INTERNATIONAL TELECOMMUNICATION UNION

ITU-T Q.771

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

(03/93)

SPECIFICATIONS OF SIGNALLING SYSTEM No. 7

SIGNALLING SYSTEM No. 7 – FUNCTIONAL DESCRIPTION OF TRANSACTION CAPABILITIES

ITU-T Recommendation Q.771

(Previously "CCITT Recommendation")

FOREWORD

The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of the International Telecommunication Union. The ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Conference (WTSC), which meets every four years, established the topics for study by the ITU-T Study Groups which, in their turn, produce Recommendations on these topics.

ITU-T Recommendation Q.771 was revised by the ITU-T Study Group XI (1988-1993) and was approved by the WTSC (Helsinki, March 1-12, 1993).

NOTES

1 As a consequence of a reform process within the International Telecommunication Union (ITU), the CCITT ceased to exist as of 28 February 1993. In its place, the ITU Telecommunication Standardization Sector (ITU-T) was created as of 1 March 1993. Similarly, in this reform process, the CCIR and the IFRB have been replaced by the Radiocommunication Sector.

In order not to delay publication of this Recommendation, no change has been made in the text to references containing the acronyms "CCITT, CCIR or IFRB" or their associated entities such as Plenary Assembly, Secretariat, etc. Future editions of this Recommendation will contain the proper terminology related to the new ITU structure.

2 In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

CONTENTS

SIGNALLING SYSTEM No. 7 – FUNCTIONAL DESCRIPTION OF TRANSACTION CAPABILITIES

(Melbourne, 1988; modified at Helsinki, 1993)

1 Introduction

1.1 General

Transaction Capabilities (TC) provide functions and protocols to a large variety of applications distributed over switches and specialized centres in telecommunication networks (e.g. databases). The support of TC by terminal equipments is for further study.

The term "Transaction Capabilities" refers to a set of communication capabilities that provide an interface between applications and a network layer service.

To date, only Signalling System No. 7 MTP plus SCCP have been considered as network layer service providers. However, any standard OSI Network Layer might be used in place of the MTP plus SCCP, provided that the requirements of the applications supported by TC (e.g. service and performance requirements) can be met. This area requires further study.

Figure 1 shows the general structure of TC.

For historical reasons, the term TC and TCAP are used interchangeably.

The remodelling of TC using OSI layering is for further study.

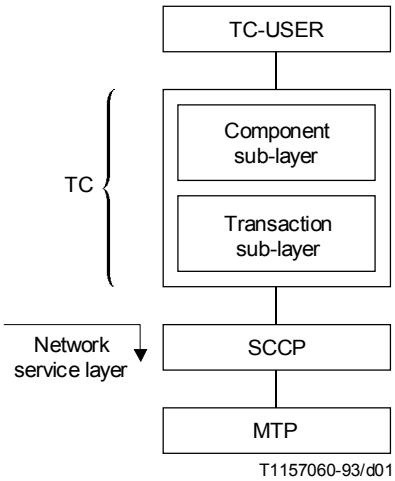


FIGURE 1/Q.771

Structure of TC in the Signalling System No. 7

1.2 Contents of the Q.771/Q.775-Series Recommendations

Recommendation Q.771 contains a general description of the service provided by the Transaction Capabilities, and the service expected from the SCCP.

Recommendation Q.772 defines the Transaction Capabilities Information Elements, and their functions.

Recommendation Q.773 defines the formats and encoding used for the Transaction Capabilities Messages, the protocol data units (PDUs) have been specified using the ASN.1 formal notation (see Recommendation X.208).

Recommendation Q.774 specifies the Transaction Capabilities procedures. Annex A to this Recommendation contains the SDL diagrams for TC.

Recommendation Q.775 contains guidelines and examples on how to define applications and their use of TC.

The present Recommendation contains both introductory information (clauses 1 and 2), and a detailed description (clause 3), using primitives, of the services provided by TC. The reader interested in the first aspect only may read the first two clauses; clause 3 contains more detailed information.

1.3 Objectives

1.3.1 Definition of Transaction Capabilities

The overall objective of Transaction Capabilities is to provide the means for the transfer of information between nodes, and to provide generic services to applications, while being independent of any of these.

1.3.2 Scope of Transaction Capabilities

Transaction Capabilities in a Signalling System No. 7 network should be considered for use between:

1) exchanges;

2) an exchange and a network service centre (e.g. data base, specialized facility unit, OA&M Centre);

3) network service centres.

The following applications have been recognized as TC-users:

– mobile service application (e.g. location of roamers);

– registration, activation and invocation of supplementary services involving specialized facility units (e.g. freephone service, credit card service);

– non circuit control-related exchange of signalling information (e.g. closed user group, look-ahead procedure);

– operation and maintenance applications (e.g. query/response).

This list is not exhaustive.

2 Overview

2.1 Terminology

The terms used throughout the Q.77x-Series Recommendations are defined in the Signalling System No. 7 glossary.

2.2 Structure of TC

2.2.1 Architectural concepts

From an end-user point of view, Transaction Capabilities lies above the network layer of the OSI model. Provision of network layer services to end-users requires communication between TC-users at various network nodes; these intra-network communications may in turn be modelled using the 7-layer reference model of OSI.

TC is structured in two sub-layers:

- the component sub-layer, deals with components, which are the application protocol data units (APDU) that convey remote operations and their responses and, as an option, the dialogue portion protocol for conveying information related to application context or user information;
- the transaction sub-layer, which deals with the exchange of messages containing components and optionally, a dialogue portion, between two TC-users.

This is illustrated by Figure 1.

2.2.2 Addressing issues

When TC is used in a Signalling System No. 7 network service, the addressing options supported by the SCCP are used.

When other network layer service providers are used, the addressing options supported by those providers will be used; further study on this area is required.

2.2.3 Management aspects

SCCP Management primitives used to inform SCCP users of availability and non-availability of SCCP (local or remote), will be passed transparently by TC to the TC-user.

2.2.4 Alignment of TCAP with Recommendations X.219 and X.229 (ROSE)

The Component sub-layer of TCAP is in partial alignment with the capabilities of the Remote Operation Service Element (ROSE). The current status of TCAP and ROSE alignment is on the basis of protocol alignment, namely the X.229 protocol is contained within the TCAP component protocol. In addition, the component sub-layer includes some extensions to ROSE.

The X.219 Remote Operation Service provides five classes of operations. Class 1 is synchronous, reporting both success and failure. Classes 2 to 5 are asynchronous and correspond to the TCAP operation classes 1 to 4. TCAP has not adopted ROSE class 1 (synchronous), because the full-duplex mode of operation is used in TCAP. However, TC-users may use the TCAP operation class 1 in a synchronous mode if appropriate. Further details are given in Recommendation Q.775.

2.2.5 Alignment of TCAP with Recommendations X.217 and X.227 (ACSE)

The dialogue control facilities of TCAP is in partial alignment with the capabilities of the Association Control Service Element (ACSE) defined in Recommendations X.217 and X.227. The current alignment is only one of protocol alignment: the abstract syntax for the dialogue control APDUs are a subset of the OSI ACSE abstract syntax.

2.3 TC based on a connectionless network service

2.3.1 Service provided by the Component sub-layer

2.3.1.1 Component

A component is the means by which TC conveys a request to perform an **operation**, or a **reply**.

An operation is an action to be performed by the remote end. It may have associated parameters. An invocation of an operation is identified by an Invoke ID; this allows several invocations of the same or different operations to be active simultaneously.

Only one reply may be sent to an operation. A reply carries an indication of success or failure in carrying out the operation.

Components are passed individually between a TC-user and the Component sub-layer. The originating TC-user may send several components to the Component sub-layer before these are transmitted (in a single message) to the remote end. Whenever several components are received in a single message, each one is delivered individually to the destination TC-user.

Components in a message are delivered to the remote TC-user in the same order as they are provided at the originating interface. The importance of the order is by prior agreement between the TC-users involved.

2.3.1.2 Dialogue

Successive components exchanged between two TC-users in order to perform an application constitute a dialogue. The Component sub-layer provides dialogue facilities, allowing several dialogues to run concurrently between two given TC-users.

Dialogue handling also allows, as an option, for the transfer and negotiation of the application context, and the transparent transfer of user information (i.e. data which are not components) between two TC-users.

Two kinds of facilities are provided: unstructured and structured.

2.3.1.2.1 Unstructured dialogue

TC-users can send components that do not expect replies without forming an explicit association between themselves. This is referred to as the unstructured dialogue case. The implicit association always exists between the communicating TC-users. Such an unstructured dialogue is supported in the protocol by the Unidirectional message. When one TC-user sends a Unidirectional message to its peers, this indicates use of the unstructured dialogue facility.

When a TC-user is a receiver of a unidirectional message, if a protocol error is to be reported, it is also returned in a Unidirectional message.

2.3.1.2.2 Structured dialogue

Alternatively, TC-users indicate the beginning, or the formation of a dialogue, the continuation, and the end of a dialogue; this is referred to as a structured dialogue. Using a structured dialogue allows two TC-users to run several dialogues concurrently with each being identified by a particular dialogue ID. Each dialogue ID has a separate Invoke ID name space, thus allowing duplication of Invoke IDs in different dialogues. Sequenced delivery of messages may be provided by means of application protocols outside the scope of TCAP, or by use of the appropriate SCCP class of service.

When using the structured dialogue service, the TC-user has to indicate one of the following four possibilities when sending a component to its peer entity:

i) a dialogue begins;

ii) a dialogue confirmation;

iii) a dialogue continues: full-duplex exchange of components is possible;

iv) a dialogue ends: the sending side will not send more components, nor will it accept any more components from the remote end.

As an option, in Phases i) and ii), an exchange of application context information and user information is possible, when this option is chosen, user information can also be sent during Phases iii) and iv).

2.3.1.3 Component correlation

The Component sub-layer provides the following facilities:

a) *Association of operations and replies*

The value of the Invoke ID, which identifies an operation invocation without ambiguity, is returned in a response to that operation invocation. A TC-user may have a number of operations active at a given time; the maximum number is dependent on the unique Invoke IDs available to a TC-User at any time.

If the "response" to this operation invocation is another operation invocation from the responding end, the original Invoke ID is returned as a Linked ID indicating that this responding operation invocation is "linked" to the original operation.

Four classes of operations are considered:

– *Class 1* – Both success and failure are reported.

– *Class 2* – Only failure is reported.

– *Class 3* – Only success is reported.

– *Class 4* – Neither success, nor failure is reported.

Where necessary, the TC-user provides segmentation of a successful response. In addition, any number of linked operations may be sent prior to the reply.

The reply may be:

– a return result indicating success;

– a return error indicating operation failure;

– a reject indicating inability to perform the operation.

The application protocol designer decides what constitutes success or failure in performing the operation.

Any component, except a reject component, may be rejected. Rejection of a result causes termination of the corresponding operation; rejection of a linked operation does not affect the linked-to operation. Rejection of a segment of a result (i.e. rejection of a Return Result – Not Last component) implies the rejection of all subsequent segments and also the entire result.

A TC-user may cancel an operation which it has previously invoked. Any subsequently received reply for this invocation will be rejected.

b) *Abnormal situations handling*

The Component sub-layer covers a number of abnormal situations in relation with a component:

– *Component reject* – When the Component sub-layer receives a malformed component, or a component which violates the rules of exchange of operations and replies, it informs the TC-user(s).

– *Operation expiry* – When the Component sub-layer detects that a class 1, 2 or 3 operation has not received a final reply after some amount of time (which is specified by the application and depends on the operation), it releases the corresponding invoke ID and informs the TC-user. Note that this situation is abnormal only in the case of a class 1 operation. Application of this to class 4 operations is a local matter.

– *Invocation cancel* – A TC-user may decide to release a given Invoke ID and decide to ignore any responses using this identifier.

2.3.1.4 Error handling

When the Component sub-layer is informed of a situation which prevents it from providing the service expected by the TC-users, it will notify the TC-user, and may terminate the pending operations.

A TC-user may also decide to abort a dialogue, which puts an end to any pending operation.

2.3.2 Service provided by the Transaction Sub-layer

The Transaction (TR) sub-layer provides the capability for the exchange of components and as an option, a dialogue portion between TR-users. The Transaction sub-layer also provides the capability to send transaction messages between peer TR-layer entities by means of the services provided by the lower layer network services. The only foreseen TR-user for the moment is the Component sub-layer. Two types of service are provided.

2.3.2.1 Unstructured dialogue

There is no explicit initiation, or termination associated with an unstructured dialogue. The only facility provided to the TC-user is the capability to send one, or several components that do not expect replies (invocation of class 4 operations) grouped in a Unidirectional message to the remote TR-user.

At the originating side, the TC-user indicates the components to be sent in a Unidirectional message by means of primitives of the request type containing a unique dialogue ID. When the TC-user issues a TC-UNI request primitive with the same dialogue ID, all the components with the same dialogue ID are sent as user data to the Transaction sub-layer by means of the TR-UNI primitive by the Component sub-layer. At the Transaction sub-layer message level, the Unidirectional message does not contain any transaction ID thereby providing no association between messages of this type. The dialogue ID is used to send a group of components in a UNI message to a particular destination address.

2.3.2.2 Structured dialogue

The structured dialogue facility allows a TC-user to start a dialogue, exchange components within this dialogue, terminate it, or abort it.

Each TR-user identifies a transaction by a separate transaction ID.

The following facilities are provided:

- *Transaction begin* – The beginning of a transaction between two TR-users causes a transaction ID to be allocated to this transaction, and permits sending TR-user information to the destination TR-user. In response to transaction begin, the destination TR-user may continue the transaction, or end it.

- *Transaction continuation* – Allows full-duplex exchange of messages between TR-users inside a transaction.

- *Transaction end* – Release the associated transaction ID, and puts an end to the exchange of messages inside this transaction. Either of the TR-users may decide to end a transaction. There are three ways for the TR-user to terminate a transaction:

  1) *pre-arranged end* – A convention exists between the TR-users: each of them may decide to terminate the transaction without having to inform the peer TR-user, which will make a similar decision on its own;

  2) *basic end* – It informs the peer TR-user, possibly sending TR-user information to it;

  3) *transaction abort* – Causes the abandonment of any message of the transaction for which transmission or delivery is pending, and ends the transaction. The reason for aborting the transaction is indicated to the remote TR-user.

- If, for some reason, no response of any kind is received to transaction begin, the Transaction sub-layer will eventually abort this transaction and inform the TR-user. This is a local option.

- *Transaction abort by TCAP* – Whenever one of a list of abnormal situations is detected, the Transaction sub-layer decides to abort the corresponding transaction and informs the TR-users.

- *Exception reporting* – The Transaction sub-layer may report to TR-users abnormal situations of which it is notified by the underlying layer.

When the TR-user is the Component sub-layer:

a) there is a one-to-one mapping between a dialogue and a transaction;

b) a message may contain zero, one or more components, within the limits of the message size supported by the underlying layer.

3 Service provided by TC based on a connectionless network service

3.1 Component sub-layer

3.1.1 Overview of Component sub-layer primitives

Tables 1 and 2 give an overview of the primitives to/from the TC-users, and contain references to the subclauses of this Recommendation where these primitives are described in detail.

Table 1 shows the primitives relating to dialogue handling. The purpose of these primitives is to request or indicate facilities of the underlying (sub)-layer, in relation with message transmission or dialogue handling. When the Transaction sub-layer is used to support the dialogue, these primitives map onto TR-primitives with the same generic name, as there is a one-to-one relationship between a dialogue and a transaction.

TABLE 1/Q.771

Primitives for dialogue handling

Name	Type	Reference (subclause)
TC-UNI	Request Indication	3.1.2.2.1
TC-BEGIN	Request Indication	3.1.2.2.2.1
TC-CONTINUE	Request Indication	3.1.2.2.2.2 and 3.1.2.2.2.3
TC-END	Request Indication	3.1.2.2.2.4
TC-U-ABORT	Request Indication	3.1.2.2.2.4
TC-P-ABORT	Indication	3.1.4.2
TC-NOTICE	Indication	3.1.2.2.3

- TC-UNI – Requests/indicates an unstructured dialogue.
- TC-BEGIN – Begins a dialogue.
- TC-CONTINUE – Continues a dialogue.
- TC-END – Ends a dialogue.

Each of the previous primitives causes any component(s) previously passed on the interface for the referenced dialogue to be delivered to the remote end (except for a TC-END with prearranged end). If the above primitives contain information related to dialogue handling, a dialogue portion is formatted and sent concatenated with the (sequence of) components(s).

- TC-U-ABORT – Allows a TC-user to terminate a dialogue abruptly, without transmitting any pending components.

– TC-P-ABORT – Informs the TC-user that the dialogue has been terminated by the service provider (i.e. TC Transaction sub-layer) in reaction to a transaction abort by the Transaction sub-layer. Any pending components are not transmitted.

– TC-NOTICE – Informs the TC-user that the Network Service Provider has been unable to provide the requested service.

Table 2 shows the TC-primitives for component handling. The main purpose of these primitives is to handle operations and replies; these primitives do not as such require facilities from the underlying (sub)-layer.

TABLE 2/Q.771

Primitives for component handling

Name	Type	Reference (subclause)
TC-INVOKE	Request Indication	3.1.3.2
TC-RESULT-L	Request Indication	3.1.3.3
TC-RESULT-NL	Request Indication	3.1.3.3
TC-U-ERROR	Request Indication	3.1.3.4
TC-L-CANCEL	Indication	3.1.3.6
TC-U-CANCEL	Request	3.1.3.6
TC-L-REJECT	Indication	3.1.4.1
TC-R-REJECT	Indication	3.1.4.1
TC-U-REJECT	Request Indication	3.1.3.5

– TC-INVOKE – Invocation of an operation, which may be linked to another operation invocation.

– TC-RESULT-L – Only result or last part of the segmented result of a successfully executed operation.

– TC-RESULT-NL – Non-final part of the segmented result of a successfully executed operation.

– TC-U-ERROR – Reply to a previously invoked operation, indicating that the operation execution failed.

– TC-L-CANCEL – Informs the TC-user locally that an operation invocation is terminated due to a timeout condition.

– TC-U-CANCEL – Causes local termination of an operation invocation, as a consequence of a TC-user decision.

– TC-L-REJECT (local reject) – Informs the local TC-user that a Component sub-layer detected invalid component was received.

– TC-R-REJECT (remote reject) – Informs the local TC-user that a component was rejected by the remote Component sub-layer.

– TC-U-REJECT – Rejection of a component by the TC-user, indicating a malformation which prevents the operation from being executed, or the reply from being understood.

In the following subclauses, the various primitives associated with component and dialogue handling are described with their parameters. The following notation is used:

M Indicates a mandatory parameter.

O Indicates provider option.

C Indicates that the parameter is "conditional" (i.e. it will always be present in the indication type primitive if it was present in the corresponding request type primitive).

U Indicates a parameter which can optionally be included by the TC-user.

FS Indicates that further study is required.

A blank Indicates that the parameter is not applicable.

(=) Indicates that the parameter must have the same value in the indication primitive as provided in the corresponding request primitive.

This notation applies throughout this Recommendation.

The parameters included in primitives must contain sufficient information to enable the construction of a valid PDU, e.g. whether an operation/error code is local or global.

3.1.2 Dialogue handling

Dialogue handling provides facilities for the exchange of components within a dialogue.

3.1.2.1 Definition of parameters

This subclause describes the parameters used with the primitives associated with dialogue handling.

- "Abort Reason" – Indicates whether a dialogue is aborted because the received application context name is not supported and no alternative one can be proposed (abort reason = application context not supported) or because of any other user problem (abort reason = user specific).
- Address parameters – Two address parameters are used: the "Destination Address" and the "Originating Address" parameters. These parameters identify respectively the destination and originating TC-user.
- "Application context Name" – The application context is the identifier of the application context proposed by the dialogue initiator or by the dialogue responder. An application context is an explicitly identified set of application-service-elements, related options and any other necessary information for the interworking of application-entities on a dialogue.
- "Components Present" – Indicates whether or not components are present. If components are present, then only syntactically valid and opportune components are delivered to the destination TC-user by TCAP in the same order they were delivered to TCAP by the source TC-user.
- "Dialogue ID" – This parameter also appears in the component handling primitives, and is used to associate components with a dialogue. The same dialogue ID must be used within the same dialogue. In a Unidirectional primitive the same dialogue ID assures that all components with the identical dialogue ID are blocked together in the same Unidirectional message destined for the same destination address. For structured dialogues, the dialogue ID is used to identify all components belonging to the same dialogue from the beginning of the dialogue to its end. The dialogue ID maps onto the transaction IDs exchanged in the messages between a pair of nodes.
- "P-ABORT" – Contains information indicating the cause for which TCAP decides to abort a Dialogue. It takes the following symbolic values; Unrecognised Transaction ID, Unrecognised Message Type, Badly Formatted Transaction Portion, Incorrect Transaction Portion, Resource Limitation, Abnormal dialogue or No Common Dialogue Portion.
- "Quality of Service" – The TC-user indicates the acceptable Quality of Service. The "Quality of Service" parameters for the connectionless SCCP network service at present consists of the following:
  - "Return Option" – Specifies whether the SCCP "return message on error" is requested.
  - "Sequence Control" – The presence of this parameter indicates the SCCP Class 1 is requested and when used in a request primitive, explictly provides the information needed to deliver a series of messages in sequence.

– "Termination" – Indicates which scenario is chosen by the TC-user for the end of the dialogue (prearranged or basic).

– "User Information" – Information which can be exchanged between TCAP users independently from the remote operation service.

– "Report Cause" – Contains information indicating the reason for the exception report, that the message was returned by the SCCP with the reason as specified in Recommendation Q.711. This parameter is required for the TC-NOTICE indication primitive.

3.1.2.2 Dialogue facilities

The dialogue facilities allow a TC-user to exchange components with a peer TC-user to perform a distributed application. The Unidirectional message facility may be used to send class 4 operation invocations and reports of protocol errors in these invocations from either TC-user using an unstructured dialogue. The structured dialogue facilities provide the capability to explicitly initiate a dialogue, exchange components within the dialogue, terminate it, or abort it.

The TCAP dialogue handling service interface is enhanced to enable the TC-user to make use of additonal services provided by the transfer and negotiation of application context and the transparent transfer of user information (i.e. data which are not components). For backward compatiblity reasons, the additional parameters in the dialogue handling primitives are optional and the resultant dialogue portion is also optional.

3.1.2.2.1 Unstructured dialogue

There is no initiation or termination associated with an unstructured dialogue. The only facility provided is the request for transmission of one, or several components invoking class 4 operations or reporting protocol errors in these invocations, grouped in a message to the remote TC-user. The TC-user is expected to ensure that only class 4 operations are sent within an unstructured dialogue, as TCAP does not verify this. The TC-user may also indicate the application context name applicable to the components to be transferred.

Components to be transmitted have been previously passed to the Component sub-layer by means of component handling primitives of the "request" type.

The use of the unstructured dialogue facility is indicated by issuing a TC-UNI primitive, as described in Table 3.

At the originating side, a TC-UNI request primitive is issued to request transmission to the remote TC-user of all the components which have been passed to the Component sub-layer with the same dialogue ID.

At the receiving side, the destination TC-user is informed that one or more component(s) have been received by means of a TC-UNI indication primitive. The parameters in this primitive apply to all the components being received; these components will actually be delivered by means of component handling primitives of the indication type.

3.1.2.2.2 Structured dialogue

The structured dialogue facility allows a TC-user to start a dialogue, exchange, as an option, application context and user information, exchange components within this dialogue, terminate it, or abort it.

3.1.2.2.2.1 Beginning of a dialogue

A TC-user begins a new dialogue by issuing a TC-BEGIN request primitive. The purpose of this primitive is

– to indicate to the Component sub-layer that a new dialogue starts, identified by the "dialogue ID" parameter of the primitive;

– to request transmission of any component(s) previously passed to the Component sub-layer by means of component handling primitives of the "request" type with the same dialogue ID.

The TC-user which begins a new dialogue may also indicate, as an option, the application context name applicable to it.

TABLE 3/Q.771

TC-UNI Primitives

Parameter	Primitive: TC-UNI	
	Request	Indication
Quality of Service	U	O (Note 3)
Destination address	M	M (Note 1)
Application context name	U	C (=)
Originating address	M (Note 1)	M (=)
Dialogue ID	M (Note 2)	
User information	U (Note 4)	C (=)
Components present		M

NOTES

1 This parameter may be implicitly associated with the access point at which the primitive is issued.

2 This parameter has only local significance.

3 When this information is made available by the underlying sub-layer, then it must also be passed up to the service user.

4 The user information can only be included if the application context name parameter is also included.

A TC-BEGIN request primitive may be issued prior to passing any component to the Component sub-layer.

At the receiving side, the destination TC-user is informed that a new dialogue starts by means of a TC-BEGIN indication primitive. The presence of component(s) is indicated by the "Components Present" parameter.

Table 4 describes the TC-BEGIN primitives.

3.1.2.2.2.2 Confirmation of the dialogue

A TC-user indicates that it wants to continue a dialogue by issuing a TC-CONTINUE request primitive. This establishes the dialogue proposed in the received TC-BEGIN indication primitive. At this stage the TC-user may optionally include an Originating Address. This optional parameter only applies to the first backward continue (i.e. the establishment of the dialogue), once a dialogue is established the addresses do not change.

If the application context name in the TC-BEGIN indication primitive is acceptable, the TC-user includes the same value in the first backward TC-CONTINUE request primitive. The TC-user may also indicate that it wants to continue a dialogue with another application context than the one proposed by the dialogue initiator.

The TC-CONTINUE primitive requests transmission of any component(s) that have been passed to the Component sub-layer for this dialogue, since the TC-BEGIN indication primitive was received for this dialogue.

At the receiving side, the TC-CONTINUE indication primitive indicates

- that the dialogue may continue;
- in case the dialogue portion was used at the start of a dialogue, the application context that will be in place for the remainder of the dialogue;
- that components are being delivered (if the "Components Present" parameter does not indicate "null").

TABLE 4/Q.771

TC-BEGIN Primitives

Parameter	Primitive: TC-BEGIN	
	Request	Indication
Quality of Service	U	O (Note 2)
Destination address	M	M (Note 1)
Application context name	U	C (=)
Originating address	M (Note 1)	M (=)
Dialogue ID	M	M
User information	U (Note 3)	C (=)
Components present		M

NOTES

1 This parameter may be implicitly associated with the access point at which the primitive is issued.

2 When this information is made available by the underlying sub-layer, then it must also be passed up to the service user.

3 The user information can only be included if the application context name parameter is also included.

Table 5 describes the TC-CONTINUE primitives when they are used to confirm a new dialogue.

TABLE 5/Q.771

TC-CONTINUE Primitives

Parameter	Primitive: TC-CONTINUE	
	Request	Indication
Quality of Service	U	O (Note 1)
Originating address	O	(Note 2)
Application context name	U	C (=)
Dialogue ID	M	M
User information	U (Note 3)	C (=)
Components present		M

NOTES

1 When this information is made available by the underlying sub-layer, then it must also be passed up to the service user.

2 This parameter is not passed to the TC-user.

3 The user information can only be included if the application context name parameter is also included.

3.1.2.2.2.3 Continuation of the dialogue

A TC-user indicates that it wants to continue an established dialogue by issuing a TC-CONTINUE request primitive. This primitive requests transmission of any components that have been passed to the Component sub-layer for this dialogue, since the last TC-CONTINUE request primitive was issued for this dialogue.

At the receiving side, the TC-CONTINUE indication primitive indicates

- that the dialogue may continue;
- that components are being delivered (if the "Components Present" parameter does not indicate "null").

Table 6 describes the TC-CONTINUE primitives when they are used to continue a dialogue that has been established.

TABLE 6/Q.771

TC-CONTINUE Primitives

Parameter	Primitive: TC-CONTINUE	
	Request	Indication
Quality of Service	U	O (Note 1)
Dialogue ID	M	M
Components present		M
User information	U (Note 2)	C (=)
NOTES 1 When this information is made available by the underlying sub-layer, then it must also be passed up to the service user. 2 This is included only if the application context name was used in the establishment phase.		

3.1.2.2.2.4 End of a dialogue

Three scenarios are provided to TC-users to end a dialogue:

- pre-arranged end;
- basic end;
- abort by the TC-user.

Normal dialogue ending (the first two scenarios) uses the TC-END request and indication primitives as described in Table 7. The "Termination" parameter in the TC-END request primitive indicates which scenario is being used for terminating the dialogue.

a) *pre-arranged end*

In this scenario, TC-users have decided by prior arrangement when to end the dialogue: the effect of the TC-END request primitive is purely local; no TC-END indication is used.

No component can be sent or received for the dialogue once the TC-END request primitive has been issued.

TABLE 7/Q.771

TC-END Primitives

Parameter	Primitive: TC-END	
	Request	Indication
Quality of Service	U	O (Note 1)
Dialogue ID	M	M
Application context name	U (Note 2)	C (=)
Components present		M
User information	U (Note 3)	C (=)
Termination	M	

NOTES

1 When this information is made available by the underlying sub-layer, then it must also be passed up to the service user.

2 These optional parameters are allowed only for the case when the TC-END request is issued in immediate response to a received TC-BEGIN indication.

3 The user information can only be included if the application context name parameter is also included or has been used at dialogue establishment.

b) *basic end*

In this scenario, the ending causes transmission of any pending components at the side which initiates it. Note, however, that any components for which transmission would be pending in the reverse direction will not be delivered.

The basic scenario uses the TC-END primitives for two purposes:

– delivery of any component(s) for which transmission is pending;

– indication that no more components will be exchanged for this dialogue in either direction.

c) *abort of a dialogue by a TC-user*

The TC-user has the ability to request immediate ending of a dialogue without taking into account any pending operation invocation (abort). An abort request by the TC-user causes all pending operations for that dialogue to be terminated. When doing so, the TC-user may provide end-to-end information indicating the cause of the abort and diagnostic information; this information is transported by TCAP without analysis.

Before the dialogue is established (i.e. before the first TC-Continue), the TC-user may also decide to abort a dialogue because the application context name proposed by the dialogue initiator cannot be supported. In this case the abort may indicate an alternative application context name which can be used by the dialogue initiator to open another dialogue for the same purpose.

The TC-U-ABORT request and indication primitives are used to indicate abort by the TC-user; Table 8 describes these primitives.

3.1.2.2.3 Exception Reporting and Message Return

The ability for TC-users to be notified of non-delivery of components is provided by the TC-NOTICE indication primitive.

TABLE 8/Q.771

TC-U-ABORT Primitives

Parameter	Primitive: TC-U-ABORT	
	Request	Indication
Quality of Service	U	O (Note 1)
Dialogue ID	M	M
Abort reason	U	C (=)
Application context name	U (Note 2)	C (=)
User information	U	C (=)

NOTES

1 When this information is made available by the underlying sub-layer, then it must also be passed up to the service user.

2 The application context name parameter shall be present if and only if the abort reason parameter indicates "application context not supported".

A TC-NOTICE indication primitive is only passed to the TC-user if the requested service (i.e. transfer of components) cannot be provided (the network layer cannot deliver the embedded message to the remote node) and the TC-user requested the return option in the Quality of Service parameter of the dialogue handling request primitive.

Table 9 describes this primitive.

TABLE 9/Q.771

TC-NOTICE Primitive

Parameter	Primitive: TC-NOTICE
	Indication
Dialogue ID	M
Report cause	M

3.1.3 Component handling

3.1.3.1 Definition of parameters

This subclause describes the parameters used with the primitives associated with component handling.

– "Class" – See 2.3.1.3.

– "Dialogue ID" – Relates components to a specific dialogue.

– "Invoke ID" – Identifies an operation invocation and its result.

– "Linked ID" – Links an operation invocation to a previous operation invoked by the remote TC-user.

– "Error" – Contains information provided by the TC-user when an operation returns failure.

– "Last Component" – Is used in primitives of the "indication" type only, to designate the last component of a message. Note that indication of the last part of the result of an operation is via the name of the primitive.

– "Operation" – Identifies the action to be executed by a TC-user on request of another TC-user.

– "Parameters" – Contains any parameters accompanying an operation, or provided in reply to an operation.

  The presence of the "Operation" and "Error" parameters is verified by the Component sub-layer and their values are not analysed by this sub-layer.

– "Problem Code" – Identifies the cause for rejecting a component.

– "Timeout" – Indicates the maximum lifetime of an operation invocation.

3.1.3.2 Operation invocation

An operation invocation is requested to the Component sub-layer by means of a TC-INVOKE request primitive. When this invocation is linked to a previous operation, the Linked ID-parameter is used.

A corresponding TC-INVOKE indication primitive is used to indicate operation activation to the destination TC-user.

Table 10 shows the primitives associated with operation invocation.

TABLE 10/Q.771

Operation invocation primitives

Parameter	Primitive: TC-INVOKE	
	Request	Indication
Dialogue ID	M	M (Note)
Class	M	
Invoke ID	M	M (=)
Linked ID	U	C (=)
Operation	M	M (=)
Parameters	U	C (=)
Last component		M
Timeout	M	
NOTE – Mandatory except for invocation of class 4 operation received in a Unidirectional message.		

3.1.3.3 Report of success

Success is reported to indicate that an operation (of class 1 or 3) has been executed by the remote TC-user. The operation is identified by the Invoke ID parameter. Several replies may be used to report success. The following primitives are used:

– TC-RESULT-L – Indicates the only or last segment of a result;

– TC-RESULT-NL – Indicates a segment of a result (with more segments to follow).

TC imposes no limitation on the number of segments. However, when the peer TC-users are certain that the Network Service used supports segmentation and reassembly of user data, the TC-RESULT-NL (RR-NL) facility is not necessary and should be avoided.

The TC-RESULT-L and TC-RESULT-NL primitives are described in Table 11. A primitive of the "request" type is used to pass a component from the TC-user to the Component sub-layer; a primitive of the "indication" type is used to deliver a component to the TC-user.

TABLE 11/Q.771

Report of success primitives

Parameter	Primitive	
	TC-RESULT-L TC-RESULT-NL request	TC-RESULT-L TC-RESULT-NL indication
Dialogue ID	M	M
Invoke ID	M	M (=)
Operation	U (Note)	C (=)
Parameters	U	C (=)
Last component		M
NOTE – Mandatory when the primitive contains the "Parameters" parameter.		

3.1.3.4 Report of failure

A TC-user receiving a (class 1 or 2) operation which it cannot execute, though it "understands" it, will issue a TC-U-ERROR request primitive, indicating the reason of the failure (Error parameter). The corresponding operation is identified by the Invoke ID parameter.

The TC-user which invoked this operation is informed by a TC-U-ERROR indication primitive.

Table 12 describes the TC-U-ERROR primitives.

3.1.3.5 Reject by the TC-user

A TC-user may reject any component (except a Reject component) generated by its peer entity, which it considers incorrect. The cause for the rejection is indicated in the Problem Code parameter. Separate parameters are available for rejection of individual component types.

TABLE 12/Q.771

Report of failure primitives

Parameter	Primitive: TC-U-ERROR	
	Request	Indication
Dialogue ID	M	M
Invoke ID	M	M (=)
Error	M	M (=)
Parameters	U	C (=)
Last component		M

Any rejection of an invocation of a result terminates the operation. When a linked operation is rejected, the linked-to operation is not affected.

A TC-user rejects a component by means of the TC-U-REJECT request primitive, and is informed of rejection by the remote TC-user by means of the TC-U-REJECT indication primitive. These primitives are described in Table 13.

TABLE 13/Q.771

User rejection primitives

Parameter	Primitive: TC-U-REJECT	
	Request	Indication
Dialogue ID	M	M (Note)
Invoke ID	M	M (=)
Problem code	M	M (=)
Last component		M
NOTE – Mandatory except for rejection of invocation of class 4 operation received in a Unidirectional message.		

3.1.3.6 Cancel of an operation

The cancel facility terminates the corresponding operation invocation. It can be requested by the TC-user, or the notification of timer expiration is indicated by the Component sub-layer. In both cases, it has only local effect – no notification is sent to the remote end.

The Component sub-layer uses the cancel facility to inform the TC-user that the timer associated with a class 1, 2 or 3 operation has expired; the TC-L-CANCEL indication primitive is used for this purpose. The timer is run for all classes, but the reporting for class 4 operations is implementation dependent. For operations of class 1, timeout is an abnormal situation; for operations of class 2, 3 or 4, timeout is a "normal" situation.

The TC-user uses the TC-U-CANCEL request primitive to inform the local Component sub-layer of a cancel decision. No component is sent.

Table 14 describes the TC-CANCEL primitives.

TABLE 14/Q.771

TC-CANCEL Primitives

Parameter	Primitive	
	TC-L-CANCEL indication	TC-U-CANCEL request
Dialogue ID	M	M
Invoke ID	M	M

3.1.3.7 Grouping of components inside a Message

A sequence of components is obtained by passing one or several components with a given Dialogue ID to the Component sub-layer between two successive requests for transmission (TC-BEGIN, TC-CONTINUE or TC-END request primitives), or before the first one (TC-BEGIN request), using the same dialogue ID.

In the case of an unstructured dialogue, the only request for transmission, TC-UNI request, causes the sending of components with the same dialogue ID as in the TC-UNI primitive.

At the originating side, a list of components is delimited by TC-UNI, TC-BEGIN, TC-CONTINUE or TC-END request primitives.

At the destination side, a sequence of components starts with an indication primitive; its end is indicated by the "Last Component" parameter of the primitives which deliver components to a TC-user. The "Components Present" parameter in the indication primitive indicates whether the sequence is empty, or not.

NOTE – Components grouped inside a message are delivered to the remote end in the same order as they are provided by the TC-user at the originating end.

3.1.4 Abnormal situations

3.1.4.1 Reject of a component by the Component sub-layer

While detecting that a received component is invalid, the Component sub-layer notifies the local TC-user by means of the TC-L-REJECT indication primitive. This primitive indicates the cause of the reject (Problem Code parameter) with sufficient information to make the retention of the failed component superfluous: whenever possible the Component Type and Component ID are indicated; otherwise a "general problem" cause is indicated. This information is passed to the TC-user, and also retained in the Component sub-layer which uses it to form a Reject component.

Any type of component can be rejected. When the component to be rejected is itself identified as a Reject component, rejection is purely local; when the rejected component is identified as an Invoke, a Return Result, or a Return Error, the whole corresponding operation is considered as terminated; when it is a linked operation, this linked operation is terminated, but the linked-to operation is not affected.

When informed of a Component sub-layer reject, the local TC-user may decide, in the case of a structured dialogue, to continue the exchange of components if the dialogue is still active. If so, the remote TC-user is informed through the reject component sent when the local TC-user issues the next dialogue control primitive.

The remote TC-user is informed of the received Reject component through a TC-R-REJECT indication primitive.

In the case of an unstructured dialogue, when the Component sub-layer detects an invalid component in a UNIDIRECTIONAL message, the TC-user is informed by means of the TC-L-REJECT indication primitive. The Component sub-layer forms a Reject component which is sent only if the TC-user chooses to inform the remote node of this occurence. In this case, the TC-user issues a TC-UNI request primitive which causes the transmission of the Reject component.

If the Component sub-layer generated Reject component combined with accumulated components from the TC-user exceeds the message length limitations, then the TC-user, being aware of the Reject component, must initiate two dialogue handling primitives. The Component sub-layer, also being aware of the length problem, will send all the components, except the Reject, with the first primitive. The Reject will be sent with the next dialogue handling primitive together with any further components provided by the TC-user.

Table 15 describes the primitives used in relation with TCAP component rejection.

TABLE 15/Q.771

Component sub-layer rejection primitive

Parameter	Primitive	
	TC-L-REJECT indication	TC-R-REJECT indication
Dialogue ID	M	M (Note)
Invoke ID	O	O
Problem code	M	M
Last component	M	M
NOTE – Mandatory except for rejection of invocation of a class 4 operation received in a Unidirectional message.		

3.1.4.2 Dialogue abort

Due to an abnormal situation, an underlying (sub-)layer may decide to abort the association between users; the structured dialogue has then to be aborted. All associated operations are terminated, and the TC-users are notified by means of TC-P-ABORT indication primitives. The P-abort parameter contains the cause for which it was decided to abort the dialogue.

The Component sub-layer will decide to abort a dialogue and discard any components received if an incorrect dialogue portion (e.g. syntactically incorrect or inopportune) is received or if no common version of the dialogue portion control APDUs can be agreed to. It informs its TC-user via a TC-P-ABORT indication primitive with the "P-Abort" in the primitive set to respectively "abnormal dialogue" or "no common dialogue portion".

Table 16 describes the TC-P-ABORT primitive.

TABLE 16/Q.771

Primitive for TCAP Abort

Parameter	Primitive
	TC-P-ABORT indication
Quality of Service	O (Note)
Dialogue ID	M
P-Abort	M
NOTE – When this information is made available by the underlying sub-layer, then it must also be passed up to the service user.	

3.1.5 Component states and state transition diagrams

For a given component ID, component correlation takes place only at the side which originates the operation; for this ID, component states and state transition diagrams are defined at this side only. The other side just reflects the value of the component ID in an Invoke or a Linked ID.

The following states are defined:

- *Idle* – No activity associated with the ID.
- *Operation Pending* – An operation has been passed to the Component sub-layer, but no request for transmission has been issued.
- *Operation Sent* – An operation has been transmitted to the remote end, but no result has been received. The timer associated with the operation invocation (with the value of "Timeout") is started when the transition from "Idle" to "Operation Sent" occurs.
- *Wait for Reject* – The result has been received; TCAP is waiting for its possible rejection by the TC-user.
- *Reject pending* – Reject of the result has been requested by the TC-user, but no request for transmission has been issued.

State transition diagrams are defined for the four classes of operations.

NOTES

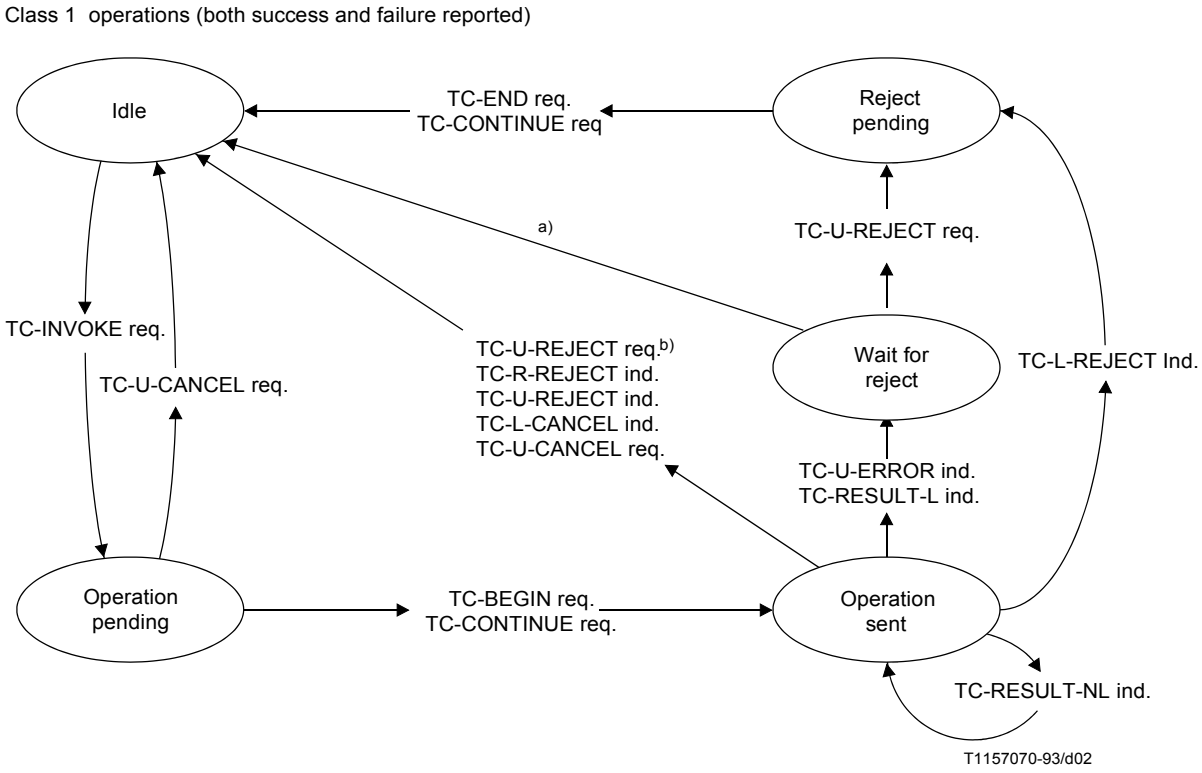
1 Each of these diagrams corresponds to one component ID: the one indicated in the Invoke ID parameter; linked operations do not alter the state machine of the linked-to operation.

2 TC-END request or indication primitives, TC-U-ABORT request or indication primitives, or the TC-P-ABORT indication primitive cause return to the "Idle" state of any component ID associated with the dialogue. Corresponding transitions are not always represented on the diagrams.

3.1.6 Mapping of Component sub-layer onto Transaction sub-layer

When mapping the Component sub-layer onto the Transaction sub-layer, a one-to-one mapping exists between a dialogue and a transaction explicity in the case of a structured dialogue, or implicitly in the case of an unstructured dialogue. It follows that there is a one-to-one relationship between dialogue handling primitives of the Component sub-layer and transaction handling primitives in the Transaction sub-layer; similar generic names have been chosen for the primitives to reflect this. The component handling primitives of the Component sub-layer have no counterpart in the Transaction sub-layer.

The correspondence between the two sub-layers is further described in Recommendation Q.774.

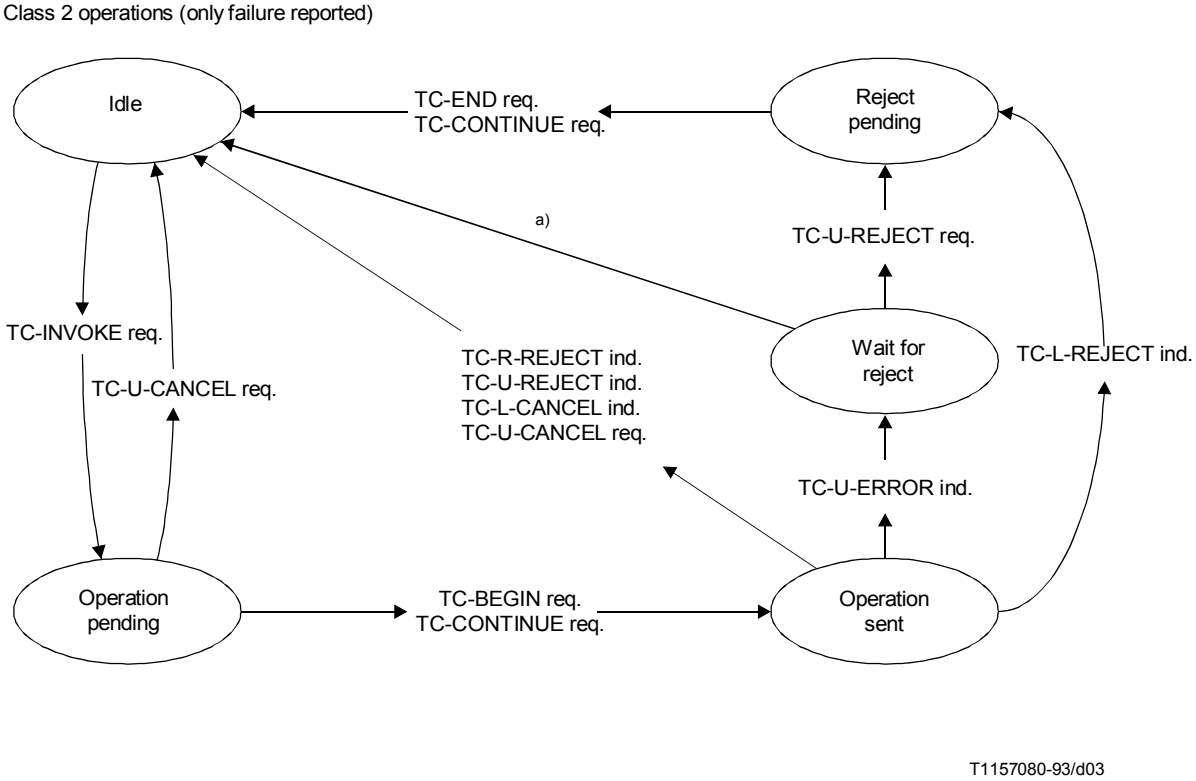


a) This transition is based on an implementation dependent mechanism. The TC-user is not informed in this case.

b) Provisionally accepted pending resolution of the issue that a TC-User can reject a segment of a result.

FIGURE 2/Q.771

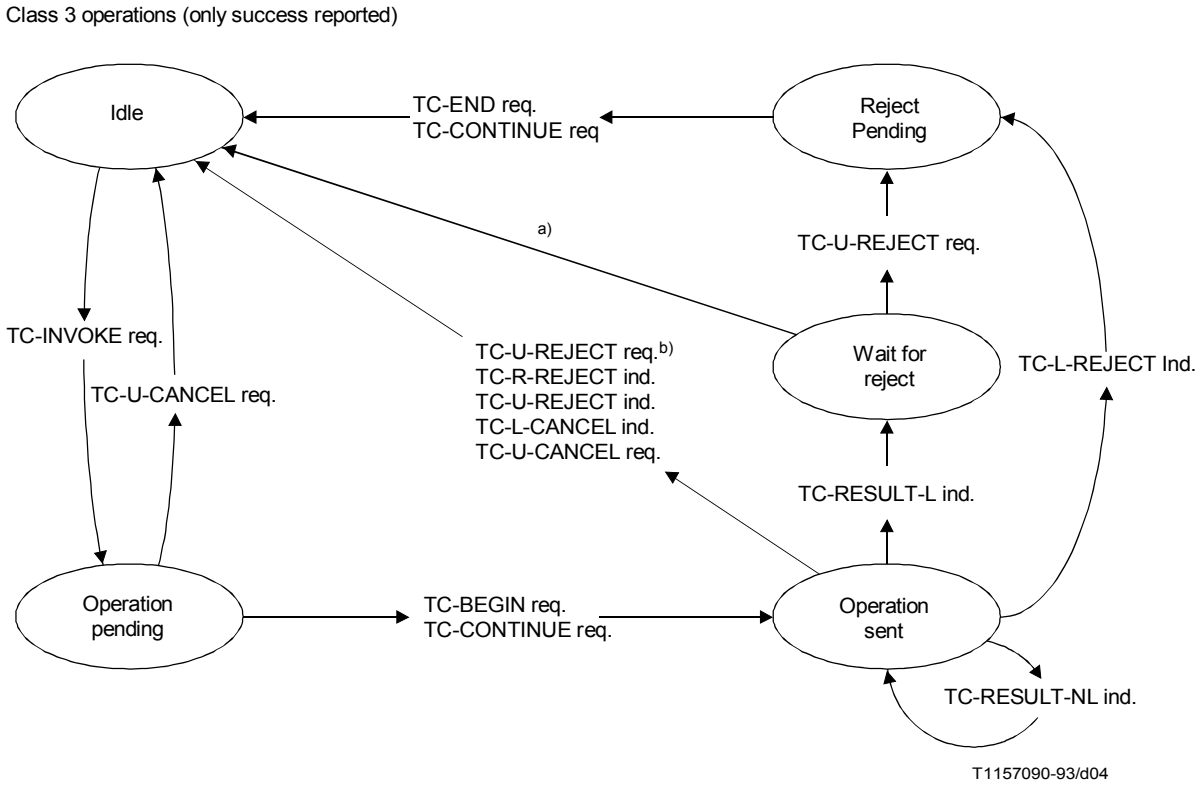
State Transition Diagram for Class 1 Operations



[a] This transition is based on an implementation dependent mechanism. The TC-user is not informed in this case.

FIGURE 3/Q.771

State Transition Diagram for Class 2 Operations

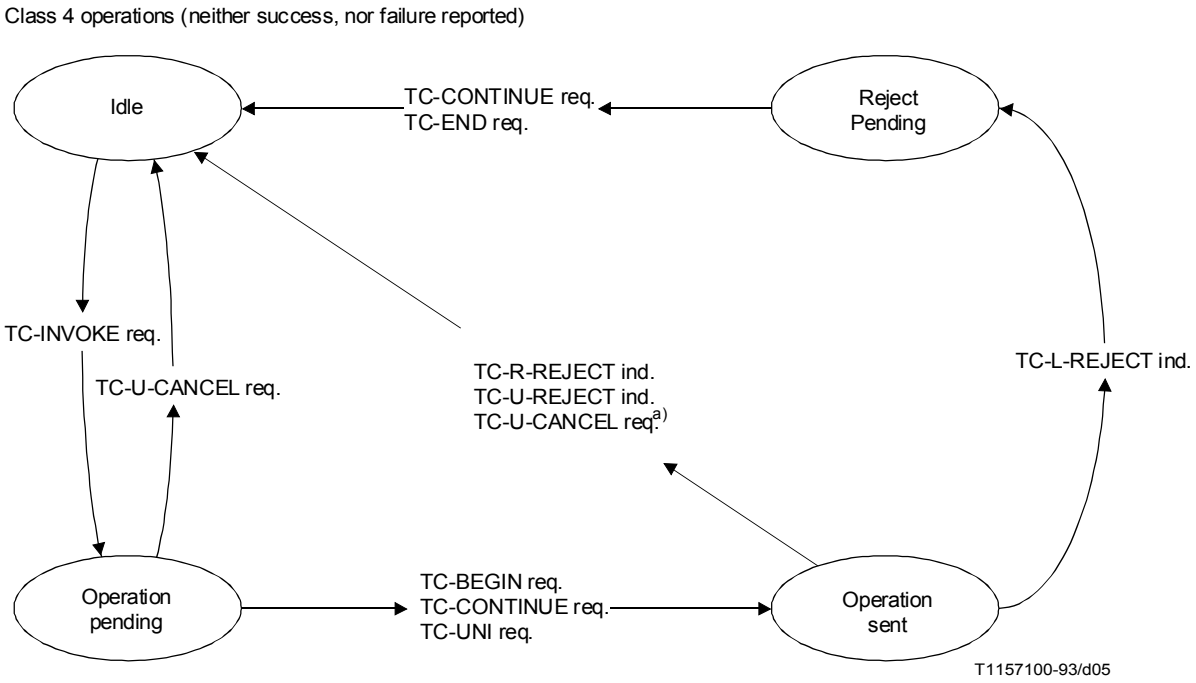


a) This transition is based on an implementation dependent mechanism. The TC-user is not informed in this case.

b) Provisionally accepted pending resolution of the issue that a TC-User can reject a segment of a result.

FIGURE 4/Q.771

State Transition Diagram for Class 3 Operations



a) This transition can occur as a result of operation timeout expiry Notification to the TC-user is a local matter.

FIGURE 5/Q.771

State Transition Diagram for Class 4 Operations

3.2 Transaction sub-layer

3.2.1 Overview of the Transaction sub-layer primitives

Table 17 gives an overview of the primitives between the TR-users and the Transaction sub-layer. A detailed description of these primitives and their parameters is given in the next subclauses. For each primitive, Table 17 indicates the relevant subclause.

TABLE 17/Q.771

Primitives for the Transaction sub-layer

Name	Type	Reference (subclause)
TR-UNI	Request indication	3.2.2
TR-BEGIN	Request indication	3.2.3
TR-CONTINUE	Request indication	3.2.4
TR-END	Request indication	3.2.5
TR-U-ABORT	Request indication	3.2.5.3
TR-P-ABORT	Indication	3.2.6.1
TR-NOTICE	Indication	3.2.7

Definition of the parameters:

- "Quality of Service" – The TR-user indicates the preferred Quality of Service. The "Quality of Service" parameters for the connectionless SCCP network service at present consist of the following:
  - "Return Option" – Specifies whether the SCCP "return message on error" is requested.
  - "Sequence Control" – The presence of this parameter indicates that SCCP class 1 is requested and when used in a request primitive, explicitly provides the information needed to deliver a series of messages in sequence.
- "Destination Address" – Identifies the destination TR-user.
- "Originating Address" – Identifies the originating TR-user.
- "P-Abort" – Indicates the cause of the abort of a transaction by Transaction sub-layer.
- "Transaction ID" – A transaction is identified by a separate transaction ID at each end.
- "Termination" – Identifies the termination scenario chosen for the transaction (pre-arranged or basic).
- "User Data" – Contains the information to be passed between TR-users.
- "Report Cause" – Contains information indicating the reason for the exception report, that the message was returned by the SCCP with the reason as specified in Recommendation Q.711. This parameter is required for the TR-NOTICE indication primitive.

3.2.2 Information transfer in unstructured dialogue

Information may be sent from one TR-user to another TR-user without their establishing an explicit association. In this case, the transaction sub-layer considers that there is no relationship among messages transmitted by this means.

The corresponding primitives are the TR-UNI request and indication primitives, described in Table 18.

TABLE 18/Q.771

TR-UNI Primitives

Parameter	Primitive: TR-UNI	
	Request	Indication
Quality of Service	U	O (Note 2)
Destination address	M	M (Note 1)
Originating address	M (Note 1)	M (=)
User data	M	M (=)

NOTES

1 This parameter may be implicitly associated with the access point at which the primitive is issued.

2 When this information is made available by the underlying layer, then it must also be passed up to the service user.

3.2.3 Transaction begin

The transaction begin facility provides means to start a transaction between two TR-users. This may be accompanied by the transfer of TR-user information (called user data in the following).

In order to begin a transaction, a TR-user issues the TR-BEGIN request primitives.

At the destination side, the TR-BEGIN indication primitive is used to inform the destination TR-user of the beginning of a transaction, and to deliver any accompanying user data.

Table 19 describes the transaction begin primitives.

Figure 6 shows the transaction state transitions during transaction begin. The following states are introduced (these apply to both the dialogue and transaction):

- Idle (I) – The transaction does not exist.
- Init sent (IS) – The transaction just started at the originating side.
- Init received (IR) – The transaction just started at the destination side.

3.2.4 Transaction continuation

3.2.4.1 Confirmation of the transaction

To confirm a transaction the TR-user initiates a TR-CONTINUE request primitive. The TR-user may optionally include an Originating Address. This optional parameter only applies to the first backward TR-CONTINUE request primitive (i.e. the confirmation), once the transaction is confirmed the addresses do not change.

TABLE 19/Q.771

Primitives for transaction begin

Parameter	Primitive: TR-BEGIN	
	Request	Indication
Quality of Service	U	O (Note 2)
Destination address	M	M (Note 1)
Originating address	M (Note 1)	M (=)
Transaction ID	M	M
User data	U	C (=)

NOTES

1 This parameter may be implicitly associated with the access point at which the primitive is issued.

2 When this information is made available by the underlying layer, then it must also be passed up to the service user.

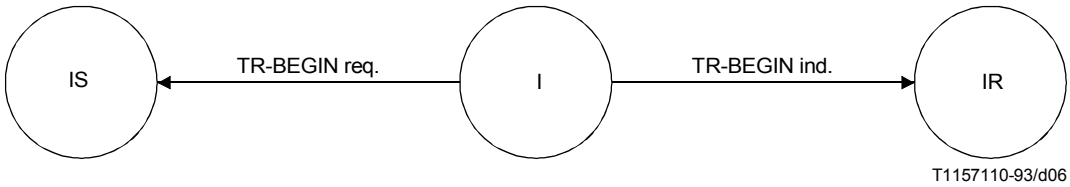


FIGURE 6/Q.771

State transitions for transaction begin

The TR-CONTINUE primitives are described by Table 20. The Transaction sub-layer does not provide segmentation/reassembly or flow control.

3.2.4.2 Continuation of the transaction

Transaction continuation allows two TR-users to exchange messages in both directions inside a transaction. The TR-CONTINUE primitives are used for this purpose. They are described in Table 21. The Transaction sub-layer does not provide segmentation/reassembly or flow control.

3.2.4.3 State transitions

State transitions associated with the continuation of a transaction are represented in Figure 7, where state A (Active) indicates that the transaction was accepted by the remote end, and the transaction can be used to exchange messages in both directions. The A state applies to both the dialogue and transaction.

TABLE 20/Q.771

Transaction Continuation Primitives

Parameter	Primitive: TR-CONTINUE	
	Request	Indication
Quality of Service	U	O (Note 1)
Originating address	O	(Note 2)
Transaction ID	M	M
User data	U	C (=)

NOTES

1 When this information is made available by the underlying layer, then it must also be passed up to the service user.

2 This parameter is not passed to the TR-user.

TABLE 21/Q.771

Transaction Continuation Primitives

Parameter	Primitive: TR-CONTINUE	
	Request	Indication
Quality of Service	U	O (Note)
Transaction ID	M	M
User data	U	C (=)

NOTE – When this information is made available by the underlying layer, then it must also be passed up to the service user.

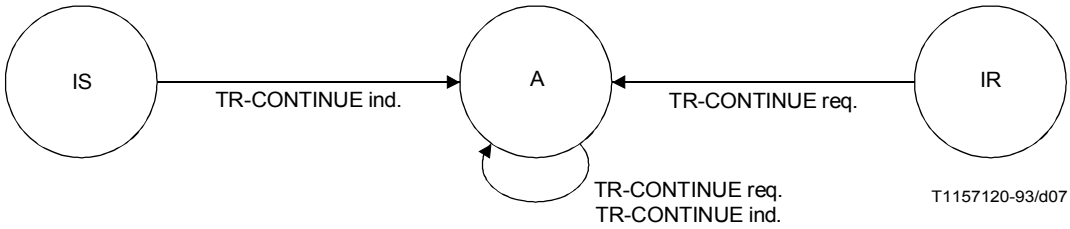


FIGURE 7/Q.771

State Transitions for Transaction Continuation

3.2.5 Transaction end

Three facilities are provided to a TR-user to end a transaction:

- pre-arranged end;
- basic end;
- abort.

The first two facilities use the TR-END primitives; the "Termination" parameter indicates which option is selected. The TR-END primitives are described in Table 22.

The last facility uses the TR-U-ABORT primitives described in Table 23.

TABLE 22/Q.771

TR-END Primitives

Parameter	Primitive: TR-END	
	Request	Indication
Quality of Service	U	O (Note)
Transaction ID	M	M
Termination	M	
User data	U	C (=)
NOTE – When this information is made available by the underlying layer, then it must also be passed up to the service user.		

3.2.5.1 Pre-arranged end

When pre-arranged end has been selected, the termination procedure is purely local. Each TR-user may decide to end the transaction at any point in time, regardless of the current transaction state. The TR-END request primitive only is used: the remote TR-user is not informed, and should request transaction termination on its own. The "User Data" parameter should not be present in this case.

Figure 8 shows the transaction state transitions for pre-arranged end of a transaction. The states are those defined in 3.2.3 and 3.2.4 above.

3.2.5.2 Basic end

When basic termination has been selected, the TR-user requests the end of the transaction by issuing the TR-END request primitive indicating this option; the primitive may then contain "User Data" which is sent to the peer entity.

At the destination side, the TR-END indication primitive is used to inform the TR-user of the end of the transaction, and deliver any accompanying "User Data".

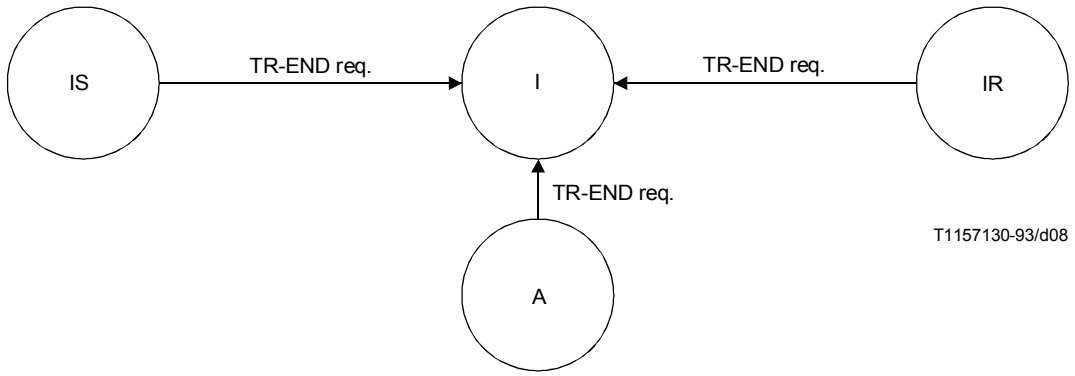


FIGURE 8/Q.771

State Transitions for the Pre-arranged End of a Transaction

Figure 9 shows the transaction state transitions for the basic end of transaction. The states are those defined in 3.2.3 and 3.2.4 above.

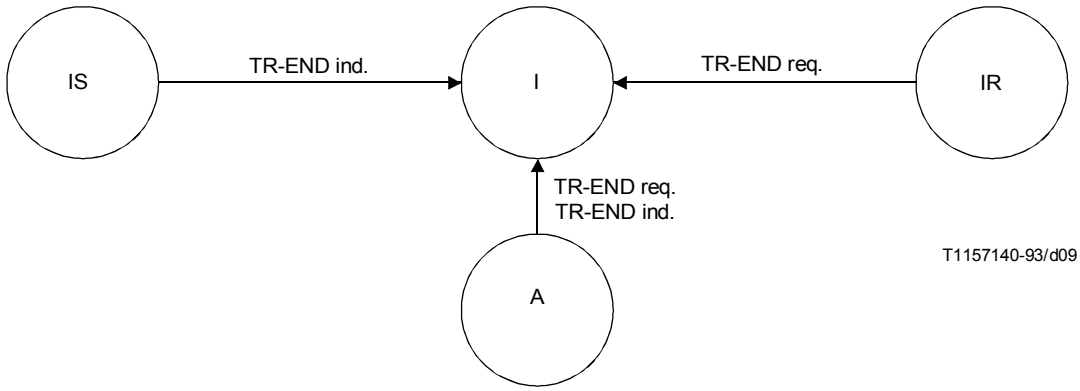


FIGURE 9/Q.771

State Transitions for the Basic End of a Transaction

3.2.5.3 Transaction abort by the TR-user

A TR-user may request the abort of a transaction at any moment; it uses for this purpose the TR-U-ABORT request primitive, which may optionally contain the cause of the abort, and/or optional end-to-end information. This information is contained in the User Abort Information parameter: it is transmitted without analysis to the peer entity. Any messages of the transaction for which transmission is pending are discarded.

A TR-user is informed of the decision of its peer entity to abort the transaction by means of the TR-U-ABORT indication primitive.

When the transaction is in the "Initiation Sent" state, i.e. a Begin message has been sent but no backward message for this transaction has been received, the result of the TR-U-ABORT request primitive is purely local. Any message subsequently received that is related to this transaction shall be handled according to the actions indicated in Table 6/Q.774.

TR-U-ABORT primitives are described by Table 23.

TABLE 23/Q.771

TR-U-ABORT Primitives

Parameter	Primitive: TR-U-ABORT	
	Request	Indication
Quality of Service	U	O (Note)
Transaction ID	M	M
User data	U	C (=)
NOTE – When this information is made available by the underlying layer, then it must also be passed up to the service user.		

3.2.6 Abnormal situations

3.2.6.1 Abort by the Transaction sub-layer

The abort facility may be invoked by the Transaction sub-layer in reaction to abnormal situations. The possible reasons for such a decision are indicated in Recommendation Q.774.

Transaction abort causes the abandonment of any message of the transaction for which transmission is pending.

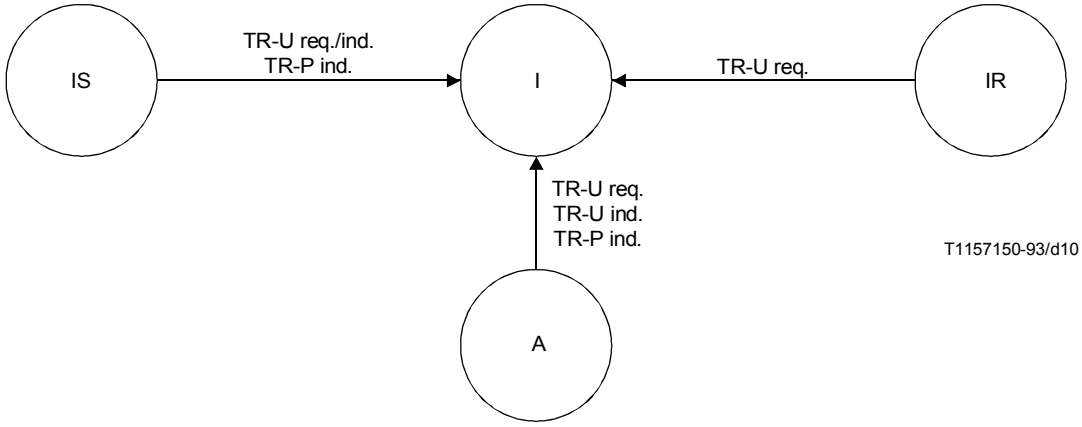
Transaction abort is made by means of the TR-P-ABORT indication primitive described by Table 24.

TABLE 24/Q.771

Transaction sub-layer abort primitive

Parameter	Primitive
	TR-P-ABORT indication
Quality of Service	O (Note)
Transaction ID	M
P-Abort	M
NOTE – When this information is made available by the underlying layer, then it must also be passed up to the service user. This is only applicable when the abort is not locally generated.	

Figure 10 shows the state transitions for transaction abort. The states are those defined in 3.2.3 and 3.2.4 above.



NOTE – TR-P stands for TR-P-ABORT, TR-U for TR-U-ABORT.

FIGURE 10/Q.771

State transitions for transaction abort

3.2.7 Exception Reporting and Message Return

The ability for TR-users to be notified of non-delivery of components is provided by the TR-NOTICE indication primitive.

A TR-NOTICE indication primitive is only passed to the TR-user if the service requested cannot be provided (the network layer cannot deliver the message to the remote node) and the TR-user requested the return option in the Quality of Service parameter.

The TR-NOTICE primitive is described in Table 25.

TABLE 25/Q.771

TR-NOTICE Primitive

Parameter	Primitive: TR-NOTICE
	Indication
Transaction ID	M (Note)
Report cause	M
NOTE – The derivability of the Transaction ID is parser dependent. If the Transaction ID cannot be derived, the N-NOTICE indication will not be delivered to the TR-user.	

3.3 Services assumed from the connectionless network layer

In the Signalling System No. 7 environment, the services assumed from the SCCP are those defined in Recommendation Q.711 (SCCP connectionless services, class 0 or class 1).

Printed in Switzerland

Geneva, 1994