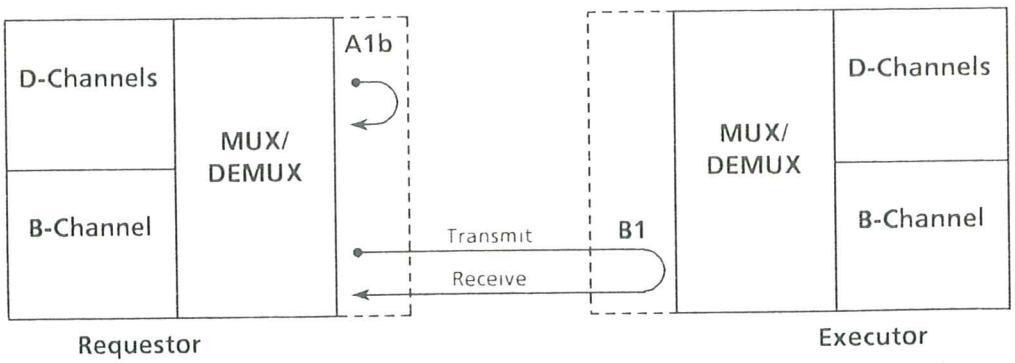


Figure 15: Loop on the Primary Rate Access (Test by Exclusion)

- The whole interface could be exercized by looping simultaneously all B-channels with A4 test loops, as described in Figure 16. This sort of tests may be used to simulate loaded environment and track faults that could be load dependent.

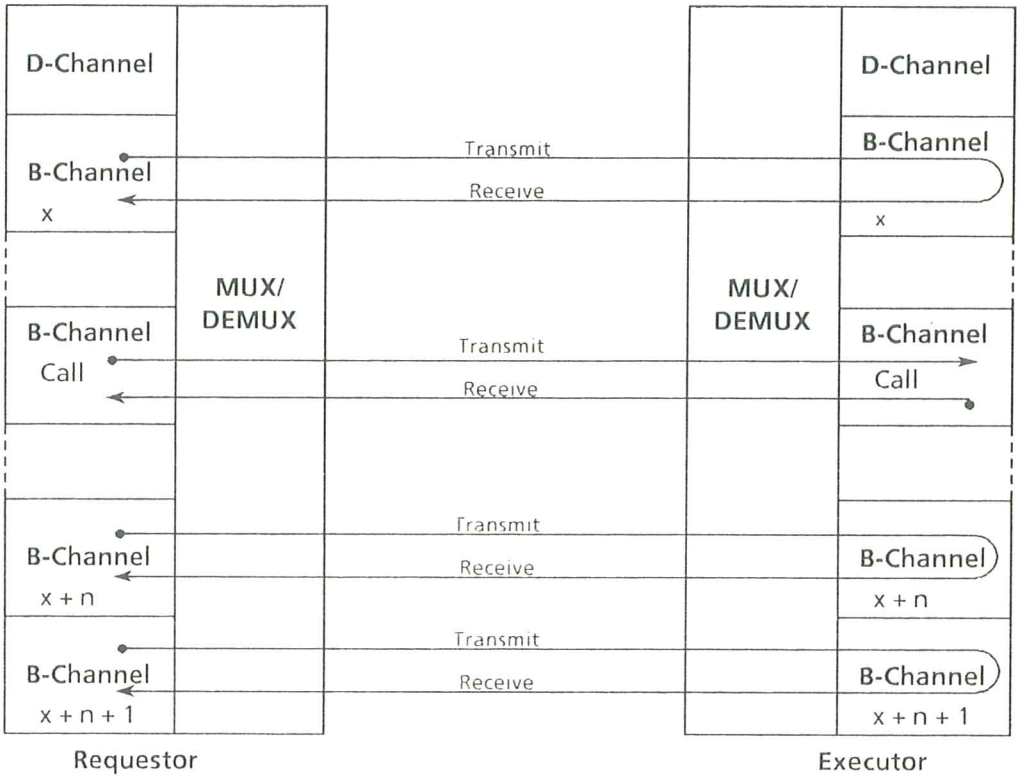


Figure 16: Loops on Idle B-Channels

Once the loop is achieved, a timer value will always be associated with the loop test in order to avoid deadlock situations.

It must be noted that the tests may be required on out-of-service equipment for the following reasons:

- The loop requestor does not necessarily control the status of the resources of a remote entity.
- Suspect equipment may have been declared out-of-service before the fault is actually localized.
- Diagnostic procedures can be used to test a piece of equipment before it is declared valid again.

The System Management will maintain channel and interface statuses for the purpose, see 9.5.

On deactivation of the Physical Layer, all test loops shall be released.

8.4.3 Self-tests

It may be the case that the DPE side of the interface supports certain self-tests; this fact is not necessarily known by the PSN side. A remote entity that has the ability to identify the configuration should be capable of activating this facility.

Subelements of the faulty interface may be tested separately by means of specific self-tests (e.g. rate adaptation mechanism).

Only self-testing of the communication functions associated with the S interface functions is within the scope of this Technical Report. The objective is to test out the DPE to the nearest possible point to the S interface and to report the results across the link.

8.5 System Management Action

On detection of an error different actions can be performed by the System Management:

- clear down the connection immediately;
- wait; if no response to indication within a given period, clear down the connection;
- leave the connection in service.

A similar set of actions applies to errors affecting a number of channels, e.g. on an S_2 link.

In any case, the result may be that the resources associated with a connection are put in the MAINTENANCE state, where they are barred against further access by the call control process, see 9.5. They remain accessible only for the SM in order to get accessed by maintenance procedures. It will be the responsibility of the SM to put them IN SERVICE or OUT-OF-SERVICE depending on the result of the maintenance procedures.

9. MECHANISMS

9.1 Maintenance as a Local Activity (Layer Management Entity)

In each DPE, a number of mechanisms will be implemented as part of Layer Management Entities (LMEs) in order to cope with the activities that are relevant to maintenance. Basically, this means that in each LME a number of counters, thresholds, alarm generators, etc. will be implemented and will be under the control of the DPE System Management (i.e. counters, and statuses will be readable and modifiable by the SM).

9.2 Layered Approach

In order to allow remote activation of maintenance facilities in a stand-alone way or as part of a call, a layered approach is employed, see Section 7. This means that the exchange of maintenance information will in principle be located above Layer 3. In practice, the maintenance information units will be enveloped in Layer 3 messages. Some reasons for the selection of this approach are:

- Maintenance activities that do not imply a B-channel.

  A normal user-to-user temporary signalling connection (with "no B-channel" indication) will be established and the available mechanism for user-to-user signalling will be used.

- Maintenance activities that imply (a) B-channel(s).

  A call is established prior to the initiation of maintenance activities. A potential B-channel selection conflict will be solved using pre-established priority rules specific to maintenance applications.

- Maintenance activities as part of an established call.

  In this case, the first phase is already established when maintenance procedures are invoked. The only difference with previous cases is the context which is normally under SM control.

- Maintenance activities between a DPE and a Maintenance Centre.

  Basic call control procedures will be used to access a Maintenance Centre Unit (MCU) that may be outside or inside the PSN. When the connection to the MCU is established, layered activities are initiated.

- Maintenance activities involving resources in the MAINTENANCE state.

  A maintenance call will be established with by-pass privileges (see 9.5.2) and the procedures will be activated once the call is established.

9.3 Local Versus Remote Testing

9.3.1 General

Figure 17 illustrates the difference between local and remote testing.

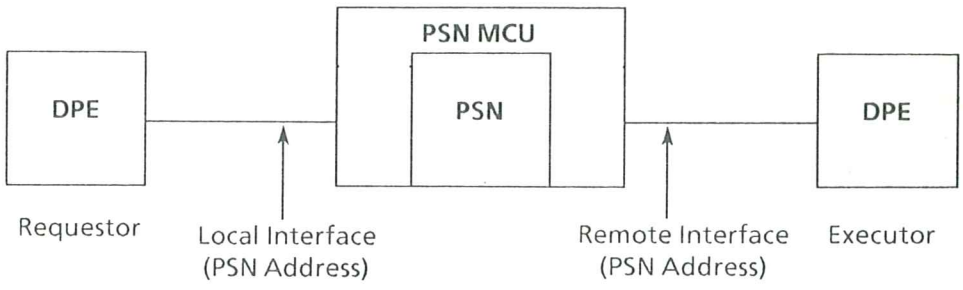


Figure 17 - Different Test Locations

Local testing refers to the interface to which a DPE is attached.

- If the PSN side is involved, the PSN MCU must be invoked to provide the necessary support for the test of the interface (loops, counters...).
- In the other case the DPE will call itself and the PSN will provide a basic network service.

Remote testing refers to any other interface.

- For the PSN side of such interface, the MCU will be addressed and the interface address will be used as a parameter in the request.
- When the DPE side of that other interface is involved, the DPE itself is addressed.

The relation between addressee and parameters is summarized in Table 5.

Type of Test	Addressee	Parameter	
Local test,PSN side	PSN MCU	Test Description	
Local test, DPE side	DPE Access	Test Description	DPE Identification (Bus Configuration)
Remote test, PSN side	PSN MCU	Test Description	Interface Address
Remote test, DPE side	DPE Access	Test Description	DPE Identification (Bus Configuration)

Table 5 - Relation between Type of Test, Addressee and Parameters

9.3.2 Terminal Identification on a Bus

When the test applies to a call already established the DPE is already logically linked to the maintenance process, and the call reference can be used as an identification parameter. However, in order to address a specific DPE (or its SM) for a fresh call, out of a number of DPEs connected to the bus on the same S_0 interface, the terminal must be identifiable unambiguously.

Several methods of terminal identification shall be studied, e.g.:

- A unique terminal identifier UTI automatically readable during the course of a maintenance enquiry can be transferred to the requesting DPE. The UIT is also applied visibly to the DPE, so that the DPE can clearly be identified on site. The unique terminal identifier could consist of the manufacturer's name and his private identification code, e.g. type and series number. ISO proposals relevant to physical terminal numbers should be taken into consideration.
- Physical terminal identification by using DDI or sub-addresses, if applicable.

- Terminal identification based on requesting and filtering the bearer capability and the low/high layer compatibility characteristics of all DPEs connected to the interface. This method, however, does not necessarily lead to an unambiguous identification.

9.4 Interface/Channel States

In order to preserve priority in collision situations, a set of states is suggested for the management of the interfaces. The management of the states itself is beyond the scope of this Technical Report, although it is mentioned that these states could be changed on operators' request or as recovery actions (on error counter overflow...).

The following states globally describe an interface:

- IN-SERVICE
- MAINTENANCE
- OUT-OF-SERVICE

The following states describe each B-channel:

- IN-SERVICE (ON):

  A B-channel is ON in its normal condition. Any call is allowed on a channel which is ON. In the case of contention between a call and a maintenance request, the call shall preempt. The solution of collisions between similar requests is given in Standard ECMA-106. This does not preclude the possibility to carry out a loop test on an established call.

- MAINTENANCE (OFF FOR MAINTENANCE):

  In this case, the B-channel cannot be used for a regular call but can be involved in a loop test. This state can be used to isolate a suspect B-channel, on which a test should be carried out. The solution of collisions between test requests is provided in Standard ECMA-106.

- OUT-OF-SERVICE (OFF):

  The B-channel is definitely unusable and any request based on this channel (call, test loop ...) will be rejected.

9.5 Priorities

A maintenance activity may not be possible due to particular configurations or may collide with another request (other maintenance or ordinary call). In these situations priority mechanisms are established to resolve conflicting requests. Contentions and priorities will be under the control of the System Management. However, on a passive bus configuration, multiple SMs exist. This gives rise to potential unresolved conflict situations.

9.5.1 Priorities in the IN SERVICE state

A normal call can only be established towards IN-SERVICE equipment; when a test call collides at a DPE with a normal call, the normal call will preempt. Collisions between maintenance calls are resolved as for normal calls, see ECMA-106.

9.5.2 Priorities in the MAINTENANCE State

When some equipment (B-channel or interface) is in the MAINTENANCE state, a maintenance call will still be possible. Special information is required as part of the basic call set-up in order to indicate to the called party that the offered call is a maintenance call. Collisions between maintenance calls are resolved as for normal calls, see ECMA-106.

9.5.3 Priorities across the PSN (Loaded Situations)

Should a call be addressed to a PSN MCU during a peak situation, the PSN will take the responsibility to execute or not the maintenance request.

When, in a similar situation, the addressee is a DPE, the PSN will not filter the call; however, the normal control mechanism of ECMA-106 will still apply.

9.6 Access Supervision

An access supervision mechanism shall be provided in order to prevent abuse and to deny unauthorized access to maintenance facilities. An example for a proposal is shown in Appendix C. The use of this mechanism is optional. When the access supervision mechanism is used, it shall

- allow enabling and disenabling of access supervision for individual operations;
- allow the application of access supervision for setting individual parameters or executing actions;
- be independent of priority mechanisms, see 9.5;
- allow or deny access or actions depending on the source of the request.

When access is denied, the reason for denial shall be returned.

9.6.1 Positioning within the Management Architecture

Access control can reside in the OSMAE, in the LME, or distributed between the two. Support in the OSMAE is appropriate when system management is accomplished via peer OSMAEs using services of all the communication protocol layers. Such application protocol, however, is beyond the scope of this Technical Report and is left to the implementor's definition.

The PDUs and primitives provided in this document allow for layer defined access control information to be optionally passed to the LME and status to be returned indicating failure to grant access. OSMAE support of access control mechanisms is for future study.

9.6.2 Implementation-specific Mechanisms

Implementation-specific restrictions on operations based on other factors are in no way precluded or prohibited by the provision of standard access control mechanisms. The standard status value should be used if access is denied for any reason.

9.7 Information Transfer Procedures

A particular OSMAE is the Maintenance Application. An application layer protocol shall be specified that defines management protocol data units (MPDU or procedure elements) and associated procedures for the transfer of information between peer System Management entities.

9.7.1 Elements

The following factors will be considered when selecting the proper procedure elements:

i) The layered approach implies that information is conveyed in Layer 3 envelopes, like the user-to-user information element, or in (a) new maintenance information element(s).

ii) Selected element(s) shall be identifiable by the PSN in order to resolve any further tarification problem.

iii) The element(s) shall be usable in any signalling phase including temporary signalling.

iv) The element(s) shall be present in the call set-up message in order to allow short signalling sequences such as "call set-up followed by a reject" ("fast connect") in order to reduce traffic over the PSN.

A list of examples for common data elements can be found in Appendix A.

9.7.2 Data Organization

In order to allow full flexibility in the end-to-end transfer of information, the data will be organized in "item lists" (i.e. each piece of data will be composed of two parts: descriptor and actual data, and a terminator will indicate the end of the list). A typical data list organization based on CCITT Rec. X.409 is indicated in Appendix B.

9.8 Primitives

The architectural concepts are discussed in Section 7. The present context requires the definition of the SM_ primitives only. These will be exchanged between the OSMAE and the Layer 3 entities in order to convey the MPDUs necessary for the implementation of maintenance facilities.

In principle, local primitives are bi-directional (i.e. a maintenance request or maintenance indication primitive may be sent either from the Maintenance Application to the Protocol Entity or vice versa), but may be uni-directional for particular maintenance and testing procedures, e.g. an indication that the loop has been activated would only be sent from the Physical Layer to the Maintenance Application.

The definition of the protocol will detail the specific content of the primitives when used for each of the procedures described in Section 10. Figure 18 shows an example of the interaction between two remote entities. Primitives and protocol flows (as described in Section 10) are indicated for a simple case.

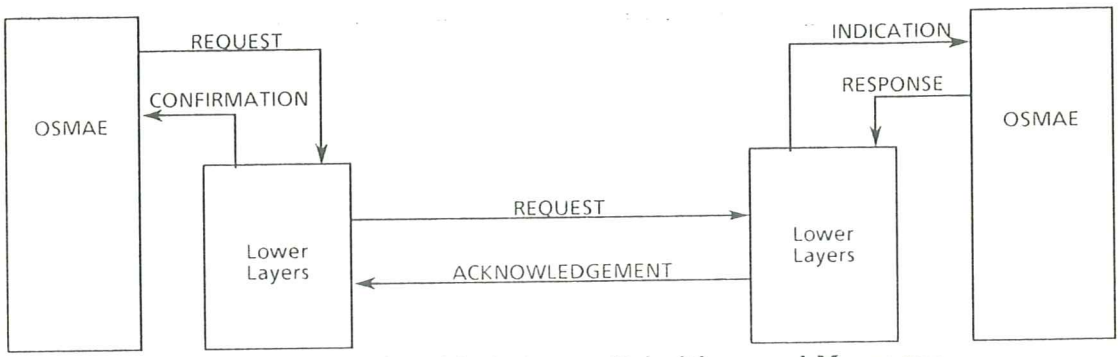


Figure 18 - Relationship between Primitives and Messages

10. MAINTENANCE PROCEDURES AND DATA FLOWS

A basic set of procedures will be required to facilitate maintenance of the interface. The procedures may have local or remote significance. They can be summarized as follows:

- Loop activation/deactivation.
- Self-test.
- Identification of access configuration (or maintenance enquiry).
- Remote monitoring of resources (counters, statuses...).
- Remote control of resources (counters, thresholds, statuses...).
- Remote reporting of events.

The different facilities that are described hereafter will not necessarily be supported by an entity (apart from loopback functions). Any request for unsupported options will be rejected with an appropriate error code.

Due to the possible absence of intermediate layers between the OSMAE and the D-channel upper layer (n^{th}, presently called 3^{rd}), the primitives will have to be mapped into layer n and will be composed of two types of parameters:

- the layer n parameters having internal significance and indicating to the layer n the service required, and
- the application parameters with global significance which form the actual MPDU.

If any other mapping were required, only the layer n parameter would have to be changed:

PRIMITIVE(Layer n param, Application param).

The application parameters form a combination of the common data elements, as listed in Appendix B.

The layer n parameters are the parameters that are necessary to identify the required Layer n service. As an example, an action reference could be used, at the Application Layer, in the same way as the call reference is used at Layer 3.

10.1 Loop Activation/Deactivation

These procedures belong to the ACTION generic family as described in 7.4.

10.1.1 Protocol Flow

In order to completely control a loop test, only the loop requestor should be allowed to cancel a loop test. A timer indication will be associated with the request and will run at the executor's side during the test. If the requestor has not required the cancellation of the loop by time-out as shown in Figure 19a, the executor will cancel the loop on his own, as shown in Figure 19b (this will be particularly the case when the interface is no longer accessible by the requestor).

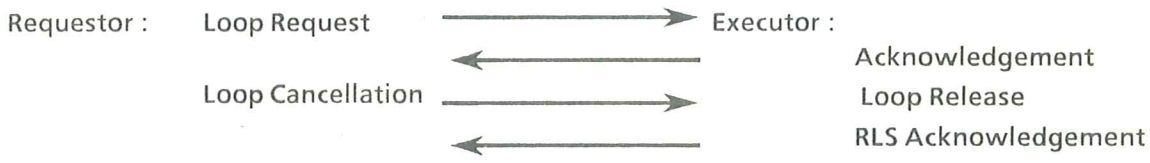


Figure 19a - Loop Activation / Cancellation by the Requestor

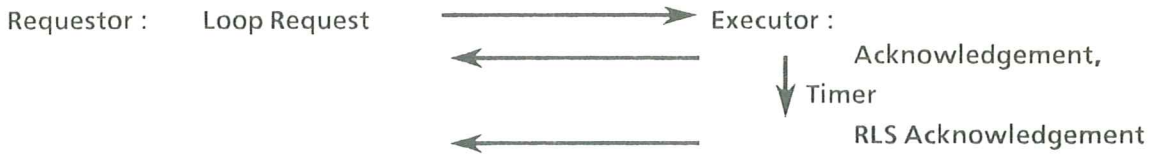


Figure 19b - "Blind" Loop Cancellation by the Executor

It may happen that the executor wants to terminate the loop for its own reasons. It will send a request for release by the requestor and start a timer, as shown in Figure 19c.

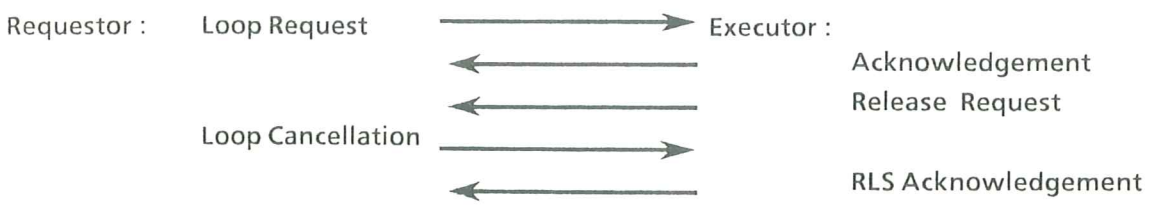


Figure 19c - Loop Cancellation on Request of the Executor

If the request for release is not satisfied within the timed period, the executor will terminate the loop as indicated in Figure 19d.

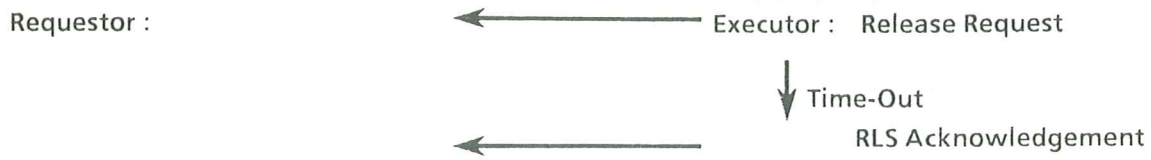


Figure 19d - Loop Release after unsuccessful Request for Release

It has to be noted that erroneous situations are not indicated in the above described flows. They will be further detailed in the protocol specification and will be based on mechanisms such as supervisory timers (e.g. the request for the loop executor to cancel the loop may fail, and the loop executor might force cancellation in such a case).

10.1.2 Primitive Parameters

The following parameters will be used for the SMDSI primitives:

ACTION_REQUEST/INDICATION (Layer 3, application)

- Layer 3 information:	Destination address	
	New call or part of an existing call	(Call reference)
- Application information:	Action reference	(New or existing call)
	Action identifier	Loop request
	Action value	Loop name and Channel ID(s)
	Action supervision information	
	Action duration	Timer value

ACTION_CONFIRMATION/RESPONSE (Layer 3, application)

- Layer 3 information:	Call reference	
	Call release required	Y/N
- Application information:	Action reference	(New or existing call)
	Action identifier	Action Acknowledgement
	Status	Accepted/Rejected
	Reason for rejection	Access failed
		Unimplemented Loop
		Unacceptable Duration
		Test impossible (Channel busy or out of service)

CANCEL_REQUEST/INDICATION (Layer 3, application)

- Layer 3 information:	Call reference	
	Call release required	Y/N
- Application information:	Action reference	(New or existing call)
	Action identifier	Loop cancellation request
	Action value	Loop name and Channel ID(s)
	Access control information	

RELEASE_REQUEST/INDICATION (Layer 3, application)

- Layer 3 information:	Call reference	

- Application information:	Action reference	(New or existing call)
	Action identifier	Loop release request
	Action value	Loop name and Channel ID(s)
	Access control information	

RELEASE_CONFIRMATION/RESPONSE (Layer 3, application)

- Layer 3 information:	Call reference	
	Call release required	Y/N
- Application information:	Action reference	(New or existing call)
	Action identifier	Release Acknowledgement
	Status	Accepted/Rejected
	Reason for rejection	Unimplemented Loop

10.1.3 Loop Test Phase

The requestor will consider the loop as active on receipt of an acknowledgement.

The loop will be considered as inactive on transmission of the cancel request or on receipt of a release acknowledgement when not preceded by any other message.

10.2 Self-Test

The procedures described in this Section belong to the ACTION generic family as described in 7.4. For the purpose of this Technical Report, self-tests apply only to the communication interface (S_0 or S_2), and a DPE will in principle not request self-tests from a PSN.

10.2.1 Protocol Flow

The whole procedure concerning activation/deactivation of self-test is optional due to the consequences of such tests on traffic loads. In any case, a system which does not support self-tests must be able to recognize and reject a self-test request. The self-test activation procedures consist of four phases of information exchange:

i) test REQUEST from the requestor;

ii) test INFORMATION from the executor, in order to inform the requesting end of the test about consequences such as time duration;

iii) test ACTIVATION from the requestor (or request CANCEL); the requestor in light of the information received can then decide whether or not the test should be activated;

iv) test activation (or cancel) ACKNOWLEDGEMENT from the executor.

During the execution of the test a timer is activated at the requesting side. An information message should be periodically returned by the executor indicating the test status. It shall also be returned at the end of the test to indicate that the executor has completed the test and returned to the NULL state. Should the requestor's timer overflow, the requestor should be able to call up a status enquiry procedure, as described in 10.4.

Different flows are possible; Figure 20 depicts the acceptance of a self-test request.

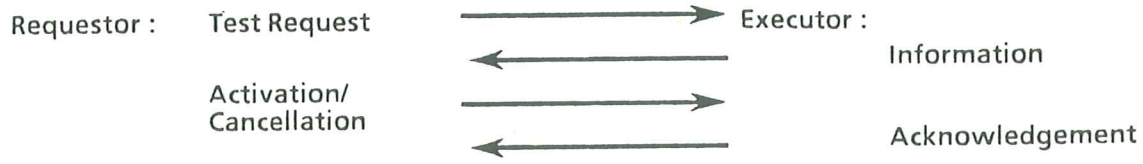


Figure 20 - Self-Test Activation / Cancellation Accepted

Figure 21 shows the rejection of a self-test request. The rejection will include an indication of the cause for the refusal, e.g. facility not supported, request not authorized (after a compatibility check), DPE busy in a call, DPE busy in a loop back test.

Figure 21 - Self-Test Activation / Cancellation Rejected

When a long test is executed, it should be divided into phases, each providing a breakpoint in the test. At that time the test executor will suspend its activity and send information on the test status. The requestor will then ask the executor to continue or cancel the test, see Figure 22.

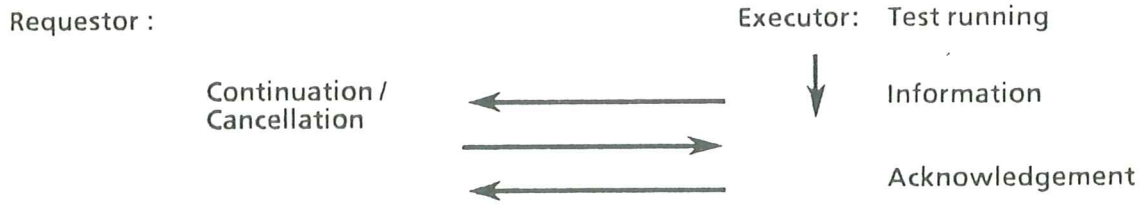


Figure 22 - Subdivision of a Test into Phases

10.2.2 Primitive parameters

The following parameters will be used for the SMDSI primitives:

ACTION_REQUEST/INDICATION (Layer 3, application)

- Layer 3 information:	Destination address	
	Part of existing call (Call reference)	
- Application information:	Action reference	(New or existing call)
	Action identifier	Self-test request
	Action value	Self-test identification
	Access control information	
	Action duration	Timer value

ACTION_CONFIRMATION/RESPONSE (Layer 3, application)

- Layer 3 information:	Call reference	
	Call release required	Y/N
- Application information:	Action reference	
	Action identifier	Action Acknowledgement
	Status	Accepted/Rejected
	Reason for rejection	Access failed
		Unimplemented feature
		Unacceptable duration
		Test impossible (interface busy)
	Action duration	Timer value

CANCEL_REQUEST/INDICATION (Layer 3, application)

- Layer 3 information:	Call reference	
	Call release required	Y/N
- Application information:	Action reference	
	Action identifier	Self-test cancellation request
	Action value	Self-test identification
	Access control information	

RELEASE_REQUEST/INDICATION (Layer 3, application

- Layer 3 information:	Call reference	
- Application information:	Action reference	
	Action identifier	Self-test release request
	Action value	Self-test identification
	Access control information	

RELEAST_CONFIRMATION/RESPONSE (Layer 3, application)

- Layer 3 information:	Call reference	
	Call release required	Y/N
- Application information:	Action reference	
	Action identifier	Releast Acknowledgement
	Status	Accepted/Rejected
	Reason for rejection	Access failed
		Unimplemented feature

STATUS_INDICATION (Layer 3, application)

- Layer 3 information:	Call reference	
- Application information	Action reference	
	Action identifier	Status Report
	Action value	Self-test identification
	Status	Status of self-test

10.2.3 Self-test Phase

Partitioning of a self-test into phases is implementation dependent and is beyond the scope of this Technical Report.

As a general rule, it should, on the one hand, not separate logically correlated sequences; on the other hand, the duration of a phase should take into account human patience (if the requestor is supposed to be a man and not an application process running in the requesting maintenance entity) and traffic performance of the PSN and, if applicable, other intervening networks.

10.3 Maintenance Enquiry

Maintenance Enquiry is a service which allows to determine an unknown access configuration as seen from the PSN side of the interface. It may or may not comprise the support of data collection by the PSN. The Maintenance Enquiry procedure is compelled and belongs to the READ generic family, as described in 7.4.

It is for further study, whether the service with or without data collection by the PSN or both shall be standardized.

i) Service with data collection by the PSN

The requestor sets up a call to the OSMAE of the PSN, thereby indicating the address of the remote access which it wants to get information about. After an acceptance procedure (which results in either acceptance or rejection of the Maintenance Enquiry request) the OSMAE will initiate a global call to all DPEs (or TEs) connected to that access. All DPEs (TEs) present will respond, indicating their characteristics by subaddress, presently assigned TEI and/or other information such as compatibility information, or - preferably - the unique terminal identifier, see 8.3.2. This information will be returned to the requestor where it can be used for the maintenance transactions themselves.

ii) Service without data collection by the PSN

The request will be forwarded to all DPEs on that bus. The DPEs will answer in the same way as described above. Their responses will, however, be transparently passed backward to the requestor who itself will run the supervisory timer and collect the data obtained.

It must be noted that a maintenance enquiry may be rejected by the DPEs, in which case the requestor would at least know about the existence of the DPE. The use of Maintenance Enquiry will be subject to priority, authorization and conformance checking rules, in order to avoid abuse and traffic overload. If a specific type of TE, e.g. DPEs of specific characteristics, are to be identified the Maintenance Enquiry call may quality this, e.g. by means of adequately narrowed bearer capability and/or low and/or high layer compatibility information.

10.3.1 Protocol Flows

i) Service with data collection by the PSN

The protocol flow consists of two parts, corresponding to two hierarchically different executors. One part covers the section between the requesting entity and the OSMAE of the PSN, and the other one covers the section between the PSN's OSMAE and the OSMAEs of the DPEs at the remote access, see Figure 23a.

The request is sent to the PSN OSMAE. It contains the access address. In its ACKNOWLEDGEMENT message, the PSN will indicate that it is undertaking the data collection and that it will return a single response.

The PSN OSMAE will broadcast, as a global call, the Maintenance Enquiry to the access and initiate a timer. Each DPE connected at the interface is given sufficient time to return information that clearly and uniquely identifies it, as indicated in 9.3.2. While the timer is running, the returned information is collected by the PSN Management Entity. At time-out, the information is packed and returned to the requestor.

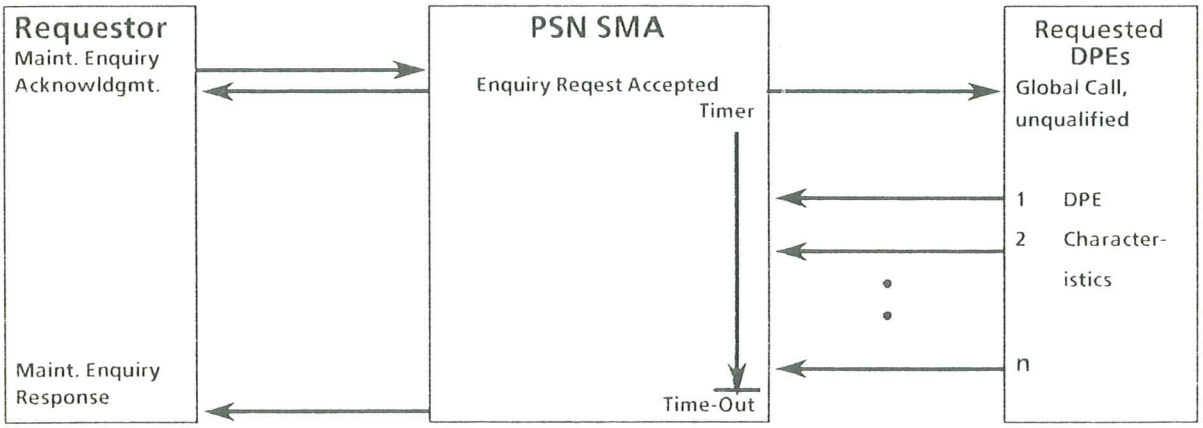


Figure 23a - Maintenance Enquiry (Accepted)

ii) Service without data collection by the PSN

The requestor announces his wish to the PSN. The PSN will indicate in its ACKNOWLEDGEMENT message that it does not undertake the data collection, see Fig. 23b.

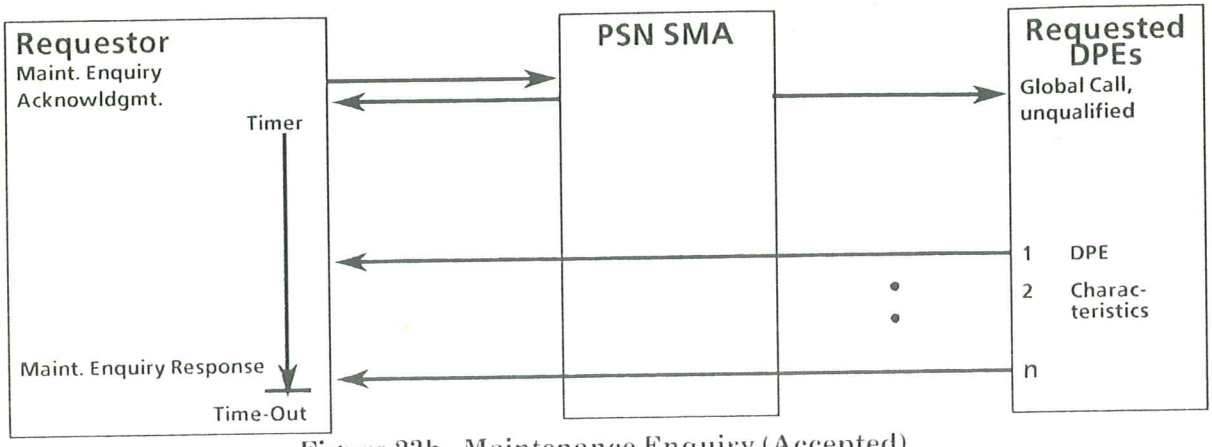


Figure 23b - Maintenance Enquiry (Accepted)

The PSN will emit a global call to the requested DPEs, as is the case with a normal call to a multi-point configuration and the requestor will initiate a timer. Each DPE connected at the interface is given sufficient time to return information that clearly and uniquely identifies it, as indicated in 9.3.2.

When the timer expires, all responses are assumed to have been received, thus terminating the procedure.

10.3.2 Primitive parameters

In both cases the following parameters will be used for the SMDSI primitives:

ACTION_REQUEST/INDICATION (Layer 3, application)

- Layer 3 information:	Destination address	
	New call	
	Call reference	
- Application information:	Action reference	(New or existing call)
	Action identifier	Maintenance enquiry request
	Access control information	

ACTION_CONFIRMATION/RESPONSE (Layer 3, application)

- Layer 3 information	Call reference	
	Call release required	Y/N
- Application information:	Action reference	
	Action identifier	Maint. Enquiry Acknowledgement
	Status	Accepted/Rejected
	Reason for rejection	Access failed Unimplemented feature

STATUS_INDICATION (Layer 3, application)

- Layer 3 information:	Call reference	
- Application information:	Action reference	
	Action identifier	Enquiry Response
	Action value	Item list describing one of the TEs identified on the access.

10.3.3 Selecting and assembling maintenance enquiry information

The maintenance enquiry information is collected either by the PSN's or by the requestor's OSMAE and then concatenated to a Status Indication serving as a maintenance enquiry report. The timer value must be set so that all DPEs (possibly including also non-DPE TEs) at the remote access have a good chance to respond to the global call. The value should consider worst case conditions for all layers since the access might be in the DEACTIVATED state and might have to set up Physical Layer, Data Link Layer and Layer 3 connections, see Standards ECMA-103, ECMA-104, ECMA-105 and ECMA-106.

10.4 Resources Monitoring/Status Enquiry

Resources monitoring is part of the READ generic family such as described in 7.4. Its purpose is to retrieve information on demand from a remote System Management.

10.4.1 Protocol flow

At any time an entity can request its peer to return information concerning the support/non-support of maintenance facilities as well as the interface state (off-line for maintenance , awaiting answer...), when a previous request (loop back test, self-test...) has been made. This request will also allow to retrieve statistical information. The requesting end will send a status enquiry request to the terminating end which will return the ad hoc information or reject the request if it is not admissible, see Figure 24.

Figure 24 - Reject of a Status Enquiry

Error situations may be covered by timers; this is considered to be part of the detailed protocol specification.

10.4.2 Primitive parameters

READ_REQUEST/INDICATION (Layer 3, application)

- Layer 3 information:	Destination address	
	New call	
	Call reference	
- Application information:	Action reference	New or existing call
	Action identifier	Read information
	Action control information	
	Action value	Data element list

READ_CONFIRMATION/RESPONSE (Layer 3, application)

- Layer 3 information:	Call reference	
- Application information:	Action reference	
	Action identifier	Read information response
	Status	Accepted/Rejected
	Reason for rejection	Access failed
		Unimplemented feature
	Action value	Item list containing the status/data information

10.5 Remote Control of Statuses/Counters/Thresholds

The remote resources control is part of the SET generic family, as described in 7.4. It is aimed at changing information on demand in a remote System Management. The request may have a global or atomistic significance in the sense that the total request may fail if at least one change fails or may be valid as long as there is at least one change accepted.

10.5.1 Protocol Flow

The function of SET REQUEST is to request the remote OSMAE to set the specified data elements and to return the status of each data element set to the local OSMAE.

A SET RESPONSE is expected for each SET REQUEST as shown in Figure 24.

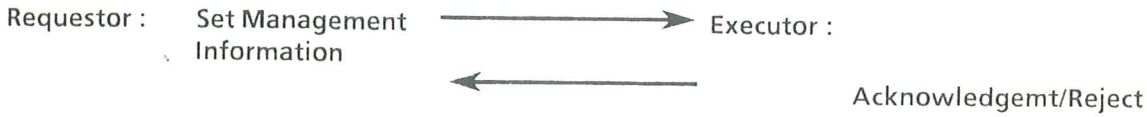


Figure 24 - Status Control Request

Error situations may be covered by timers; this is considered to be part of the detailed protocol specification.

10.5.2 Primitive Parameters

SET_REQUEST/INDICATION (Layer 3, application)

- Layer 3 information:	Destination address	
	New call	
	Call reference	
- Application information:	Action reference	New or existing call
	Action identifier	Set management information
	Status	Accepted/Rejected
	Reason for rejection	Access failed Unimplemented feature
	Action value	Data element list

SET_CONFIRMATION/RESPONSE (Layer 3, application)

- Layer 3 information	Call reference	
- Application information	Action reference	
	Action identifier	Set information response
	Status	Accepted/Rejected
	Reason for rejection	Access failed Unimplemented feature
	Action value	Item list with the status/ data information

10.5.3 Changing Remote Parameters

When an OSMAE receives an MDPU containing a SET REQUEST, it checks the Resource Identifier to determine in which layer(s) the data elements are to be set. Subsequently, the Access Control field will be checked. In the Data Element List each data element identified is set to the value specified for that data element. If the data element identified is not a valid data element for the SET REQUEST operation,

the reason for the failure of setting will be returned in the status field of the SET RESPONSE MDPU.

An OSMAE generates a RESPONSE MDPU containing a SET RESPONSE to respond to a SET REQUEST. The information in the Resource Identifier field is the same as that in the corresponding SET REQUEST. The SET RESPONSE may contain either a single status value indicating an invalid request (unsupported operation, bad layer internal selector, no access authentication, etc.) or a Data Element List containing the value appropriate for the corresponding request.

10.6 Event Reporting

This function is to notify the remote System Management of any events that have been detected by the local System Management. It is part of the M_REPORT family described in 7.4.

An EVENT ACKNOWLEDGEMENT MDPU is not expected for each EVENT MDPU, but may be requested in the EVENT MPDU.

The function of an EVENT ACKNOWLEDGEMENT MPDU is to acknowledge the receipt of an EVENT MPDU by a System Management Application. The system reporting the event has the option of requesting that its EVENT MPDU be acknowledged.

10.6.1 Protocol Flow

The event message is sent by a local System Management whenever a particular change of a state occurs that has been appointed as an event to be reported remotely. The receipt of an event MPDU is unsolicited in the sense that the receiving SM does not make requests for events.

The information provided by the layers (LMEs), the request for acknowledgement and the type of event are supplied in appropriate fields. Multiple events may be associated with different layers contained in a single MPDU. The inclusion of the event's time is optional.

The receiving SM will not return any acknowledgement MPDU unless it is asked to do so. In this case an EVENT ACKNOWLEDGEMENT MDPU is returned at the earliest opportunity in order to cancel the timer that may have been started by the SM reporting the event. This is shown in Figure 25.

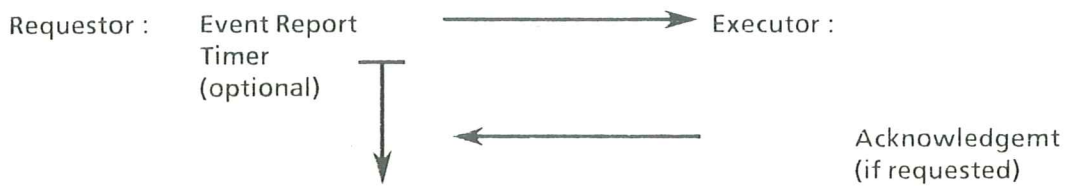


Figure 25 - Event Reporting

10.6.2 Primitive parameters

REPORT_REQUEST/INDICATION (Layer 3, application)

- Layer 3 information:	Destination address New call Call reference	
- Application information:	Action reference Action identifier Action value	New or existing call Event reporting Item list composed of several Event information Elements
	Acknowledgement Required Indication Access control information	

REPORT_CONFIRMATION/RESPONSE (Layer 3, application)

- Layer 3 information:	Call reference	
- Application information:	Action reference Action identifier Status	Event acknowledgement Status codes:
	Action value	. Resource ID (Layer inf.) . Event ID (indication which event is acknowledged)

11. TEST WITHIN A CALL

While a call is established, a Maintenance Entity will be able to invoke test facilities to be executed by its peer entity by the same mechanisms as described above. Some more erroneous situations may result from collisions or incompatible priorities. These fill be fixed by the detailed specification of the protocol.

Any exchange of information between two OSMAEs will be referred to by an action reference. The action reference shall be systematically part of the MDUs. A mapping between the action reference and the call reference used at the Layer 3 protocol entity needs to be maintained at the SMDSI.

When maintenance activity is to be carried out on an already existing call, the OSMAE will have to indicate to Layer 3 which call is to be involved. A means for this might be the Layer 3 call reference whose significance, however, is local only. This may require the use of additional indicators.

12. SUBJECTS FOR STANDARDIZATION

Possible subjects for standardization within the scope of this Technical Report are:

- A signalling system for maintenance of the DPE-to-PSN interface
- A guide for running maintenance and test procedures at DPE-to-PSN interfaces.

APPENDIX A

COMMON DATA ELEMENT DEFINITIONS

AccessSupervision

This is an optional field of unspecified form to be used as an input to access supervision functions.

DefinedParameter

This field contains the parameters according to their definitions in the layer specific implementations.

ActionReference

This optional field correlates each response with its request. Mainly it will be used to identify the source or the recepient of the data element. ActionReference shall be included in responses to requests which contain an action reference. The value returned is the value that was contained in ActionReference.
It is not the purpose of ActionReference to protect against duplication or loss of application layer messages, but merely to identify each maintenance transaction. The protocol places no constraints on the method applied for generating ActionReference values.

Layer

This field indicates the layer to which the operation applies. The layers are defined in accordance with the OSI Reference Model. The Management Layer is the Layer in which the System Management protocol is used. The Physical Layer is layer 1 and the Data Link Layer is layer 2 in the OSI Reference Model.

LayerInfo

This field contains all of the layer information that is required for the interpretation of the operation by the OSMAE addressed. The information specifies the layer, sublayer, instance (in the case of management or multiple instances by the same entity, eg. a bridge) and the entity within the layer to which the request operation of event applies, to which responses apply or in which the event was detected.

NonSpecific

This field is of unspecified form, but is intended to contain the information used to control access to functions and parameters within an OSMAE.

Parameter

A parameter contains either the status or the identifier of a defined parameter (layer or implementation-specific). If the operation on the parameter was successful, then the status is returned as success and the value of the parameter is returned only if appropriate (e.g. for a Get). If the operation on the parameter failed, then the parameter is returned in StatusInfo and the status of the failure is returned in Status. If the operation on the parameter was only partially successful, then that information is returned.

The values carried with the parameters are :

ReadRQ
Parameter id. "Empty" values, such as integer=0, zero length octet or bitstring, may be used except where required for complex (array) data access, such as indices (parameter specific).

ReadRSP
Parameter identity and current value of specified parameter.

SetRQ
Parameter identity and value to which the parameter is to be set.

SetRSP
Parameter identity and value to which the parameter was set (or not set if an error occurred).

CompareAndSetRQ
Parameters are the types defined for Expected Value and for Value. Expected Value contains the parameter identity of the parameter to be compared and the value of that parameter. Value contains the parameter identity of the parameter which is to be set and the value to which it is to be set.

CompareAndSetRSP
Parameters are the types defined for Expected Value and for Value. Expected Value contains the parameter identity of the parameter used for comparison and its value. Value contains the parameter identity of the parameter which was set and its value.

The definitions for parameters are found in the layer or implementation specifications selected by the ResourceID of the operation.

ParameterList

This field contains a list of parameters which are to be used according to the semantics of the function declared. In the case of a ReadRQ, this list is a list of the parameters whose values are to be read.

ResourceID

This field specifies the resource that is the subject of the maintenance information exchange. For the event exchanges, the placement of this field inside each operation allows that events affecting multiple layers be described in the same PDU.

ResourceTypeID

The ResourceTypeID is used to verify the context of the Layer (this is used in the case where a connectionless protocol is used).

Status

This field indicates the success or failure of the Request PDU.

StatusCode

This field contains the code that identifies the status of the operation requested : success or failure (identifies the general type of failure).

success : the operation was performed correctly and any value returned can be considered as valid.

badOperation : the operation specified (e.g. Request, Read) is not supported.

badLayer : the layer specified is not supported.

badLayerInstance : the LayerInstance specified is not supported.

badParameter : the ParameterID specified is not supported.

badParameterOperation : the indicated parameter may not be operated on by the specified operation (e.g. a parameter is not readable). The operation may not be allowed or supported.

pduTooLong : the PDU is too long (the entire PDU cannot be interpreted or received).

badParameterValue : the value of the ParameterValue specified is out of the range of permissible values or is not supported by this implementation.

badExpectedValue : the value of ParamExpValue was not equal to the current value of the parameter.

badAccessRights : the request associated with this response was not executed, because either the value of AccessSupervision was not acceptable or the access was denied for other reasons.

badAction : the action indicated is not supported.

negativeAcknowledge : the PDU cannot be accepted (for any reason). This is to be used for event notification acknowledgement.

badRequestInfo : RequestInfo is not supported or is not defined.

badResourceID : ResourceID is not supported or is not defined.

badLayerInfo : LayerInfo is not supported or is not defined.

StatusInfo

This optional field contains any additional information (in addition to StatusCode) that the Agent can include about the status returned in the StatusCode, for example, this field may include the portion of the PDU sent that could not be interpreted. Implementation specific values shall use Private Use identifiers.

UniversalTerminalIdentifier

The UTI indicates the physical identity of the device OSMAE, e.g. an indication of the manufacturer and his type and series number.

APPENDIX B

DATA LIST ORGANIZATION

B.1 Syntax

Due to the variety of information types, events and counters will be organized in item lists when required for transfer to a remote entity. Each item is identified by an item descriptor which may be used recursively when necessary. The item (data element) will be composed of three basic components and will comply with the structure described in CCITT Recommendation X.409. The three components are :

* Item Type : governs the interpretation of the contents,
* Item Length : describes the length of the contents,
* Item Value : is the information itself which the item is to convey.

Figure B-1 indicates the structure of an item.

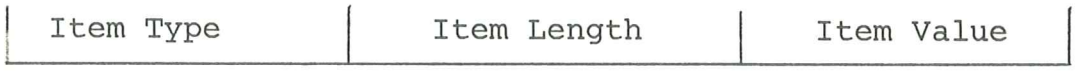


Figure B-1 : Structure of an Item

Although the mechanism principally allows for long item lists, a practical limit is imposed by lower layer protocol constraints (e.g. maximum Layer 3 message size, which in turn is limited by the Data Link Layer maximum frame size).

The proposed structure allows the presence (or absence) of any element in any order.

Due to limitation of the list length imposed by protocol constraints, only the "X.409 short form" will be used for the definition of the item length.

The presence or absence of values and the format associated to the item will be implicitely defined by the code provided in the item identifier (e.g. an event may have no value or in a REQUEST is a counter name required only).

There are 4 classes of item types : universal, application wide, context specific and private use. The universal class covers the following types :

Boolean
Integer
Bit String
Octet String
Null Element
Sequence
Set
Numeric String
Printable String
T.61 String

Videotex String
IA5 String
UTC Time
Generalized Time

For the definition of counters and events other classes will have to be specified and the item type coding will itself be structured as described in Figures B-2a and B-2b.

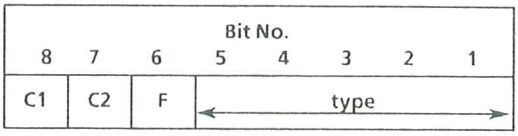


Figure B-2a : Coding Structure of Item Type

The first element in a list will be used to indicate the origin, i.e. the maintainer of the counter or the originator of the events. The other codes will be used to indicate a counter within this category, see Figure B-2b.

List 1	Element = ORIGIN xx
	Item 1 = COUNTER $\in$ xx
	Item 2 = COUNTER $\in$ xx
	Item 3 = COUNTER $\in$ xx
List 2	Element = ORIGIN xy
	Item 1 = COUNTER $\in$ xy
	Item 2 = COUNTER $\in$ xy

Figure B-2b : Indication of the Origin of the Counters by the 1st Element of a List

If the range of type codes exceeds 32, the extension mechanism provided by CCITT Rec. X.409 will be used as indicated in Figure B-2c.

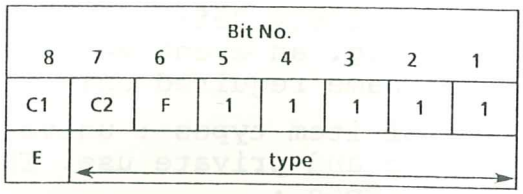


Figure B-2c : Extension Mechanism for Item Definition

B.2 Possible Item Types

As a proposal and only to give examples, some item types are listed subsequently. Each of them has to be justified.

B.2.1 Physical Layer

Possible Item types for the Physical Layer are listed in Table B-1.

Item Type	Event	Counter
FCS failure (if implemented, see ECMA-103 and ECMA-104	X	X
Loss of synchronization	X	X
Loss of framing	X	X
Remote Alarm received		X
Accumulator of faulty situations		

Table B-1 : Physical Layer Information Items

B.2.2 Data Link Layer

Possible Item types for the Data Link Layer are listed in Table B-2.

B.2.3 Layer 3

Possible Item types for Layer 3 are listed in Table B-3.

B.2.4 System Management

Possible Item types for the System Management are listed in Table B-4.

B.3 Examples for Item Lists

Examples for item lists are given in Tables B-5 and B-6. Table B-5 shows a simple list with two items while Table B-6 shows a recoursive list.

Item Type	Status	Event	Counter	Accu-mulator
Link state	X			
RX and TX bytes				X
RX and TX frames				X
RX and TX REJECT				X
RNR RX (remote buffer error)				X
RNR TX (local buffer error)				X
RX and TR FRMR				X
Protocol failures Time-Outs Address Errors Invalid frame type, RX (this may duplicate the frame count?)		X	X X	X
FCS errors (if no Physical Layer counters are used)				X

Table B.2 : Data Link Layer Information Items

Item Type	Status	Event	Counter	Accu-mulator
Call state	X			
RX and TX calls			X	
RX call resource errors			X	
Protocol failures Time-Outs RX Invalid message type Valid but unimplemented message, RX Errors code received for different classes		X	X X X X	

Table B.3 : Layer 3 Information Items

Item Type	Status	Event	Counter	Accu-mulator
Alarms received				X
Interface status	X			
Channel occupation Channel status	X X			

Table B.4 : System Management Information Items

Information Element			Coding
type = list			C_1 C_1 F S S S S S
list length			0 0 0 0 0 1 1 0
value	<	item 1 type	C_1 C_1 F S S S S S
		item 1 length	0 0 0 0 0 0 1 0
		item 1 value	S S S S S S S S
		item 2 type	C_1 C_1 F S S S S S
		item 2 length	0 0 0 0 0 0 1 0
		item 2 value >	S S S S S S S S

Table B.5 : System Management Information Items - 2 Items

Information Element	Coding
type = list	C_1 C_1 F S S S S S
list length	0 0 0 0 1 1 0 0
value < item 1 type	C_1 C_1 F S S S S S
item 1 length	0 0 0 0 0 0 1 0
item 1 value	S S S S S S S S
< item 1.1 type	C_1 C_1 F S S S S S
item 1.1 length	0 0 0 0 0 0 0 1
item 1.1 value	S S S S S S S S
item 1.2 type	C_1 C_1 F S S S S S
item 1.2 length	0 0 0 0 0 0 0 1
item 1.2 value >	S S S S S S S S
item 2 type	C_1 C_1 F S S S S S
item 2 length	0 0 0 0 0 0 1 0
item 2 value >	S S S S S S S S

Table B.6 : System Management Information Items
Recoursive List

APPENDIX C

EXAMPLE OF A PROPOSAL FOR AN ACCESS SUPERVISION MECHANISM

The following proposes an access supervision mechanism that is under study for inclusion in the standard. It is compatible with the existing constraints and uses the existing PDUs and services.

It is specified as an algorithm and set of parameters to support the algorithm.

C.1 General

Access supervision is based on an access class, a property associated with each operation and parameter combination and each action. The requested operation is performed if the required class is the same or lower than the class assigned to and announced by the requestor.

Verification of the legitimacy of the requestor's use of the class is accomplished via the access supervision information of the requestPDU, an unstructured string of octets, which is compared to the entries of a class validation table, effectively implementing a password check.

Access is further limited by the source of the request being recognized as local or remote, where the validation table entry carries local and remote source permission values. The actual source address is not used to avoid the problems of manager reassignment due to failure or reconfiguration.

The OSMAE centralizes a large part of the access supervision mechanism, fundamentally providing the control function shown in Figure C-1.

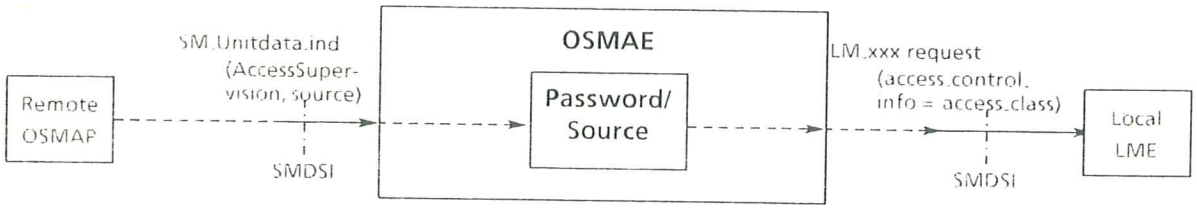


Figure C-1 : OSMAE Access Supervision Function

C.2 Control Function

The OSMAE performs the function producing the access class using an algorithm applied to the information or the request and a set of System Management parameters.

C.2.1 Parameters

The parameter associated with the access control function is the class validation table.

C.2.1.1 Syntax

This parameter would be an addition to the Management Parameter choice.

```
ClassValidationEntry::= SEQUENCE{ Index[0] IMPLICIT
                        INTEGER,--1..8
                        LocalSourcePermitted[1] IMPLICIT
                        BOOLEAN,
                        RemotSourcePermitted[2] IMPLICIT
                        BOOLEAN,
                        Password[3]IMPLICIT OCTETSTRING}
```

C.2.1.2 Semantics

The Class Validation Table consists of eight entries, indexed zero to seven for classes zero to seven. The table is accessed via an entry, where the index specifies which table entry is being accessed.

- LocalSourcePermitted specifies whether local access is valid for the corresponding class.
- RemoteSourcePermitted specifies whether remote access is valid for the corresponding class.
- Password specifies the value of the AccessSupervision field of the SM-PDU required for the corresponding class.

Read/Write support of this parameter is optional.

C.2.2 Algorithm

Upon receipt of a request PDU, the table is searched from entry seven down to entry one, searching for an exact match and permission of source. If a match is found, the index of the matching entry is passed as the highest allowed access class. If the class is not found, the access class passed to the LME is "none". If the AccessSupervision field is not present in the PDU, it is treated as an OCTETSTRING of zero length.

Implementation-specific limits or mechanisms may need to be provided to prevent all access from being denied in certain configurations.

C.3 Access Classes

The Access Class passed to the LME includes ("none", 1..8) where class 8 is the highest "security", 1 is the lowest validated class and "none" is the class implying that a match was not found, possibly because it was not supplied and no table entry was configured for the default.

C.3.1 LayerSpecific

The LME is responsible for acting on the access class information passed in the LM-xxx, request. This is layer specific and can be accomplished by specifying the minimum required access class for an operation, either via a parameter or by previous definition. The operation may be defined in terms of parameters, groups of parameters, operations permitted on those parameters, or actions performed.

C.3.1.1 Syntax

```
AccessGroupClass::= IMPLICIT AccessClass

AccessClass::[APPLICATION X] INTEGER{ NoneRequired(0),
                                      ClassRequired(1),
                                      Class2Required(2),
                                      Class3Required(3),
                                      Class4Required(4),
                                      Class5Required(5),
                                      Class6Required(6),
                                      Class7Required(7),
                                      Class8Required(8)}
```

Where "AccessGroup" would specify the parameters or operations regulated by the parameter.

C.3.1.2 Semantics

AccessGroupClass determines permission to perform the operations associated with the specified access group.

C.3.2 System Management Specific

The Management entity parameters are protected by two access class parameters.

C.3.2.1 Syntax

These parameters would be additions to the Management Parameter choice.

```
AccessSupervisionGroupClass::= IMPLICIT AccessClass

WriteSupervisionGroupClass ::= IMPLICIT AccessClass

AccessClass::[APPLICATION X]INTEGER{ NoneRequired(0),
                                     Class1Required(1),
                                     Class2Required(2),
                                     Class3Required(3),
                                     Class4Required(4),
                                     Class5Required(5),
                                     Class6Required(6),
                                     Class7Required(7) }
```

APPENDIX D

LIST OF ACRONYMS

CSMA	Carrier Sensing Multiple Access
DEMUX	Demultiplexer
DDI	Direct Dialling Inwards
DPE	Data Processing Equipment
FCS	Frame Check Sequence
HDB3	High Density Bipolar 3
LAP	Link Access Protocol
LME	Layer Management Entity
LMI	Layer Management Interface
MCU	Maintenance Centre Unit
MIB	Management Information Base
MPDU	Management Protocol Data Unit
MUX	Multiplexer
OSI	Open Systems Interconnection
OSMAE	Open System Management Application Entity
PE	Protocol Entity
PCSN	Private Circuit Switching Network
PDU	Protocol Data Unit
PPSN	Private Packet Switching Network
PSN	Private Switching Network
PT	Private Switching Network Termination
SM	System Management
SMDSI	System Management Data Service Interface
T	Timer
TA	Terminal Adaptor
TE	Terminal Equipment
TEI	Terminal Equipment Identifier
TS	Time Slot
UTI	Universal Terminal Identifier