E C M A

EUROPEAN COMPUTER MANUFACTURERS ASSOCIATION

STANDARD ECMA-165

PRIVATE TELECOMMUNICATION NETWORKS (PTN)

-

SIGNALLING BETWEEN PRIVATE TELECOMMUNICATION EXCHANGES

-

GENERIC FUNCTIONAL PROTOCOL FOR THE SUPPORT OF SUPPLEMENTARY SERVICES

(QSIG-GF)

March 1992

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

Phone +41 22 735 36 34 Fax: +41 22 786 52 31

E C M A

EUROPEAN COMPUTER MANUFACTURERS ASSOCIATION

STANDARD ECMA-165

PRIVATE TELECOMMUNICATION NETWORKS (PTN)

-

SIGNALLING BETWEEN PRIVATE TELECOMMUNICATION EXCHANGES

-

GENERIC FUNCTIONAL PROTOCOL FOR THE SUPPORT OF SUPPLEMENTARY SERVICES

(QSIG-GF)

March 1992

Brief History

This Standard is one of a series of ECMA Standards defining services and signalling protocols applicable to Private Telecommunication Networks (PTNs). The series uses ISDN concepts as developed by CCITT and conforms to the framework of Standards on Open Systems Interconnection as defined by ISO. It has been produced under work items 5.1.2.2 and 5.1.3 of the supplement to ITSTC Memorandum M-IT-05 (Issue 1, November 1989) with the intention of submitting to ETSI as a proposed ETS.

This Standard defines the signalling protocol for use at the Q reference point between two PTNXs for the transport of protocol information as part of Supplementary services and/or Additional Network Features (ANFs) within a Private Telecommunication Network.

The generic functional procedures provide a flexible and open ended approach to the provision of Supplementary service and ANF protocols. These procedures provide:

- generic protocols which may be utilised in the provision of Supplementary services and ANFs, both related to existing calls and separate from existing calls where appropriate to the capability required;
- a dialogue identification protocol to enable Supplementary service or ANF information flows to be tied together to form a dialogue;
- Supplementary service and ANF transparency across a PTN, whereby transit PTNXs need have no knowledge of the capability provided to the PTN user or PTN itself unless involved in the provision of that capability; and
- the capability for standardised and manufacturer specific capabilities to coexist in both single and multi-vendor PTNs.

The protocol defined in this Standard is based upon that described in CCITT Recommendation Q.932 [Blue Book, 1988] including subsequent revisions during the 1989-1992 Plenary period and draft ETS 300196 under preparation in ETSI.

This Standard is based on the practical experience of ECMA member companies and results from their active and continuous participation in the work of ISO, CCITT, ETSI and other national and international standardisation bodies. It represents a pragmatic and widely based consensus.

This Standard has been submitted to ETSI for adoption as an ETS.

Adopted as an ECMA Standard by the General Assembly, February 1992.

Table of Contents

1 Scope

This ECMA Standard defines the signalling protocol for the control of Supplementary services and Additional Network Features (ANFs) at the Q reference point. The protocol is part of signalling system ECMA QSIG. The Q reference point exists between Private Telecommunication Network Exchanges (PTNX) connected together within a Private Telecommunication Network (PTN) and is defined in Standard ECMA-133. Detailed procedures applicable to individual Supplementary services and ANFs are beyond the scope of this Standard and will be specified by other ECMA Standards for those services which ECMA standardises and by individual manufacturers for proprietary services using the capabilities defined in this Standard.

ECMA-143 defines the Layer 3 protocol for circuit-switched call control at the Q reference point. This Standard defines additional protocol procedures, to be used in conjunction with those defined in ECMA-143 for the control of Supplementary services and ANFs. The protocol defined in this Standard can also be used for the transport of Manufacturer Specific Information (MSI) between PTNXs.

NOTE 1

Typical examples of the application of these generic functional procedures to some Supplementary services are provided in Annex A, for explanatory and illustrative purposes only.

NOTE 2

Other ECMA standards define protocols for use at other reference points, notably at the S reference point between a Terminal Equipment (TE) and a PTN.

NOTE 3

Specific Supplementary services and Additional Network Features may require additional information transfer mechanisms which are service or feature specific and are beyond the scope of this Standard.

2 Conformance

In order to conform to this Standard, a PTNX shall satisfy the requirements identified in the Protocol Implementation Conformance Statement (PICS) proforma in Annex I.

3 References

ECMA-133 (1989)	Reference Configurations for Calls Through Exchanges of Private Telecommunication Networks.
ECMA-142 (1990)	Specification, Functional Model and Information Flows for Control Aspects of basic Services in Private Telecommunication Networks.
ECMA-143 (1990)	Layer 3 Protocol for Signalling Between Exchanges of Private Telecommunication Networks for the Control of Circuit Switched Calls.
ENV 41007 (1989)	CENELEC pre-Standard: Definitions of Terms in Private Telecommunication Networks
CCITT Rec. I.112 (1988)	Vocabulary of Terms for ISDNs
CCITT Rec. I.210 (1988)	Principles of Telecommunication Services Supported by an ISDN and the Means to Describe Them
CCITT Rec. X.208 (1988)	Specification of Abstract Syntax Notation One (ASN.1)
CCITT Rec. X.209 (1988)	Encoding Rules for Abstract Syntax Notation One (ASN.1)
CCITT Rec. X.219 (1988)	Remote Operations Model, Notation and Service
CCITT Rec. X.229 (1988)	Remote Operations Protocol Specification

4 Definitions

For the purposes of this Standard, the following definitions apply:

4.1 External definitions

This Standard uses the following terms defined in other documents:

Connection	(ENV 41007)
Link	(ENV 41007)
Private	(ENV 41007)
Private Telecommunication Network Exchange (PTNX)	(ENV 41007)
Service	(CCITT Rec. I.112)
Signalling	(CCITT Rec. I.112)
Terminal, Terminal Equipment	(ENV 41007)
User	(ECMA-142)

4.2 Additional Network Feature (ANF)

A capability provided by a PTN, not generally directly to a User, over and above that of the Basic call.

4.3 Adjacent PTNX

A PTNX as considered from another PTNX to which it is directly connected via one or more inter-PTNX links.

4.4 Application Protocol Data Unit (APDU)

A sequence of data elements exchanged between peer application layer entities, e.g. DSE APDUs and ROSE APDUs

4.5 Call, Basic call

An instance of the use of a basic service.

4.6 Call independent signalling connection

A signalling connection established between SS-Control entities located in different PTNXs that does n have an associated user-information connection.

4.7 Call independent

A property of information which is conveyed across the Q reference point in a message which does not use a call reference which has an associated user-information connection (that is, using a Connectionless or Connection oriented transport mechanism as defined in 7.2 or 7.3).

4.8 Call related

A property of information which is conveyed across the Q reference point in a message which uses a call reference which has an associated user-information connection.

4.9 Connection oriented

Communication between peer protocol entities by means of a connection or association established by an underlying layer.

4.10 Connectionless

Communication between peer protocol entities by means of an unacknowledged, unidirectional transport mechanism provided by an underlying layer.

4.11 Coordination Function

An entity which provides coordination between various SS-Control entities, ROSE, DSE, GFT-Control and Call Control for different Supplementary services (see clause 6).

4.12 Destination PTNX

In the context of a single one-way exchange of information between two SS-Control entities, the PTNX where the receiving SS-Control entity is located.

4.13 DSE APDU

An APDU defined by the Dialogue Service Element.

4.14 Dialogue Service Element (DSE)

A service element which provides services to SS-Control via the Coordination Function that associate ROSE APDUs which are not implicitly associated by an underlying network layer connection.

4.15 ECMA QSIG

The generic name given to the signalling protocol that exists conceptually at the 'Q' reference point and is defined in this and other ECMA standards. This protocol is visible and indirectly testable at the 'C' reference point (see Appendix A of ECMA-133).

4.16 End PTNX

In the context of a particular call, an Originating or Terminating PTNX. It can also be a Gateway PTNX, dependent on the capabilities of the signalling system being interworked (i.e. unless it transports APDUs unchanged to or from the other signalling system).

4.17 Gateway PTNX

Sub-clause 5.5 of ECMA-143 shall apply. Dependent on the capabilities of the signalling system being interworked by the Gateway PTNX, it can act as a Transit or an End PTNX in the context of the Supplementary services APDUs. That is, it can either transport the APDUs unchanged to or from the other signalling system, perhaps embedded in some other protocol unit, or process the APDUs and perform an interworking function of the information flows and encoding of the Supplementary service concerned.

4.18 Generic Functional Transport Control (GFT-Control) entity

The entity that exists within a PTNX and provides a range of services (defined in clause 6) to SS-Control, ROSE and DSE via the Coordination Function.

4.19 Incoming side

In the context of a Call independent signalling connection, the Side which receives the request for connection establishment from the Preceding PTNX.

4.20 Interpretation APDU

An APDU defined by the Coordination Function.

4.21 Invocation

A request by a SS-Control entity to perform an operation in a remote SS-Control entity.

4.22 Link significance

A property of a Facility information element which does not contain a Network Facility Extension octet group. It indicates that the element has only significance on a single inter-PTNX link - i.e. only between two Adjacent PTNXs.

4.23 Mistyped

A property of an APDU whose structure does not conform to the structure defined in clause 11 of this Standard or the structure defined for a particular Supplementary service.

4.24 Network significance

A property of a Facility information element which includes a Network Facility Extension octet group. It indicates that the element has significance between two PTNXs which are not necessarily Adjacent.

4.25 Next PTNX

An Adjacent PTNX to which an APDU is to be sent in the context of an existing signalling connection (related to a call or independent of a call).

4.26 Notification

A piece of protocol information which has the following properties:

- it is intended to be delivered only to terminals and is therefore passed on transparently by PTNXs;
- it does not cause a change of state on either side of the Q reference point;
- it represents a one-way flow of information that requires no response; and
- it provides additional information that can be discarded without the need for significant error recovery if it is unrecognised by the terminal.

4.27 Originating PTNX

Sub-clause 5.4 of ECMA-143 shall apply. In addition, the term is also applied to a PTNX which originates a Call independent signalling connection.

4.28 Outgoing side

In the context of a Call independent signalling connection, the Side which sends the request for connection establishment to the Next PTNX.

4.29 Preceding PTNX

Sub-clause 5.6 of ECMA-143 shall apply. In addition, the term is also applied in a similar way to a PTNX participating in a Call independent signalling connection.

4.30 Protocol Control

An entity which exists within a PTNX and provides a range of services (defined in clause 6) to the Generic Functional Transport Control entity.

4.31 ROSE APDU

An APDU defined by the Remote Operations Service Element (ROSE) - see 11.3.

4.32 Side

The Protocol Control entity within a PTNX at one end of an inter-PTNX link.

4.33 Source PTNX

In the context of a single one-way exchange of information between two SS-Control entities, the PTNX where the sending SS-Control entity is located.

4.34 Subsequent PTNX

Sub-clause 5.6 of ECMA-143 shall apply. In addition, the term is also applied in a similar way to a PTNX participating in a Call independent signalling connection.

4.35 Supplementary service

Section 2.4 of CCITT Recommendation I.210 shall apply.

For the purpose of this Standard, ANFs shall be regarded as Supplementary services.

4.36 Supplementary Services Control (SS-Control) entity

An entity that exists within a PTNX and provides the procedures associated with the support of a particular Supplementary service.

4.37 Terminating PTNX

Sub-clause 5.4 of ECMA-143 shall apply. In addition, the term is also applied to a PTNX which terminates a Call independent signalling connection.

4.38 Transit PTNX

Sub-clause 5.4 of ECMA-143 shall apply. In addition, the term is also applied to a PTNX which participates in the provision of a Call independent signalling connection, but does not originate or terminate that connection.

4.39 Unrecognised

A property of a message, information element, APDU or operation value whose type identifier is not one supported by the Destination PTNX.

5 List of acronyms

AE	Application Entity
ANF	Additional Network Feature
APDU	Application Protocol Data Unit
ASN.1	Abstract Syntax Notation One
DSE	Dialogue Service Element
DSS1	Digital Subscriber Signalling no. 1
FIE	Facility information element
GFT	Generic Functional Transport
ICD	International Code Designator
MSI	Manufacturer Specific Information
NFE	Network Facility Extension
PICS	Protocol Implementation Conformance Statement
PTN	Private Telecommunication Network
PTNX	Private Telecommunication Network Exchange
RO	Remote Operations
ROSE	Remote Operations Service Element
SS	Supplementary Service

6 General principles

The generic functional protocol defined in this Standard provides the means to exchange signalling information for the control of Supplementary services over a PTN. It does not by itself control any Supplementary service but rather provides generic services to specific SS-Control entities. Procedures for individual Supplementary services based on these generic procedures are defined in other standards or may be manufacturer-specific.

The generic functional protocol operates at the Q reference point between two PTNXs in conjunction with a Layer 3 protocol for Basic call control (ECMA-143). Together these use the services of the Data Link Layer.

The generic functional protocol provides mechanisms for the support of Supplementary services which relate to existing basic calls or are entirely independent of any existing basic calls. In performing a Supplementary service, whether Call independent or Call related, use may be made of both the Call related (7.1) and Call independent (7.2 and 7.3) information transfer procedures.

If a particular Supplementary service comprises Call related and Call independent information transfe procedures or relates to several basic calls at the same time it is - for the purpose of this Standard - deemed t consist of separate instances of Call related (one for each call) and Call independent services respectively. Th combined use of two or more instances of Call related and/or Call independent procedures in support of a particular Supplementary service is outside the scope of this Standard.

6.1 Application Association

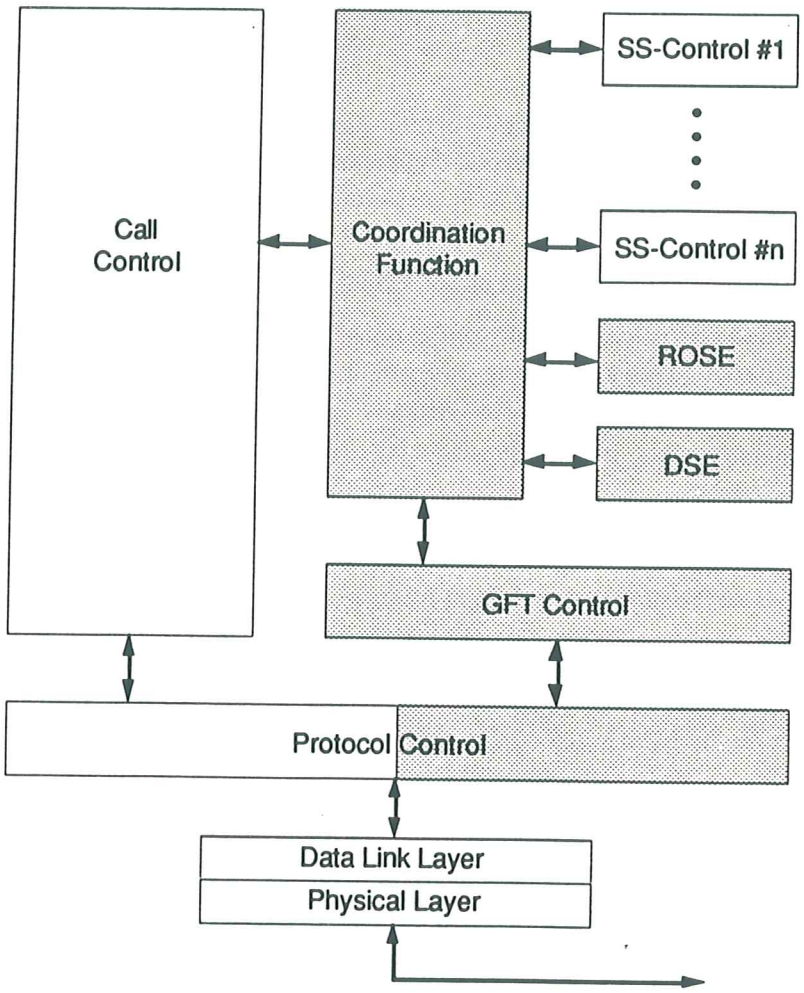
The use of explicit Application Association control by means of the Association Control Service Elemen (ACSE, X.217/227) is beyond the scope of this Standard. However, Supplementary service operation require an association between the respective peer SS-Control entities. This Standard provides two mean by which this association can be implicitly achieved:

(a) by the network layer connection in the case of Call related connections and for call-independen signalling connections; or

(b) by the application layer dialogue service, in which case the association is independent of t underlying network layer connections and can use a combination of different mechanisms, includi Call independent Connectionless information transfer.

6.2 Protocol Model

Figure 1 shows the conceptual model for the generic functional protocol and its relation to the Basic cal model defined in ECMA-143.



NOTE

The capabilities defined in this Standard are indicated by shading, i.e. GFT-Control, DSE, ROSE and extensions to Protocol Control. Part of the functions of the Coordination Function are also defined in this Standard, but the remainder of this element governs Supplementary service specific interactions which are beyond the scope of this Standard.

Figure 1 - ECMA-165 Protocol Model

At the top layer (the application layer) the actual Supplementary service protocol operates between peer Supplementary Services Control (SS-Control) entities which are service-specific. The operation of specific SS-Control entities is beyond the scope of this Standard.

SS-Control entities use the services of the Remote Operations Service Element (ROSE) and the Dialogue Service Element (DSE) at the application layer via the Coordination Function. These entities use the services of Generic Functional Transport Control (GFT-Control) at the network layer via the Coordination Function. GFT-Control uses the services of Protocol Control at the network layer.

The Remote Operations Service Element (ROSE) is defined in X.219.

NOTE 4

In the application of ROSE for the support of Supplementary services in ECMA-QSIG, the underlying services used by ROSE are those provided by GFT-Control and not those provided by the Association Control Service Entity (ACSE) and the Reliable Transport Service Entity (RTSE).

The Dialogue Service Element (DSE) provides a means of associating ROSE APDUs which are not implicitly associated by an underlying network layer connection.

The Coordination Function provides coordination between GFT-Control, the various SS-Control entities ROSE, DSE and Call Control for different Supplementary services. The relationships it coordinates ar beyond the scope of this Standard. It also provides functions to support the handling of unrecognise APDUs.

GFT-Control provides two distinct types of service via the Coordination Function:

- transport services for the carriage of Notifications, ROSE APDUs and DSE APDUs between SS-Contro entities in different PTNXs, including transparent relaying through Transit PTNXs. These services ca be related to a Call or independent of a Call; and,
- establishment and release of Call independent signalling connections.

Protocol Control is an extension of the existing ECMA-143 Protocol Control entity. It provides services t GFT-Control for:

- the transport of APDUs between Adjacent PTNXs;
- the establishment and release of signalling connections (Call independent Connection oriented servic between Adjacent PTNXs.

This entity builds on the ECMA-143 (Basic call) Protocol Control in the following way:

- the Call related transfer of APDUs uses the call reference established for the call by ECMA-14 Protocol Control. This can be either by:
  - . the combination of Basic call control information and APDUs in the same ECMA-143 message i they appear concurrently at the Protocol Control service access points; or,
  - . the transfer of APDUs in an ECMA-QSIG-GF message associated with the call reference, when n Call Control primitive appears at the Protocol Control service access point.
- Call independent signalling connections use the call reference mechanism of ECMA-143 Protocol Control and some of the messages and procedures.

6.3 Application of the protocol model to communication between SS-Control entities in non-Adjacent PTNXs

Figure 2 shows the application of the protocol model to the case where communication occurs between S Control entities in two PTNXs via a single Transit PTNX. It may be applied to communication via mo than one Transit PTNX by simple replication.

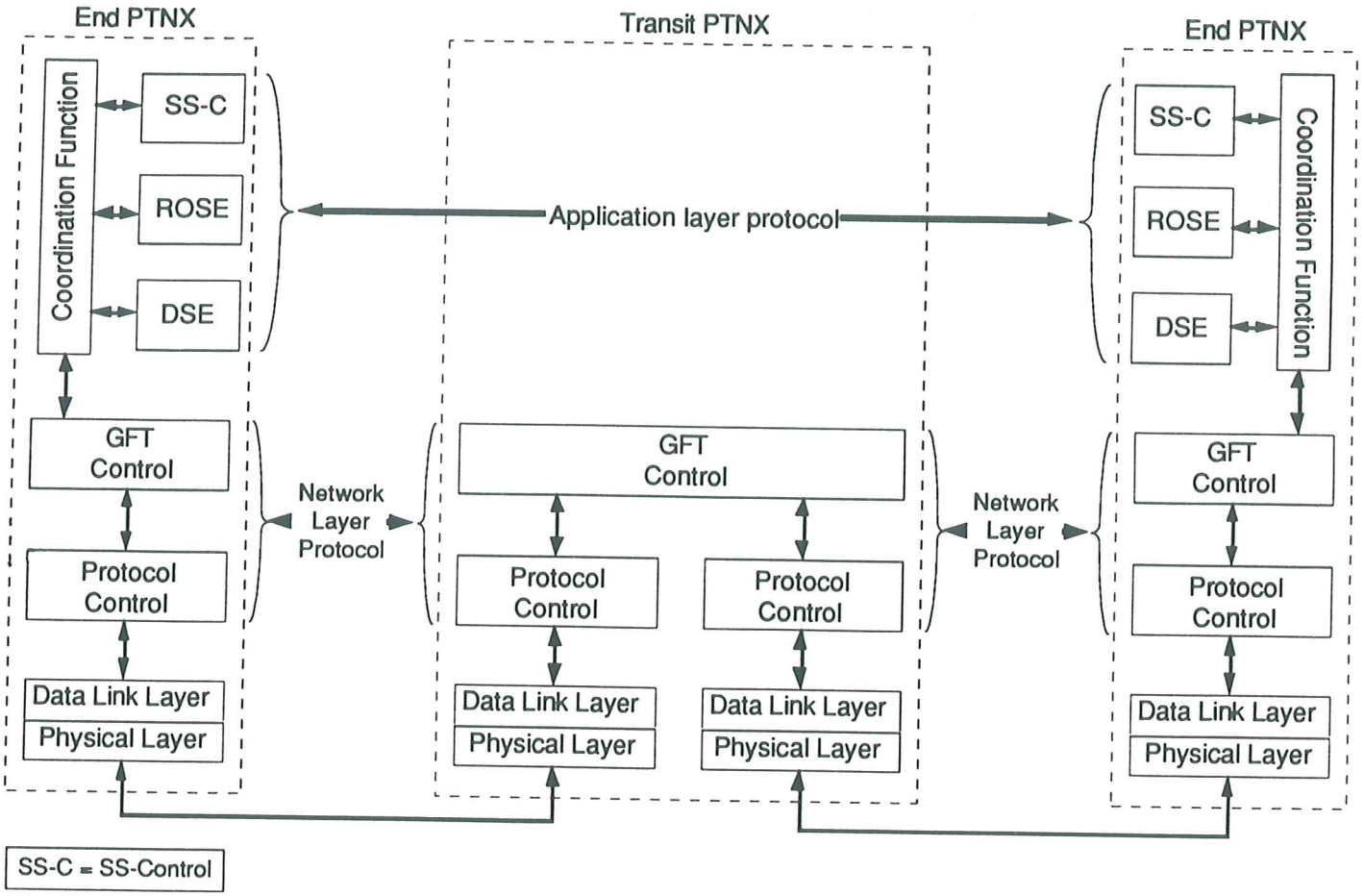


Figure 2 - Application of the protocol model to communication between non-Adjacent PTNXs

In figure 2, relaying functions at the Transit PTNX are performed by GFT-Control.

If communication is Call related, each of the PTNXs in which the SS-Control entities are located may be either an End or a Transit PTNX. For simplicity, the Call Control entities are not shown.

If communication is in the context of a Call independent signalling connection, one of the PTNXs in which the SS-Control entities are located is the Originating PTNX and the other is the Terminating PTNX.

6.4 Services provided by ROSE

ROSE provides a set of services to SS-Control to support the ROSE protocol. Primitives for these services are specified in X.219 and relate to the following ROSE APDUs: Invoke, ReturnResult, ReturnError and Reject.

6.5 Services provided by DSE

DSE provides the following services to SS-Control via the Coordination Function:

- **Dialog Begin** Request/Indication
- **Dialog Continue** Request/Indication
- **Dialog End** Request/Indication
- **Dialog Abort** Request/Indication

These services are used for creating and terminating a Dialogue which associates peer SS-Control entities and for exchanging ROSE APDUs within such an association.

6.6 Services provided by GFT-Control

This entity provides the following services to SS-Control, ROSE and DSE via the Coordination Function.

6.6.1 Connection oriented services

The following services are provided:

- **GF-Setup** Request/Indication/Response/Confirm
- **GF-Release** Request/Indication
- **GF-Reject** Request/Indication

These services can contain one or more APDUs.

These services are used for the control of the establishment and clearing of a Call independent signalling connection between the PTNXs in which the peer SS-Control entities exist.

- **GF-Data** Request/Indication

This service contains one or more APDUs.

This service is used for the conveyance of APDUs on a signalling connection (Call related or C independent) between the PTNXs in which the peer SS-Control entities exist.

6.6.2 Connectionless transport services

The following service is provided:

- **GF-Unitdata** Request/Indication

This service contains one or more APDUs.

This service is used to effect the transport of APDUs between two peer SS-Control entities without the use of a network layer connection. It is an unconfirmed service.

6.6.3 Notification services

The following service is provided to SS-Control via the Coordination Function:

- **GF-Notify** Request

This service is used to effect the transport of notifications associated with the network layer signalling connection of a Call.

6.7 Services provided by Protocol Control to GFT-Control

The following services are provided:

6.7.1 Connection oriented transport services

The following services provide the Connection oriented network service for Call independent Supplementary service control:

- **PC-Setup** Request/Indication/Response/Confirmation
- **PC-Release** Request/Indication
- **PC-Reject** Request/Indication

NOTE 5

These primitives are similar to the primitives defined in 6.2 of ECMA-143 for provision of services to Call Control.

These services are used for the establishment and clearing of Call independent signalling connections between Adjacent PTNXs. These primitives may include APDUs.

The following service is provided to GFP-Control:

- **PC-Data** Request/Indication

This service contains one or more APDUs.

The service is used for the conveyance of APDUs between Adjacent PTNXs in association with a Basic call or Call independent signalling connection.

6.7.2 Connectionless transport service

The following service is provided to GFT-Control:

- **PC-Unitdata** Request/Indication

This service contains one or more APDUs.

This service is used to effect the transport of APDUs between two Adjacent PTNXs without the use of a network layer connection.

6.7.3 Notification services

The following service is provided to GFT-Control:

- **PC-Notify** Request/Indication

This service is used to effect the transport of notifications between Adjacent PTNXs in association with the network layer signalling connection of a Call.

6.8 Services required of the Data Link Layer

The services required by Protocol Control are as specified in 6.3 of ECMA-143.

7 Protocol Control and GFT-Control Requirements

7.1 Call related Procedures for the transport of APDUs

This clause describes the procedures required to transport Call related APDUs.

NOTE 6

The APDUs need not directly relate to the provision or state of the Call which provides the signalling connection over which the information is carried. If the Call fails and the connection is cleared down for any reason, APDUs that are in the process of being sent may never reach their destination. In such a case, the APDUs will be discarded. It is the responsibility of the Supplementary service protocol to cater for this eventuality.

7.1.1 Protocol Control requirements

7.1.1.1 Sending the Facility information element

When requested by GFT-Control, the Facility information element may be sent at any time during a call (i.e. where a call reference exists) subject to the following conditions:

- If a call establishment or a call clearing message that may contain a Facility information element (see clause 10) is to be sent in the context of a Basic call, the Facility information element shall be included in that message.
- If no suitable call establishment or call clearing message is to be sent, the Facility information element shall be carried in a FACILITY message.

Three exceptions where the Facility information element shall not be sent and an indication of transmission failure given to GFT-Control are:

- when no response has been received to a previously sent SETUP message (as defined in 8.1 of ECMA-143)
- when the Facility information element is of network significance and a call clearing message has already been sent or received on the inter-PTNX link; or

- if no call establishment or clearing message is to be sent and a RELEASE or RELEASI COMPLETE message has been sent or received on the inter-PTNX link.

NOTE 7

Further actions by the GFT-Control entity in such a situation (e.g. if the Facility information elemen was received from the Subsequent PTNX) are implementation dependent. In designing protocols fo Supplementary services in a PTN, account should be taken of the fact that an end to end Call relate signalling relationship cannot be guaranteed until the receipt of the first end to end Basic cal message.

NOTE 8

In the case where the Facility information element is sent to a PTNX which does not conform to thi Standard, the Facility information element will be discarded by that PTNX and a STATUS messag (see clause 11 in ECMA-143) may be received. The STATUS message will indicate that either: th Facility information element was unrecognised; or, that the message (FACILITY) was unrecognis In such cases, the recovery action, if any, is an implementation specific matter.

7.1.1.2 Receiving the Facility information element

A PTNX receiving a Facility information element in a valid call clearing or call establishmen message (see clause 10) or a FACILITY message shall pass the entire contents of that informatio element to GFT-Control.

7.1.2 Generic Functional Transport Control requirements

7.1.2.1 Actions at a Source PTNX

On receipt of a request for APDU transport from the Coordination Function, the APDUs to b transported shall be encoded in a Facility information element, as defined in 11.3.3.

APDUs may be of two basic types:

- Those which have only Link significance, i.e. over a *single* link of the PTN, between two Adjacen PTNXs; or,
- Those which have Network significance, between two PTNXs in the PTN which are *no necessarily* adjacent, and which can be, but need not be, the End PTNXs involved in the call.

If the APDUs have link significance, the Network Facility Extension (NFE), defined in 11.3.3.1, neec not be included in the Facility information element (although it may optionally be included, explicitl identifying the Adjacent PTNX);

If the APDUs have network significance, the NFE shall be included, encoded as described in table 1.

NOTE 9

The Facility information element may contain one or more APDUs. If more than one APDU i contained in a single Facility information element, they will all be processed by the Destinatio PTNX. How and if these requests are related is beyond the scope of this Standard.

Table 1 - Encoding of NFE

Case No.	Communication between ..	Encoding of sourceEntity	Encoding of sourceEntityAddress	Encoding of destinationEntity	Encoding of destinationEntityAddress
1	End PTNX (origination or destination) → End PTNX (destination or origination, depending on direction of FIE)	endPTNX (NOTE 2)	NOT Included	endPTNX	NOT Included
2	End PTNX (origination or destination) → addressed PTNX	endPTNX (NOTE 2)	NOT Included	anyTypeOfPTNX	PTNX address
3	End PTNX (origination or destination) → Next PTNX which understands contents	endPTNX (NOTE 2)	NOT Included	anyTypeOfPTNX	NOT Included
4	Transit PTNX → Destination or Originating PTNX (depending on direction of FIE)	anyTypeOfPTNX	PTNX Address	endPTNX	NOT Included
5	Transit PTNX → addressed PTNX	anyTypeOfPTNX	PTNX Address	anyTypeOfPTNX	PTNX Address
6	Transit PTNX → Next PTNX which understands contents	anyTypeOfPTNX	PTNX address	anyTypeOfPTNX	NOT Included

NOTE 10

*In principle, an End PTNX can encode the **sourceEntity** element as **anyTypeOfPTNX**, but only if the **sourceEntityAddress** element is included. This could be used to unambiguously identify the End PTNX and avoid any interception of a response APDU by a Transit PTNX trying to act as an End PTNX.*

The Facility information element shall be delivered to Protocol Control.

7.1.2.2 Actions at a Receiving PTNX

A PTNX receiving a Facility information element (in one of the messages listed in clause 10) shall determine whether or not it is the Destination PTNX for that Facility information element.

It shall accomplish this by examination of the header of the Facility information element.

If the Facility information element header does not contain an NFE, the PTNX shall become the Destination PTNX for that Facility information element.

If the received Facility information element contains an NFE, the PTNX shall determine whether it is a Transit PTNX or End PTNX in the context of the Basic call and act as described below.

If the received Facility information element contains more than one NFE, the PTNX shall process the first NFE as a valid NFE and discard all others.

7.1.2.2.1 End PTNX actions

If the receiving PTNX is an End PTNX, and the encoding of the received NFE complies with the encoding and structure defined in clause 11, the following actions shall apply:

- if the **destinationEntity** element of the NFE indicates **endPTNX** or **anyTypeOfPTNX** and no **destinationEntityAddress** element is included, it shall become the Destination PTNX for that Facility information element;
- if the **destinationEntity** element of the NFE indicates **anyTypeOfPTNX** and includes a **destinationEntityAddress** element, it shall compare the received address to its own address. If the addresses match, the PTNX shall become the Destination PTNX for that Facility information element;
- if the **destinationEntity** element of the NFE indicates **endPTNX** and erroneously includes a **destinationEntityAddress** element, the PTNX shall become the Destination PTNX for that Facility information element;
- in all other cases, the received Facility information element shall be discarded.

If the received NFE does not conform to the encoding and structure defined in clause 11, the entire Facility information element shall be discarded.

7.1.2.2.2 Transit PTNX actions

If the receiving PTNX is a Transit PTNX, and the encoding of the received NFE complies with the encoding and structure defined in clause 11, the following actions shall apply:

- if the **destinationEntity** element of the NFE indicates **anyTypeOfPTNX** and a **destinationEntityAddress** element is included, it shall compare the received address to its own address. If the addresses match, the PTNX shall become the Destination PTNX for that Facility information element;
- if the **destinationEntity** element of the NFE indicates **anyTypeOfPTNX** and no **destinationEntityAddress** element is included, the PTNX may become the Destination PTNX for that Facility information element if it understands the contents;
- if the **destinationEntity** element of the NFE indicates **endPTNX** and erroneously includes a **destinationEntityAddress** element, the PTNX shall ignore the contents of the **destinationEntityAddress** field and treat the contents of the Facility information element as if only the **destinationEntity** element was present;
- if the **destinationEntity** element of the NFE indicates **endPTNX,** and the Transit PTNX is capable of acting as an End PTNX for all services indicated in the Facility information element, it may become the Destination PTNX for that Facility information element.;

  NOTE 11

  In this case, the source of the information will have no knowledge that the information has been intercepted, as the Transit PTNX will act as if it were an End PTNX. This may occur, for example, when a PTNX at a PTN numbering domain boundary wishes to translate numbering information contained within an APDU.

- in all cases where the PTNX does not become the Destination PTNX, the Facility information element shall be passed on unchanged to the Next PTNX.

If the received NFE does not conform to the encoding and structure defined in clause 11, the entire Facility information element shall be discarded and no Facility information element shall be passed on to the Next PTNX.

NOTE 12

Processing of a Facility information element at a Transit PTNX does not preclude another Facility information element, which may have similar contents to that received by the Transit PTNX, being sent to the Next PTNX as a result of that internal processing.

7.1.2.3 Actions at a Destination PTNX

All APDUs shall be delivered to the appropriate SS-Control entity via the Coordination Function at a Destination PTNX in the order in which they were received in the Facility information element.

7.1.2.4 Dynamic description (SDL) of Generic Functional Transport Control

Figures 4 and 5 show SDL diagrams describing the actions of the GFT-Control entity, as specified in 7.1.2. Figure 3 is the key to these SDL diagrams.

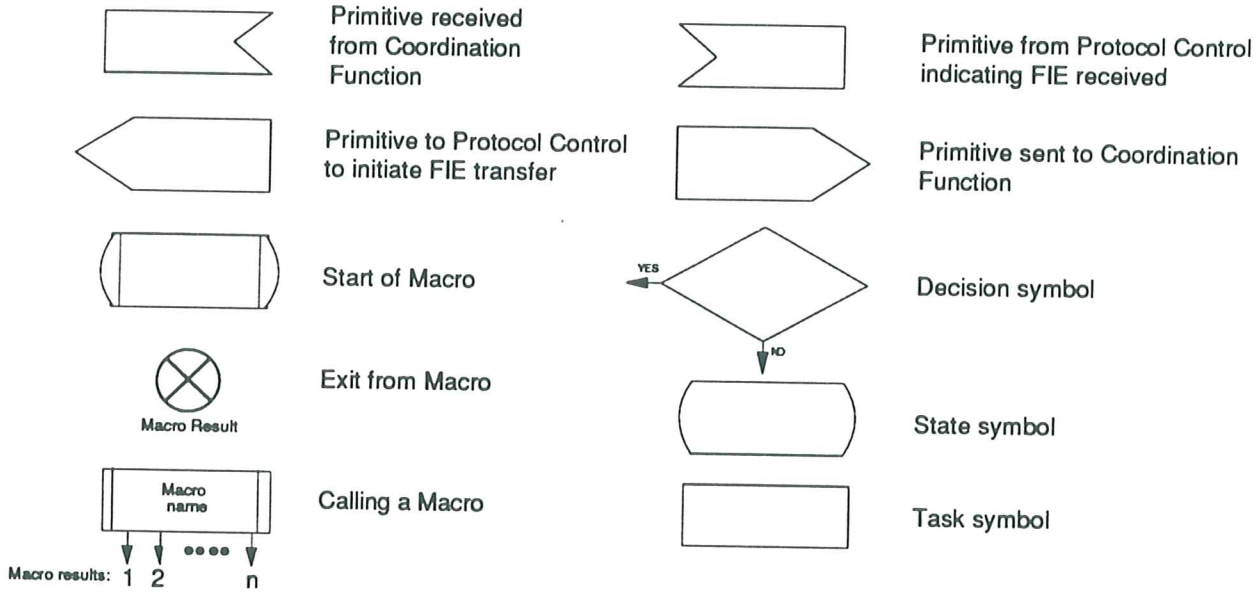
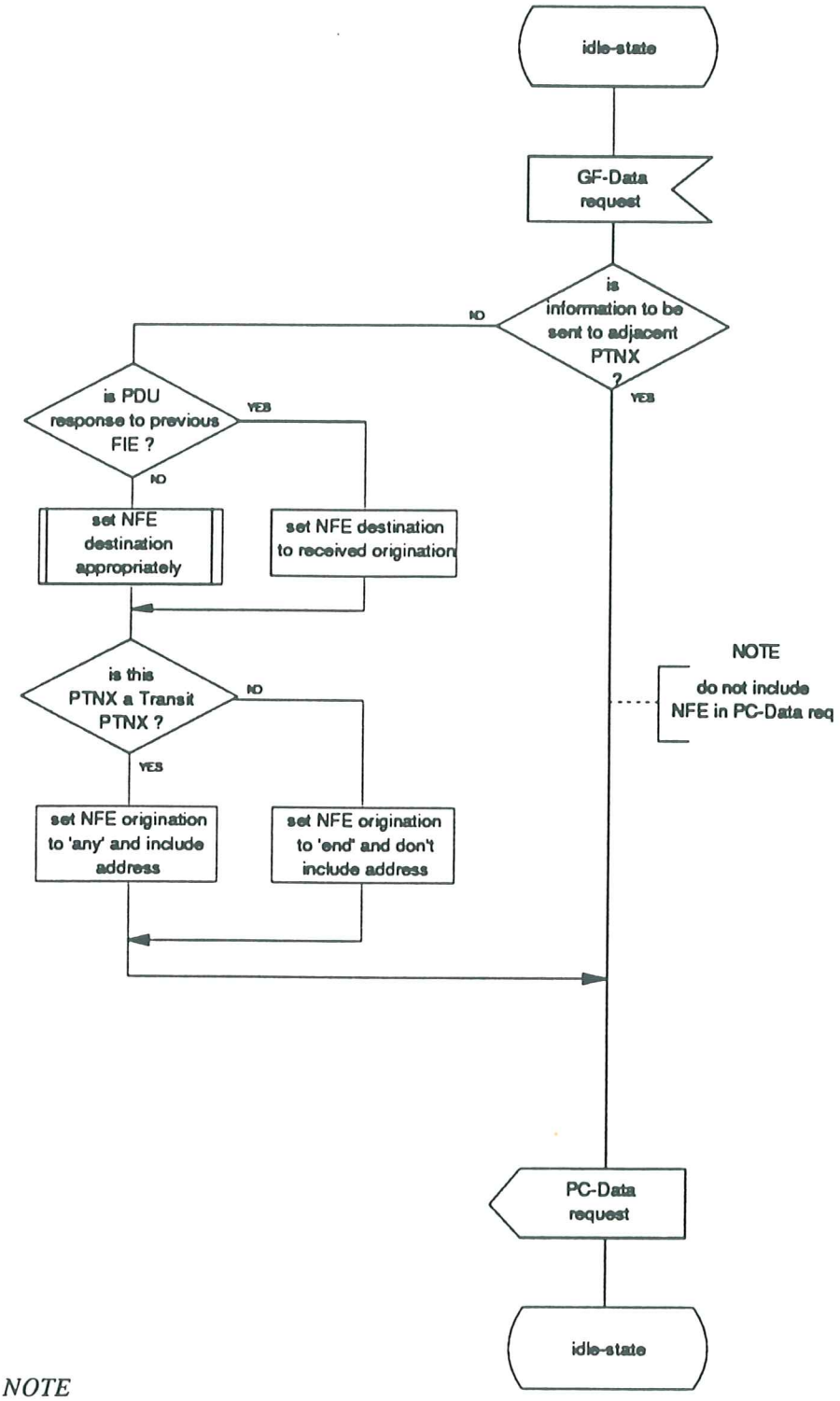


Figure 3 - Key to SDL Diagrams in figures 4 and 5



NOTE

In principle, including the NFE to explicitly identify the Adjacent PTNX is not precluded by the procedures in this Standard.

Figure 4 - Actions at a Source PTNX (sheet 1 of 2)

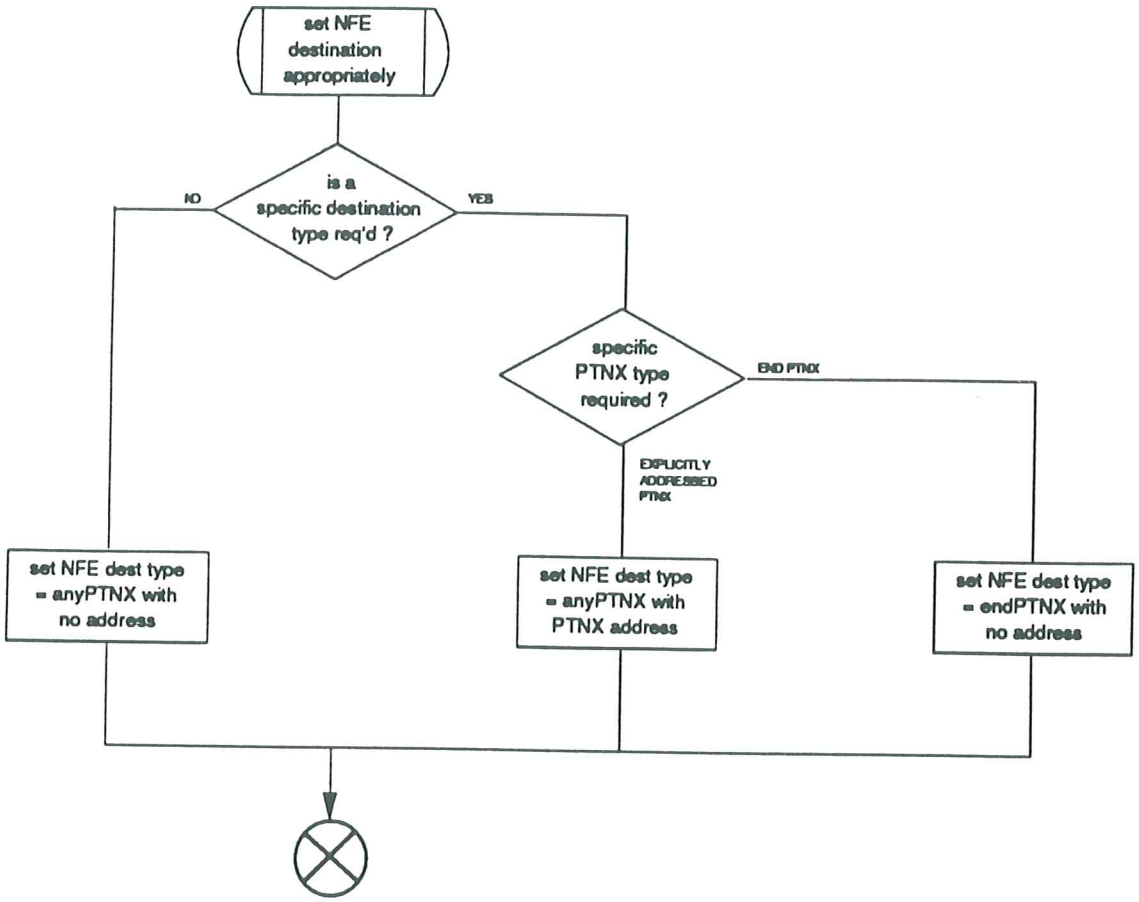
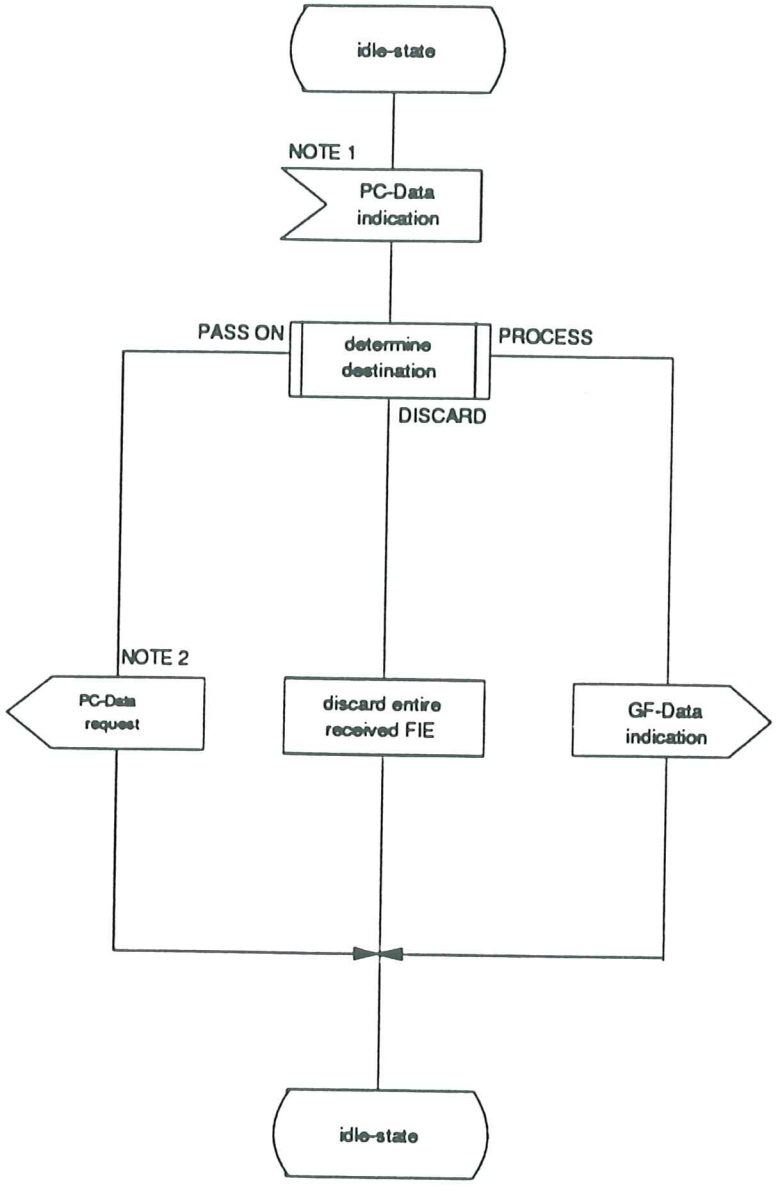


Figure 4 - Actions at a Source PTNX (sheet 2 of 2)



NOTE 1

This primitive indicates that Protocol Control has received a Facility information element from the Adjacent PTNX in the direction of the Source PTNX.

NOTE 2

This primitive to the Protocol Control entity causes a Facility information element to be sent to the Next PTNX in the direction of the Destination PTNX.

Figure 5 - Actions at a Receiving PTNX (sheet 1 of 2)

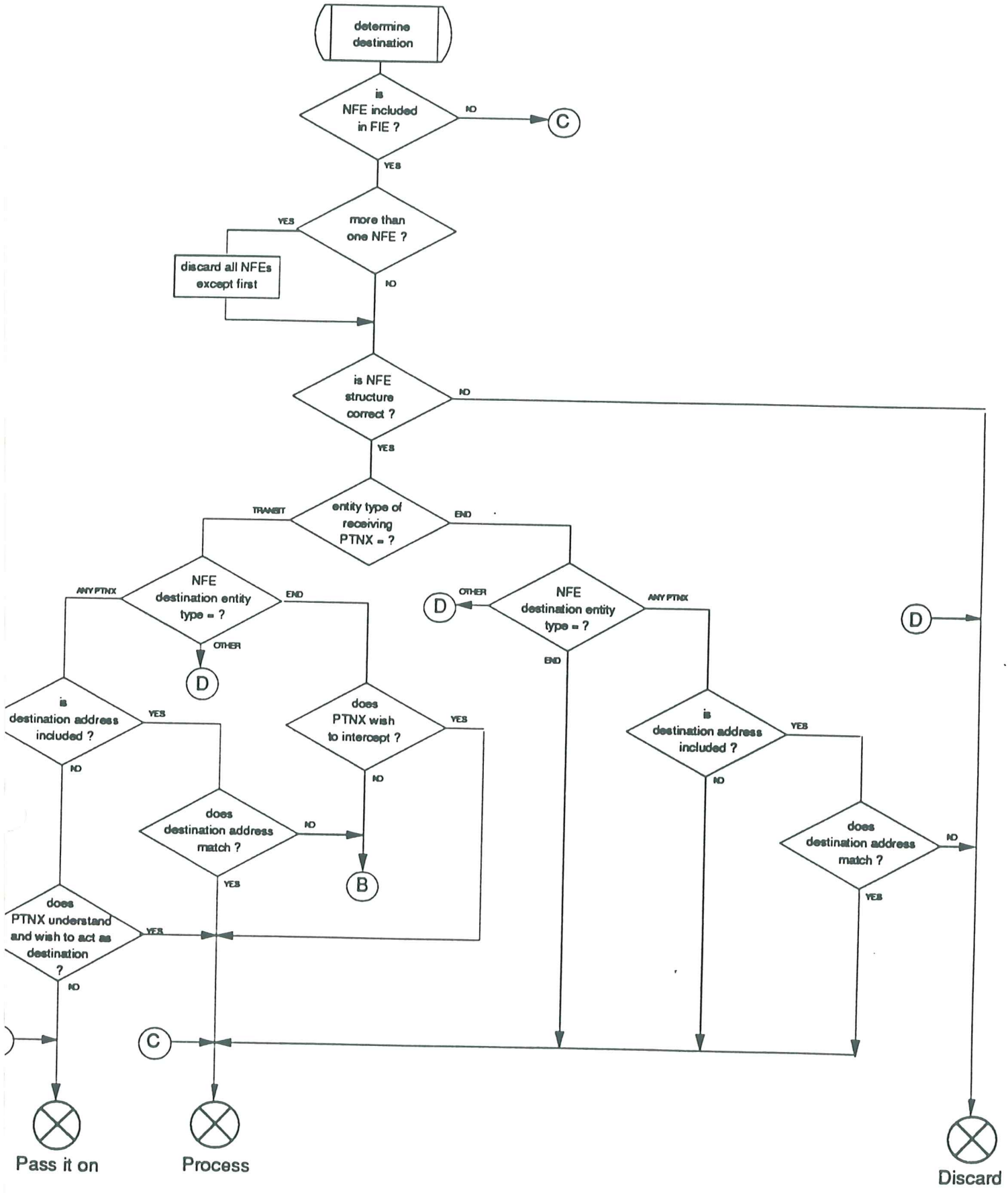


Figure 5 - Actions at a Receiving PTNX (sheet 2 of 2)

7.2 Connectionless APDU Transport Mechanism

The procedures defined in this clause describe a Connectionless network layer service which provide: APDU transfer between PTNXs outside the context of a call.

7.2.1 Protocol Control requirements

7.2.1.1 Requirements for sending a Connectionless message

When requested by GFT-Control to send APDUs using Connectionless transport, Protocol Contro shall first ensure that a Data Link connection exists on the relevant inter-PTNX link. If a Data Lin connection does not exist, Protocol Control shall establish a data link connection according to th procedures described in 7.1.1 of ECMA-143. Once this Data Link is established, Protocol Contro shall transfer the APDUs (encoded in a Facility information element) across the interface by sending FACILITY message (defined in 10.7) containing the Dummy call reference (defined in 11.2), and th Calling and Called party number information elements as provided by GFT-Control.

NOTE 13

In the case where the FACILITY message is sent to a PTNX which does not support Connectionles APDU transport, the FACILITY message will be discarded by that PTNX in accordance with 7.3 o ECMA-143 and a call clearing message containing the Dummy call reference can be returned. Th call clearing message shall be discarded as described in 7.2.1.2.

7.2.1.2 Requirements for receiving a Connectionless message

On receipt of a valid FACILITY message containing the Dummy call reference the Facilit information element shall be passed to GFT-Control.

If a FACILITY message containing the Dummy call reference contains any of the following errors, i shall be discarded:

- unrecognised information element which is encoded "comprehension required";
- missing mandatory information element; or,
- mandatory information element content error.

If a FACILITY message containing the Dummy call reference contains any unrecognised informatio elements that are not encoded "comprehension required"; or optional information elements w invalid contents, these information elements shall be discarded and the remainder of the FACILIT message processed as valid.

On receipt of any messages containing the dummy call reference, other than the FACILITY message the message shall be discarded.

7.2.2 GFT-Control requirements

7.2.2.1 Actions at a Source PTNX

On receipt of a request from the Coordination Function to send APDUs using Connectionless transport, accompanied by the address of the Destination PTNX, GFT-Control shall:

- if a route to the destination can be selected, select the appropriate inter-PTNX link based on the destination address given in the request from the Coordination Function and inform Protocol Control to send a FACILITY message which shall contain:
  - a Calling party number information element, identifying the address of the Source PTNX;
  - a Called party number information element identifying the address of the Destination PTNX; and,
  - a Facility information element which shall not contain an NFE.

- if no route to the Destination PTNX can be selected, ignore the request.

7.2.2.2 Actions at a Receiving PTNX

If a PTNX receives a FACILITY message containing the Dummy call reference on an ECMA QSIG link from an Adjacent PTNX, it shall examine the contents of the Called party number information element to determine whether or not the FACILITY message is to be terminated at that PTNX. If the Called party number identifies another PTNX, and the receiving PTNX can route the FACILITY message based on this Called party number, the FACILITY message (with contents as received) shall be sent on the appropriate ECMA QSIG link. If the Called party number information element contains an address identifying the receiving PTNX, it shall act as the Destination PTNX for the FACILITY message.

If a received FACILITY message containing the Dummy call reference contains a Called Party number information element that does not identify the receiving PTNX or a PTNX to which the FACILITY message can be passed on, the PTNX shall discard the FACILITY message.

NOTE 14

It is the responsibility of the appropriate specification for the Supplementary service utilising these transport procedures to ensure that the service can cope gracefully if the FACILITY message is discarded during routing.

7.2.2.3 Actions at a Destination PTNX

If the received FACILITY message is destined for the receiving PTNX, the contents of the Facility information element and the address of the Source PTNX shall be passed to the appropriate SS-Control entity via the Coordination Function.

NOTE 15

It is the responsibility of SS-Control (i.e. the specific Supplementary service) in the Destination PTNX to store the Calling party number information element to enable response to the service request to be made using a further Connectionless message.

If the received Facility information element contains an NFE, the receiving PTNX shall ignore the contents of that NFE.

7.3 Connection oriented APDU transport mechanism

The procedures in this clause describe a Connection oriented network layer service which provides APDU transfer between PTNXs outside the context of a call.

7.3.1 Protocol Control requirements

The description of the Protocol Control requirements for Connection oriented APDU transport uses a subset of the states defined in 6.4 of ECMA-143.

7.3.1.1 Actions in the Null state

When asked to initiate a Call independent signalling connection by GFT-Control, the Outgoing side Protocol Control shall:

- ensure that a Data Link connection exists on the relevant inter-PTNX link. If a Data Link connection does not exist, Protocol Control shall establish a Data Link connection according to the procedures described in 7.1.1 of ECMA-143;
- send a SETUP message on the appropriate inter-PTNX link which shall contain only:
  - a Call reference, selected according to 12.3 of ECMA-143;
  - a Sending complete information element, as defined in 12.5 of ECMA-143;

- a Bearer capability information element indicating the additional codepoints defined in 11.3.1 i.e. Coding standard indicating 'other international standard', Information transfer capability indicating 'unrestricted digital information', Transfer mode indicating 'circuit mode', and Information transfer rate indicating 'Call independent signalling connection';
- a Channel identification information element indicating 'no-channel' in the channel selection field and 'exclusive' in the preferred/exclusive field, as defined in 11.3.2;
- a Called party number information element containing a number at least sufficient to identify the Terminating PTNX;
- optionally, a Calling party number information element containing a number at least sufficient to identify the Originating PTNX;
- optionally, one or more Facility information elements; and
- optionally, a Transit counter information element as defined in 12.6 of ECMA-143.

- start timer T303; and,
- enter the Call initiated state.

On receipt of a SETUP message relating to establishment of a Call independent signalling connection the Incoming side shall:

- if the request is valid and can be processed, return a CALL PROCEEDING message to the Outgoing side, indicate the connection request to GFT-Control and enter the Incoming call proceeding state; or,
- if the request is invalid or cannot be accepted by the PTNX, return a RELEASE COMPLETE message to the Outgoing side, release the call reference and remain in the Null state.

7.3.1.2 Actions in the Call initiated state

On receipt of a CALL PROCEEDING message from the Incoming side, the Outgoing side shall stop T303, start timer T310, if applicable, and enter the Outgoing call proceeding state.

If no response is received from the Incoming side before timer T303 expires, the SETUP message may optionally be retransmitted and timer T303 restarted. If no response is received before timer T303 expires for a second time, the Outgoing side shall send a RELEASE COMPLETE message the Incoming side; inform GFT-Control of the failure of the signalling connection request; and enter the Null state.

NOTE 16

The RELEASE COMPLETE message should contain cause no. 102 "Recovery on Timer Expiry".

NOTE 17

If the Connection oriented procedures are not supported by a PTNX which receives a SETUP message requesting a Call independent signalling connection, it will respond with a call clearing message indicating, for example, that the Bearer capability cannot be provided or that the message has contained an information element content error. This will initiate connection release in accordance with 7.3.1.7.

7.3.1.3 Actions in the Incoming call proceeding state

When receiving an indication that the Call independent signalling connection is established from GFT-Control, the Incoming side shall: send a CONNECT message to the Outgoing side and either: enter the Active state, or start timer T313 and enter the Connect request state.

7.3.1.4 Actions in the Outgoing call proceeding state

On receipt of a CONNECT message from the Incoming side, the Outgoing side shall: stop timer T310 (if applicable), inform GFT-Control that the signalling connection is established, send a CONNECT ACKNOWLEDGE message to the Incoming side and enter the Active state.

If timer T310 expires, the Outgoing side shall indicate that the signalling connection request has failed to GFT-Control and initiate release of the connection as described in 7.3.1.7.

NOTE 18

The cause sent to the Incoming side should be no. 102 "Recovery on Timer Expiry".

7.3.1.5 Actions in the Connect request state

On receipt of a CONNECT ACKNOWLEDGE message, the Incoming side shall: stop timer T313 and enter the Active state.

If timer T313 expires before a CONNECT ACKNOWLEDGE message is received the Incoming side shall: indicate failure of connection establishment to GFT-Control and initiate release of the connection as described in 7.3.1.7.

NOTE 19

The cause sent to the Outgoing side should be no. 102 "Recovery on Timer Expiry".

7.3.1.6 Actions in the Active state

On receipt of a FACILITY message from a peer Protocol Control entity, an indication shall be given to GFT-Control.

On receipt of a request to send Supplementary services related information by GFT-Control, Protocol Control shall send a FACILITY message to the peer Protocol Control entity.

A received CONNECT ACKNOWLEDGE message shall be ignored.

7.3.1.7 Connection release

When Protocol Control is requested by GFT-Control to release a Call independent signalling connection, Protocol Control shall:

- if in the Release request state, ignore the request from GFT-Control; or
- if in any other Protocol Control state, send a RELEASE message with an appropriate cause value, start timer T308 and enter the Release request state.

When Protocol Control makes a local decision to release a Call independent signalling connection (e.g. due to a protocol error), it shall, if not in the Release request state: inform GFT-Control that the signalling connection has been released, send a RELEASE message with an appropriate cause value, start timer T308 and enter the Release request state.

On receipt of a RELEASE message in any state other than the Release request state, Protocol Control shall indicate to GFT-Control that the signalling connection has been released, send a RELEASE COMPLETE message, release the call reference and enter the Null state.

On receipt of a RELEASE COMPLETE message in any state other than the Release request state, Protocol Control shall indicate to GFT-Control that the signalling connection has been released, release the call reference and enter the Null state.

7.3.1.8 Actions in the Release request state

On receipt of a RELEASE or a RELEASE COMPLETE message, Protocol Control shall: stop timer T308, release the call reference and enter the Null state.

If timer T308 expires for the first time, the RELEASE message shall be retransmitted and timer T30 shall be restarted. If timer T308 expires a second time, Protocol Control shall release the cal reference and enter the Null state.

7.3.1.9 Transport of APDUs associated with a Call independent signalling connection

Sub-clause 7.1.1 shall apply, with the exception that the term 'call' shall be interpreted as 'Cal independent signalling connection'.

7.3.1.10 Protocol error handling

7.3 of ECMA-143 shall apply with the following modifications:

- actions regarding the handling of B-channels are not applicable;
- actions regarding the handling of the DISCONNECT message (not defined for use with Cal independent connections) are not applicable.;
- if a SETUP ACKNOWLEDGE, ALERTING, DISCONNECT or PROGRESS message (defined ECMA-143) is received in any state (except the Null state, where invalid call reference erro procedures apply) it shall be treated as an unexpected or unrecognised message in accordance wit 7.3.4 of ECMA-143;

7.4 of ECMA-143 shall apply for the generation and request of Call independent connection stat information.

7.3.1.11 Protocol timer values

Table 2 defines the values and attributes of the protocol timers required for Connection oriente Protocol Control.

In Table 2, the following conventions are used to indicate the applicability of the protocol timers t an incoming or outgoing side Protocol Control entity in a PTNX:

M: The support of the timer is Mandatory

O: The support of the timer is Optional

M(I): The support of the timer is Mandatory if the associated (optional) procedures ar implemented.

All timer values given in table 2 shall have a tolerance of 10%. Where minimum and maximun values are given, the choice of value is an implementation matter, within the range specified, with a tolerance of 10% below the minimum value and 10% above the maximum value.

NOTE 20

The use of timer T314 for message segmentation procedures (see 7.2 of ECMA-143) is beyond th scope of this Standard.

Table 2 - Protocol Control timer values

Timer number	Timer value	Call state	Cause for start	Normally terminated	Action to be taken when timer expires	Incoming side	Outgoing side
T303	Minimum 4 s, Maximum 6 s	Call initiated	On Sending SETUP	On receipt of CALL PROCEEDING, CONNECT or RELEASE COMPLETE	Retransmit SETUP and restart T303 or release the connection as specified in 7.3.1.7		M
Second T303	Minimum 4 s, Maximum 6 s	Call initiated	On retransmission of SETUP	On receipt of CALL PROCEEDING, CONNECT or RELEASE COMPLETE	Release connection as specified in 7.3.1.7		O
T308	Minimum 4 s, Maximum 6 s	Release request	On Sending RELEASE	On Receiving RELEASE or RELEASE COMPLETE	Retransmit RELEASE, restart T308	M	M
Second T308	Minimum 4 s, Maximum 6 s	Release request	On expiry of T308	On receiving RELEASE or RELEASE COMPLETE	Release Call Reference	M	M
T309	90 s	Any State	Data Link disconnection. Connections in Stable states are not lost.	On Data Link re-establishment	Release connection and call reference	M	M
T310	30 s	Outgoing call proceeding	On receipt of CALL PROCEEDING	On Receipt of CONNECT or RELEASE	Release the connection as specified in 7.3.1.7		M (Optional for a Transit PTNX)
T313	Minimum 4 s, Maximum 6 s	Connect request	On sending CONNECT	On receipt of CONNECT ACKNOWLEDGE	Release the connection as specified in 7.3.1.7	O	
T322	Minimum 4 s, Maximum 6 s	Any connection state except Null.	STATUS ENQUIRY sent	STATUS, RELEASE or RELEASE COMPLETE received	STATUS ENQUIRY may be transmitted several times - implementation dependant.	M (I)	M (I)

7.3.2 Dynamic Description (SDL) of Connection oriented Protocol Control procedures

Figure 7 contains a dynamic description of the Connection oriented Protocol Control procedures in 7.3.1. It is based on the SDL description of the Basic call, defined in 8.4 of ECMA-143 and is not intended to be complete. It is to be used as an aid to the interpretation of the text, which shall be the prime source should a conflict occur.

Figure 6 shows the key to the symbols used in figure 7. Table 3 describes the naming convention used for primitives shown in the SDL diagram.

Table 3 - Key to primitive names used in figure 7

Prefix	Primitive from/to:
Event_	An entity which provides Protocol Control with notification of protocol related events other than receipt of incoming messages or primitives from GFT-Control or the Data Link Layer

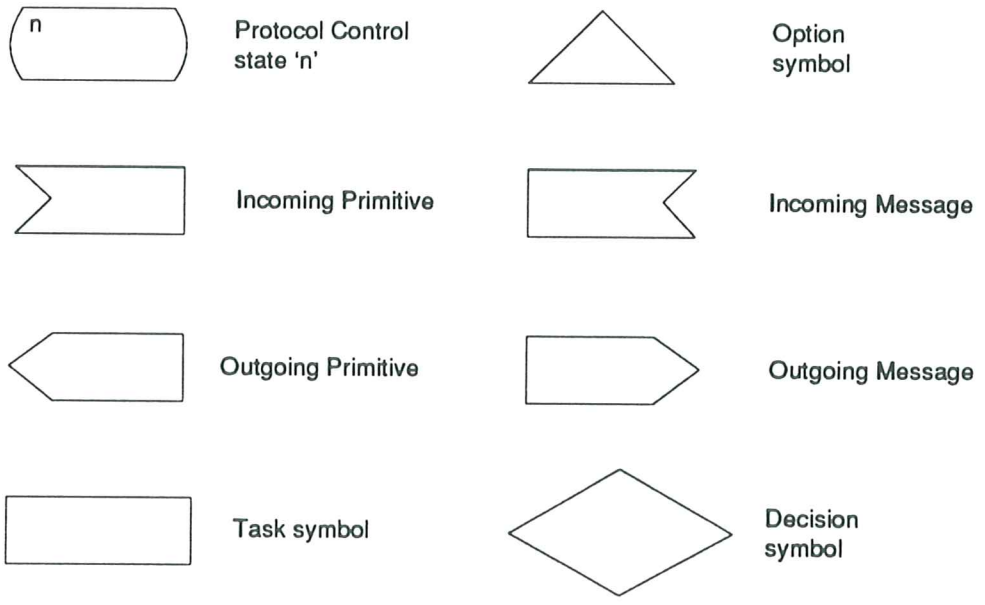


Figure 6 - Key to symbols used in the SDL diagram for Connection oriented Protocol Control

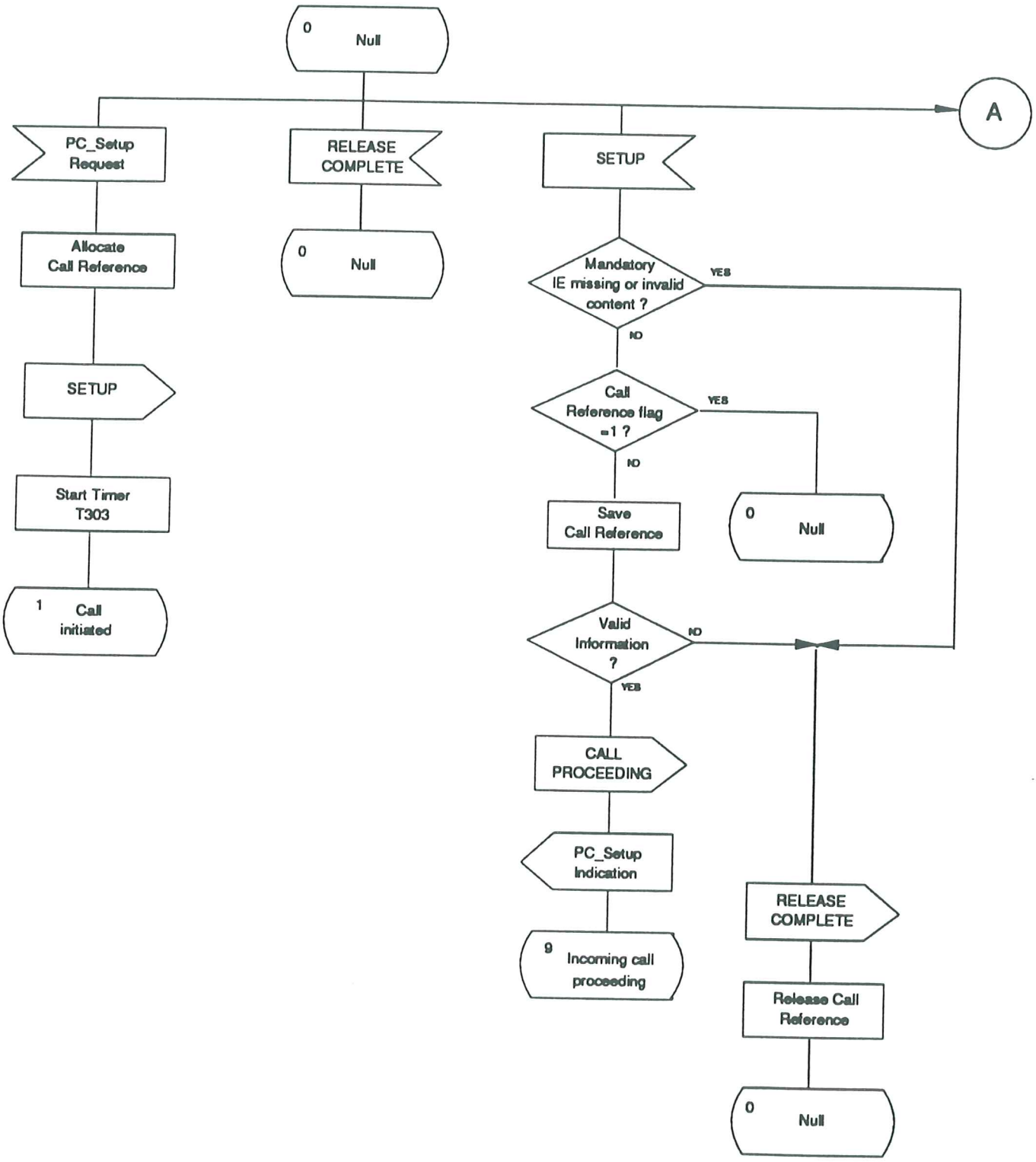


Figure 7 - Connection oriented Protocol Control SDL (sheet 1 of 10)

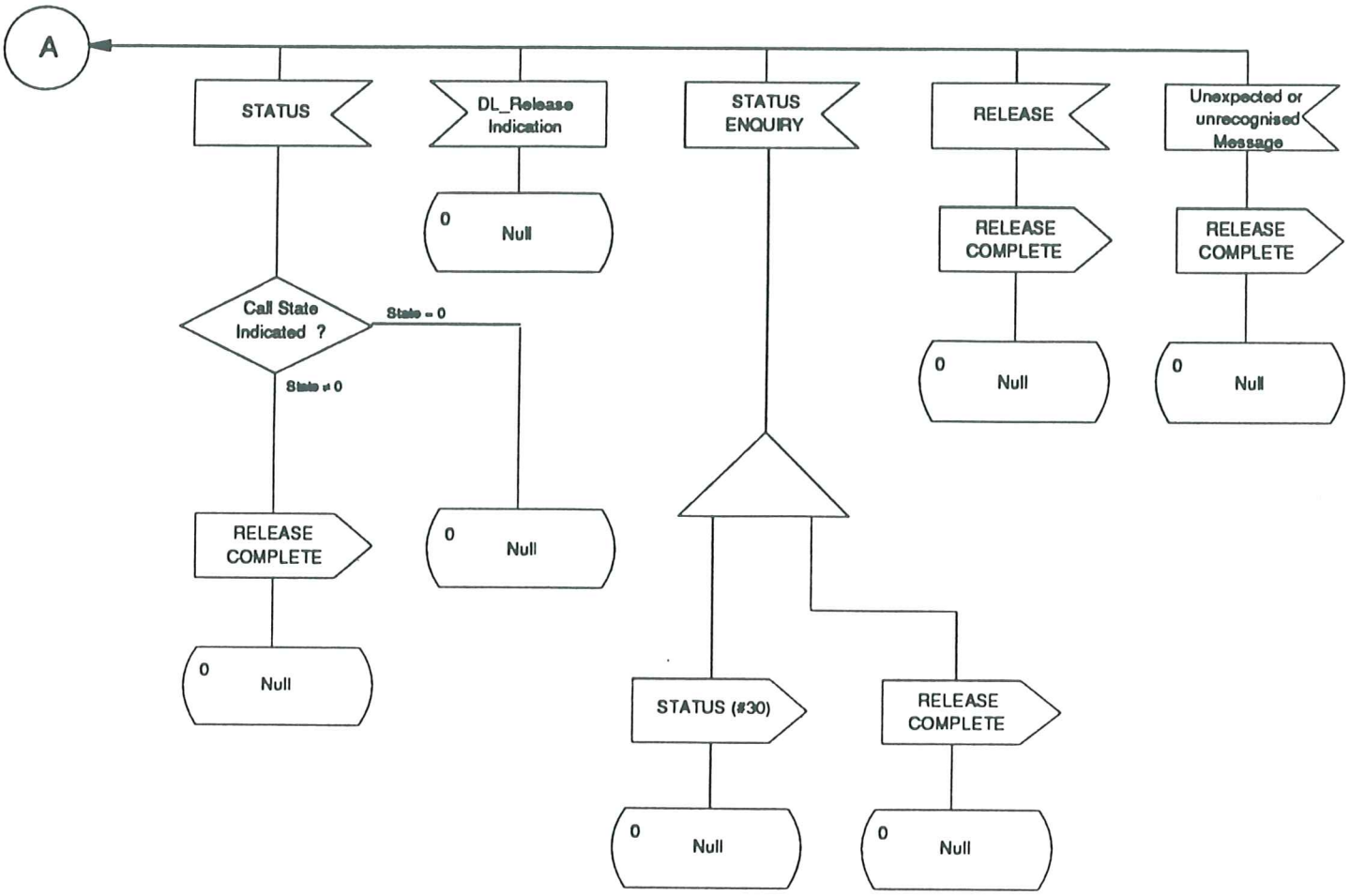


Figure 7 - Connection oriented Protocol Control SDL (sheet 2 of 10)

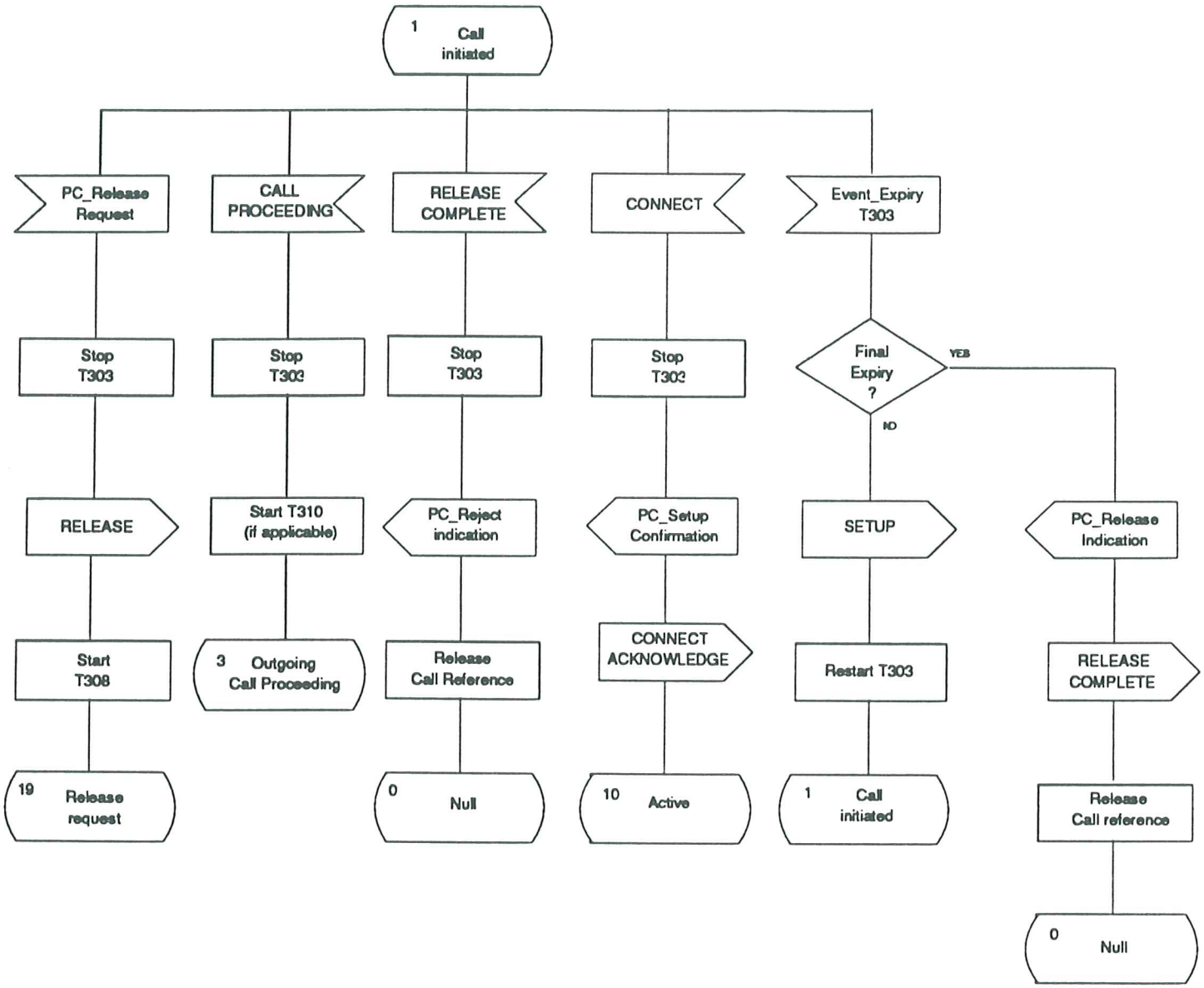


Figure 7 - Connection oriented Protocol Control SDL (sheet 3 of 10)

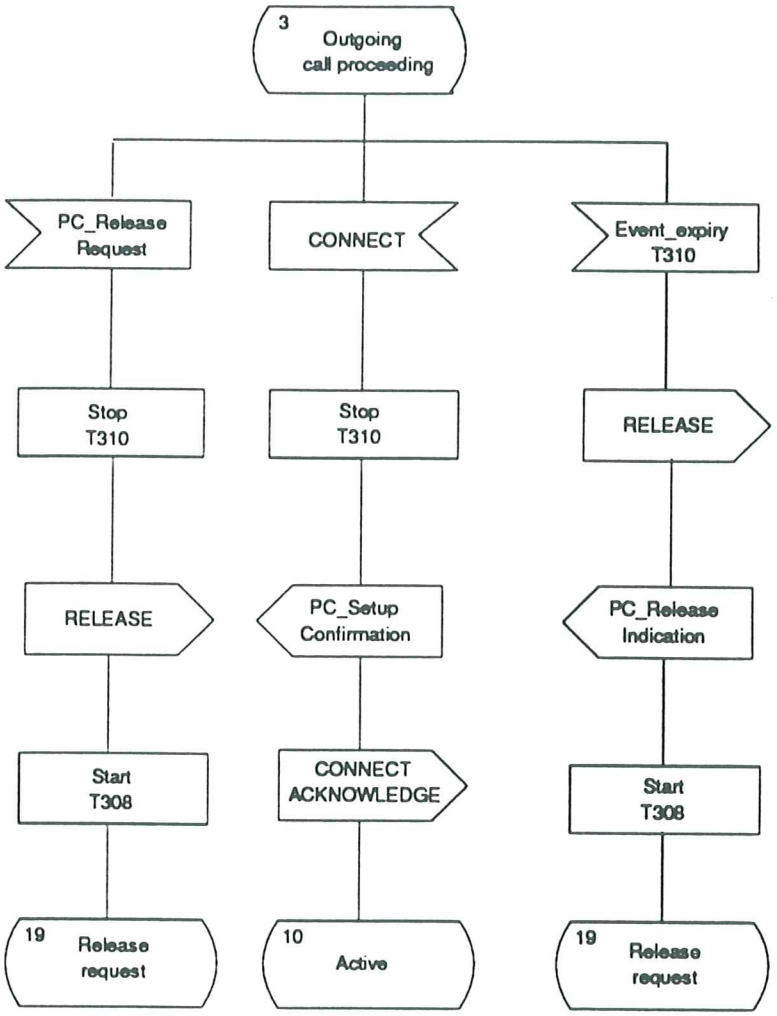


Figure 7 - Connection oriented Protocol Control SDL (sheet 4 of 10)

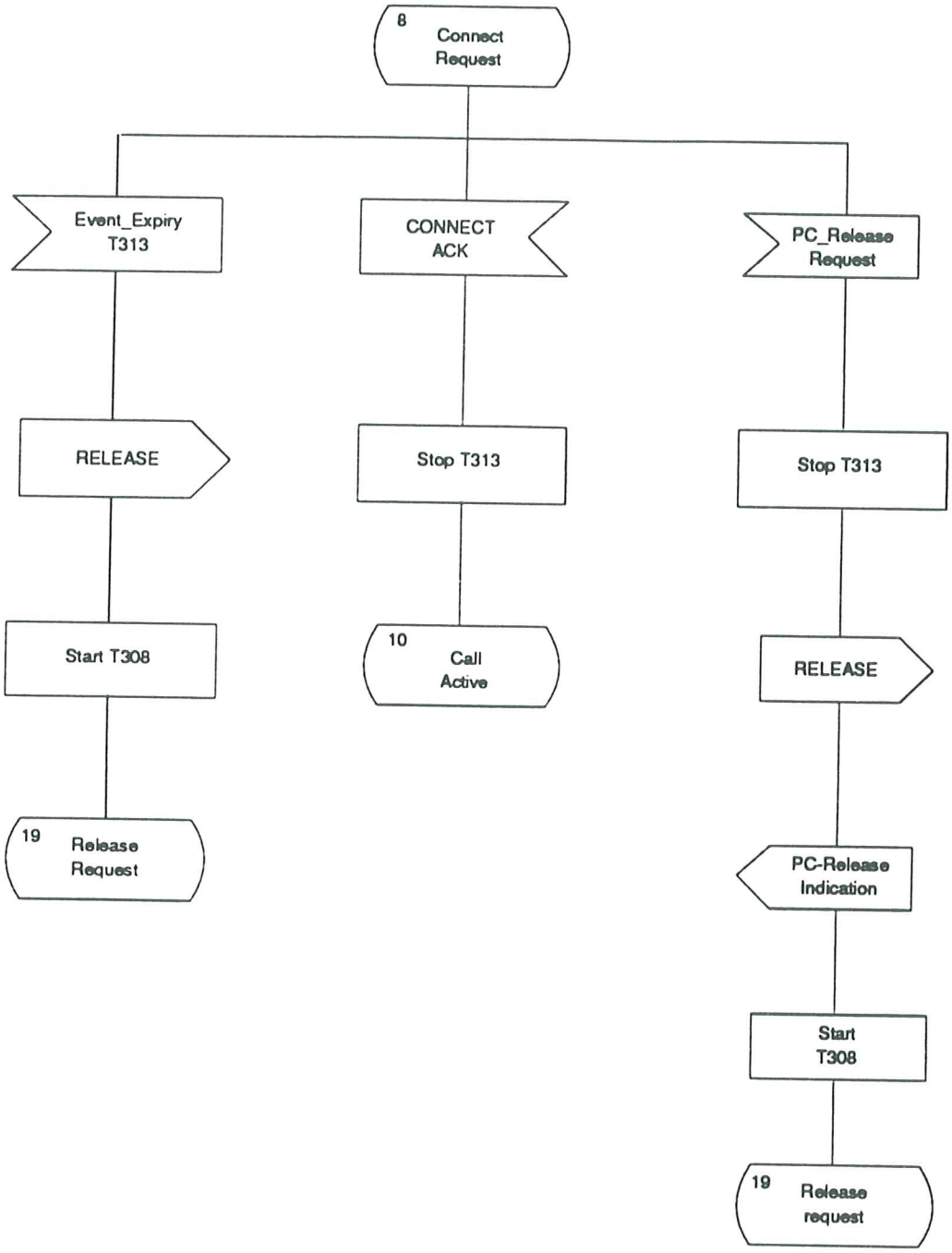


Figure 7 - Connection oriented Protocol Control SDL (sheet 5 of 10)

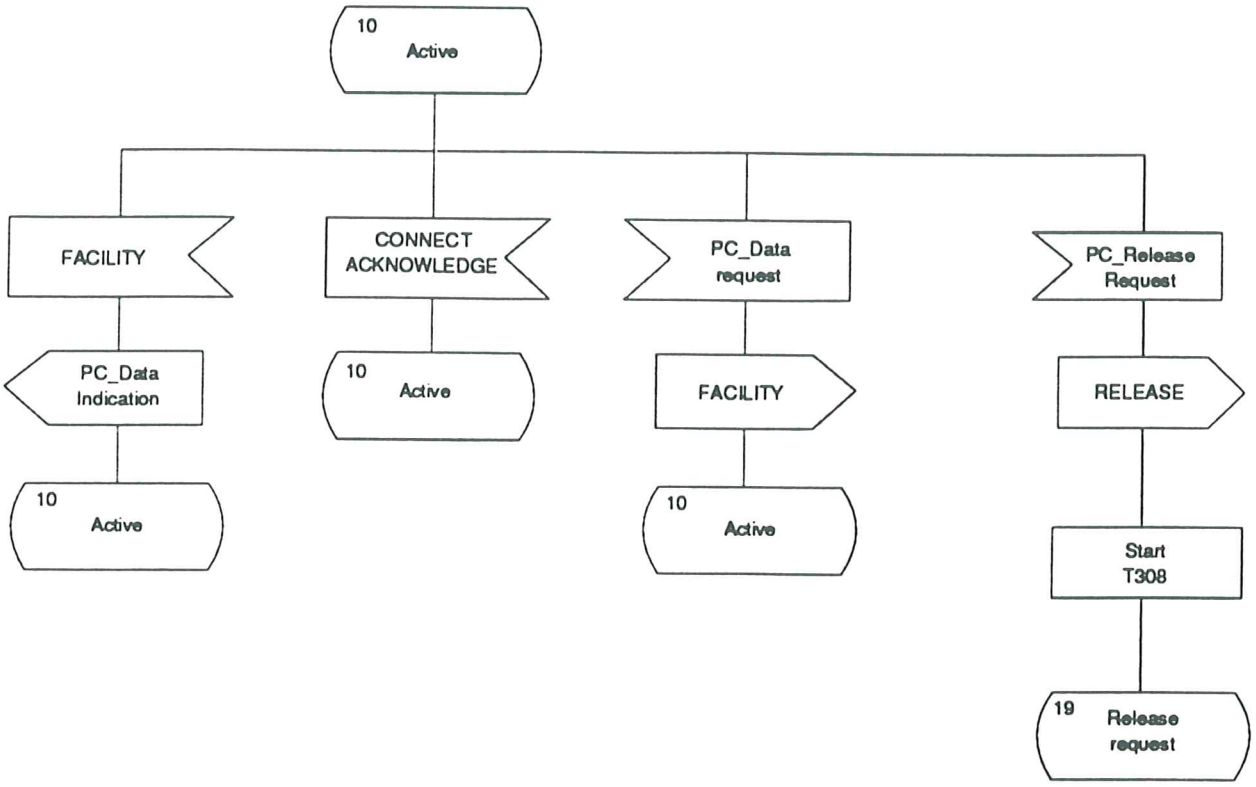


Figure 7 - Connection oriented Protocol Control SDL (sheet 6 of 10)

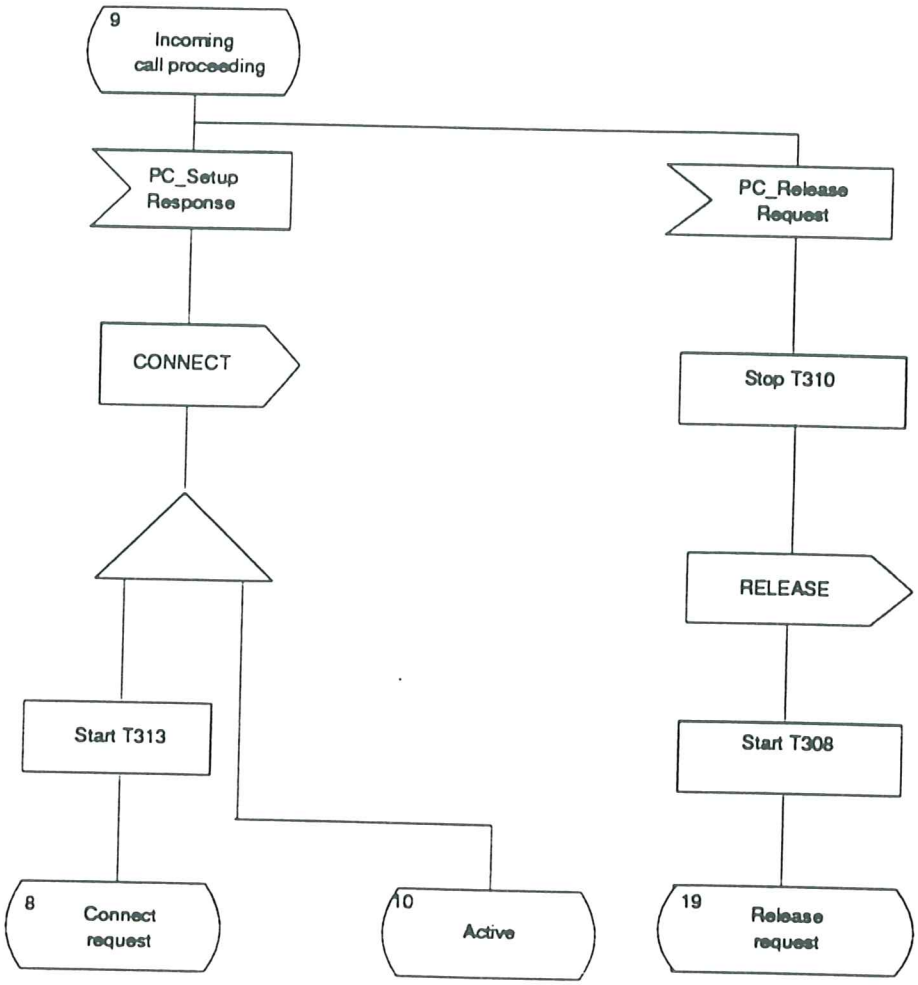


Figure 7 - Connection oriented Protocol Control SDL (sheet 7 of 10)

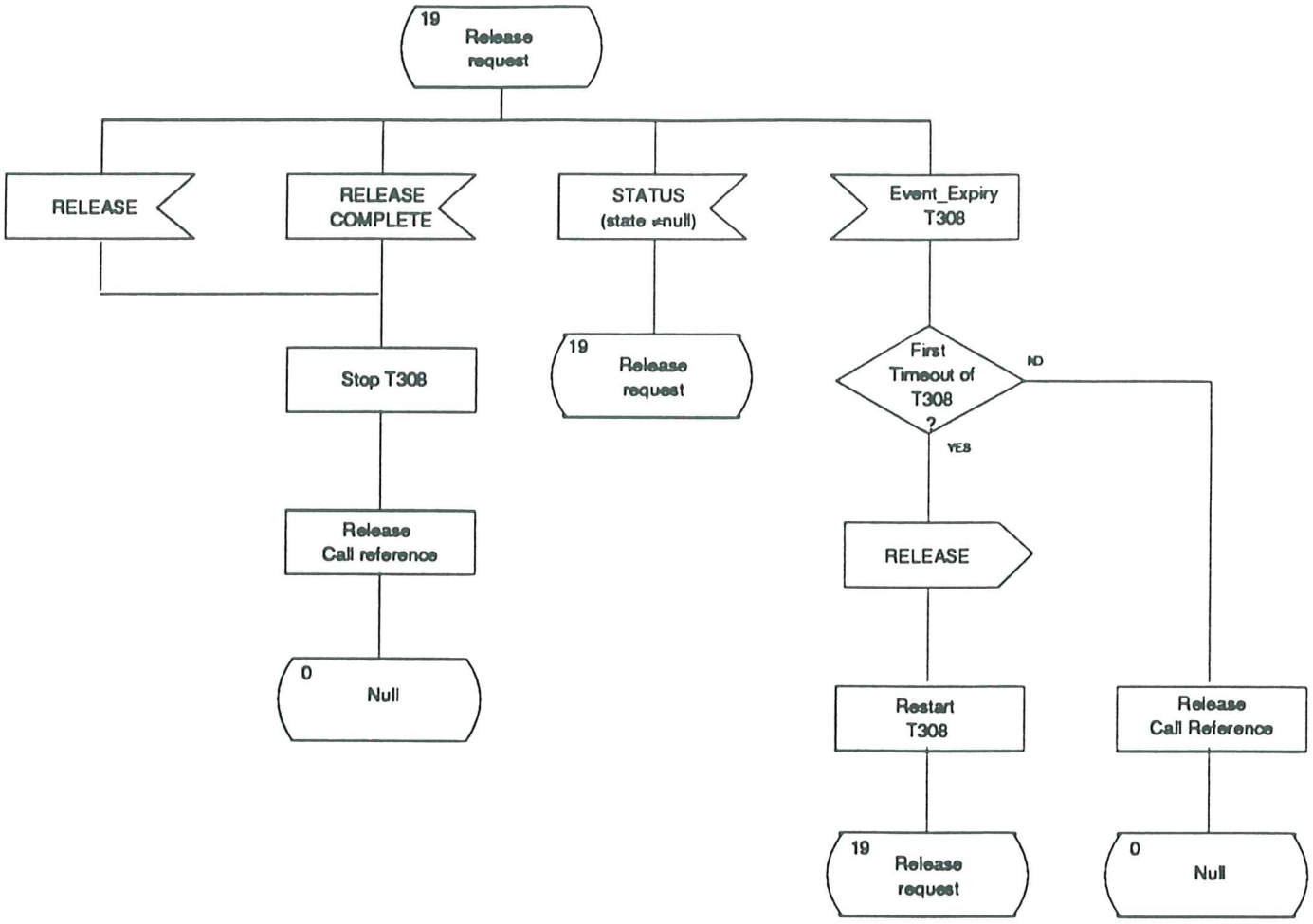


Figure 7 - Connection oriented Protocol Control SDL (sheet 8 of 10)

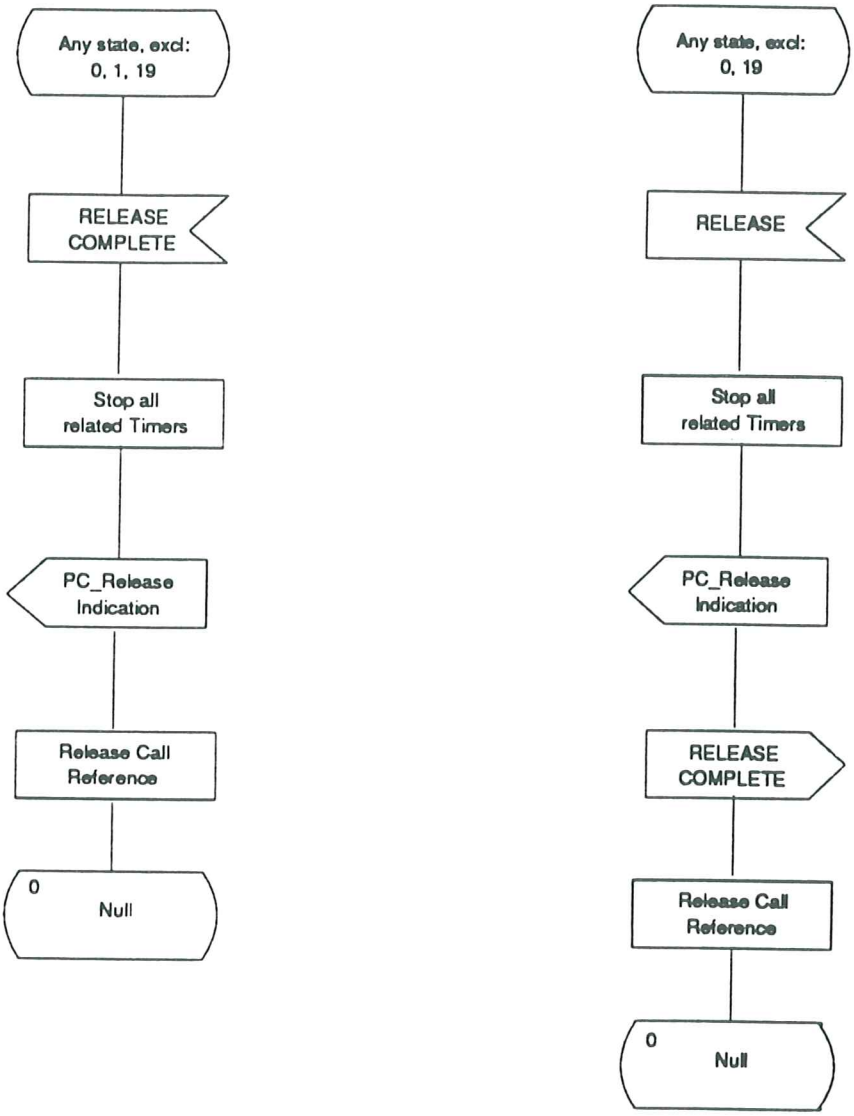


Figure 7 - Connection oriented Protocol Control SDL (sheet 9 of 10)

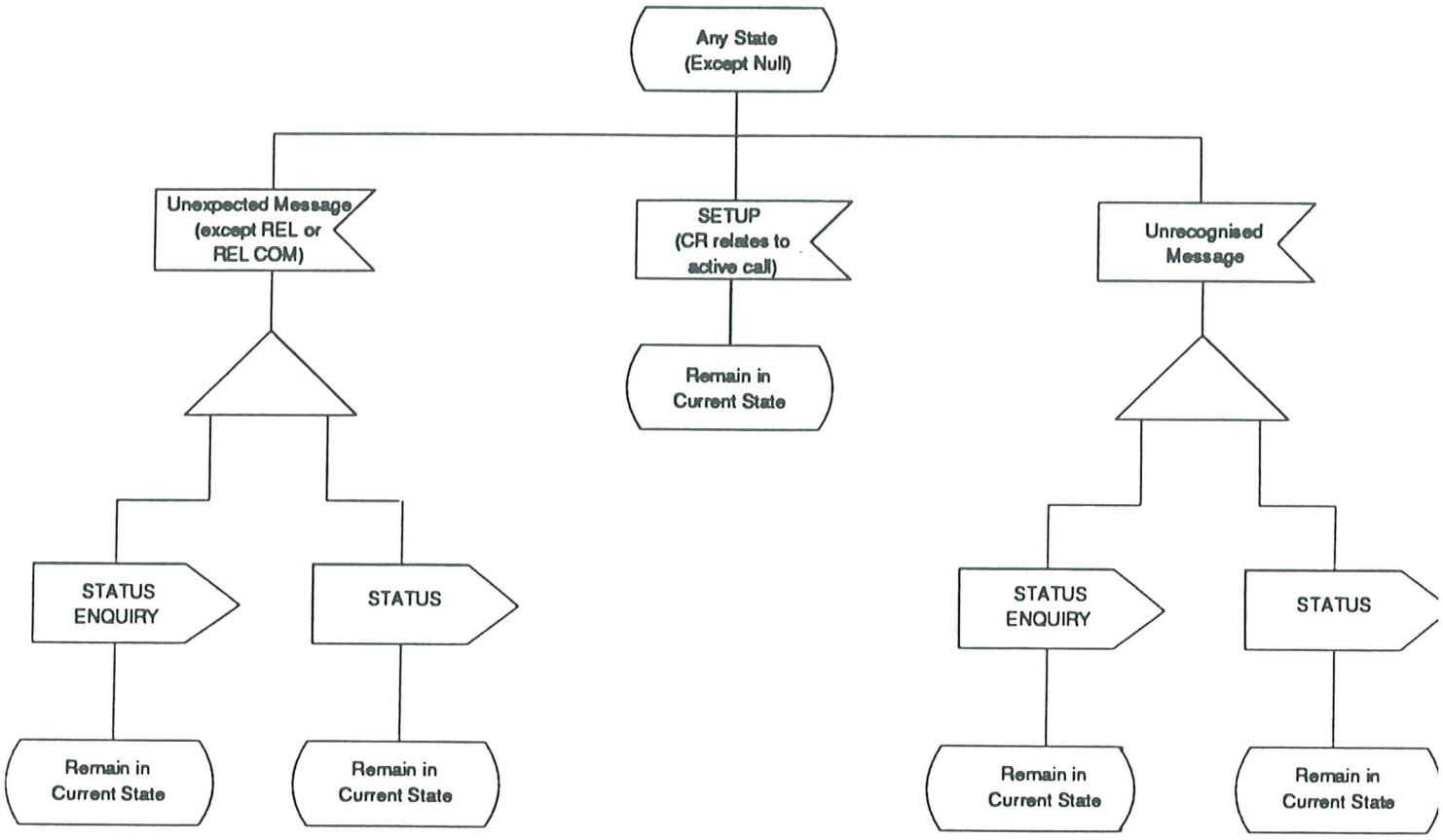


Figure 7 - Connection oriented Protocol Control SDL (sheet 10 of 10)

7.3.3 Generic Functional Transport Control requirements

The procedures describing the requirements of the GFT-Control entity for Call independent signalling connection control are defined in terms of a number of states. These states are conceptual states that are used to enable straightforward description of the dynamic aspects of the GFT-Control procedures.

The states used are separated into states that exist at an Originating PTNX, a Transit PTNX and a Terminating PTNX. A brief description of the states is as follows:

Originating PTNX GFT-Control States:

Originating_connection_idle:	no connection exists
Originating_connection_request:	connection establishment has been requested, but no response has been received from the Terminating PTNX.
Originating_connection_active:	the connection is active

Transit PTNX GFT-Control States:

Transit_connection_idle:	no connection exists
Transit_connection_request:	connection establishment request has been received from the Preceding PTNX and forwarded to the Subsequent PTNX, but no response has been received from the Subsequent PTNX.
Transit_connection_active:	the connection is active

Terminating PTNX GFT-Control States:

Incoming_connection_idle:	no connection exists
Incoming_connection_active:	the connection is active

7.3.3.1 Actions at an Originating PTNX

7.3.3.1.1 Actions in the Originating_connection_idle state

When a request for establishment of a Call independent signalling connection to a remote PTNX is received from the Coordination Function, GFT-Control shall: request the Outgoing side Protocol Control to send a SETUP message, including the address of the Terminating PTNX, and optionally the Transit count information element with the transit count field set to zero, and enter the Originating_connection_request state.

7.3.3.1.2 Actions in the Originating_connection_request state

If Protocol Control informs GFT-Control that a RELEASE or RELEASE COMPLETE message has been received, GFT-Control shall inform the Coordination Function that the connection has failed and enter the Originating_connection_idle state.

If Protocol Control informs GFT-Control that a CONNECT message has been received, GFT-Control shall enter the Originating_connection_active state.

7.3.3.1.3 Actions in the Originating_connection_active state

If a request for transfer of APDUs on the connection is received from the Coordination Function, GFT-Control shall instruct Protocol Control to send a FACILITY message to the Subsequent PTNX, containing a Facility information element in accordance with 7.3.3.4.

If Protocol Control informs GFT-Control that a FACILITY message has been received, the PTNX shall become the Destination PTNX for the received Facility information element in accordance with 7.3.3.5.

If Protocol Control informs GFT-Control that a RELEASE message has been received, GFT-Control shall inform the Coordination Function that the connection has been released and enter the Originating_connection_idle state.

If a request that the connection be released is received from the Coordination Function, GFT Control shall: request that Protocol Control send a RELEASE message and enter the Originating_connection_idle state.

7.3.3.2 Actions at a Transit PTNX

If GFT-Control receives indication from Protocol Control of a received SETUP message from the Preceding PTNX, it shall examine the contents of the Called party number information element. If the Called party number information element matches that of the Receiving PTNX, the PTNX shall become a Terminating PTNX, otherwise it shall follow the procedures of this clause.

If GFT-Control receives any APDUs from Protocol Control in any of the messages which may contain a Facility information element (see clause 10), it shall examine the header of the Facility information element for the presence of an NFE:

- If no NFE is present, or an NFE is present and does not indicate the following:

  sourceEntity: end PTNX ; and,

  destinationEntity: end PTNX

  (with no **sourceEntityAddress** and **destinationEntityAddress** elements present), the PTNX shall discard the entire received Facility information element;

- In all other cases, the Transit PTNX shall instruct Protocol Control to pass on the received APDUs in the message sent to the Next PTNX.

7.3.3.2.1 Actions in the Transit_Connection_idle state:

If the destination address contained in the SETUP message is that of another PTNX and a connection to that PTNX is possible, GFT-Control shall: request Protocol Control to send a SETUP message on the appropriate inter-PTNX link to the Subsequent PTNX, associate the incoming and outgoing connections and enter the Transit_connection_request state.

If the received SETUP message contains a Transit counter information element in which the transit count field has a value that is less than the acceptable (network dependent) limit, that information element shall be included in the SETUP message sent to the Subsequent PTNX. The value of the transit count field in the outgoing Transit counter information element shall be one greater than the value received.

If the received SETUP message contains a Transit counter information element in which the transit count field has a value that is greater than or equal to the acceptable (network dependent) limit of Transit PTNXs through which the call may be routed, and the PTNX is unable to become the Terminating PTNX, GFT-Control shall: request Protocol Control to release the connection by sending a RELEASE message to the Preceding PTNX and remain in the Transit_connection_idle state.

If the received SETUP message does not contain a Transit counter information element, the Transit PTNX may include a Transit counter information element in the SETUP message sent to the Subsequent PTNX. The value of the transit count field in this element shall be set to an initial value.

If the contents of the Destination address information element contained in the SETUP message is not sufficient to enable routing onto a further inter-PTNX link, GFT-Control shall: request Protocol Control to release the connection by sending a RELEASE message to the Preceding PTNX and remain in the Transit_connection_idle state.

7.3.3.2.2 Actions in the Transit_Connection_request state

When Protocol Control informs GFT-Control of a CONNECT message received from the Subsequent PTNX, GFT-Control shall: request Protocol Control to send a CONNECT message to the Preceding PTNX and enter the Transit_connection_active state.

When Protocol Control informs GFT-Control that a RELEASE or RELEASE COMPLETE message has been received from the Subsequent PTNX, GFT-Control shall: request Protocol Control to send a RELEASE message to the Preceding PTNX and enter the Transit_connection_idle state.

When Protocol Control informs GFT-Control that a RELEASE message has been received from the Preceding PTNX, GFT-Control shall: request Protocol Control to send a RELEASE message to the Subsequent PTNX and enter the Transit_connection_idle state.

7.3.3.2.3 Actions in the Transit_Connection_active state

If Protocol Control informs GFT-Control of the receipt of a FACILITY message from the Subsequent PTNX, and if it contains a valid NFE, GFT-Control shall request Protocol Control to send a FACILITY message (exactly as received) to the Preceding PTNX.

If Protocol Control informs GFT-Control of the receipt of a FACILITY message from the Preceding PTNX, and it contains a valid NFE, GFT-Control shall request Protocol Control to send a FACILITY message (exactly as received) to the Subsequent PTNX.

If Protocol Control informs GFT-Control of the receipt of a RELEASE message from the Subsequent PTNX, GFT-Control shall request Protocol Control to send a RELEASE message to the Preceding PTNX and shall enter the Transit_connection_idle state.

If Protocol Control informs GFT-Control of the receipt of a RELEASE message from the Preceding PTNX, GFT-Control shall request Protocol Control to send a RELEASE message to the Subsequent PTNX and shall enter the Transit_connection_idle state.

7.3.3.3 Actions at a Terminating PTNX

7.3.3.3.1 Actions in the Incoming_Connection_idle state:

If Protocol Control notifies GFT-Control of a received SETUP message that is to be terminated on the receiving PTNX, and resources for the connection are available, GFT-Control shall request Protocol Control to send a CONNECT message and enter the Incoming_connection_active state.

If no resources for the connection are available, GFT-Control shall: request Protocol Control to send a RELEASE message; and remain in the Incoming_connection_idle state.

7.3.3.3.2 Actions in the Incoming_Connection_active state

If SS-Control requests transfer of APDUs on the connection, GFT-Control shall instruct Protocol Control to send a FACILITY message to the Preceding PTNX.

If Protocol Control informs GFT-Control that a FACILITY message has been received from the Preceding PTNX, GFT-Control shall become the Destination PTNX for those received APDUs (see 7.3.3.5).

If Protocol Control informs GFT-Control that a RELEASE message has been received from the Preceding PTNX, it shall inform SS-Control that the connection has been released and enter the Incoming_connection_idle state.

If SS-Control requests that the connection be released, GFT-Control shall: request that Protocol Control send a RELEASE message; and enter the Incoming_connection_idle state.

7.3.3.4 Actions at a Source PTNX

If the Source PTNX (Originating PTNX or Terminating PTNX) wishes to send APDUs related to the Call independent signalling connection, GFT-Control shall request Protocol Control to send this information in a Facility information element in conjunction with a relevant (state dependent) message on the Call independent signalling connection. This Facility information element shall include an NFE (as defined in 11.3.3.1) which indicates the following:

sourceEntity: end PTNX; and
destinationEntity: end PTNX

The **sourceEntityAddress** and **destinationEntityAddress** elements shall not be included in the NFE.

7.3.3.5 Actions at a Destination PTNX

If GFT-Control at an Originating PTNX or Terminating PTNX receives any APDUs from Protocol Control in any of the messages which may contain a Facility information element (see clause 10), it shall examine the header of the Facility information element for the presence of an NFE:

- If no NFE is present, or an NFE is present and does not indicate the following:

  sourceEntity: end PTNX ; and,
  destinationEntity: end PTNX

  (with no **sourceEntityAddress** and **destinationEntityAddress** elements present), the PTNX shall discard the entire received Facility information element;

- In all other cases, the APDUs shall be passed to the Coordination Function.

7.4 Call related procedures for the transport of Notifications

This clause defines the functional signalling procedures that support the delivery of notifications over the PTN in association with a Basic call.

7.4.1 Categories of notifications

Procedures are defined for the delivery of three types of notification information as follows:

- the delivery of simple notification indicators based on the Notification Indicator information element as described in 11.3.4;
- the delivery of notification 'parameters' that are specified as information elements using t. encoding scheme defined in clause 12 of ECMA-143 within the **qsigIeNotification** Notification defined in Table 26 of 11.3.3.
- the delivery of notification components using an extension codepoint in octet 3 of the Notification indicator information element and ASN.1 encoded data structure in subsequent octets.

7.4.2 Protocol Control requirements

7.4.2.1 Sending notification information

The transport of notifications shall make use of the call reference of a Basic call and its underlying data link layer connection. Notifications shall be sent using the Notification indicator information element.

If the delivery of the notification information coincides with the sending of the FACILITY message or any of the Basic call messages listed in clause 10 in which the Notification indicator information element is permitted, the notification may be carried in that message. Otherwise, the notification shall be delivered in a NOTIFY message.

However:

- if a SETUP message has been sent, but no response has been received from the Next PTNX (i.e. the B-channel has not yet been agreed on the Outgoing side of the PTNX); or
- if a clearing message has already been sent to or received from the Next PTNX

the notification information shall be discarded.

NOTE 21

In the case where the Notification indicator information element is sent to a PTNX which does not conform to this Standard, the Notification indicator information element will be discarded by that PTNX and a STATUS message (see 11 in ECMA-143) can be received. The STATUS message will indicate that either: the Notification indicator information element was unrecognised; or, that the message (NOTIFY or FACILITY) was unrecognised. In such cases, no further action should be taken.

7.4.2.2 Receiving notification information

On receipt of a Notification indicator information element, in the NOTIFY message or in any of the other messages listed in clause 10 in which the Notification indicator information element is permitted, it shall be passed to GFT-Control.

7.4.3 GFT-Control requirements

7.4.3.1 Actions at a PTNX which generates notifications

A PTNX which wishes to generate a notification shall request Protocol Control to send a Notification indicator information element.

7.4.3.2 Actions at a Transit PTNX

If a Transit PTNX receives a Notification indicator information element from the Preceding PTNX, it shall request Protocol Control to send the Notification indicator information element to the Subsequent PTNX.

If a Transit PTNX receives a Notification indicator information element from the Subsequent PTNX, it shall request Protocol Control to send the Notification indicator information element to the Preceding PTNX.

7.4.3.3 Actions at a Receiving End PTNX

If an End PTNX receives a Notification indicator information element, at any time during a Call, it shall convey the information it contains to the PTN user - dependent on the ability of the PTN user's equipment to receive such information.

NOTE 22

Further (implementation specific) actions of a PTNX receiving a notification (e.g. changing the state of a local non-Standard state machine) are not precluded and are beyond the scope of this Standard.

8 Application layer requirements

8.1 Coordination Function requirements

The behaviour of the Coordination Function in passing information between the various SS-Control entities, ROSE, DSE, Call Control and GFT-Control is beyond the scope of this Standard, with the exception of the provisions in 8.1.1 and 8.1.2 relating to the handling of the Interpretation APDU and error handling at a Destination PTNX.

8.1.1 Inclusion of an Interpretation APDU at a Source PTNX

If a Source PTNX wishes to include additional information to facilitate handling of unrecognised ROSI APDUs of type **InvokePDU** (see 11.3.3.4) at a Destination PTNX, it shall include an Interpretatio APDU (see 11.3.3.2) as the first APDU in the sequence of APDUs sent to GFT-Control.

8.1.2 Handling of APDUs at a Destination PTNX

An APDU which is received by the Destination PTNX and is not recognised as a supported APDU shal be discarded.

If an Interpretation APDU is received by the Destination PTNX as the first APDU of a sequence o APDUs from GFT-Control, it shall examine any ROSE APDU of type **RejectPDU** generated as a resul of the processing of these APDUs. If the element **problem** in the **RejectPDU** is of type **InvokeProble** and has value **unrecognisedOperation** the action taken shall depend on the contents of the Interpretatio APDU as follows:

- If the Interpretation APDU indicates **rejectUnrecognisedInvokePdu** the ROSE APDU of t **RejectPDU** shall be delivered to GFT-Control;
- If the Interpretation APDU indicates **clearCallIfAnyInvokePduNotRecognised** the ROSE APDU o type **RejectPDU** shall be delivered to GFT-Control and Call Control shall be requested to clear th Basic call to which the **InvokePDU** was related.
- If the Interpretation APDU indicates **discardAnyUnrecognisedInvokePDU** the ROSE APDU of typ **RejectPDU** shall be discarded.

If no Interpretation APDU is received, any ROSE APDUs of type **RejectPDU** shall be delivered to GFT Control.

If an Interpretation APDU is received that is not the first APDU in the sequence of APDUs receive from GFT-Control, or does not conform to the structure in 11.3.3.2, it shall be ignored.

8.2 ROSE requirements

The procedures specified in section 7 of X.229 for sending and receiving ROSE APDUs shall apply, witl the exception that the Transfer services used shall be those provided by GFT-Control.

As a minimum, a PTNX shall recognise received ROSE APDUs and reject those whose operation val are not supported. Additional requirements relating to the use of ROSE are Supplementary service speci... and are beyond the scope of this Standard.

8.3 DSE requirements

The DSE may be used to create a dialogue between two PTNXs, to enable service requests and responses t be correlated, particularly when they do not exist within the context of the same network layer connection.

The DSE uses the underlying services provided by GFT-Control via the Coordination Function.

The coding requirements for the DSE APDUs are defined in 11.3.3.3.

Any DSE APDUs, with the exception of a DialogAbortPDU, may contain one or more ROSE APDUs.

A state machine shall be associated with each dialogue within a PTNX. Four dialogue states are defined:

- Idle: no dialogue exists;
- Initiate sending: a **DialogBeginPDU** has been sent, a **DialogContinuePDU** is awaited from the pee PTNX;
- Initiate receiving: a **DialogBeginPDU** has been received, a request from the Coordination Function i awaited to continue or terminate the dialogue;

- Active: the dialogue is established.

8.3.1 Actions at the PTNX which initiates the dialogue (PTNX A)

8.3.1.1 Idle state procedures

When a request from the Coordination Function to initiate a dialogue is received, PTNX A shall:

- send a **DialogBeginPDU** to the PTNX identified in the request (PTNX B). The element of type **OriginationDialogId** shall contain a dialogue identifier selected by PTNX A that is sufficient to distinguish the dialogue from any others in which PTNX A is involved. The **DialogBeginPDU** may also contain one or more ROSE APDUs relating to a particular Supplementary service or services.;
- start timer T_Originating_Dialogue (T_OD); and
- enter the Initiate sending state.

The selected dialogue identifier shall be included in the element of type **OriginationDialogId** in all further **DialogContinuePDUs** sent from PTNX A to PTNX B for the duration of the dialogue.

8.3.1.2 Initiate sending state procedures

On receipt of a **DialogContinuePDU**, PTNX A shall:

- cancel timer T_Originating_Dialogue;
- store the value of the element of type **OriginationDialogID**. This is the dialogue identifier selected by PTNX B and shall be included in all DSE APDUs sent from PTNX A to PTNX B in the element of type **DestinationDialogueId** for the duration of the dialogue;
- provide an indication of dialogue continuation to the Coordination Function; and,
- enter the Active state.

On receipt of a **DialogEndPDU**, PTNX A shall consider the dialogue to be terminated, release the locally assigned dialogue identifier, inform the Coordination Function, cancel timer T_Originating_Dialogue and enter the Idle state.

On receipt of a **DialogAbortPDU**, PTNX A shall consider the dialogue to be aborted, inform the Coordination Function, cancel timer T_Originating_Dialogue and enter the Idle state.

If a request to abort the dialogue is received from the Coordination Function, PTNX A shall cancel timer T_Originating_Dialogue, release the locally assigned dialogue identifier and enter the Idle state.

If timer T_Originating_Dialogue expires, PTNX A shall consider the dialogue to be aborted, inform the Coordination Function that the dialogue has been aborted, release the dialogue identifier assigned locally by PTNX A and enter the Idle state.

8.3.2 Actions at the PTNX which terminates the dialogue (PTNX B)

8.3.2.1 Idle state procedures

On receipt of a **DialogBeginPDU** from PTNX A, PTNX B shall:

- check that the value of the element of type **OriginationDialogId** in the **DialogBeginPDU** is valid. If it is not valid, PTNX B shall discard the **DialogBeginPDU** and remain in the idle state;
- save the value of the element of type **OriginationDialogId** in the **DialogBeginPDU**. This is the dialogue identifier selected by PTNX A and shall be included in all DSE APDUs sent from PTNX B to PTNX A in the element of type **DestinationDialogueId** for the duration of the dialogue;
- inform the Coordination Function; and,
- enter the Initiate receiving state.

8.3.2.2 Initiate receiving state procedures

If PTNX B wishes to continue the dialogue, it shall:

- send a **DialogContinuePDU** to PTNX A containing, in the element of type **OriginationDialogueID** a dialogue identifier selected by PTNX B to be sufficient to distinguish the dialogue from any others in which PTNX B is involved, and in the element of type **DestinationDialogID** the value received in the element of type **OriginationDialogueID** in the **DialogBeginPDU** from PTNX A; and
- enter the Active state.

If PTNX B cannot accept the dialogue, it shall send **DialogAbortPDU** to PTNX A, release the stored dialogue identifier and enter the Idle state.

If PTNX B wishes to end the dialogue, it shall send **DialogEndPDU** to PTNX A, release the stored dialogue identifier and enter the Idle state.

8.3.3 Dialogue Continuation in the Active State

If a PTNX wishes to continue the dialogue, it shall: send a **DialogueContinuePDU** to the peer PTNX and remain in the active state. The **DialogContinuePDU** may also contain one or more ROSE APDUs.

On receipt of a **DialogueContinuePDU**, the PTNX shall indicate dialogue continuation to the Coordination Function, together with any ROSE APDUs contained in the received **DialogContinuePDU**.

On receipt of a **DialogEndPDU**, the PTNX shall consider the dialogue to be terminated, inform the Coordination Function, release the dialogue identifier assigned locally and the identifier received from the peer PTNX, and enter the Idle state.

On receipt of a **DialogAbortPDU**, the PTNX shall consider the dialogue to be aborted, inform the Coordination Function, release the dialogue identifier assigned locally and the identifier received from the peer PTNX, and enter the Idle state.

If a request to terminate the dialogue is received from the Coordination Function, the PTNX shall send a **DialogEndPDU** to the peer PTNX, release the dialogue identifier assigned locally and the identifier received from the peer PTNX, and enter the idle state.

If a request to abort the dialogue is received from the Coordination Function, the PTNX shall send a **DialogAbortPDU** to the peer PTNX, release the dialogue identifier assigned locally and the identifier received from the peer PTNX, and enter the idle state.

8.3.4 Dialogue Protocol Timers

Table 4 - Dialogue Protocol Timers

Timer	State	Value	Normal Start	Normal Termination	Actions on expiry
T_OD	Initiate sending	Implementation dependent	On sending Dialog BeginPDU	On receipt of a Dialog ContinuePDU, DialogEndPDU or DialogAbortPDU	Indicate to SS-Control that dialogue is aborted. Enter idle state.

8.3.5 Error procedures relating to dialogue control

If a PTNX receives any DSE APDUs relating to the same dialogue from the peer PTNX that are incorrectly formatted they shall be discarded.

If a PTNX receives a **DialogEndPDU** or a **DialogAbortPDU** in the Idle state, the APDU shall be discarded and it shall remain in the Idle state.

If a PTNX receives a **DialogContinuePDU** in the Idle state, it shall send a **DialogAbortPDU** containing an element of type **DestinationDialogId** which has the same value as the element of type **OriginationDialogId** in the received **DialogContinuePDU**, and remain in the Idle state.

If a PTNX receives a **DialogContinuePDU** in the Active state which contains, in the element of type **OriginationDialogId** an unrecognised dialogue identifier, it shall send a **DialogAbortPDU** containing an element of type **DestinationDialogId** which has the same value as the element of type **OriginationDialogId** in the received **DialogContinuePDU**, and remain in the Active state.

8.3.6 Example of a dialogue

Figure 8 shows an example of a dialogue between two PTNXs, illustrating the usage and values of the origination and destination dialogue identifiers.

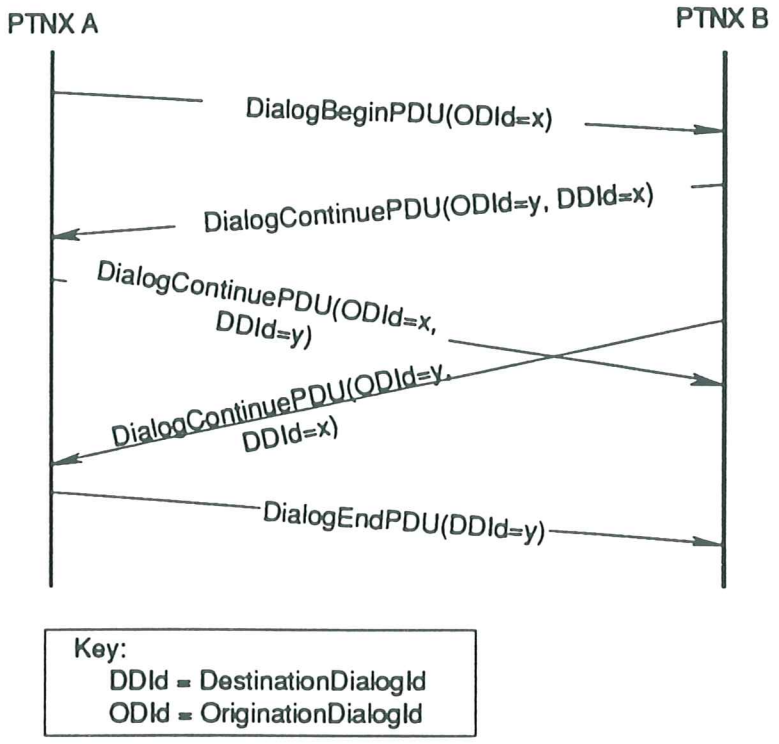


Figure 8 - A simple example of a Dialogue

8.3.7 Dynamic Description (SDL) of Dialogue Identification Protocol Procedures

Figure 10 provides an SDL representation of the dynamic aspects of the DSE protocol. Figure 9 contains a description of the elements used in figure 10.

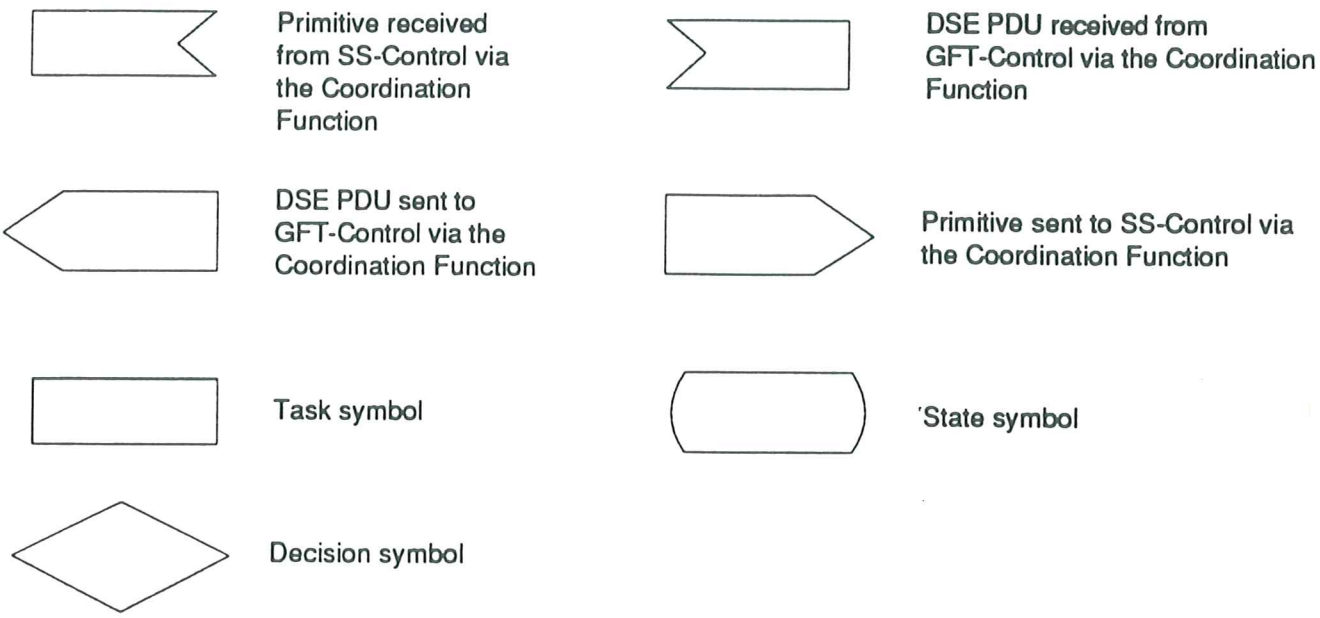


Figure 9 - Key to Dialogue SDL diagram in figure 10

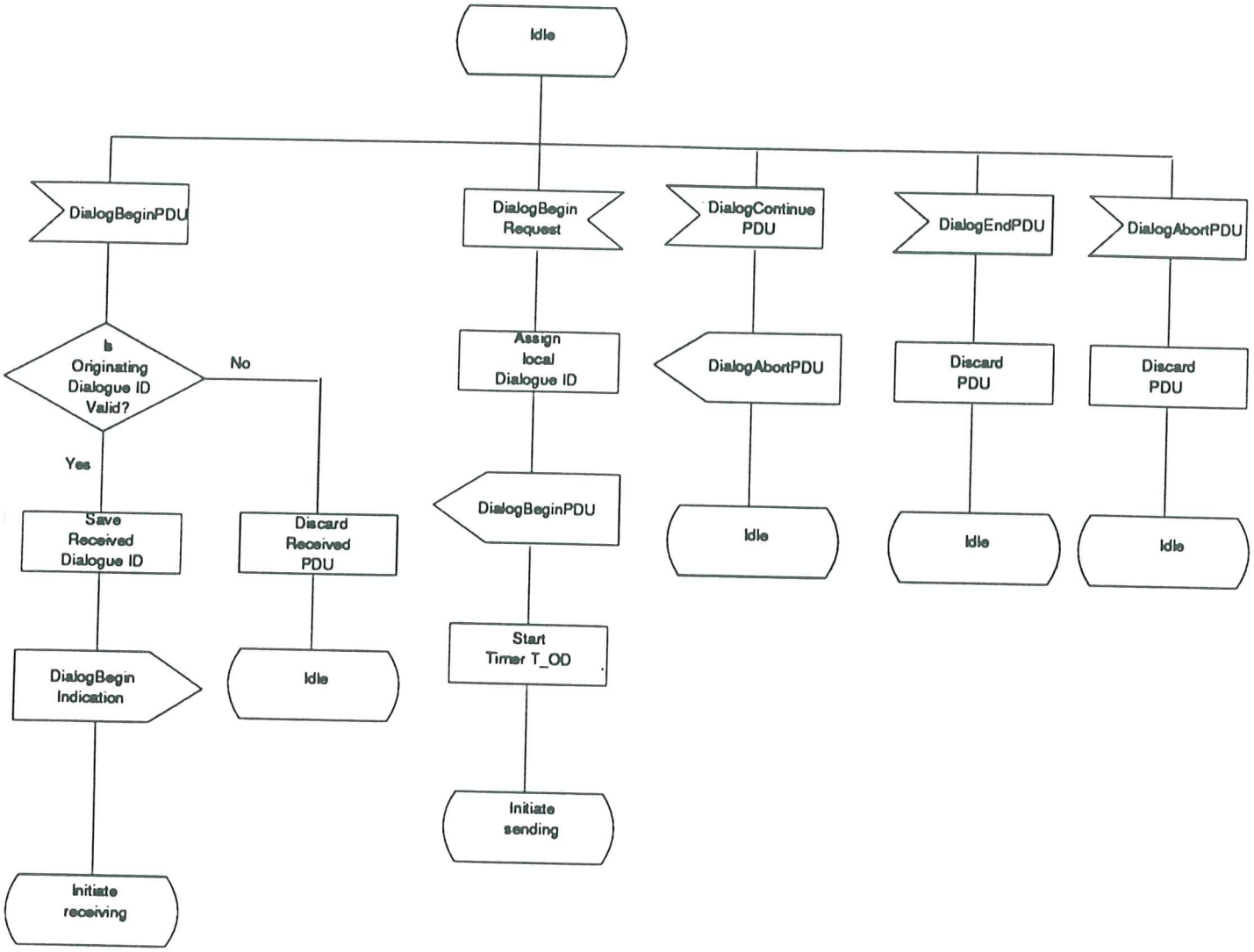


Figure 10 - Dialogue procedures dynamic description (sheet 1 of 4)

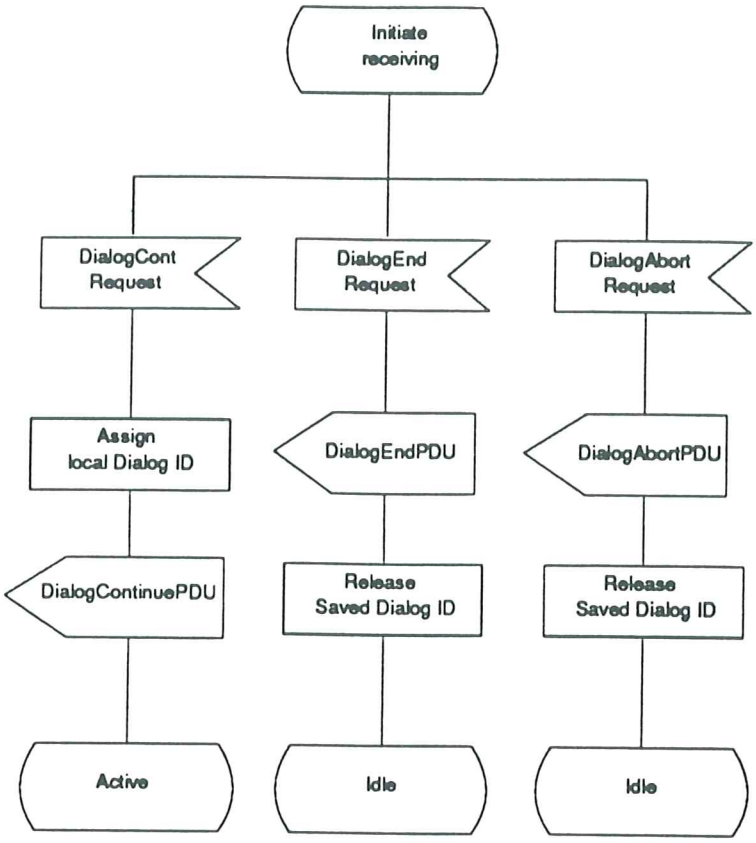


Figure 10 - Dialogue procedures dynamic description (sheet 2 of 4)

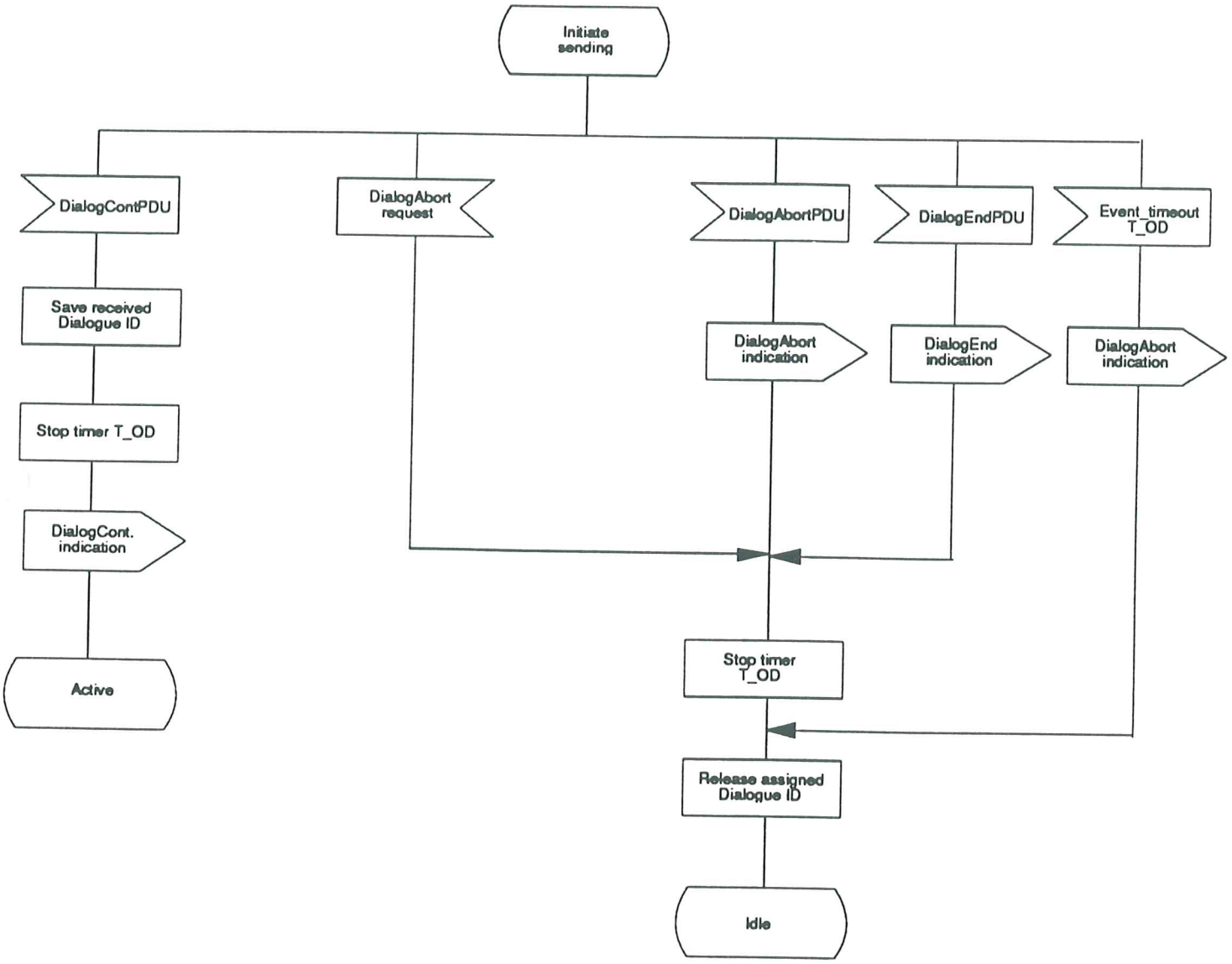


Figure 10 - Dialogue procedures dynamic description (sheet 3 of 4)

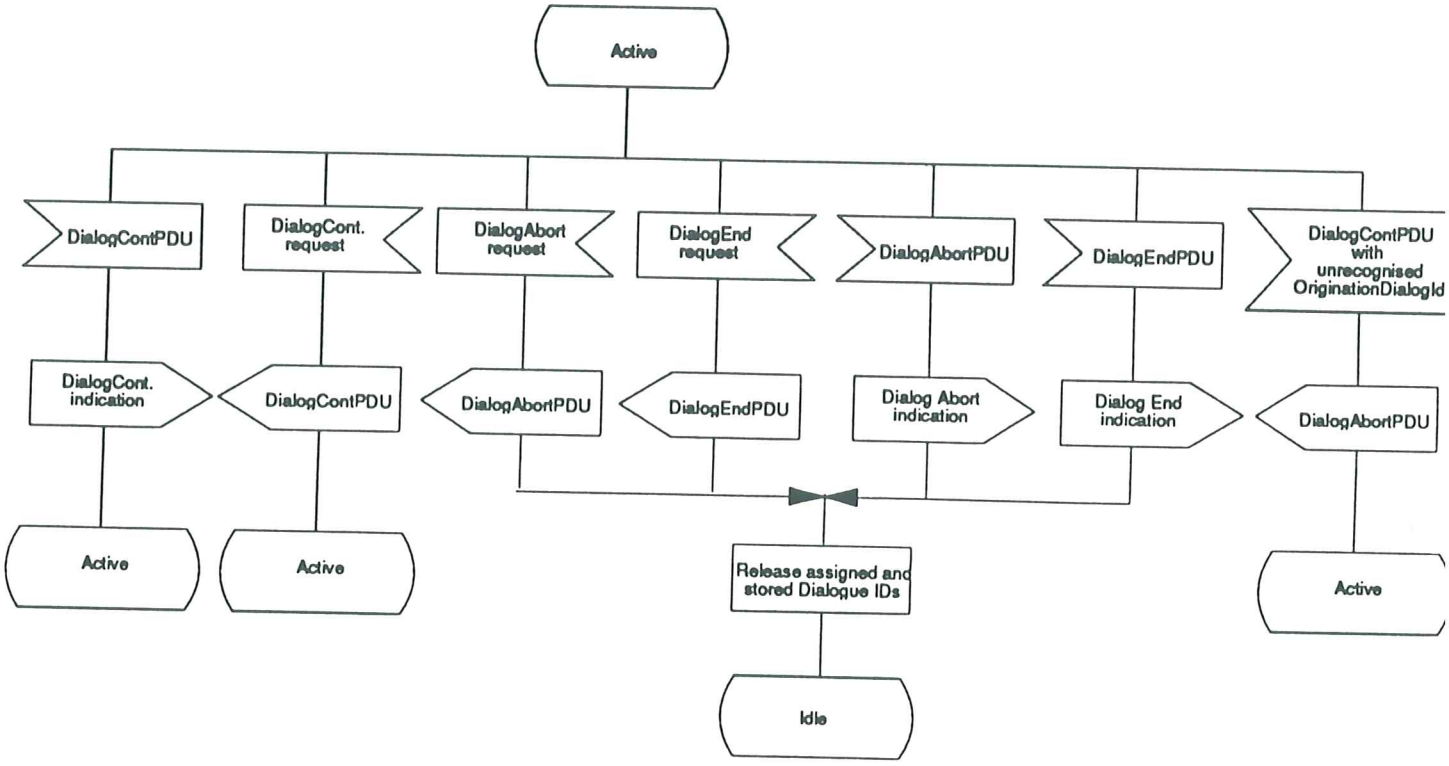


Figure 10 - Dialogue procedures dynamic description (sheet 4 of 4)

8.4 SS-Control requirements

The requirements for SS-Control are Supplementary service specific and are beyond the scope of this Standard.

9 Manufacturer Specific Information

ECMA QSIG permits the inclusion in messages of non-standardised information which is specific to a particular design of PTNX or a particular network etc. This information is known as Manufacturer Specific Information (MSI)

Manufacturer specific information may exist in the PTN as a result of the following:

- manufacturer specific Supplementary services;
- manufacturer specific extensions to ECMA Standard Supplementary services; or
- manufacturer specific notifications

In all these cases, any information which is manufacturer specific shall be encoded in such a way that it can be uniquely identified. Apart from the use of information elements belonging to codesets 6 or 7, as described in Annex C of ECMA-143 for conveyance of MSI to an Adjacent PTNX, any manufacturer specific information generated by a PTNX conforming to this Standard shall be encoded in conformance with the contents of this clause.

9.1 Manufacturer specific operations

Manufacturer specific operations shall conform to the encoding and transport rules defined for standardised operations in other clauses of this Standard, but in addition shall make use of operation values which are

unique to that manufacturer - i.e. of type **OBJECT IDENTIFIER**. If any non-standardised error values are to be included in a manufacturer specific operation, they shall be of type **OBJECT IDENTIFIER**. Examples of how manufacturer specific operations may be encoded are shown in Annex D.

NOTE 23

*All operations and errors specific to ECMA will be of type **OBJECT IDENTIFIER**.*

9.2 Manufacturer specific additions to standardised operations

As an alternative to the definition of a manufacturer specific operation, a manufacturer may wish to use an enhanced form of a standardised operation defined by ECMA.

NOTE 24

This may be used, for example, to include additional parameters which are manufacturer specific as part of the Standard service (e.g. information describing the detailed location of a party involved in the service).

To allow for this possibility, ECMA Standards for Supplementary services will include 'placeholders' for manufacturer specific extensions. Each placeholder will be an optional CHOICE construct containing an element of type **Extension** or a sequence of elements of type **Extension** (as defined in Table 5) with the argument, result or error parameter of an operation. This placeholder may be included in the ROSE APDU if MSI is to be conveyed. An element of type **Extension** shall contain an element of type **OBJECT IDENTIFIER** to uniquely identify the MSI.

If the Destination PTNX identifies an element of type **Extension** or a sequence of elements of type **Extension** in an ECMA standardised operation, when processing the contents of a received Facility information element in accordance with the relevant ECMA Supplementary service Standard, it shall act on an element of type **Extension** only if it recognises the value in the element of type **OBJECT IDENTIFIER** (see table 5). Otherwise the entire element of type **Extension** shall be discarded. In the case of a sequence of elements of type **Extension** (i.e. where multiple extensions to the service are defined) the PTNX shall consider each element of type **Extension** separately - that is, only those elements of type **Extension** containing an unrecognised value in the element of type **OBJECT IDENTIFIER** shall be discarded.

Table 5 - ECMA Manufacturer specific extension mechanism

```
ECMA-manufacturer-specific-service-extension-definition
      { iso( 1) identified-organization( 3) icd-ecma( 0012) standard( 0) qsig-
generic-          procedures( yyy) msi-definition( 0) }

DEFINITIONS ::=
EXPORTS     Extension, EXTENSION;
BEGIN
EXTENSION MACRO ::=
BEGIN
            TYPE NOTATION  ::= Argument
            VALUE NOTATION ::= Value (VALUE(OBJECT IDENTIFIER))
            Argument       ::= "Argument" NamedType
            NamedType      ::= identifier type|type
END  -- of EXTENSION macro

Extension    ::=   SEQUENCE
                       {  manufacturer    EXTENSION,
                          ANY DEFINED BY  manufacturer
                       }

END   -- of ECMA-manufacturer-specific-service-extension-definition
```

An example of the use of the **Extension** type is shown in annex D.

9.3 Manufacturer specific notifications

Manufacturer specific notifications may occur in the PTN as part of manufacturer specific Supplementary services or as additions to standardised Supplementary services. If provided, they shall be encoded and transported across the PTN in accordance with the rules for ECMA standardised notifications (see 7.4, 10 and 11.3.4).

Manufacturer specific notifications shall make use of the type **NotificationDataStructure** in octet 3.1 of the Notification indicator information element (see 11.3.4). Elements of type **NotificationDataStructure** shall include an element **notificationTypeID** of type **OBJECT IDENTIFIER**. Additional information accompanying standardised notifications shall be included in element **notificationArgument**.

Manufacturer specific notifications shall not make use of the notification description field (octet 3) of the Notification indicator information element, other than to include the 'discriminator for notification extension' codepoint (see 11.3.4).

10 Message functional definitions and contents

This clause describes additions to the call control messages defined in clause 11 of ECMA-143 and a number of new messages. The tables in this clause follow the conventions described in the introduction of clause 11 of ECMA-143.

Table 6 summarises the messages that may also be used for the transport of APDUs and notification information, including those already defined in ECMA-143.

Table 6 - ECMA QSIG messages used for the transport of APDUs and notification Information

Call establishment messages	**Reference:**
ALERTING	10.1
CONNECT	10.2
SETUP	10.3
Call clearing messages	**Reference:**
DISCONNECT	10.4
RELEASE	10.5
RELEASE COMPLETE	10.6
Miscellaneous messages	**Reference:**
FACILITY	10.7
NOTIFY	10.8
PROGRESS	10.9

10.1 ALERTING

11.2.1 of ECMA-143 shall apply, with the following modification:

- the information elements shown in table 7 may also be included:

Table 7 - ALERTING message content

Information Element	Reference	Type	Length
Facility	11.3.3	O	3 - *
Notification Indicator	11.3.4	O	3 - *

10.2 CONNECT

Sub-clause 11.2.3 of ECMA-143 shall apply, with the following modification:

- the information elements shown in table 8 may also be included:

Table 8 - CONNECT message content

Information Element	Reference	Type	Length
Facility	11.3.3	O	3 - *
Notification Indicator	11.3.4	O	3 - *

10.3 SETUP

Sub-clause 11.2.10 of ECMA-143 shall apply, with the following modification:

- the information elements shown in table 9 may also be included:

Table 9 - SETUP message content

Information Element	Reference	Type	Length
Facility	11.3.3	O	3 - *
Notification Indicator	11.3.4	O	3 - *

NOTE

Because of additional coding possibility in 11.3.2, the length of the Channel identification information element can be 3 octets.

10.4 DISCONNECT

Sub-clause 11.2.5 of ECMA-143 shall apply, with the following modification:

- the information elements shown in table 10 may also be included:

Table 10 - DISCONNECT message content

Information Element	Reference	Type	Length
Facility	11.3.3	O	3 - *
Notification Indicator	11.3.4	O	3 - *

10.5 RELEASE

Sub-clause 11.2.8 of ECMA-143 shall apply, with the following modification:

- the information elements shown in table 11 may also be included:

Table 11 - RELEASE message content

Information Element	Reference	Type	Length
Facility	11.3.3	O	3 - *

10.6 RELEASE COMPLETE

Sub-clause 11.2.9 of ECMA-143 shall apply, with the following modification:

- the information elements shown in table 12 may also be included:

Table 12 - RELEASE COMPLETE message content

Information Element	Reference	Type	Length
Facility	11.3.3	O	3 - *

10.7 FACILITY

This message, as shown in table 13, may be sent to transport APDUs. For the use of this message, refer to clause 7.

Table 13 - FACILITY message content

Message Type: FACILITY

Direction: Both

Information Element	Reference	Type	Length
Protocol Discriminator	12.2/ECMA-143	M	1
Call Reference	11.2	M	1 - 3 (note 1)
Message Type	11.1	M	1
Facility	11.3.3	M	3 -*
Notification Indicator	11.3.4	O	3 - *
Calling party number	12.5/ECMA-143	O (note 2)	4 - *
Called party number	12.5/ECMA-143	O (note 2)	4 - *

NOTE 1:

When the FACILITY message is used in a Connectionless manner, the dummy call reference (see 11.2) shall be used

NOTE 2:

This information element is mandatory when the FACILITY message is used in a Connectionless manner, otherwise it shall not be included.

10.8 NOTIFY

This message may be sent by a PTNX to provide notifications to a user, in association with a Basic call.

For the use of this message, see 7.4.

Table 14 - NOTIFY message content

Message Type: NOTIFY

Direction: Both

Information Element	Reference	Type	Length
Protocol Discriminator	12.2/ECMA-143	M	1
Call Reference	11.2	M	3
Message Type	11.1	M	1
Notification Indicator	11.3.4	M	3 - *

10.9 PROGRESS

Sub-clause 11.2.7 of ECMA-143 shall apply, with the following modification:

- the information elements shown in table 15 may also be included:

Table 15 - PROGRESS message content

Information Element	Reference	Type	Length
Facility	11.3.3	O	3 - *
Notification Indicator	11.3.4	O	3 - *

11 General Message Format and Information Element Coding

This clause describes information element coding in addition to that defined in clause 12 of ECMA-143.

Where the contents of an information element field are described using ASN.1 notation, the encoding of thi field shall be in accordance with the Basic Encoding Rules (BER) defined in X.209.

Any message can be subject segmentation in accordance with the procedures of 7.2 of ECMA-143.

11.1 Message Type

The following message type codings are additional to those defined in 12.4 of ECMA-143 and are used fo the Supplementary service specific messages defined in clause 10.

Table 16 - Message types applicable Over the PTN

Bits
8 7 6 5 4 3 2 1
0 1 1 - - - - - Miscellaneous Messages Group
- - - 0 0 0 1 0 FACILITY
- - - 0 1 1 1 0 NOTIFY

11.2 Call reference

Sub-clause 12.3 of ECMA-143 shall apply, with the following addition:

- The dummy call reference defined in figure 11 shall be used when a FACILITY message is sent ir accordance with the procedures of 7.2.

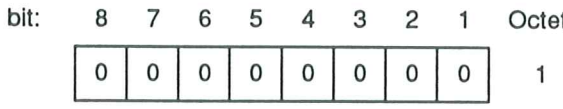


Figure 11 - Dummy Call Reference

11.3 Other Information Elements

For the information elements defined in this clause, the coding and presentation rules defined in 12.5 of ECMA-143 shall apply. Table 17 lists the information element codings in codeset zero defined in this Standard in addition to those defined in table 24, 12.5 of ECMA-143 .

Table 17 - Additional codeset zero information elements

Bits 8 7 6 5 4 3 2 1	
0 - - - - - - -	Variable Length Information Elements
- 0 0 1 1 1 0 0	Facility
- 0 1 0 0 1 1 1	Notification Indicator
All other values are reserved	

11.3.1 Bearer capability

Sub-clause 12.5.5 of ECMA-143 shall apply with the additional codepoints in table 18:

Table 18 - Additional codepoints defined for Channel identification

Coding standard (octet 3)

Bits 7 6	
01	other international standard (Note)

Information transfer capability (octet 3) for coding standard 'other international standard'

Bits 5 4 3 2 1	
0 1 0 0 0	unrestricted digital information
All other values are reserved	

Transfer mode (octet 4) for coding standard 'other international standard'

Bits 7 6	
0 0	circuit mode
All other values are reserved	

Information transfer rate (octet 4, bits 5 to 1) for coding standard 'other international standard'

Bits 5 4 3 2 1	
0 0 0 0 0	Call independent signalling connection
All other values are reserved	

Note When this coding standard is indicated, the coding defined in 12.5.5 of ECMA-143 shall apply for octets 1 to 2. Information transfer capability, Transfer mode and Information transfer rate shall be encoded as indicated and no other octets shall be included.

11.3.2 Channel identification

Sub-clause 12.5.12 of ECMA-143 shall apply with the additional codepoints in table 19:

Table 19 - Additional codepoints defined for Channel Identification

Information channel selection (octet 3)	
Bits	
2 1	
0 0	no channel (Note)
Note	When this coding is indicated, octets 3.2 and 3.3 shall be omitted.

11.3.3 Facility

This clause defines only the structure and coding of the Facility information element. The purpose of Facility information element is to convey an optional Interpretation APDU and one or more ROSE APDU and/or DSE APDUs.

All APDUs contained in the Facility information element will be delivered to the same PTNX (as identified by the NFE). If the different APDUs are to be processed by different PTNXs, they shall be included in different Facility information elements.

The Facility information element may be repeated in a given message. The maximum length of the Facility information element is application dependent. The Facility information element is defined in figure 12 and tables 20 through 26.

bit: 8	7	6	5	4	3	2	1	Octet:
	Facility Information							
0	0	0	1	1	1	0	0	1
	element identifier							
Length of Information element contents								2
1 ext	0 spare	0 spare	Service Discriminator					3
NetworkFacilityExtension (note 1)								4 * etc
InterpretationApdu (note 2)								*
ComponentPart (note 3)								

NOTE 1

*An element of type **NetworkFacilityExtension** may be included, in accordance with the procedures of clause 7.*

NOTE 2

*An element of type **InterpretationApdu** may be included, in accordance with the procedures in 8.1.*

NOTE 3

*One or more elements of type **ComponentPart** shall be included.*

Figure 12 - Facility information element

Table 20 - Service Discriminator Coding (octet 3)

Bits 5 4 3 2 1	
1 0 0 0 1	Discriminator for Supplementary service Applications
All Other values are reserved.	

Table 21 - Component part Coding

```
Component-part-definition
            { iso( 1) identified-organization( 3) icd-ecma( 0012) standard( 0)
                  qsig-generic-procedures( yyy) component-part-definition( 1)
}

DEFINITIONS  ::=
EXPORTS              ComponentPart;
IMPORTS              RoseAPDU            FROM  Functional-protocol-notation
                     {iso( 1) identified-organization( 3) icd-ecma( 0012)
standard( 0)                                qsig-generic-procedures( yyy) remote-
operations-pdus( 5) };
                     DseAPDU              FROM  Dialogue-identification-pdus
                     {iso( 1) identified-organization( 3) icd-ecma( 0012)
standard( 0)                                qsig-generic-procedures( yyy) dialog-
identification-pdus( 4) };
BEGIN

ComponentPart        ::=  CHOICE
                            { roseApdu   RoseAPDU,
                              dseApdu    DseAPDU
                            }

END -- of Component part definition
```

11.3.3.1 Network Facility Extension Coding

Table 22 describes the encoding of the element of type **NetworkFacilityExtension**. This provides a means of routing the contents of the Facility information element within the context of a call across the PTN, and a means of identifying the originator of the information, in accordance with the procedures of clause 7.

Table 1 in 7.1.2.1 describes the particular encodings of the element of type **NetworkFacilityExtension**.

Table 22 - Network Facility Extension Coding

```
Network-Facility-Extension
            { iso( 1) identified-organization( 3) icd-ecma( 0012) standard( 0)
                  qsig-generic-procedures( yyy) Network-facility-extension(
2) }

DEFINITIONS  ::=
EXPORTS      NetworkFacilityExtension;
IMPORTS      PartyNumber FROM Addressing-Data-Elements
                         { ccitt( 0) identified-organization
                            etsi( 0) 196 addressing-data-elements( 6) };
BEGIN

NetworkFacilityExtension ::=    [10] IMPLICIT SEQUENCE
            { sourceEntity           [0] IMPLICIT EntityType,
              sourceEntityAddress    [1] IMPLICIT AddressInformation OPTIONAL,
              destinationEntity          [2] IMPLICIT EntityType,
              destinationEntityAddress   [3] IMPLICIT AddressInformation
      OPTIONAL
            }

EntityType   ::=   ENUMERATED
              { endPTNX( 0),
                anyTypeOfPTNX( 1)
              }

AddressInformation     ::=       PartyNumber

END     -- of ECMA Network Facility Extension
```

11.3.3.2 Interpretation APDU

Table 23 describes the encoding of the element of type **InterpretationAPDU**. This APDU provides a means whereby the originator can include optional instructions to the receiving PTNX for use in the event that it does not understand the operation value of an **invokePDU** contained in an element of type **ComponentPart** of the Facility information element.

Sub-clause 8.1 describes the use of the element of type **InterpretationAPDU.**

Table 23 - Interpretation APDU Coding

```
Interpretation-Apdu
       { iso( 1) identified-organization( 3) icd-ecma( 0012) standard( 0)
             qsig-generic-procedures( yyy) Interpretation-apdu( 3) }

DEFINITIONS            ::=
EXPORTS               InterpretationApdu;
BEGIN

InterpretationApdu :=     [11] IMPLICIT ENUMERATED
                           { discardAnyUnrecognisedInvokePdu( 0),
                             clearCallIfAnyInvokePduNotRecognised( 1),
                             rejectAnyUnrecognisedInvokePdu( 2)
                             -- this coding is implied by the absence of an
                             -- interpretation APDU.
                           }

END     -- of ECMA Interpretation-Apdu
```

11.3.3.3 DSE APDU

Table 24 provides the formal ASN.1 (X.208) definition of the DSE APDUs.

Table 24 - Formal Definition of DSE APDUs (sheet 1 of 2)

```
Dialog-Service-Pdus
      {iso( 1) identified-organization( 3) icd-ecma( 0012) standard( 0)
            qsig-generic-procedures( yyy) dialog-service-pdus( 4) }

DEFINITIONS ::=
BEGIN
EXPORTS      DseAPDU;
IMPORTS      RoseAPDU            FROM  Functional-Protocol-Notation
      {iso( 1) identified-organization( 3) icd-ecma( 0012) standard( 0)
            qsig-generic-procedures( yyy) remote-operations-pdus( 5) },
      Extension           FROM
         {iso( 1) identified-organization( 3) icd-ecma( 0012) standard( 0)
            qsig-generic-procedures( yyy) msi-definition( 0) };

DseAPDU      ::=    CHOICE
                        { begin      [12] IMPLICIT DialogBeginPDU,
                          end        [14] IMPLICIT DialogEndPDU,
                          continue   [15] IMPLICIT DialogContinuePDU,
                          abort      [17] IMPLICIT DialogAbortPDU
                        }

DialogBeginPDU ::= SEQUENCE
                        { OriginationDialogId,
                          RemoteOperationsPortion       OPTIONAL
                        }

DialogEndPDU ::=    SEQUENCE
                        { DestinationDialogId,
                          RemoteOperationsPortion       OPTIONAL
                        }

DialogContinuePDU ::= SEQUENCE
                        { OriginationDialogId,
                          DestinationDialogId,
                          RemoteOperationsPortion       OPTIONAL
                        }

DialogAbortPDU ::= SEQUENCE
                        { DestinationDialogId,
                          CHOICE
                              { P-AbortCause,
                                UserAbortInformation
                              }
                  }
```

Table 24 - Formal Definition of DSE APDUs (sheet 2 of 2)

```
OriginationDialogId          ::= [0] IMPLICIT OCTET STRING SIZE(0..8)

DestinationDialogId          ::= [1] IMPLICIT OCTET STRING SIZE(0..8)

P-AbortCause                 ::= [2] IMPLICIT INTEGER
                                 { unrecognisedDseApdu( 0),
                                   unrecognisedDialogId( 1),
                                   badlyFormattedDseApdu( 2),
                                   incorrectDseApdu( 3),
                                   resourceLimitation( 4)
                                 }  (0..255)

UserAbortInformation     ::=     [3] IMPLICIT Extension

RemoteOperationsPortion  ::=     [4] IMPLICIT SEQUENCE OF RoseAPDU

END    -- of ECMA Dialog-service-pdus
```

11.3.3.4 ROSE APDUs

This clause defines the contents and form of the ROSE APDUs that will be used to control Supplementary services over the PTN. The protocol used by Supplementary services shall be in conformance with the ROSE protocol (see clause 7 of X.229). The ROSE APDUs defined in this Standard in order to support

the ROSE protocol are based on the RO-APDUs defined in X.229. They are:

- InvokePDU (based on ROIV-APDU)
- ReturnResultPDU (based on RORR-APDU)
- ReturnErrorPDU (based on RORE-APDU)
- RejectPDU (based on RORJ-APDU)

The structure and contents of these ROSE APDUs are defined in Table 25.

NOTE 25

The definitions in Table 25 are equivalent to those contained in clause 9 of X.229 with the exception that a number of the ASN.1 types in Table 25 (e.g. ***InvokeIdType****) are size delimited to enhance interoperability in a multivendor PTN.*

NOTE 26

Annex B gives a general overview of the ROSE protocol and its constituent parts. Annex E provides definitions of the problem codes for use in the ***RejectPDU*** *types.*

ROSE APDUs used in the context of a Supplementary service shall be defined and encoded in accordance with ASN.1 rules (see X.208 and X.209). Definitions will appear in the relevant Supplementary service specifications (which can be ECMA Standards or manufacturer specific).

Certain Supplementary services may require the use of existing information elements within ROSE APDUs encoded according to the rules of 12.5 of ECMA-143 (with the exception of the Facility information element, which shall not be included as a parameter in this way). In such a case, these information elements shall be included within an element of type **QSIGInformationElement**. In this way, the ECMA-143 encoding for these information elements may be retained. If more than one information element is to be included as part of the same parameter, all the information elements

shall be grouped together within the same element of type **QSIGInformationElement**. The type **QSIGInformationElement** is encoded as shown in table 26.

Table 25 - ROSE APDU Encoding (sheet 1 of 2)

```
Remote-Operations-Apdus
      { iso( 1) identified-organization( 3) icd-ecma( 0012) standard( 0)
        qsig-generic-procedures( yyy) remote-operations-apdus( 5)  }
DEFINITIONS ::=
BEGIN

EXPORTS      RoseAPDU;
IMPORTS      OPERATION, ERROR FROM Remote-Operations-Notation
             { joint-iso-ccitt( 2) remote-operations( 4) notation( 0) };

RoseAPDU     ::=   CHOICE
                   {     invoke       [1] IMPLICIT InvokePDU,
                         retResult    [2] IMPLICIT ReturnResultPDU,
                         retError     [3] IMPLICIT ReturnErrorPDU,
                         reject       [4] IMPLICIT RejectPDU
                   }

InvokePDU    ::=   SEQUENCE
                   {     invokeID           InvokeIDType,
                         linkedID           [0] IMPLICIT InvokeIDType OPTIONAL,
                         operationValue     OPERATION,
                         argument           ANY DEFINED BY
                                            operationValue           OPTIONAL
                   }

ReturnResultPDU ::=      SEQUENCE
                   {     invokeID           InvokeIDType,
                         SEQUENCE
                         {     operationValue     OPERATION,
                               result             ANY DEFINED BY
                                                  operationValue }   OPTIONAL
                   }

ReturnErrorPDU ::= SEQUENCE
                   {     invokeID           InvokeIDType,
                         errorValue         ERROR,
                         parameter          ANY DEFINED BY
                                            errorValue   OPTIONAL
                   }

RejectPDU    ::=   SEQUENCE
                   {     invokeID     CHOICE
                                      { InvokeIDType,
                                        NULL },
                         problem      CHOICE
                                      { [0] IMPLICIT GeneralProblem,
                                        [1] IMPLICIT InvokeProblem,
                                        [2] IMPLICIT ReturnResultProblem,
                                        [3] IMPLICIT ReturnErrorProblem }
                   }
```

Table 25 - ROSE APDU Encoding (sheet 2 of 2)

```
InvokeIDType ::=    INTEGER(0..65535)

GeneralProblem      ::=    INTEGER
                    {      unrecognisedPDU( 0),
                           mistypedPDU( 1),
                           badlyStructuredPDU( 2)
                    } (0..255)

InvokeProblem       ::=    INTEGER
                    {      duplicateInvocation( 0),
                           unrecognisedOperation( 1),
                           mistypedArgument( 2),
                           resourceLimitation( 3),
                           initiatorReleasing( 4),
                           unrecognisedLinkedIdentifier( 5),
                           linkedResponseUnexpected( 6),
                           unexpectedChildOperation( 7)
                    } (0..255)

ReturnResultProblem       ::=    INTEGER
                    {      unrecognisedInvocation( 0),
                           resultResponseUnexpected( 1),
                           mistypedResult( 2)
                    } (0..255)

ReturnErrorProblem ::=    INTEGER
                    {      unrecognisedInvocation( 0),
                           errorResponseUnexpected( 1),
                           unrecognisedError( 2),
                           mistypedParameter( 3)
                    } (0..255)

END      -- of ECMA Remote-Operations-Apdus
```

Table 26 - Formal definition of Generic QSIG Parameter tags

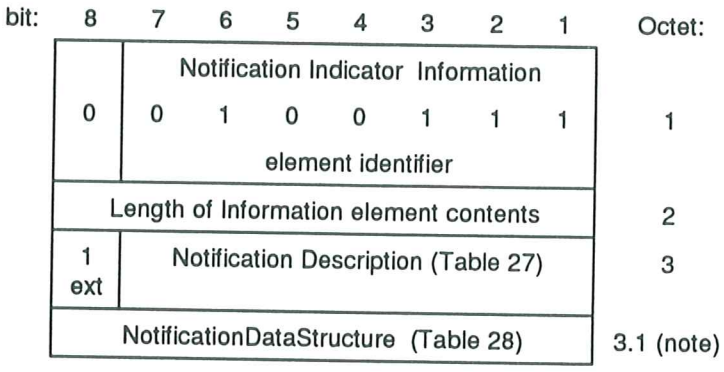
```
QSIG-generic-parameters-definition
      { iso( 1) identified-organisation( 3) icd-ecma( 0012) standard( 0)
            qsig-generic-procedures( yyy) qSIG-generic-parameters( 6)}

DEFINITIONS                ::=
BEGIN

EXPORTS      QSIGInformationElement, qsigIeNotification;

IMPORTS      NOTIFICATION FROM  Notification-Data-Structure
                  { iso( 1) identified-organisation( 3) icd-ecma( 0012)
standard( 0)                              qsig-generic-procedures( yyy)
                                notification-data-structure( 7)};

QSIGInformationElement          ::=    [APPLICATION  0] IMPLICIT OCTET STRING

-- this notification is used to convey information elements used as
notifications
-- across a PTN
qsigIeNotification       NOTIFICATION
                                ARGUMENT   QSIGInformationElement
                                ::= { iso( 1) idenitified-organisation( 3)
                                      icd-ecma( 0012)
                                      private-ISDN-signalling-domain( 9)
                                                qsig-ie-notification( 2001)
}

END  -- of QSIG-generic parameters definition
```

11.3.4 Notification Indicator

The purpose of the Notification indicator information element is to convey a notification.

The Notification indicator information element is coded as shown in figure 13 and tables 27 and 28. The maximum length of the information element is application dependent.

The Notification indicator information element may be repeated in a message.



NOTE

Octet 3.1 shall only be included when the notification description indicates the "discriminator for notification extension"

Figure 13 - Notification Indicator Information Element

Table 27 - Notification Description Encoding

Bits 7 6 5 4 3 2 1	
0 0 0 0 0 0 0	user suspended
0 0 0 0 0 0 1	user resumed
0 0 0 0 0 1 0	reserved
0 0 0 0 0 1 1	discriminator for notification extension
1 1 0 0 0 0 0	call is a waiting call
1 1 1 1 0 0 1	remote hold
1 1 1 1 0 1 0	remote retrieval

All other values are reserved, but shall be treated as valid.

Table 28 - ASN.1 encoded data structure

```
Notification-Data-Structure
      { iso( 1) idenitified-organisation( 3) icd-ecma( 0012) standard( 0)
            qsig-generic-procedures( yyy) notification-data-structure( 7) }

DEFINITIONS                ::=
BEGIN

EXPORTS      NOTIFICATION, NotificationDataStructure;

NOTIFICATION MACRO         ::=
BEGIN

TYPE NOTATION                    ::=  Argument
VALUE NOTATION             ::=  value ( VALUE CHOICE
                                        { localValue INTEGER,
                                          globalValue      OBJECT
IDENTIFIER
                                        }
Argument                   ::=  "ARGUMENT" NamedType
NamedType                  ::=  identifier type | type

END -- of NOTIFICATION MACRO

NotificationDataStructure       ::=  SEQUENCE
                                { notificationTypeID           NOTIFICATION,
                                  notificationArgument    ANY DEFINED BY
                                                          notificationTypeID
                                }

END -- of Notification-Data-Structure
```

Annex A

(informative)

Application of the Functional Protocol

A.1 Examples of the use of the functional protocol over the PTN

This Annex contains examples of the use and encoding of the functional protocol (as defined in clauses 6 to 9 of this Standard. It is intended as an example of the potential application or use of the protocol and is not intended to constrain the definition of particular Supplementary services.

A.2 Call related Supplementary services

In the figures in this clause, the notation shown in figure A.1 is used when referring to messages between nodes.

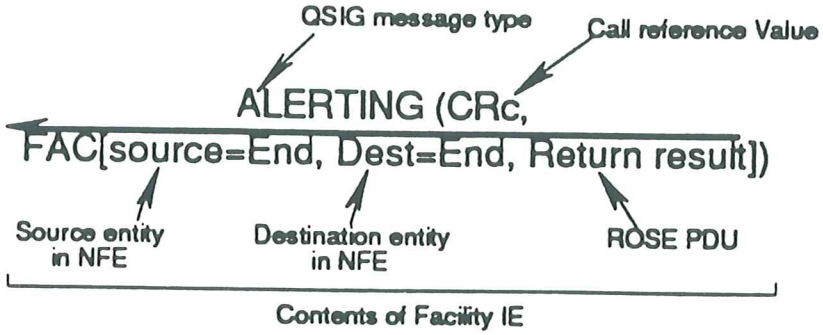


Figure A.1 - Notation for Call related supplementary services example message flows

The abbreviations 'end', and 'any' indicate the entity types 'endPTNX', and 'anyTypeOfPTNX' as defined in 11.3.3.1.

A.2.1 Call Establishment

A.2.1.1 End to end Service request

In this example, a service invocation is passed between the End PTNXs involved in a call, during call establishment. The Supplementary service used as an example is the 'Hypothetical-service-operation' as defined in Annex D, without any manufacturer specific extension.

NOTE A.1

Depending on the particular service, the result of processing the invocation may cause the call setup to fail in some circumstances.

Figure A.2 shows the transport of the end to end service request and response during call setup. Figure A.4 shows the encoding of the Facility information element sent in the original SETUP message. It contains an **InvokePDU** with a single integer argument (**hypotheticalParameter1**) and the operation value is give by its object identifier:

```
{ iso(1) identified-organization(3) icd-ecma(12) Standard(0)
      hypothetical-Standard(999) hypothetical-operation(1) }
```

This results in an object identifier of 6 octets in length, encoded in accordance with clause 22 of X.209 .

The **invokeIdentifier** chosen for this example was the arbitrary value '2'. This identifier is generated by the originator of the **InvokePDU** so that the any response received via the same underlying association (in this case the Basic call) can be correlated with the originally sent **InvokePDU**. The encoding of the **ReturnResultPDU** (sent in the ALERTING message of figure A.2) in figure A.5 illustrates the use of the **invokeIdentifier** to perform this correlation.

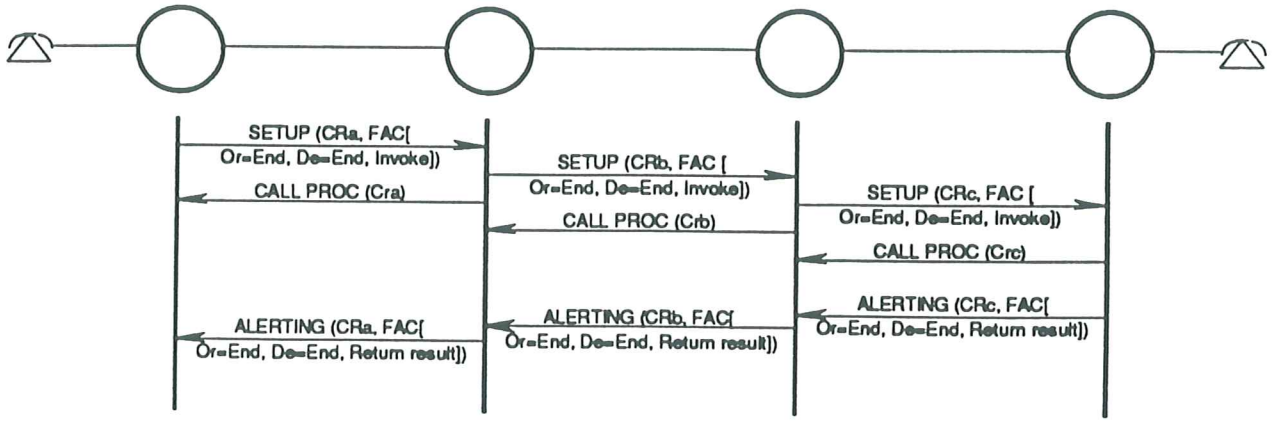


Figure A.2 - End to end service invocation on call setup

A.2.1.2 Link Service Request

Figure A.3 shows an example of a link by link service request and response during call setup. The service request is between two transit nodes and does not contain a Facility Network Extension octet group.

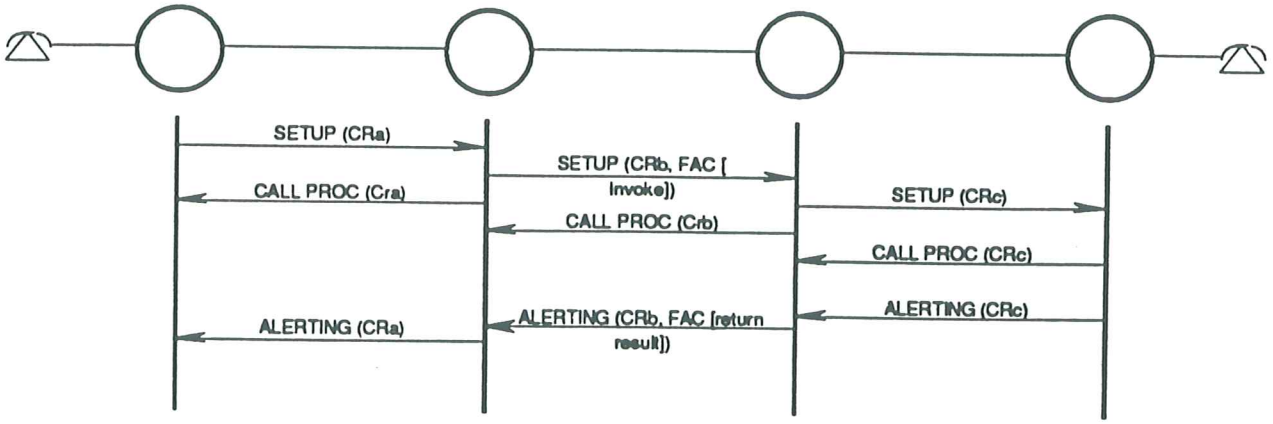


Figure A.3 - Link service request on call setup

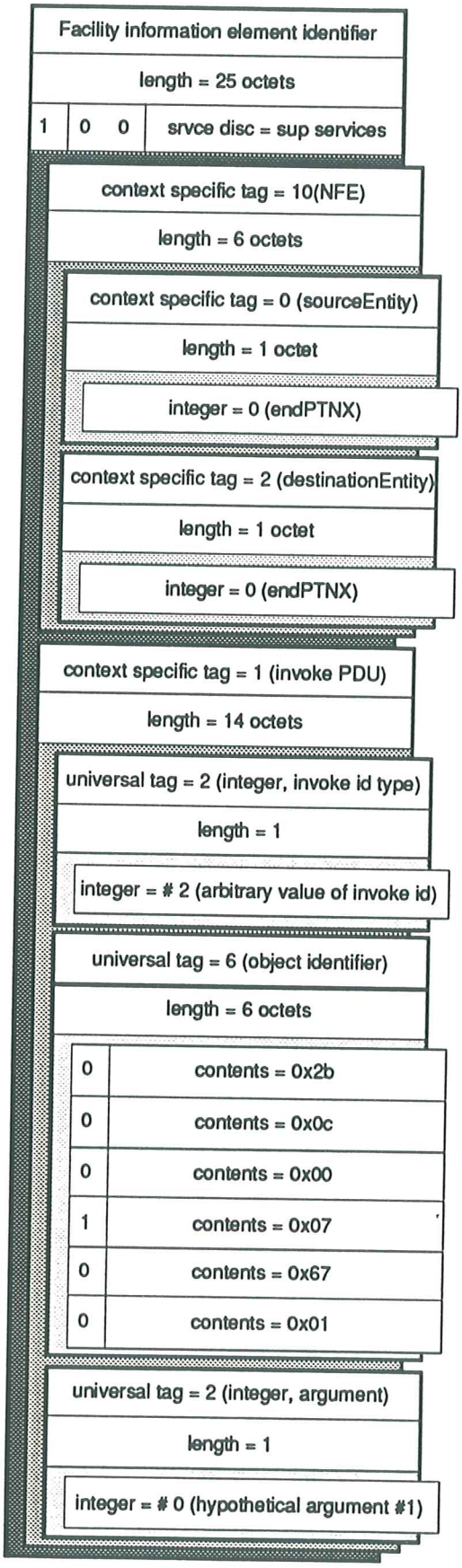


Figure A.4 - Encoding of InvokePDU from figure A.2

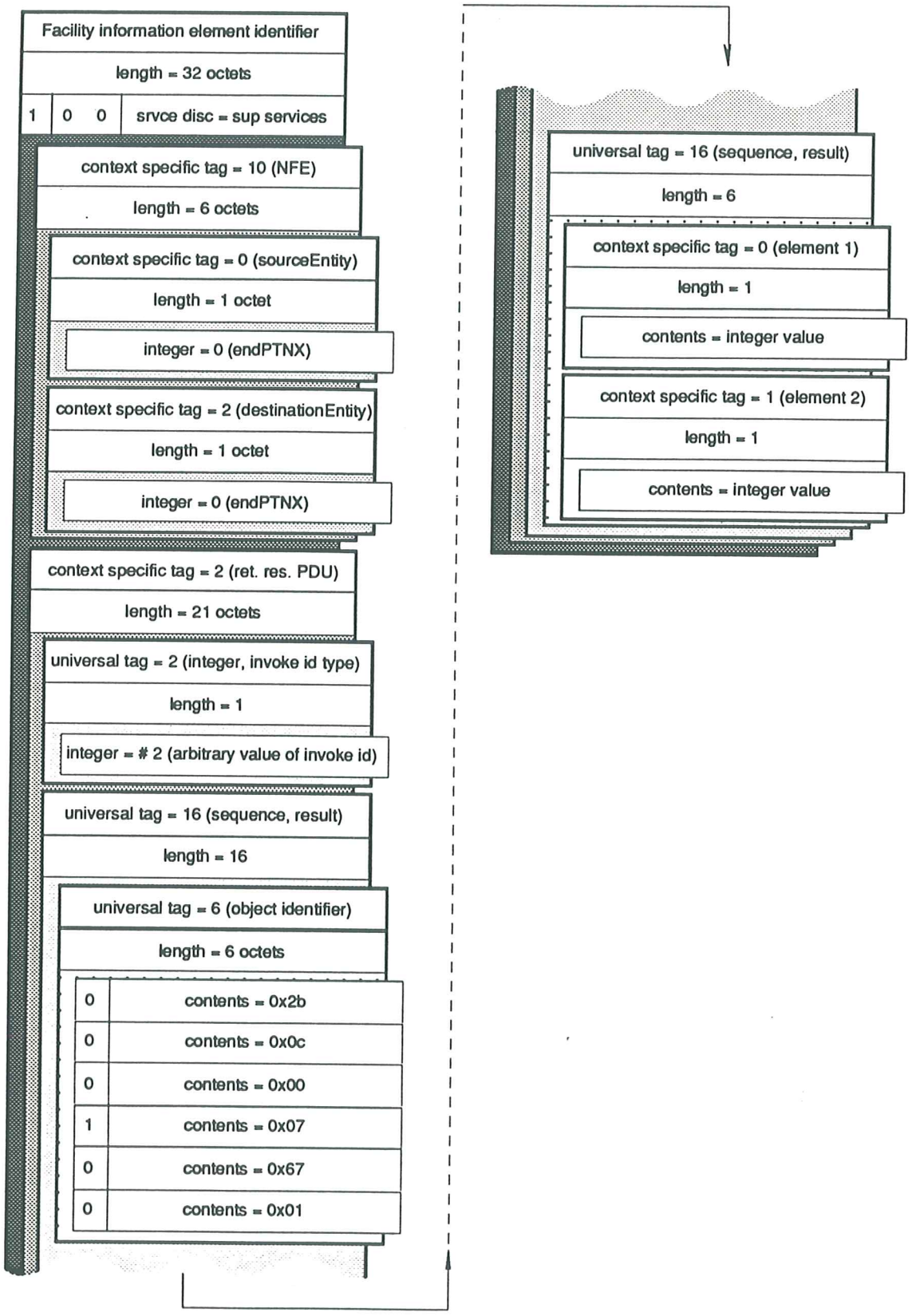


Figure A.5 - Encoding of ReturnResultPDU from ALERTING message of figure A.2

A.2.2 Call Clearing

A.2.2.1 End to End Request

Figure A.6 shows a call being cleared across the network, with an end to end service request. This request is a Class 5 ROSE operation which requires no response.

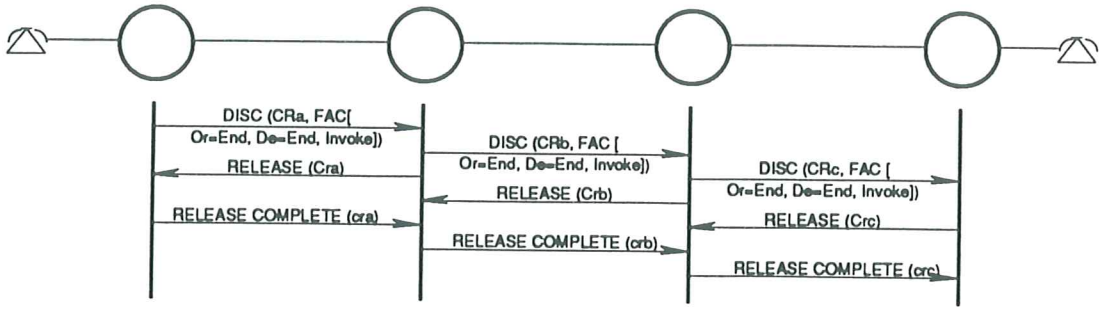


Figure A.6 - End to end service request on call clearing

A.2.2.2 Link Service Request

Figure A.7 shows a call being cleared across the network, with a link service request between two Transit PTNXs.

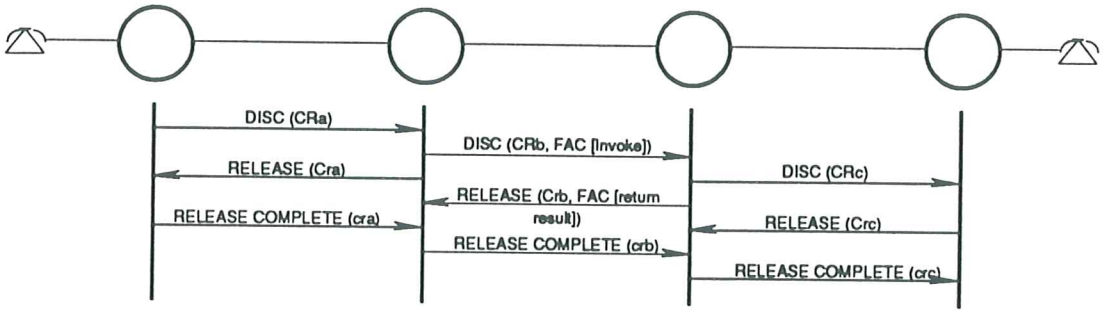


Figure A.7 - link service request on call clearing

A.2.3 Call Active

A.2.3.1 End to End Request

Figure A.8 shows an end to end service request and response during the active state of a call.

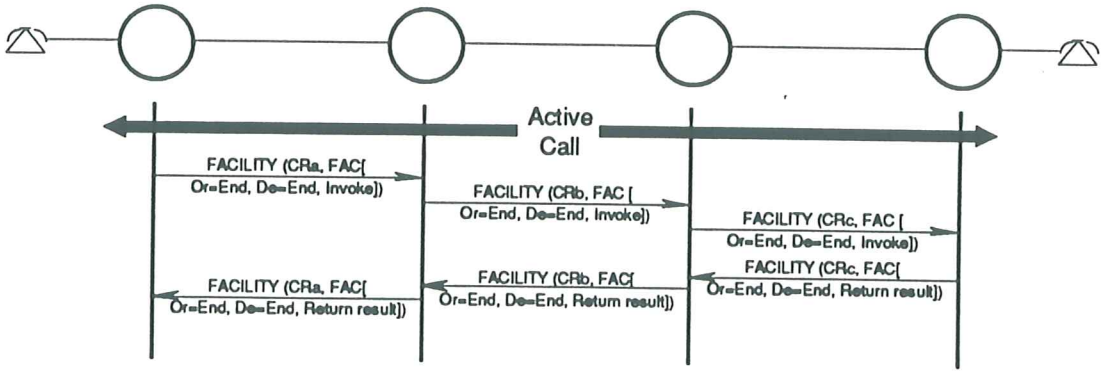


Figure A.8 - End to end service request during active call

A.2.3.2 Link Service Request

Figure A.9 shows a link by link service request and response during the active state of a call.

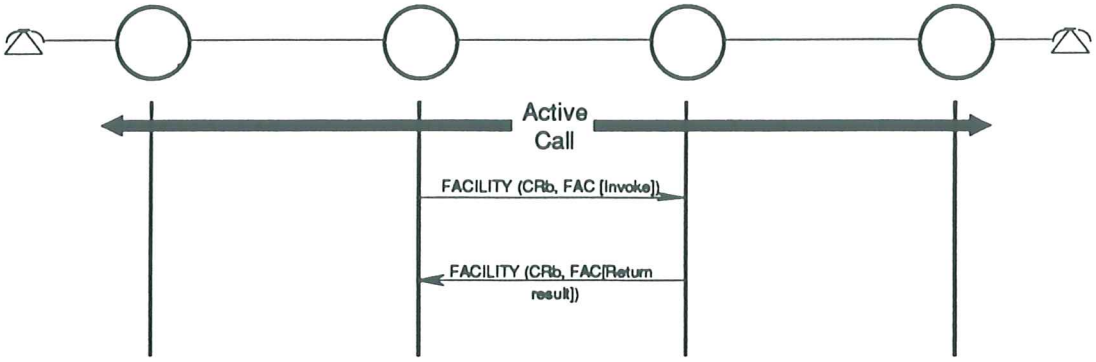


Figure A.9 - Link by Link service request during active call

A.3 Call independent Supplementary services

In this clause, the notation shown in figures A.10 and A.11 is used when referring to messages between nodes:

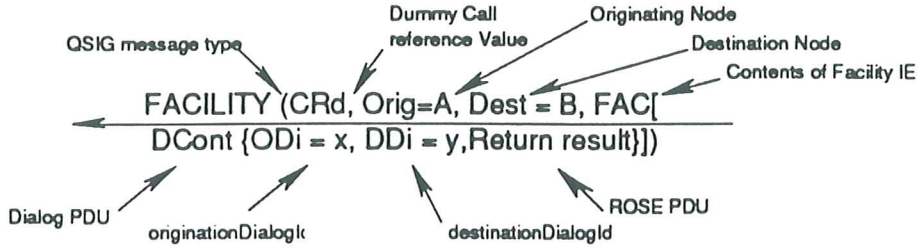


Figure A.10 - Notation for Connectionless Call independent message sequence examples

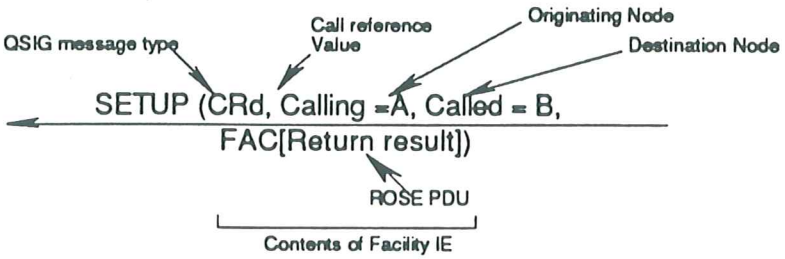


Figure A.11 - Notation for Connection oriented Call independent message sequence examples

The abbreviations DCont, DBeg and DEnd refer to the DialogContinuePDU, DialogBeginPDU and DialogEndPDU respectively, defined in clause 9.

A.3.1 Connectionless Transport

Figure A.12 shows service requests which are passed between two PTNXs

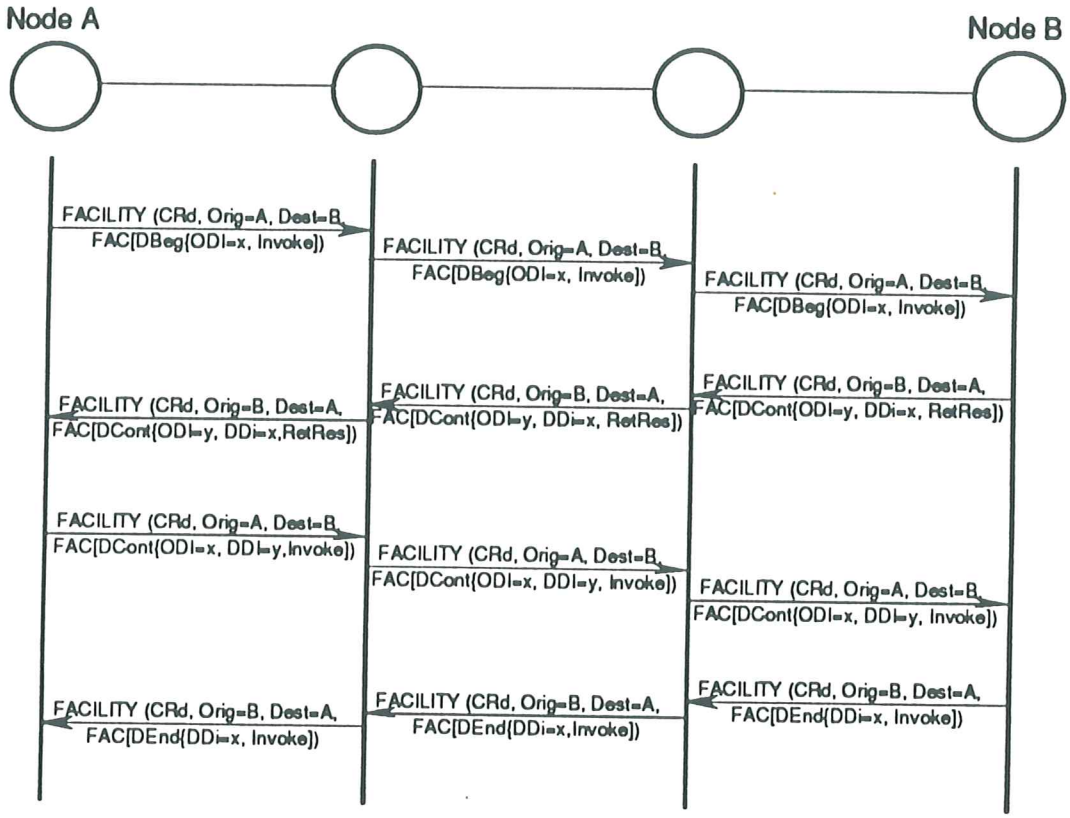


Figure A.12 - Connectionless end to end service request

A.3.2 Connection oriented Transport

Figure A.13 shows the establishment, active and clearing phases of a Call independent signalling connection between two PTNXs.

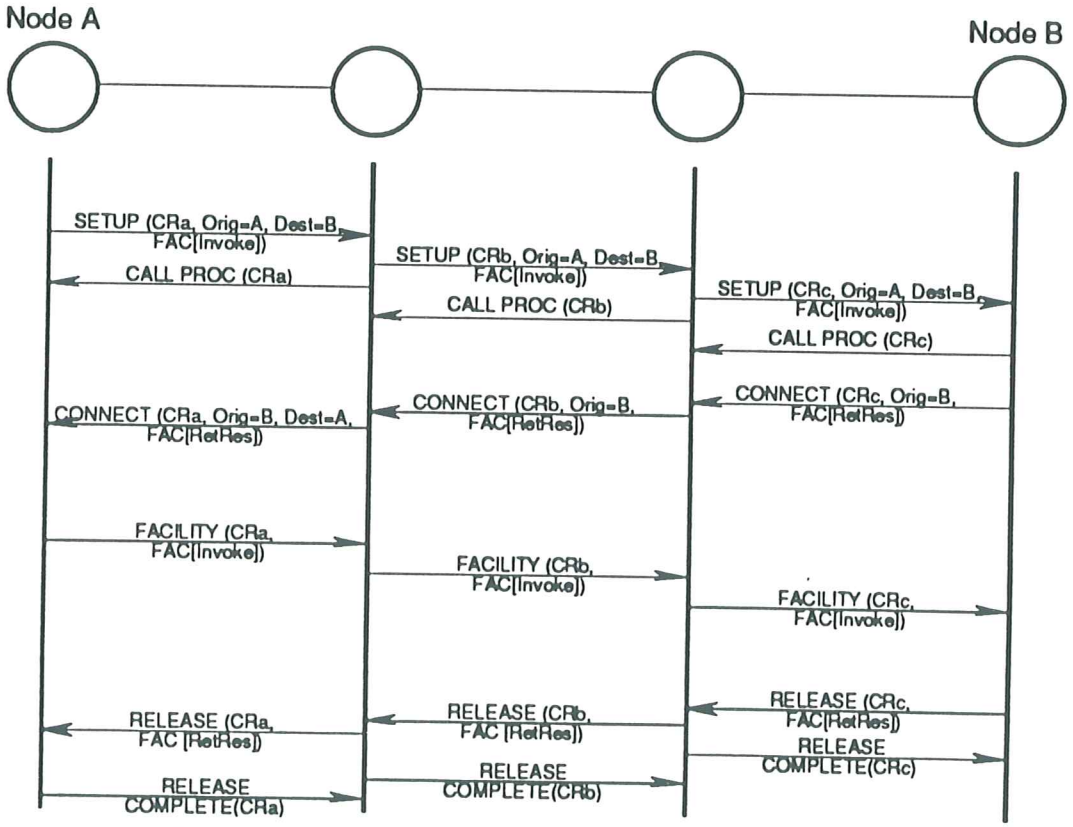


Figure A.13 - Connection oriented signalling connection

Annex B

(informative)

Remote Operations Protocol

The remote operations (RO) protocol is defined in CCITT Recommendations X.219/ X.229. The generic procedures defined in this ECMA Standard provide an encoding mechanism for the transport and use of this RO protocol in the PTN environment for the provision of Supplementary services or additional network features.

In the OSI environment, communication between application processes is represented in terms of communication between a pair of application entities (AEs). Communication between application entities are inherently interactive. Typically, one entity requests that a particular operation be performed; the other entity attempts to perform the operation and then reports the outcome of the attempts. The concept of Remote Operations is a vehicle for supporting interactive applications of this type.

The generic structure of an operation is an elementary request/reply interaction. Operations are carried out within the context of an application-association.

Figure B.1 models this view.

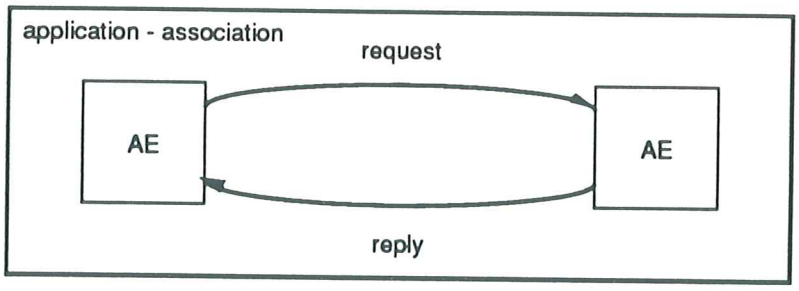


Figure B.1 - Remote Operations Model

Operations invoked by one AE (the invoker) are performed by the other AE (the performer). Operations may be classified according to whether the performer of an operation is expected to report its outcome:

- in the case of success or failure (a result reply is returned if the operation is successful, an error reply is returned if the operation is unsuccessful);
- in case of failure only (no reply is returned if the operation is successful, an error reply is returned if the operation is unsuccessful);
- in case of success only (a result reply is returned if the operation is successful, no reply is returned if the operation is unsuccessful);
- or not at all (neither a result nor an error reply is returned, whether the operation was successful or not).

Operations may also be classified according to two possible operation modes: synchronous, in which the invoker requires a reply from the performer before invoking another operation; and asynchronous, in which the invoker may continue to invoke further operations without awaiting a reply.

The following Operation Classes are defined:

Operation Class 1:	Synchronous, reporting success or failure (result or error).
Operation Class 2:	Asynchronous, reporting success or failure (result or error).
Operation Class 3:	Asynchronous, reporting failure (error) only, if any.
Operation Class 4:	Asynchronous, reporting success (result) only.
Operation Class 5:	Asynchronous, outcome not reported.

The Operation Class of each operation has to be agreed between application entities (e.g. in an Applicatior Protocol Standard).

In some cases, it is useful to group operations into a set of linked operations which is formed by one paren operation and one or more child operations. The performer of the parent operation may invoke none, one, o more child operations during the execution of the parent operation. The invoker of the parent operation is th performer of the child operations. A child operation may be a parent operation of another set of linkec operations in a recursive manner. Figure B.2 models this concept.

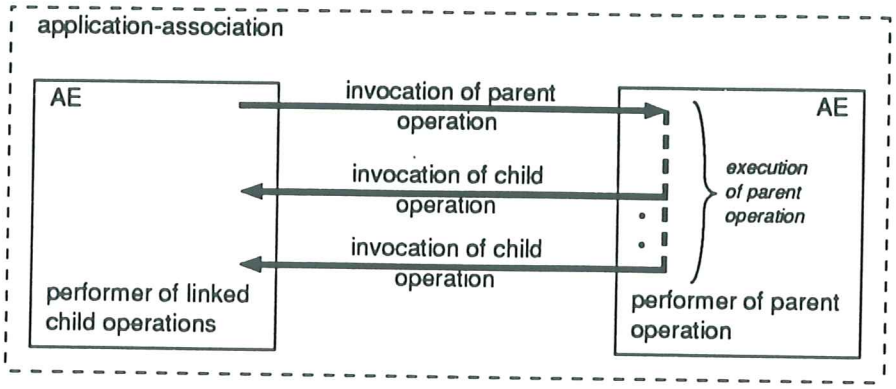


Figure B.2 - Linked Operations

An application association defines the relationship between a pair of AEs, and is formed by the exchange of application (in this case Supplementary services) Protocol Control information through the use of the services of underlying layers. The AE that initiates an association is called the association initiating AE, or the association initiator, while the AE that responds to the initiation of an application association by another is called the association responding AE, or the association responder.

NOTE B.1

In the application of ROSE for the support of Supplementary services in ECMA-QSIG, the underlying services used by ROSE are those provided by GFT-Control and not those provided by the Association Control Service Entity (ACSE) and the Reliable Transport Service Entity (RTSE).

Application associations are clasified by which application-entity is allowed to invoke operations:

Association Class 1:	Only the association-initiating application-entity can invoke operations.
Association Class 2:	Only the association-responding application-entity can invoke operations.
Association Class 3:	Both the association-initiating and the association-responding application-entities can invoke operations.

This Standard assumes Application associations of Association Class 3.

Annex C

(informative)

Formal Rose Definitions

Table C.1 in this Annex is an extract from CCITT Recommendation X.219 which describes the OPERATION and ERROR macros used for Remote operations. It also specifies the BIND and UNBIND macros, but these are not applicable to the protocol described in this Standard.

Table C.1 - Formal Definition of Data Types (sheet 1 of 2)

(extract from X.219, Blue Book)

```
Remote-Operation-Notation
      { joint-iso-ccitt( 2) remote-operations( 4) notation( 0) }

DEFINITIONS          ::=
BEGIN

EXPORTS      BIND, UNBIND, OPERATION, ERROR;

BIND MACRO           ::=
BEGIN
TYPE NOTATION                ::=   Argument Result Error
VALUE NOTATION       ::=     Argument-value | Result-value | Error-value

Argument             ::=     empty | "ARGUMENT" Name type (Argument-type)
Result               ::=     empty | "RESULT"   Name type (Result-type)
Error                ::=     empty | "BIND-ERROR"      Name type (Error-type)
Name                 ::=     empty | identifier
Argument-value       ::=     empty | "ARGUMENT" value (Arg-value Argument-type)
                             <VALUE [16] EXPLICIT Argument-type    ::= Arg-value>
Result-value         ::=     empty | "RESULT"   value (Res-value Result-type)
                             <VALUE [17] EXPLICIT Result-type      ::= Res-value>
Error-value          ::=     empty | "ERROR"    value (Err-value Error-type)
                             <VALUE [18] EXPLICIT Error-type::= Err-value>

END    -- of BIND macro


UNBIND MACRO         ::=
BEGIN
TYPE NOTATION                ::=   Argument Result Errors
VALUE NOTATION       ::=     Argument-value | Result-value | Error-value

Argument             ::=     empty | "ARGUMENT" Name type (Argument-type)
Result               ::=     empty | "RESULT"   Name type (Result-type)
```

Table C.1 - Formal Definition of Data Types (sheet 2 of 2)

(extract from X.219, Blue Book)

```
Error                ::=   empty | "UNBIND-ERROR" Name type (Error-type)
Name                 ::=   empty | identifier
Argument-value       ::=   empty | "ARGUMENT" value (Arg-value Argument-type)
                           <VALUE [19] EXPLICIT Argument-type    ::= Arg-value>
Result-value         ::=   empty | "RESULT"   value (Res-value Result-type)
                           <VALUE [20] EXPLICIT Result-type      ::= Res-value>
Error-value          ::=   empty | "ERROR"    value (Err-value Error-type)
                           <VALUE [21] EXPLICIT Error-type::= Err-value>

END    -- of UNBIND macro


OPERATION MACRO      ::=
BEGIN
TYPE NOTATION              ::=   Argument Result Errors LinkedOperations
VALUE NOTATION       ::=   value ( VALUE CHOICE
                                   { localValue INTEGER,
                                     globalValue      OBJECT IDENTIFIER
                                   }
                                 )

Argument             ::=   "ARGUMENT"   NamedType | empty
Result               ::=   "RESULT"     ResultType | empty
Errors               ::=   "ERRORS"     "{" ErrorNames "}" | empty
LinkedOperations     ::=   "LINKED"     "{" LinkedOperationNames "}" | empty
NamedType            ::=   identifier type | type
ResultType           ::=   NamedType | empty
ErrorNames           ::=   ErrorList | empty
ErrorList            ::=   Error | ErrorList "," Error
Error                ::=   value (ERROR) | type
LinkedOperationNames       ::=   OperationList | empty
OperationList              ::=   Operation | OperationList "," Operation
Operation            ::=   value (OPERATION) | type

END    -- of OPERATION MACRO

ERROR MACRO          ::=
BEGIN
TYPE NOTATION              ::=   Parameter
VALUE NOTATION       ::=   value ( VALUE CHOICE
                                   { localValue INTEGER,
                                     globalValue      OBJECT IDENTIFIER
                                   }
                                 )


Parameter      ::=   "PARAMETER"  NamedType | empty

NamedType      ::=   identifier type | type

END    -- of ERROR MACRO

END    -- of Remote Operations Notations
```

Annex D

(informative)

Examples of the Use of Manufacturer Specific Information

D.1 Manufacturer Specific Object Identifiers in Operation Values

As defined in 9.1, manufacturers who wish to provide manufacturer specific Supplementary services in a standardised manner should use unique operation values, constructed using manufacturer specific object identifiers.

Manufacturer specific object identifiers may be constructed in the following way. Manufacturers requiring an assigned identification may apply to a "Sponsoring and Issuing organisation" according to ISO 6523 and be assigned an organisation identifier. The manufacturer should then use that organisation identifier in an object identifier (as the root of the manufacturer specific service operation value) according to the structure defined by the issuing organisation.

ECMA is a "Sponsoring and Issuing organisation" for identification of organisations and has been assigned an International Code Designator (ICD). The use of ECMA issued organisation identifiers in object identifiers is as shown in table D.1. PTNXs conforming to this Standard need not make use of an organisation identifier issued by ECMA as an organisation identifier issued by any "sponsoring and issuing organisation" will produce a unique object identifier to uniquely identify the service concerned.

Table D.1 - Structure of ECMA Object Identifier

```
level 1: iso( 1)
level 2: identified-organization( 3)
level 3: icd-ecma( 0012)
level 4: a)  Standard( 0)
         b)  technical-report( 1)
         c)  member-company( 2)
         d)  private-ISDN-signalling-domain( 9)
         e), f) other common domains as required
level 5: for a) and b) of Level 4:
             number of Standard or technical report (3
             digits from 001 to 999).
         for c) of level 4:
             organisation identifier assigned by ECMA
         for d) of level 4:
                0 -  999  operation values assigned by ECMA TC32-TG6
             1000 - 1999  error values assigned by ECMA TC32-TG6
             2000 - 2999  notification values assigned by ECMA TC32-TG6
level 6: this level and others below it are used to suit the purpose of
         the respective level 4 a), b) or c).
```

Thus, according to table D.1, the ECMA object identifier for a company with the assigned organisation code '1999' (all organisation codes issued by ECMA have 4 digits of which the first is always '1'), may be structured as shown in table D.2. The contents of level 6 is manufacturer specific and may identify a company specific operation value or may not exist at all. In this example, level 6 provides a manufacturer specific operation value.

Table D.2 - ECMA Object Identifier for hypothetical manufacturer specific service operation

```
Object identifier for hypothetical manufacturer specific
service operation value:

HypotheticalManufacturerSpecificSupplementaryService ::=
{ iso( 1) identifier-organisation( 3) icd-ecma( 0012)
            member-company( 2) hypothetical-manufacturer( 1999)
              hypothetical-manufacturer-service( 1) }

In pure numeric notation, this would be:

{ 1 3 0012 2 1999 1 }

(This shall be encoded as described in CCITT recommendation X.209)
```

This object identifier value would then be used in the definition of the manufacturer specific oper: (internally to that manufacturer). An example of a manufacturer specific operation definition is shown i table D.3.

D.2 Manufacturer specific extensions to standardised Supplementary services

An example of the use of the element of type **Extension** (defined in 9.2) in an ECMA standardise Supplementary services definition is given in table D.4, for a Supplementary service Standard which i published as Standard ECMA-999. In the operation definitions for ECMA standardised Supplementar services, the following constructs are used:

- where the standardised parameter (argument of **InvokePDU**, result of **ReturnResultPDU**) is a singl value (e.g. INTEGER), the Standard can instead specify a SEQUENCE containing a CHOICE of a element of type **Extension** or a SEQUENCE of elements of type **Extension.** Thus, the parameter woul then become:

```
Parameter ::= CHOICE { INTEGER,
                       SEQUENCE { INTEGER,
                                  CHOICE {
                                  [1] IMPLICIT Extension,
                                  [2] IMPLICIT SEQUENCE OF Extension }
                       OPTIONAL }
```

- where the parameter is a SEQUENCE type, this would be replaced by a SEQUENCE containing CHOICE of an element of type **Extension** or a SEQUENCE of elements of type **Extension.** Thus, th parameter would then become:

```
Parameter ::= SEQUENCE { List-of-Standard-parameter-types,
                         CHOICE {
                         [1] IMPLICIT Extension,
                         [2] IMPLICIT SEQUENCE OF Extension }
                         OPTIONAL }
```

- where there is no defined parameter, a parameter should be added as shown below:

```
Parameter ::= CHOICE { NULL,
                       [1] IMPLICIT Extension,
                       [2] IMPLICIT SEQUENCE OF Extension }
```

NOTE D.1

The use of implicit tagging within the CHOICE construct containing elements of type ***Extension*** *should be used consistent with the context specific tags used in the remainder of the SEQUENCE in which it is contained.*

In this way, manufacturer specific additions to standardised Supplementary services may be included in a generic and backwards compatible manner. The manufacturer object identifier (shown in table D.2 above) should be encoded in the same manner as described in 9.1.

The use of a SEQUENCE of elements of type Extension allows the coexistence of a number of different extensions to the standardised Supplementary service. It also allows for future versions of the standardised service to be backwards compatible with, and to coexist with, manufacturer-specific additions to the original Supplementary service.

Table D.3 - Example of manufacturer specific operation

```
Hypothetical-service-operation
      { iso identified-organization icd-ecma member-company
            hypothetical-manufacturer hypothetical-service-offering }

DEFINITIONS        ::=
BEGIN

IMPORTS      OPERATION FROM Remote-Operation-Notation
                  { joint-iso-ccitt( 2) remote-operations( 4) notation( 0) };

hypotheticalService      OPERATION
                  ARGUMENT     HypotheticalArgument
                  RESULTHypotheticalResult
                  ::= { iso( 1) identified-organization( 3) icd-ecma( 0012)
                              member-company( 2) hypothetical-manufacturer(
1999)
                                 hypothetical-manufacturer-service( 1) }

HypotheticalArgument      ::=    INTEGER
                                 { hypotheticalParameter1( 0),
                                   hypotheticalParameter2( 1)
                                 }

HypotheticalResult ::=    INTEGER
                                 { hypotheticalResult1( 0),
                                   hypotheticalResult2( 1)
                                 }

END                  -- of hypothetical-manufacturer-service-operation
```

Table D.4 - Example definition of ECMA operation with elements of type Extension

```
Hypothetical-service-operation
      { iso( 1) identified-organization( 3) icd-ecma( 0012) standard( 0)
            hypothetical-Standard(999) }

DEFINITIONS        ::=
BEGIN

IMPORTS      OPERATION FROM Remote-Operation-Notation
                  { joint-iso-ccitt( 2) remote-operations( 4) notation( 0) }
             Extension FROM ECMA-manufacturer-specific-service-extension-
definition
                  { iso( 1) identified-organization( 3) icd-ecma( 0012)
standard                              qsig-generic-procedures( yyy) msi-
definition( 5)};


hypotheticalService      OPERATION
                  ARGUMENT CHOICE
                           { NormalIntegerArgument,
                             SEQUENCE
                             { NormalIntegerArgument,
                               extension CHOICE
                                        { [2] IMPLICIT Extension
                                          [3] IMPLICIT SEQUENCE OF Extension
                                        } OPTIONAL
                             }
                           }

                  RESULT SEQUENCE
                           { ListOfNormalResultSequenceElements,
                             extension CHOICE
                                        { [2] IMPLICIT Extension
                                          [3] IMPLICIT SEQUENCE OF Extension
                                        } OPTIONAL
                           }

                  ::= { iso(1) identified-organization(3) icd-ecma(0012)
                              Standard(0) hypothetical-Standard(999)
                                          hypothetical-operation(1) }

NormallIntegerArgument   ::=    INTEGER
                                { hypotheticalParameter1( 0),
                                  hypotheticalParameter2( 1)
                                }

ListOfNormalResultSequenceElements ::= { normalResultSequenceElement1
                                                  [0] IMPLICIT INTEGER,
                                         normalResultSequenceElement2
                                                  [1] IMPLICIT INTEGER }

END                -- of hypothetical-service-operation
```

Annex E

(informative)

Problem Code Definitions

Table E.1 - Problem Code Definitions (sheet 1 of 2)

General Problem:	
- unrecognisedPDU	signifies that the type of the APDU as evidenced by its Type identifier, is not defined in clause 11.
- mistypedPDU	signifies that the structure of the APDU does not conform to that defined in clause 11.
- badlyStructuredPDU	signifies that the structure of the APDU does not conform to the Standard notation and encoding rules, defined in CCITT Recommendations X.208 and X.209.
Invoke problem:	
- duplicatedInvocation	signifies that the Invoked-identifier parameter violates the assignment rules of CCITT Recommendation X.219.
- unrecognisedOperation	signifies that the type of the operation is not one of those supported.
- mistypedArgument	signifies that the type of the operation argument supplied is not expected.
- resourceLimitation	the performing PTNX is not able to perform the invoked operation due to resource limitation.
- initiatorReleasing	the association initiator is not willing to perform the invoked operation because it is about to attempt to release the application association.
- unrecognisedLinkedId	signifies that there is no operation in progress with an Invoke identifier equal to the specified Linked identifier.
- linkedResponseUnexpected	signifies that the invoked operation referred to by the Linked identifier is not a parent operation.
- unexpectedChildOperation	signifies that the invoked child operation is not one that the invoked parent operation referred to by the Linked identifier allows.
Return result problem:	
- unrecognisedInvocation	signifies that no operation with the specified invoke identifier is in progress
- resultResponseUnexpected	signifies that the invoked operation does not report a result
- mistypedResult	signifies that the type of the Result parameter supplied is not expected.

Table E.1 - Problem Code Definitions (sheet 2 of 2)

Return error problem:	
- unrecognisedInvocation	signifies that no operation with the specified invoke identifier is in progress
- error responseUnexpected	signifies that the invoked operation does not report failure.
- unrecognisedError	signifies that the reported error is not one expected.
- unexpectedError	signifies that the reported error is not one that the invoked operation ma report.
- mistypedParameter	signifies that the type of the error parameter supplied is not one that i expected.

Annex F

(informative)

Bibliography

ISO 6523	(1984) Data Interchange - Structures for the Identification of Organisations
ETS 300196	(ETSI ETS under preparation) - Integrated Services Digital Network (ISDN) Digital Subscriber Signalling one (DSS1) Protocol Generic Procedures for the Support of Supplementary Services
CCITT Rec.Q.932	(1988) Generic Procedures for the Control of ISDN Supplementary Services
CCITT Rec. X.217	(1988) Association control service definition for Open Systems Interconnection for CCITT Applications
CCITT Rec. X.227	(1988) Association control protocol specification for Open Systems Interconnection for CCITT Applications

Annex G

(informative)

ASN.1 Definition of PartyNumber

Table G.1 is an extract from ETS 300196 which describes the contents and structure used for PartyNumber as used in the NFE.

Table G.1 - Encoding of PartyNumber (sheet 1 of 3)

```
Addressing-Data-Elements
            {ccitt( 0) identified-organization etsi( 0) 196
                  addressing-data-elements( 6)}

DEFINITIONS EXPLICIT TAGS ::=

BEGIN

EXPORTS                    PresentedAddressScreened,
                           PresentedAddressUnscreened,
                           PresentedNumberScreened,
                           PresentedNumberUnscreened,
                           Address, PartyNumber, PartySubaddress,
                           ScreeningIndicator, PresentationAllowedIndicator;

PresentedAddressScreened ::=    CHOICE {
                           presentationAllowedAddress [0] IMPLICIT AddressScreened,
                           presentationRestricted [1] IMPLICIT NULL,
                           numberNotAvailableDueToInterworking [2] IMPLICIT NULL,
                           presentationRestrictedAddress [3] IMPLICIT
                           AddressScreened}

PresentedAddressUnscreened      ::=    CHOICE {
                           presentationAllowedAddress [0] IMPLICIT
                           AddressUnscreened,
                           presentationRestricted [1] IMPLICIT NULL,
                           numberNotAvailableDueToInterworking [2] IMPLICIT NULL,
                           presentationRestrictedAddress [3] IMPLICIT
                           AddressScreened}

PresentedNumberScreened  ::=    CHOICE {
                           presentationAllowedAddress [0] IMPLICIT NumberScreened,
                           presentationRestricted [1] IMPLICIT NULL,
                           numberNotAvailableDueToInterworking [2] IMPLICIT NULL,
                           presentationRestrictedAddress [3] IMPLICIT
                           NumberScreened}

PresentedNumberUnscreened       ::=    CHOICE {
                           presentationAllowedAddress [0] IMPLICIT PartyNumber,
                           presentationRestricted [1] IMPLICIT NULL,
                           numberNotAvailableDueToInterworking [2] IMPLICIT NULL,
                           presentationRestrictedAddress [3] IMPLICIT PartyNumber}

AddressScreened          ::=    SEQUENCE {
                                PartyNumber,
                                ScreeningIndicator,
                                PartySubaddress OPTIONAL}


NumberScreened           ::=    SEQUENCE {
                                      PartyNumber,
                                      ScreeningIndicator}
```

Table G.1 - Encoding of PartyNumber (sheet 2 of 3)

```
Address                         ::=    SEQUENCE {
                                PartyNumber,
                                PartySubaddress OPTIONAL)

PartyNumber                     ::=    CHOICE {
                                unknownPartyNumber [0] IMPLICIT NumberDigits,
                                -- the numbering plan is the default numbering
                                -- plan of the network. It is recommanded that
                                -- this value is used.
                                publicPartyNumber [1] IMPLICIT PublicPartyNumber,
                                -- the numbering plan is according to
                                -- Recommendation E.163 and E.164.
                                dataPartyNumber [3] IMPLICIT NumberDigits,
                                -- not used, value reserved.
                                telexPartyNumber [4] IMPLICIT NumberDigits,
                                -- not used, value reserved.
                                privateNumber [5] IMPLICIT PrivateNumber,
                                nationalStandardPartyNumber [8] IMPLICIT
NumberDigits}
                                -- not used, value reserved.

PublicPartyNumber               ::=    SEQUENCE {
                                publicTypeOfNumber PublicTypeOfNumber,
                                publicNumberDigits NumberDigits}

PrivatePartyNumber              ::=    SEQUENCE {
                                privateTypeOfNumber      PrivateTypeOfNumber,
                                privateNumberDigits      NumberDigits}

NumberDigits                    ::=    NumericString (SIZE(1..20))

PublicTypeOfNumber              ::=    ENUMERATED {
                                unknown (0),
                                -- if used number digits carry prefix indicating
                                -- type of number according to national.
                                -- recommendations
                                internationalNumber (1),
                                nationalNumber (2),
                                networkSpecificNumber (3),
                                -- not used, value reserved
                                subscriberNumber (4),
                                abbreviatedNumber (6)}
                                -- valid only for called party number at the
                                -- outgoing access, network substitutes appropriate
                                -- number.

PrivateTypeOfNumber                   ::=    ENUMERATED {
                                unknown (0),
                                level2RegionalNumber (1),
                                level1RegionaNumber (2),
```

Table G.1 - Encoding of PartyNumber (sheet 3 of 3)

```
                                pTNSpecificNumber (3),
                                localNumber (4),
                                level3RegionalNumber (5),
                                abbreviatedNumber (6)}

PartySubaddress                 ::=    CHOICE {
                                UserSpecifiedSubaddress,
                                -- not recommended.
                                NSAPSubaddress}
                                -- according to Recommendation X.213.

UserSpecifiedSubaddress         ::=    SEQUENCE {
                                SubaddressInformation,
                                oddCountIndicator BOOLEAN OPTIONAL}
                                -- used when the coding of subaddress is BCD

NSAPSubaddress                  ::=    OCTET STRING (SIZE(1..20))
                                -- specified according to X.213. Some networks may
                                -- limit the subaddress value to some other length
                                -- e.g. 4 octets

SubaddressInformation           ::=    OCTET STRING (SIZE(1..20))
                                -- coded according to user requirements. Some
                                -- networks may limit the subaddress value to some
                                -- other length e.g. 4 octets

ScreeningIndicator              ::=    ENUMERATED {
                                userProvidedNotScreened (0),
                                -- number was provided by a remote user terminal
                                -- equipment, and has been screened by a network
                                -- that is not the local public or the local
                                -- private network.
                                userProvidedVerifiedAndPassed (1),
                                -- number was provided by a remote user terminal
                                -- equipment (or by a remote private network), and
                                -- has been screened by the local public or the
                                -- local private network.
                                userProvidedVerifiedAndFailed (2),
                                -- not used, value reserved.
                                networkProvided (3)}
                                -- number was provided by local public or local
                                -- private network.

PresentationAllowedIndicator    ::=    BOOLEAN

END -- of Addressing-Data-Elements
```

Annex H

(informative)

Object Identifiers Used in ECMA-165

This Annex lists the module identifiers used in ECMA-165 and which data types are exported from each. It also lists any other object identifiers defined in the text. All the object identifiers in this standard are defined using the ECMA object identifier tree. For module names, this means that each object identifier value is assigned in the tree:

```
qsigGfObjectIdTree :: = iso( 1) identified-organization( 3)
      icd-ecma( 0012) standard( 0) qsig-generic-procedures( yyy)
```

NOTE H.1:

'yyy' is the number of the ECMA standard number for QSIG-GF.

The values for module names have been assigned in ascending order throughout the standard as values in the tree above. That is:

```
qsigGfModuleName ::= { qsigGfObjectIdTree moduleNumber }
```

Table H.1 lists the module number values and the data types and Macros which are exported from these modules.

Table H.1 - ASN.1 Module Object identifiers used in ECMA-165

Module number	Name of Module	Data types/values/macros exported
0	ECMA-manufacturer-specific-service-extension-definition	Extension, EXTENSION
1	Component-part-definition	ComponentPart
2	Network-Facility-Extension	NetworkFacilityExtension
3	Interpretation-Apdu	InterpretationApdu
4	Dialogue-Service-Pdus	DseAPDU
5	Remote-Operations-Apdus	RoseAPDU
6	QSIG-generic-parameters-definition	QSIGInformationElement qsigIeNotification
7	Notification-Data-Structure	NOTIFICATION NotificationDataStructure

Annex I

(normative)

Protocol Implementation Conformance Statement (PICS) For ECMA-165

I.1 Introduction

The supplier of a protocol implementation which is claimed to conform to Standard ECMA-165 shall complete the following Protocol Implementation Conformance Statement (PICS) proforma.

A completed PICS proforma is the PICS for the implementation in question. The PICS is a statement of which capabilities and options of the protocol have been implemented. The PICS can have a number of uses, including use:

- by the protocol implementor, as a check list to reduce the risk of failure to conform to the standard through oversight;
- by the supplier and acquirer - or potential acquirer - of the implementation, as a detailed indication of the capabilities of the implementation, stated relative to the common basis for understanding provided by the standards PICS proforma;
- by the user or potential user of the implementation, as a basis for initially checking the possibility of interworking with another implementation

  NOTE I.1

  While interworking can never be guaranteed, failure to interwork can often be predicted from incompatible PICS's.
- by a protocol tester, as the basis for selecting appropriate tests against which to assess the claim for conformance of the implementation.

I.2 Instructions for completing the PICS proforma

I.2.1 General structure of the PICS proforma

The PICS proforma is a fixed format questionnaire divided into sub-clauses each containing a group of individual items. Each item is identified by an item number, the name of the item (question to be answered), and the reference(s) to the clause(s) that specifies (specify) the item in the main body of this standard.

The "Status" column indicates whether an item is applicable and if so whether support is mandatory or optional. The following terms are used:

m	mandatory (the capability is required for conformance to the protocol);
o	optional (the capability is not required for conformance to the protocol, but if the capability is implemented it is required to conform to the protocol specifications);
o.<n>	optional, but support of at least one of the group of options labelled by the same numeral <n> is required;
x	prohibited;

c.<cond>	conditional requirement, depending on support for the item or items listed in condition <cond>;
<item>:m	simple conditional requirement, the capability being mandatory if item number <item> is supported,otherwise not applicable;
<item>:o	simple conditional requirement, the capability being optional if item number <item> is supported, otherwise not applicable.

Answers to the questionnaire items are to be provided either in the "Support" column, by simply marking an answer to indicate a restricted choice (Yes or No), or in the "Not Applicable" column (N/A).

I.2.2 Additional information

Items of Additional Information allow a supplier to provide further information intended to assist ... interpretation of the PICS. It is not intended or expected that a large quantity will be supplied, and a PICS can be considered complete without any such information. Examples might be an outline of the ways in which a (single) implementation can be set up to operate in a variety of environments and configurations.

References to items of Additional Information may be entered next to any answer in the questionnaire, and may be included in items of Exception information.

I.2.3 Exception information

It may occasionally happen that a supplier will wish to answer an item with mandatory or prohibited status (after any conditions have been applied) in a way that conflicts with the indicated requirement. No pre-printed answer will be found in the Support column for this: instead, the supplier is required to write into the support column an x.<i> reference to an item of Exception Information, and to provide the appropriate rationale in the Exception item itself.

An implementation for which an Exception item is required in this way does not conform to ECMA-165.

NOTE I.2

A possible reason for the situation described above is that a defect in the Standard has been reported, a correction for which is expected to change the requirement not met by the implementation.

I.3 PICS Proforma

I.3.1 Implementation Identification

Supplier	
Contact point for queries about the PICS	
Implementation Name(s) and Version(s)	

NOTE I.3

Only the first three items are required for all implementations; other information may be completed as appropriate in meeting the requirement for full identification.

NOTE I.4

The terms Name and Version should be interpreted appropriately to correspond with a suppliers terminology (e.g. Type, Series, Model).

Other information necessary for full identification - e.g., name(s) and version(s) for machines and/or operating systems; System name(s)	

I.3.2 Protocol Summary, ECMA-165

Protocol version	1.0
Addenda Implemented (if applicable)	
Amendments Implemented	
Have any exception items been required (see I.3.3) ?	No [ ] Yes [ ] (The answer Yes means that the implementation does not conform to ECMA-QSIG-GF)

Date of Statement	

I.3.3 Call Related Protocol Control and GFT-Control Requirements

Item	Question/feature	References	Status	N/A	Support
A1	Can the implementation act as a Source PTNX for APDUs?	7.1.1.1	o		Yes[ ] No [ ]
A3	Sending the Facility information element	7.1.1.1	m		Yes [ ]
A4	Receiving the Facility information element	7.1.1.2	m		Yes [ ]
A5	Actions at a Source PTNX	7.1.2.1	**A1**:m	[ ]	Yes [ ]
A6	Actions at a receiving PTNX	7.1.2.2	m		Yes [ ]
A7	Can the PTNX act as an End PTNX?	7.1.2.2.1	o		Yes[ ] No [ ]
A8	End PTNX actions	7.1.2.2.1	**A7**:m	[ ]	Yes [ ]
A9	Actions at a Destination PTNX	7.1.2.3	m		Yes [ ]
A10	Transit PTNX actions	7.1.2.2.2	m		Yes [ ]
A11	Can the implementation generate notification information ?	7.4	o		Yes[ ] No [ ]
A12	Sending notification information	7.4.2.1	**A11**:m	[ ]	Yes [ ]
A13	Receiving notification information	7.4.2.2	m		Yes [ ]
A14	Actions at a PTNX which generates notifications	7.4.3.1	**A11**:m	[ ]	Yes [ ]
A15	Actions at a Transit PTNX	7.4.3.2	m		Yes [ ]
A16	Actions at a receiving End PTNX	7.4.3.3	m		Yes [ ]

I.3.4 Connectionless APDU transport mechanism

Item	Question/feature	References	Status	N/A	Support
B1	Does the PTNX support Connectionless APDU transport?	7.2	o		Yes[ ] No [ ]
B2	Requirements for sending a Connectionless message	7.2.1.1	**B1**:m	[ ]	m: Yes [ ]
B3	Requirements for Receiving a Connectionless message	7.2.1.2	**B1**:m	[ ]	m: Yes [ ]
B4	Actions at a receiving PTNX	7.2.2.2	**B1**: m	[ ]	m: Yes [ ]
B6	Actions at a Destination PTNX	7.2.2.3	**B1**: o	[ ]	o: Yes [ ] No [ ]
B7	Actions at a Source PTNX	7.2.2.1	**B1**: o	[ ]	o: Yes [ ] No [ ]

I.3.5 Connection oriented APDU transport mechanism

Item	Question/feature	References	Status	N/A	Support
C1	Does the PTNX support connection-oriented APDU transport?	7.3	o		Yes[ ] No [ ]
C2	Connection oriented transport mechanism - Protocol Control requirements	7.3.1	C1:m	[ ]	m: Yes [ ]
C3	Actions at an Originating PTNX	7.3.3.1	C1: o	[ ]	o: Yes [ ] No [ ]
C4	Actions at a Transit PTNX	7.3.3.2	C1: m	[ ]	m: Yes [ ]
C5	Actions at a Terminating PTNX	7.3.3.3	C1: o	[ ]	o: Yes [ ] No [ ]

I.3.6 Coordination Function requirements

Item	Question/feature	References	Status	N/A	Support
D1	Inclusion of an Interpretation APDU at a Source PTNX	8.1.1	o		Yes [ ] No [ ]
D2	Handling of APDUs at a destination PTNX	8.1.2	m		Yes [ ]

I.3.7 ROSE requirements

Item	Question/feature	References	Status	N/A	Support
E1	ROSE requirements	8.2	m		Yes [ ]

I.3.8 DSE requirements

Item	Question/feature	References	Status	N/A	Support
F1	Does implementation support the DSE protocol?	8.3	o		Yes [ ] No [ ]
F2	Actions at the PTNX which initiates the dialogue	8.3.1	F1:o.1	[ ]	o: Yes [ ] No [ ]
F3	Actions at the PTNX which terminates the dialogue	8.3.2	F1:o.1	[ ]	o: Yes [ ] No [ ]
F4	Actions for dialogue continuation	8.2.3	F1: m	[ ]	m: Yes [ ]
F5	T_Originating_Dialogue	8.3.4	F1: m	[ ]	m: Yes [ ] value [s]
F6	Error procedures relating to dialogue control	8.3.5	F1: m	[ ]	m: Yes [ ]

I.3.9 Manufacturer specific information

Item	Question/feature	References	Status	N/A	Support
H1	Manufacturer specific operations	9.1	o		Yes [ ] No[ ]
H2	Manufacturer specific additions to standardised operations	9.2	o		Yes [ ] No[ ]
H3	Manufacturer specific notifications	9.3	o		Yes [ ] No[ ]

I.3.10 Encoding

Item	Question/feature	References	Status	N/A	Support
I1	General message format and information element coding	11	m		Yes [ ]
I2	Message type	11.1	m		Yes [ ]
I3	Dummy Call reference	11.2	**B1**:m	[ ]	Yes [ ]
I4	Bearer Capability	11.3.1	**C1**:m	[ ]	Yes [ ]
I5	Channel identification	11.3.2	**C1**:m	[ ]	Yes [ ]
I6	Facility information element structure	11.3.3	m		Yes [ ]
I7	Network-Facility-Extension encoding	11.3.3.1	m		Yes [ ]
I8	Interpretation APDU encoding	11.3.3.2	m		Yes [ ]
I9	DSE APDU encoding	11.3.3.3	**F1**:m	[ ]	m :Yes [ ]
I10	ROSE APDU encoding	11.3.3.4	m		Yes [ ]
I11	Notification indicator encoding	11.3.4	m		Yes [ ]

I.3.11 Implemented parameters in ECMA-165 messages

NOTE

In the following clauses, the headings 'orig' and 'Rx' should be interpreted as follows:

'orig': *the capability to originate the element specified - i.e. create the element and send it on an ECMA QSIG link; **not** relay the element having received it from a Preceding PTNX.*

'Rx': *the capability to correctly receive and process the specified element as a valid element from a Preceding PTNX; **including** relay of the element to a Subsequent PTNX if acting as a Transit PTNX for the related call or connection.*

I.3.11.1 ALERTING message

Item	Question/feature	References	Status	N/A	Support
J1	Facility information element - Orig	10.1, 11.3.3	o		o: Yes [ ] No [ ]
J2	Facility information element - Rx	10.1, 11.3.3	m		m: Yes [ ]
J3	Notification indicator information element - Orig	10.1, 11.3.4	o		o: Yes [ ] No [ ]
J4	Notification indicator information element - Rx	10.1, 11.3.4	m		m: Yes [ ]

I.3.11.2 CONNECT message

Item	Question/feature	References	Status	N/A	Support
K1	Facility information element - Orig	10.2, 11.3.3	o		o: Yes [ ] No [ ]
K2	Facility information element - Rx	10.2, 11.3.3	m		m:Yes [ ]
K3	Notification indicator information element - Orig	10.2, 11.3.4	o		o: Yes [ ] No [ ]
K4	Notification indicator information element - Rx	10.2, 11.3.4	m		m: Yes [ ]

I.3.11.3 SETUP message

Item	Question/feature	References	Status	N/A	Support
L1	Facility information element - Orig	10.3, 11.3.3	o		o: Yes [ ] No [ ]
L2	Facility information element - Rx	10.3, 11.3.3	m		m: Yes [ ]
L3	Notification indicator information element - Orig	10.3, 11.3.4	o		o: Yes [ ] No [ ]
L4	Notification indicator information element - Rx	10.3, 11.3.4	m		m: Yes [ ]

I.3.11.4 DISCONNECT message

Item	Question/feature	References	Status	N/A	Support
M1	Facility information element - Orig	10.4, 11.3.3	o		o: Yes [ ] No [ ]
M2	Facility information element - Rx	10.4, 11.3.3	m		m: Yes [ ]
M3	Notification indicator information element - Orig	10.4, 11.3.4	o		o: Yes [ ] No [ ]
M4	Notification indicator information element - Rx	10.4, 11.3.4	m		m: Yes [ ]

I.3.11.5 RELEASE message

Item	Question/feature	References	Status	N/A	Support
N1	Facility information element - Orig	10.5, 11.3.3	o		o: Yes [ ] No [ ]
N2	Facility information element - Rx	10.5, 11.3.3	m		m: Yes [ ]

I.3.11.6 RELEASE COMPLETE message

Item	Question/feature	References	Status	N/A	Support
O1	Facility information element - Orig	10.6, 11.3.3	o		o: Yes [ ] No [ ]
O2	Facility information element - Rx	10.6, 11.3.3	m		m: Yes [ ]

I.3.11.7 FACILITY message

Item	Question/feature	References	Status	N/A	Support
P1	Protocol discriminator- Orig	10.7, (12.2 ECMA-143)	m		m: Yes [ ]
P2	Protocol discriminator- Rx	10.7, (12.2 ECMA-143)	m		m: Yes [ ]
P3	Call reference-Orig	10.7, 11.2	m		m: Yes [ ]
P4	Call reference-Rx	10.7, 11.2	m		m: Yes [ ]
P5	Message type-Orig	10.7, 11.1	m		m: Yes [ ]
P6	Message type-Rx	10.7, 11.1	m		m: Yes [ ]
P7	Calling party number - Orig	10.7, 12.5 of ECMA-143	**B1**: m	[ ]	m: Yes [ ]
P8	Calling party number - Rx	10.7, 12.5 of ECMA-143	**B1**: m	[ ]	m: Yes [ ]
P9	Called party number - Orig	10.7, 12.5 of ECMA-143	**B1**: m	[ ]	m: Yes [ ]
P10	Called party number - Rx	10.7, 12.5 of ECMA-143	**B1**: m	[ ]	m: Yes [ ]
P11	Facility information element - Orig	10.7, 11.3.3	m		m: Yes [ ]
P12	Facility information element - Rx	10.7, 11.3.3	m		m: Yes [ ]
P13	Notification indicator information element - Orig	10.7, 11.3.4	o		o: Yes [ ] No [ ]
P14	Notification indicator information element - Rx	10.7, 11.3.4	m		m: Yes [ ]

I.3.11.8 NOTIFY message

Item	Question/feature	References	Status	N/A	Support
Q1	Protocol discriminator - Orig	10.8, (12.2 ECMA-143)	m		m: Yes [ ]
Q2	Protocol discriminator- Rx	10.8, (12.2 ECMA-143)	m		m: Yes [ ]
Q3	Call reference - Orig	10.8, 11.2	m		m: Yes [ ]
Q4	Call reference - Rx	10.8, 11.2	m		m: Yes [ ]
Q5	Message type - Orig	10.8, 11.1	m		m: Yes [ ]
Q6	Message type - Rx	10.8, 11.1	m		m: Yes [ ]
Q7	Notification Indicator - Orig	10.8, 11.3.4	m		m: Yes [ ]
Q8	Notification Indicator - Rx	10.8, 11.3.4	m		m: Yes [ ]

I.3.11.9 PROGRESS message

Item	Question/feature	References	Status	N/A	Support
R1	Facility information element - Orig	10.9, 11.3.3	o		o: Yes [ ] No [ ]
R2	Facility information element - Rx	10.9, 11.3.3	m		m:Yes [ ]
R3	Notification indicator information element - Orig	10.9, 11.3.4	o		o: Yes [ ] No [ ]
R4	Notification indicator information element - Rx	10.9, 11.3.4	m		m: Yes [ ]