



ATIS-1000114.2004(\$2019)

Signalling System Number 7 (SS7) – Transaction
Capabilities Application Part (TCAP)

AMERICAN NATIONAL STANDARD FOR TELECOMMUNICATIONS



As a leading technology and solutions development organization, ATIS brings together the top global ICT companies to advance the industry's most-pressing business priorities. Through ATIS committees and forums, nearly 200 companies address cloud services, device solutions, emergency services, M2M communications, cyber security, ehealth, network evolution, quality of service, billing support, operations, and more. These priorities follow a fast-track development lifecycle – from design and innovation through solutions that include standards, specifications, requirements, business use cases, software toolkits, and interoperability testing.

ATIS is accredited by the American National Standards Institute (ANSI). ATIS is the North American Organizational Partner for the 3rd Generation Partnership Project (3GPP), a founding Partner of oneM2M, a member and major U.S. contributor to the International Telecommunication Union (ITU) Radio and Telecommunications sectors, and a member of the Inter-American Telecommunication Commission (CITEL). For more information, visit www.atis.org.

AMERICAN NATIONAL STANDARD

Approval of an American National Standard requires review by ANSI that the requirements for due process, consensus, and other criteria for approval have been met by the standards developer.

Consensus is established when, in the judgment of the ANSI Board of Standards Review, substantial agreement has been reached by directly and materially affected interests. Substantial agreement means much more than a simple majority, but not necessarily unanimity. Consensus requires that all views and objections be considered, and that a concerted effort be made towards their resolution.

The use of American National Standards is completely voluntary; their existence does not in any respect preclude anyone, whether he has approved the standards or not, from manufacturing, marketing, purchasing, or using products, processes, or procedures not conforming to the standards.

The American National Standards Institute does not develop standards and will in no circumstances give an interpretation of any American National Standard. Moreover, no person shall have the right or authority to issue an interpretation of an American National Standard in the name of the American National Standards Institute. Requests for interpretations should be addressed to the secretariat or sponsor whose name appears on the title page of this standard.

CAUTION NOTICE: This American National Standard may be revised or withdrawn at any time. The procedures of the American National Standards Institute require that action be taken periodically to reaffirm, revise, or withdraw this standard. Purchasers of American National Standards may receive current information on all standards by calling or writing the American National Standards Institute.

Notice of Disclaimer & Limitation of Liability

The information provided in this document is directed solely to professionals who have the appropriate degree of experience to understand and interpret its contents in accordance with generally accepted engineering or other professional standards and applicable regulations. No recommendation as to products or vendors is made or should be implied.

NO REPRESENTATION OR WARRANTY IS MADE THAT THE INFORMATION IS TECHNICALLY ACCURATE OR SUFFICIENT OR CONFORMS TO ANY STATUTE, GOVERNMENTAL RULE OR REGULATION, AND FURTHER, NO REPRESENTATION OR WARRANTY IS MADE OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR AGAINST INFRINGEMENT OF INTELLECTUAL PROPERTY RIGHTS. ATIS SHALL NOT BE LIABLE, BEYOND THE AMOUNT OF ANY SUM RECEIVED IN PAYMENT BY ATIS FOR THIS DOCUMENT, AND IN NO EVENT SHALL ATIS BE LIABLE FOR LOST PROFITS OR OTHER INCIDENTAL OR CONSEQUENTIAL DAMAGES. ATIS EXPRESSLY ADVISES THAT ANY AND ALL USE OF OR RELIANCE UPON THE INFORMATION PROVIDED IN THIS DOCUMENT IS AT THE RISK OF THE USER.

NOTE - The user's attention is called to the possibility that compliance with this standard may require use of an invention covered by patent rights. By publication of this standard, no position is taken with respect to whether use of an invention covered by patent rights will be required, and if any such use is required no position is taken regarding the validity of this claim or any patent rights in connection therewith. Please refer to [<http://www.atis.org/legal/patentinfo.asp>] to determine if any statement has been filed by a patent holder indicating a willingness to grant a license either without compensation or on reasonable and non-discriminatory terms and conditions to applicants desiring to obtain a license.

ATIS-1000114.2004(S2019), *Signalling System Number 7 (SS7) – Transaction Capabilities Application Part (TCAP)*

Is an American National Standard developed by the **Signaling, Architecture, and Control (SAC)** Subcommittee under the **ATIS Packet Technologies and Systems Committee (TMOC)**.

Published by

Alliance for Telecommunications Industry Solutions
1200 G Street, NW, Suite 500
Washington, DC 20005

Copyright © 2019 by Alliance for Telecommunications Industry Solutions
All rights reserved.

No part of this publication may be reproduced in any form, in an electronic retrieval system or otherwise, without the prior written permission of the publisher. For information contact ATIS at 202.628.6380. ATIS is online at <http://www.atis.org>.

American National Standard for Telecommunications

**SIGNALLING SYSTEM NUMBER 7 (SS7) –
TRANSACTION CAPABILITIES APPLICATION PART (TCAP)**

Secretariat

Alliance for Telecommunications Industry Solutions

Approved September 3, 2004

American National Standards Institute, Inc.

Abstract

This document is based on T1.114-2000, and allows functions similar to those in ITU-T Recommendations Q.771 through Q.774 of the White Book specification of Signalling System No. 7 for international use, issued by the ITU-T Study Group XI (Vol. VI Fascicle VI.9).

FOREWORD

The information contained in this Foreword is not part of this American National Standard (ANS) and has not been processed in accordance with ANSI's requirements for an ANS. As such, this Foreword may contain material that has not been subjected to public review or a consensus process. In addition, it does not contain requirements necessary for conformance to the Standard.

This document is entitled *Signalling System Number 7 (SS7) – Transaction Capabilities Application Part (TCAP)*. It is based on T1.114-2000, and allows functions similar to those in ITU-T Recommendations Q.771 through Q.774 of the White Book specification of Signalling System No. 7 for international use, issued by the ITU-T Study Group XI (Vol. VI Fascicle VI.9).

A change bar on the right margin indicates a change from the 2000 American National Standard. These change bars are advisory only, and reflect the editors' views of which textual changes constitute significant technical changes. Because of the differences in style and content between this standard and the ITU-T Recommendations, it is not possible to indicate differences using margin marks.

This standard is intended for use in conjunction with T1.110-1999, *Signalling System Number 7 (SS7) – General information*, which includes an overview of SS7, a glossary, and a chapter on abbreviations.

Information contained in an informative annex in these specifications is not considered part of this standard but is rather auxiliary to the standard. Similarly, footnotes are not officially part of this standard.

ANSI guidelines specify two categories of requirements: mandatory and recommendation. The mandatory requirements are designated by the word *shall* and recommendations by the word *should*. Where both a mandatory requirement and a recommendation are specified for the same criterion, the recommendation represents a goal currently identifiable as having distinct compatibility or performance advantages.

The Alliance for Telecommunication Industry Solutions (ATIS) serves the public through improved understanding between carriers, customers, and manufacturers. The Packet Technologies and Systems Committee (PTSC) – formerly T1S1 -- develops and recommends standards and technical reports related to services, architectures, and signaling, in addition to related subjects under consideration in other North American and international standards bodies.. The PTSC coordinates and develops standards and technical reports relevant to telecommunications networks in the U.S., reviews and prepares contributions on such matters for submission to U.S. ITU-T and U.S. ITU-R Study Groups or other standards organizations, and reviews for acceptability or per contra the positions of other countries in related standards development and takes or recommends appropriate actions..

Suggestions for improvement of this document are welcome. They should be sent to the Alliance for Telecommunications Industry Solutions, PTSC Secretariat, 1200 G Street NW, Suite 500, Washington, DC 20005.

At the time it approved this document, PTSC, which is responsible for the development of this Standard, had the following members:

- R. Hall, T1S1 Chair
- J. Zebarth, T1S1 Vice-Chair
- S. Carioti, ATIS Disciplines
- S. Barclay, ATIS Secretariat
- C. Underkoffler, ATIS Chief Editor
- S. Goldman, T1S1 Technical Editor

Organization Represented	Name of Representative
Alcatel USA Inc.	Ken Biholar
AT&T	Martin Dolly George Stanek (Alt.)
AT&T Wireless Services, Inc.	Peter Musgrove Brian Daly (Alt.)
Bellsouth Telecommunications	David M. Brady
C.S.I Telecommunications	Michael S. Newman Thomas G. Croda (Alt.)
Cisco Systems	Rajiv Kapoor Chip Sharp (Alt.)
Defense Info. Systems Agency	Chris Fitzgerald
Ericsson Incorporated	Susana Sabater-Maroto Stephen Hayes (Alt.)

Organization Represented	Name of Representative
FBI ESTS	Gregory Milonovich Eric Mason (Alt.)
Harris Corporation	Marlis Humphrey
Hewlett-Packard	Steve Mills
Intelsat	Mark T. Neibert
Lucent Technologies	Stuart O. Goldman
National Communications System	Nicholas Andre Jean Trakinat (Alt.)
Nortel Networks	Joseph A. Zebarth David Bench (Alt.)
Qwest	Steve Showell Michael Fargano (Alt.)

ATIS-1000114.2004 (S2019)

Organization Represented	Name of Representative
SBC Communications, Inc.	B.S. Sambasivan Bob Hall (Alt.)
Siemens Info & Comm Ntwks, Inc.	Ron Franks David E. Francisco (Alt.)
Sprint Corporation	Mark L. Jones
Telcordia Technologies	Wesley Downum Cliff Halevi (Alt.)

Organization Represented	Name of Representative
Tellabs Operations, Inc.	William A. Walker
Tridea Works	Greg Ratta
Turin Networks	Matthew Nguyen
Verizon Communications	Dave Morris Wendy Pugh (Alt.)

The Interoperability Subcommittee (PTSC-IOP) – formerly T1S1.3, the Transaction Capabilities and Application Layer Working Group -- was responsible for the development of this document and had the following members:

- Wesley Downum, T1S1.3 Chair
- Brian Foster, T1S1.3 Chair
- Stuart Goldman, T1S1.3 (TC & AL) convener and T1S1.3 Vice Chair
- Stuart Goldman, T1S1.3Senior Technical Editor
- Wesley Downum, T1S1.3 Editor
- Paul Garvey, T1S1.3 Editor
- Alex Leung, T1S1.3 Editor
- Carol Moylan, T1S1.3 Editor
- Stewart Patch, T1S1.3 Editor
- Arnette Schultz, T1S1.3 Editor

Bjorn Ahle
 Nicholas E. Andre
 Shridharan Balachandran
 Dick Boblin
 Janey Cheu
 Koan S. Chong
 Don Conrad
 Jeff Copley
 Ranga Dendi
 Amar Deol

Martin Dolly
 Christopher Fisher
 Rakesh Gupta
 Jinsha He
 Rich Hemmeter
 Peter Kelleher
 Ceyhan Lennon
 Anne Marie Livingstone
 Clarence Nurse
 Mike McGrew

Yatendra Pathek
 Mike Pierce
 Andre H. Rausch
 Selvam Rengasami
 Walt Roehr
 John Roquet
 Kraig Sanders
 N. Sandesara
 Viqar Shaikh
 Ray Singh

Glenn Sisson
 Al Varney
 Stan Wainberg
 Volnie Whyte
 Albert Yuhan
 Yi Zhao

American National Standard for Telecommunications –

Signalling System Number 7 (SS7) – Transaction Capabilities Application Part (TCAP)

This standard contains the following five chapters:

- ◆ T1.114.1, *Functional Description of Transaction Capabilities*
- ◆ T1.114.2, *Definitions and Functions of Transaction Capabilities Messages*
- ◆ T1.114.3, *TC Formats and Codes*
- ◆ T1.114.4, *Transaction Capabilities Procedures*
- ◆ T1.114.5, *Definitions and Functions of Transaction Capabilities Operations, Parameters, and Error Codes*

Chapter T1.114.1

Functional Description of Transaction Capabilities

TABLE OF CONTENTS

1 SCOPE, PURPOSE, AND APPLICATION	1
1.1 GENERAL	1
1.2 DEFINITION OF TRANSACTION CAPABILITIES	1
1.3 SCOPE OF TRANSACTION CAPABILITIES	1
1.4 SCOPE OF THE SPECIFICATION OF TRANSACTION CAPABILITIES.....	2
2 ARCHITECTURAL CONCEPTS AND TERMINOLOGY.....	2
2.1 APPLICATION OF OSI REFERENCE MODEL.....	2
2.2 CONSIDERATIONS.....	2
2.2.1 <i>Addressing of Upper Layer Entities</i>	2
2.2.2 <i>Management of Upper Layer Entities</i>	2
2.2.3 <i>Layered vs. Nonlayered</i>	3
2.2.4 <i>Architecture Model vs. Implementation</i>	3
3 OVERVIEW OF TC FUNCTIONS AND PROCEDURES.....	3
3.1 FRAMEWORK FOR TRANSACTION CAPABILITIES PROTOCOL	3
3.2 DISCUSSION	3
3.2.1 <i>Application</i>	3
3.2.2 <i>Presentation</i>	4
3.2.3 <i>Session</i>	4
3.2.4 <i>Transport</i>	4
3.3 IDENTIFYING SERVICES REQUIRED OF EACH LAYER	4
3.4 GENERAL DESCRIPTION OF TCAP PROCEDURES.....	5
3.4.1 <i>Types of Transactions</i>	5
3.4.2 <i>Initiation of Transactions</i>	5
3.4.3 <i>Termination of Transactions</i>	5
3.4.4 <i>Association of Application Process Transactions</i>	5
3.4.5 <i>Correlation of Components</i>	5
3.5 GENERAL COMPONENT PROCEDURES	6
3.5.1 <i>Operation Succeeds</i>	6
3.5.2 <i>Operation Fails</i>	6
3.5.3 <i>Protocol Error at Component Level</i>	6
3.5.4 <i>Protocol Error at TCAP Message Level</i>	6
3.6 SERVICE PROCEDURES	6
4 LAYER SERVICE CHARACTERISTICS	7
4.1 LAYER SERVICES ASSUMED FROM THE SCCP.....	7
4.1.1 <i>Description</i>	7
4.1.2 <i>Primitives and Parameters</i>	7
4.2 PRIMITIVES AND LAYER SERVICES FOR ASP LAYERS	7
4.3 LAYER SERVICES PROVIDED TO THE APPLICATION PROCESS.....	8
5 STRUCTURE OF T1.114.....	8

TABLE OF FIGURES

FIGURE 1/T1.114.1 - TRANSACTION CAPABILITIES	9
FIGURE 2/T1.114.1 - IDENTIFYING LAYERS AND SERVICES WITH HEADERS.....	9
FIGURE 3/T1.114.1 - EXAMPLE STATE DIAGRAM	10
FIGURE 4/T1.114.1 - CONNECTIONLESS TRANSACTION PRIMITIVES	10

Functional Description of Transaction Capabilities

1 SCOPE, PURPOSE, AND APPLICATION

1.1 General

This chapter specifies Transaction Capabilities (TC) for Signalling System Number 7 (SS7). The term, *Transaction Capabilities* refers to the Application layer protocol, called Transaction Capabilities Application Part (TCAP), plus the supporting Presentation, Session, and Transport layers, called the Application Service Part (ASP). To date, only SS7 MTP plus SCCP transport has been considered.

NOTE - For MTP, see –T1.111-2001, *Signalling System Number 7 (SS7) - Message Transfer Part (MTP)*. For SCCP, see T1.112-2001, *Signalling System Number 7 (SS7) - Signalling Connection Control Part (SCCP)*.

Any standard OSI Network layer may be used in place of the MTP plus SCCP, provided that performance requirements of the services being supported by the higher layers can be met.

This chapter may contain requirements that reference other American National Standards. If so, when the American National Standards referenced in the requirements are superseded by revisions, the revisions shall apply.

1.2 Definition of Transaction Capabilities

Transaction Capabilities in the SS7 protocol are functions that control non-circuit-related information transfer between two or more signalling nodes via a signalling network.

1.3 Scope of Transaction Capabilities

Transaction Capabilities in a SS7 network should be considered for use between:

- (1) Exchanges;
- (2) Exchanges and network service centers -- e.g., databases, service control points, Operation, Administration, Maintenance and Provisioning (OAM&P) centers; and
- (3) Subscribers and network service centers (in conjunction with the subscriber access protocol -- e.g., CCITT Recommendation Q.931¹).

Although Transaction Capabilities in a SS7 network could be considered for use between subscribers, the standardization of subscriber-to-subscriber information content is outside the scope of SS7.

Furthermore, Transaction Capabilities in a SS7 network may interwork with a transaction-oriented information transfer originated in or destined for networks using other data communications protocols.

Transaction Capabilities provides a set of procedures that can be used for a variety of services, thereby avoiding the inefficiency of creating specific procedures tailored to a particular need. Thus, Transaction Capabilities provides a framework for a common approach to new services within a network as well as a framework for service architecture for cooperative internetwork services.

¹ T1.607 corresponds to CCITT Recommendation Q.931. All CCITT (now ITU-T) Recommendations are available from the International Telecommunications Union. < <http://www.itu.int/ITU-T/> >.

1.4 Scope of the Specification of Transaction Capabilities

This specification is intended to provide, in an open-ended manner, the capabilities needed to support present and near-term ISDN and non-ISDN services requiring transactions among exchanges, service control points, and databases. Extension to other cases should be straightforward within the framework of Transaction Capabilities.

This specification addresses TC procedures relying on a connectionless network service.

The specification itself is general in nature. Some internetwork services are seen to have sufficient interest across multiple networks that common agreements as to how they will operate across network boundaries are seen as desirable.

2 ARCHITECTURAL CONCEPTS AND TERMINOLOGY

2.1 Application of OSI Reference Model

The layered reference model is recognized as a useful tool in developing protocol specifications. From an end user point of view, Transaction Capabilities for initially planned services lie within the Network layer of the OSI model. For example, from the point of view of an exchange querying a database, the exchange and the database may also be seen as "end users". The nature of anticipated services to be supported suggests that the protocol may require the equivalent of many of the functions provided in OSI layers 6, 5, and 4, called the Application Service Part (ASP) in SS7. The ASP consists of Presentation, Session, and Transport layers. Hence there should be advantages in adopting these concepts to a protocol for network transactions.

Processes providing a transaction-oriented service may appear at more than one signalling point. The architecture shall be able to support global addressing of a transaction service and provide the functions needed to route signals to the appropriate points. The architecture shall also provide management functions to handle congestion and failure of transaction processes.

Figure 1/T1.114.1 illustrates two ways in which the architecture of Transaction Capabilities may be modeled. Part (a) of Figure 1/T1.114.1 shows separate ASP entities for each process supported. Part (b) of Figure 1/T1.114.1 shows a common TCAP and ASP supporting more than one transaction process.

2.2 Considerations

2.2.1 Addressing of Upper Layer Entities

The model uses the subsystem number (SSN) at the SCCP layer to identify the particular Application Process. This allows the SCCP global title translation functions to be used to support global addressing of transaction services, in addition to point code and subsystem number addressing.

2.2.2 Management of Upper Layer Entities

Use of SSNs allows the SCCP management procedures to be used to handle independently failing or congesting Application Processes.

2.2.3 Layered vs. Nonlayered

Some transaction-oriented services may not require any functions of the ASP. This suggests that a nonlayered approach may be preferable. However, as new transaction-oriented services are defined, a nonlayered approach may complicate the protocol design. As well, with a nonlayered approach, the benefits of commonality with other protocols could be lost. Transaction Capabilities have therefore been specified so as to allow inclusion of ASP functions readily when required.

Since not all defined Presentation, Session, and Transport functions are likely to be required at any one node in the network, it is not necessary to provide all the possible ASP functions at every node. Suitable classes, options, and functional units should be selected for each node as the needs of that node dictate. These can be expanded as required.

2.2.4 Architecture Model vs. Implementation

From the OSI point of view, a particular process has its own Application, Presentation, Session, and Transport layers. This is illustrated in part (a) of Figure 1/T1.114.1. Each "stack" is independent of the other "stacks". This concept is applied to the Transaction Capabilities protocol. Note that there is an implicit agreement to use the MTP and SCCP as common elements in this structure.

Within any particular implementation, it is possible to implement each block shown independently or to combine blocks vertically or horizontally or both where desired. Thus part (b) of Figure 1/T1.114.1 may represent an example of an implementation in which all the blocks are combined horizontally, while retaining the protocol model of part (a) of Figure 1/T1.114.1. As stated in 1.3, the approach is not intended to constrain any implementation of a service, whether it be intranetwork or internetwork.

3 OVERVIEW OF TC FUNCTIONS AND PROCEDURES

3.1 Framework for Transaction Capabilities Protocol

The initial Transaction Capabilities protocol, consisting of TCAP and the ASP, is based on the following CCITT Red Book Recommendations:

- ◆ *Application*: X.409, X.410
- ◆ *Presentation*: X.409, X.410
- ◆ *Session*: X.225
- ◆ *Transport*: X.224

3.2 Discussion

3.2.1 Application

Section 2 (Remote Operations) of CCITT Recommendation X.410-1984, *Message Handling Systems: Remote Operations and Reliable Transfer Server*, provides the basis for TCAP for foreseen services. It provides four Operation Protocol Data Units (OPDUs):

- ◆ *Invoke*: {Invoke ID, Correlation ID,² Operation, Argument}
- ◆ *Return Result*: {Correlation ID,³ Result}
- ◆ *Return Error*: {Correlation ID,³ Error, Parameter}
- ◆ *Reject*: {Correlation ID,³ Problem, Parameter⁴}

The Invoke OPDU requests that an operation be performed. The Return Result OPDU reports the successful completion of an operation. The Return Error OPDU reports the unsuccessful completion of an operation. The Reject OPDU reports the receipt and rejection of an incorrect OPDU. These OPDUs may be viewed as tools for constructing a Transaction Capabilities Application Protocol.

An OPDU as extended for TCAP is referred to as a Component. Zero, one, or more Components are carried in a TCAP message.

The interpretation of the operations, parameters, and errors carried in a component is determined by the Application Context. The Dialogue Portion of TCAP may identify the version of T1.114 and the Application Context used during an interaction between two peer applications. The Dialogue Portion of TCAP also may contain Security information.

3.2.2. Presentation

The purpose of the Presentation layer is to provide functions to negotiate and select a transfer syntax and carry-out transfer syntax transformations. These capabilities are not required for presently envisaged services.

3.2.3 Session

Services to be supported using Transaction Capabilities are not expected to require any Session layer services. Hence, TC may operate in a connectionless mode in this case.

3.2.4 Transport

Transport Class 0 service (simple, i.e., no enhancements to the service provided by the Network Layer) may be appropriate since, for many transaction-oriented services, the SS7 network layer (provided by the MTP and SCCP) will provide a sufficiently high degree of reliability.

3.3 Identifying Services Required of Each Layer

Figure 2/T1.114.1 illustrates the general concept of how services of the various layers are activated. As a message passes down through the layers, each layer places a header -- and possibly a trailer -- on it

² In TCAP, Remote Operations, as described in CCITT Recommendation X.410-1984, has been extended to include an optional ID to allow correlation of Invoke OPDUs to other Invoke OPDUs.

³ Section 2 of CCITT Recommendation X.410-1984 calls this ID the Invoke ID; it is being understood that it is the reflection of the Invoke ID appearing in the Invoke OPDU. For TCAP, it is renamed Correlation ID for the OPDUs noted.

⁴ In TCAP, Remote Operations, as described in CCITT Recommendation X.410-1984, has been extended to include a mandatory parameter for the Reject OPDU.

that is specific to the functions that layer performs. These functions may be grouped into classes or functional sets. The original Protocol Data Unit (PDU), plus headers of higher layers, are treated as user information and are not examined by lower layers as the message is passed down. Similarly, when a message is being passed to a higher layer, each layer strips off its header before passing the message up as user data using the appropriate primitives. Some of the information contained in the headers is preserved as parameters in the interlayer primitives -- for example, the called and calling addresses.

For some transaction services, some layers will be null. If a particular transaction service does not require any of the Presentation, Session, or Transport layer services, then layer protocol headers are not required. This is the case when a connectionless mode of operation is used.

3.4 General Description of TCAP Procedures

3.4.1 Types of Transactions

Transactions can be viewed at two levels: 1) Application-Process-to-Application-Process; and 2) TCAP-to-TCAP. An Application Process transaction is defined by the Application Process. A TCAP transaction consists of one or more TCAP messages between two Application Processes. TCAP transactions include one-way and two-way transactions -- i.e., the transaction involves communication in one direction only, or interactive communication. The Application Process initiating the TCAP transaction indicates which case applies, from its point of view, at the start of the transaction.

3.4.2 Initiation of Transactions

An Application Process transaction is initiated with the assignment of a Transaction ID. A TCAP transaction is initiated by sending or receiving an initiating TCAP message.

3.4.3 Termination of Transactions

An Application Process transaction is terminated with the release of the Transaction ID. A TCAP transaction is terminated by either a terminating TCAP message or at the discretion of the Application Processes at both ends (without an explicit message being sent) by informing their respective TCAPs.

3.4.4 Association of Application Process Transactions

Application Process transactions are identified through Transaction IDs. These are carried in the TCAP message. Each Application Process involved in the TCAP transaction may assign Transaction IDs in any manner convenient to it.

3.4.5 Correlation of Components

Within a single transaction, one or more operations may take place. For each operation, one or more Components may be involved.

Return Result, Return Error, and Reject Components are correlated with Invoke Components through a Correlation ID that is set equal to the Invoke ID appearing in the Invoke Component (see 3.2.1). In the case where an Invoke Component must refer to another Invoke Component, it also contains a

Correlation ID (set equal to the Invoke ID of the Invoke Component to which it needs to be correlated) as well as its own Invoke ID.

3.5 General Component Procedures

Depending on the service being supported, an invoked operation may report success or failure, failure only, success only, or neither. Figure 3/T1.114.1 contains an overview state diagram for an Invoke Component that succeeds and reports both success and failure. The state transition diagrams in Chapter T1.114.4 provide further details.

3.5.1 Operation Succeeds

When an originating node invokes an operation at a remote node, and success is reported, a Return Result or Invoke Component is sent.

3.5.2 Operation Fails

When no protocol error exists, and the operation invoked is not able to reach a successful result, a Return Error Component is sent indicating the reason for failure. This applies when the operation reports failure.

3.5.3 Protocol Error at Component Level

When a Component other than a Reject Component is incorrect, its receipt and rejection is reported using a Reject Component. Causes for rejection may include undetected bit errors, badly structured Components, duplicate invocations (when applicable to the service being supported), and unrecognized operations.

3.5.4 Protocol Error at TCAP Message Level

When an incorrect TCAP message is received (e.g., two Transaction IDs are indicated but only one is provided), an Abort Package Type with a P-Abort Cause or User Abort Information data element is returned if the return address is available. The P-Abort Cause or User Abort Information data element is coded to reflect an incorrect TCAP message as opposed to an incorrect Component.⁵

3.6 Service Procedures

Service procedures are defined using Components in sequences appropriate to the service being supported. Operations and parameters are defined as required for the service, including error conditions within the service. Such error conditions are not protocol errors but are part of the service definition. An example of such an error is incomplete customer input.

⁵ Applications using T1.114-1988 report Transaction Portion problems using a Reject Component. It is preferred that other applications report these problems using the Abort Package Type.

A set of operations and parameters are provided in Chapter T1.114.5 for use in developing procedures for specific services. Operations and parameters not included may be defined for the service in question, as required, following the pattern of those provided.

4 LAYER SERVICE CHARACTERISTICS

4.1 Layer Services Assumed from the SCCP

The SCCP is specified in -T1.112-2001, *Signalling System Number 7 (SS7) -- Signalling Connection Control Part (SCCP)*.

4.1.1 Description

The initial services supported by Transaction Capabilities require only SCCP Class 0 -- basic connectionless service -- or Class 1 -- sequenced (MTP) connectionless service. Connection-oriented SCCP classes 2 through 4 will be required for anticipated transaction-oriented services in the medium-to long-term. (Note that SCCP service classes and service classes of other layers do not have a one-to-one correspondence.)

4.1.2 Primitives and Parameters

The primitives and parameters between the SCCP and a user of the SCCP are described in 2.2.2 of Chapter T1.112.1 and Table 8/T1.112.1. Of these, N-UNITDATA shall be required to support the initial set of transaction services.

The primitives N-COORD, N-STATE, N-TRAFFIC, and N-PCSTATE pass between SCCP management and the user of the SCCP. They are related to the management of service resources in specific network architectures. The use of these primitives and the N-NOTICE primitive for Transaction Capabilities are not addressed.

4.2 Primitives and Layer Services for ASP Layers

Primitives for connectionless operation of the ASP are needed. T-, S- and P-UNITDATA primitives are shown in Figure 4/T1.114.1 with the parameters for N-UNITDATA (defined in Table 8/T1.112.1), namely Called Address, Calling Address, Quality of Service Parameter Set, User Data, and optional parameters.

These primitives require no functionality at the indicated layers but preserve the layered structure of the protocol for future, more complex layer services. These more complex layer services will require primitives appropriate to the layer services requested of the ASP layers. These primitives are defined in the CCITT Recommendations listed in 3.1.

Primitives for management purposes in a connectionless mode of operation analogous to N-COORD and the like are not addressed.

Each layer of the architecture introduces its own Protocol Data Units (PDUs) when required. Each PDU is carried as user data by lower layers. Thus, a Presentation PDU (PPDU) carries as user data the Application PDU (APDU). The Session PDU (SPDU) carries the PPDU as user data, and so on. This applies when services of the ASP are used by TCAP.

When a connectionless mode of operation is used, no ASP services are required, therefore, no fields in UNITDATA messages are allocated for Transport, Session, or Presentation PDUs.

4.3 Layer Services Provided to the Application Process

Transaction Capabilities provides the means for an Application Process at a given node to access Application Processes at other nodes via the SS7 network. Transaction Capabilities consists of a set of service elements, each of which accepts and processes requests for provision of some Transaction Capabilities from the Application Process, or provides to the Application Process some response as a result of a stimulus from Transaction Capabilities.

The Application layer is responsible for the transfer of information between Application Processes. It is the highest layer of the OSI reference model and provides the sole means for Application Processes to access the capabilities available from the Open Systems Interconnect environment.

Users of the Application layer services (TC-user) are Application Processes. Application Process data is data transferred between two application entities on behalf of the Application Processes for whom the application entities are providing services. Application Process data consists of zero, one or more Components, and any dialogue portion information as described in 3.2.1. A set of parameters that are associated with or required for the performance of processing functions is included in each Component.

5 STRUCTURE OF T1.114

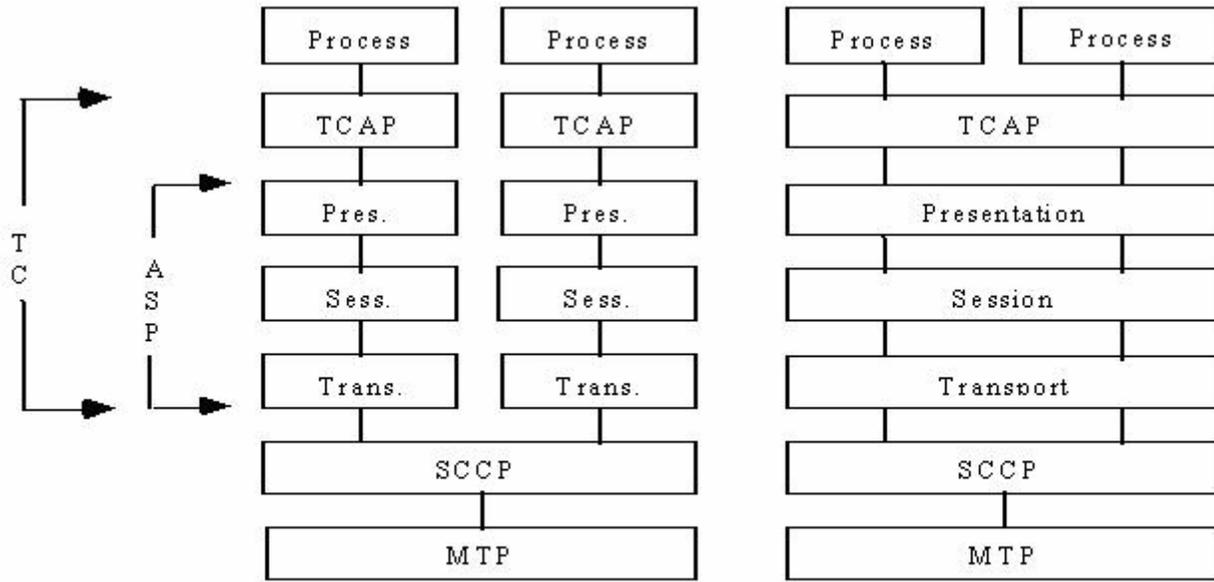
SS7 Transaction Capabilities are specified in Chapters T1.114.1 to T1.114.5.

The elements and functions of the signalling information used within Transaction Capabilities are described in Chapter T1.114.2. Message formats and codings are specified in Chapter T1.114.3. TCAP procedures are specified in Chapter T1.114.4. Operations, parameters, and error codes are specified in Chapter T1.114.5.

The specifications include operations, parameters, and procedures viewed as common to several services.

Where a service requires functionality of the supporting ASP layers, such services are cross-referenced to the appropriate CCITT Recommendations as listed in 3.1 of Chapter T1.114.1.

Chapter T1.114.1



(a) Individual ASP Functions

(b) Common ASP Functions

Figure 1/T1.114.1 - Transaction Capabilities

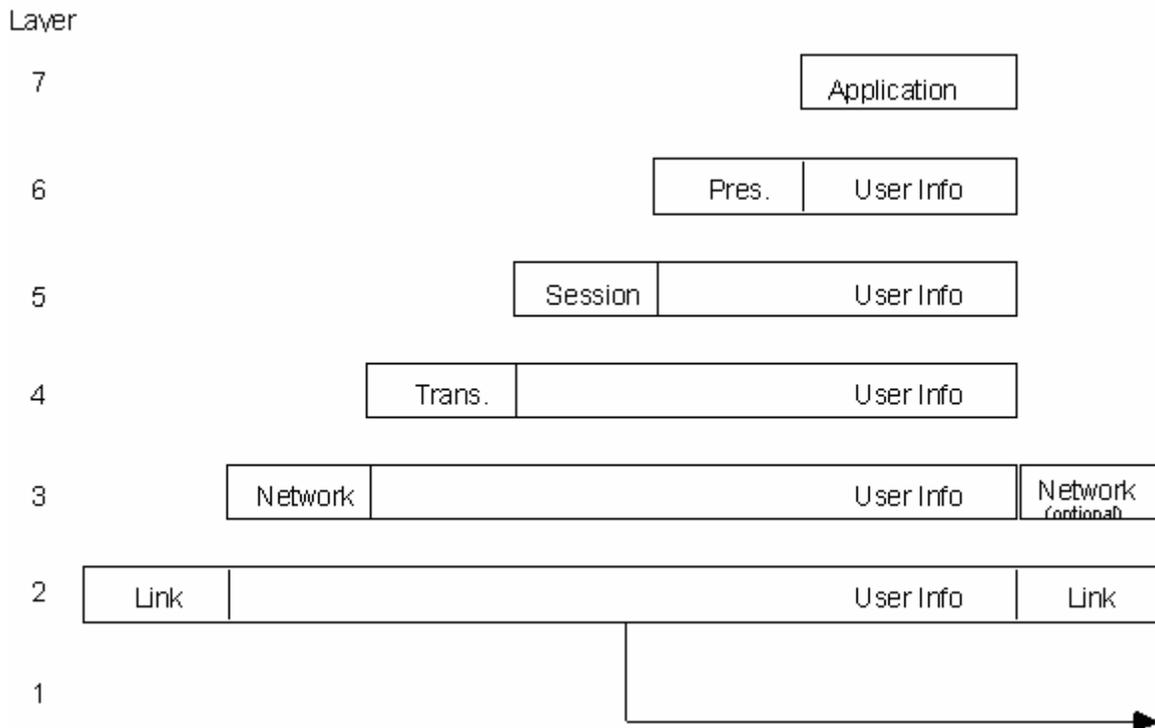
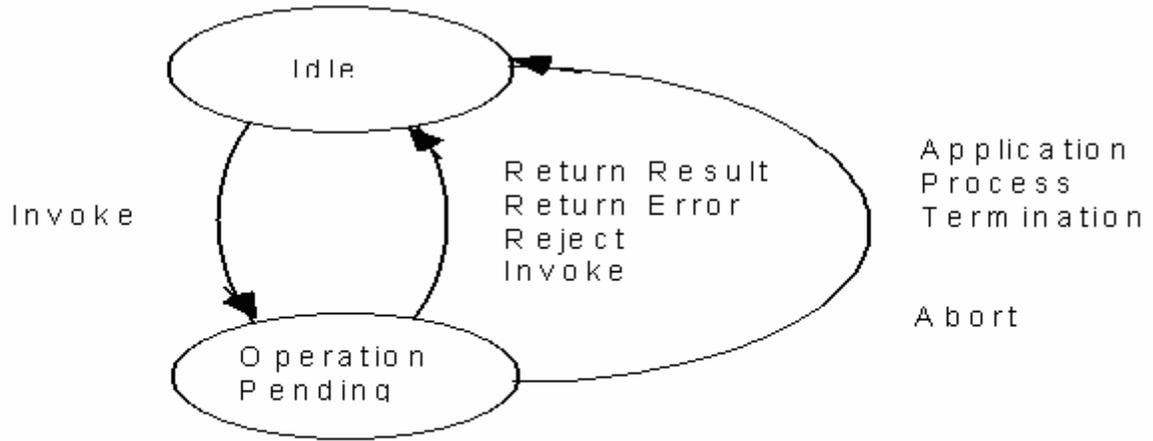


Figure 2/T1.114.1 - Identifying Layers and Services with Headers



NOTE - Abort terminates the Transaction

Figure 3/T1.114.1 - Example State Diagram

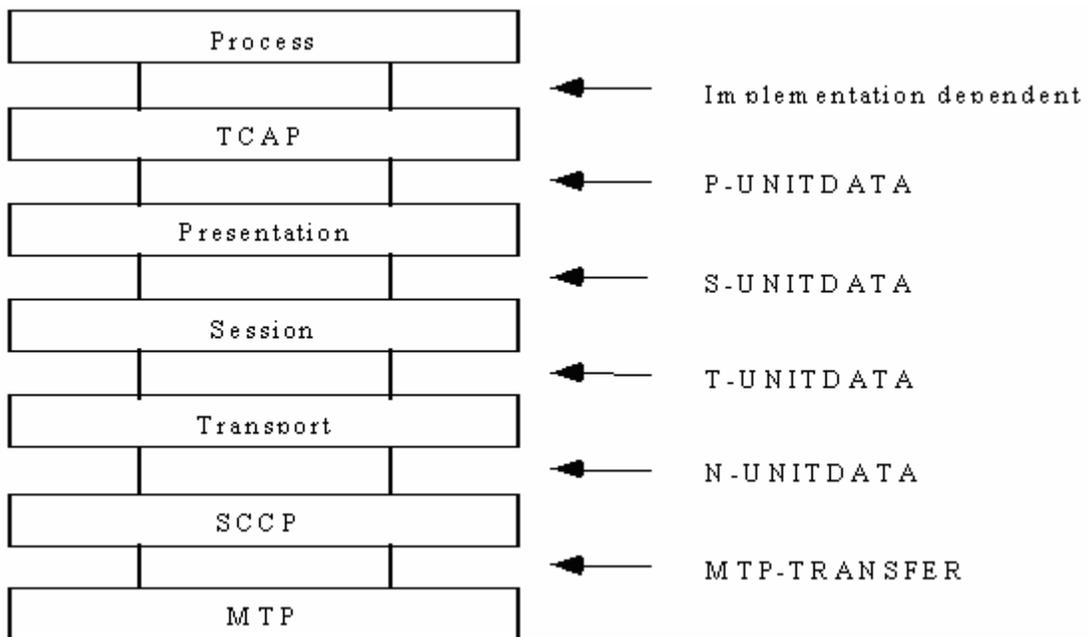


Figure 4/T1.114.1 - Connectionless Transaction Primitives

Chapter T1.114.2

Definition and Functions of Transaction Capabilities Messages

TABLE OF CONTENTS

1 SCOPE, PURPOSE, AND APPLICATION	1
2 PROTOCOL MESSAGE REQUIREMENTS.....	1
3 TRANSACTION PORTION	1
3.1 PACKAGE TYPE IDENTIFIER	1
3.2 TOTAL TCAP MESSAGE LENGTH.....	2
3.3 TRANSACTION ID IDENTIFIER.....	2
3.4 TRANSACTION ID LENGTH	2
3.5 TRANSACTION ID	2
3.5.1 <i>Originating Transaction ID</i>	2
3.5.2 <i>Responding Transaction ID</i>	2
3.6 P-ABORT CAUSE IDENTIFIER	2
3.7 P-ABORT CAUSE LENGTH	2
3.8 P-ABORT CAUSE	3
3.9 USER ABORT INFORMATION IDENTIFIER	3
3.10 USER ABORT INFORMATION LENGTH	3
3.11 USER ABORT INFORMATION	3
3.12 COMPONENT SEQUENCE IDENTIFIER	3
3.13 COMPONENT SEQUENCE LENGTH	3
4 DIALOGUE PORTION.....	4
4.1 DIALOGUE PORTION IDENTIFIER	4
4.2 DIALOGUE PORTION LENGTH	4
4.3 PROTOCOL VERSION IDENTIFIER	4
4.4 PROTOCOL VERSION LENGTH.....	4
4.5 PROTOCOL VERSION	4
4.6 APPLICATION CONTEXT IDENTIFIER	4
4.7 APPLICATION CONTEXT LENGTH.....	4
4.8 USER INFORMATION IDENTIFIER	4
4.9 USER INFORMATION LENGTH	4
4.10 SECURITY CONTEXT IDENTIFIER.....	5
4.11 SECURITY CONTEXT LENGTH	5
4.12 CONFIDENTIALITY IDENTIFIER.....	5
4.13 CONFIDENTIALITY LENGTH	5
5 COMPONENT PORTION	5
5.1 COMPONENT SEQUENCE IDENTIFIER	5
5.2 COMPONENT SEQUENCE LENGTH	5
5.3 COMPONENT TYPE IDENTIFIER	5
5.4 COMPONENT LENGTH.....	6
5.5 COMPONENT ID IDENTIFIER	6
5.6 COMPONENT ID LENGTH.....	6
5.7 COMPONENT ID	6
5.7.1 <i>Invoke ID</i>	6
5.7.2 <i>Correlation ID</i>	6
5.8 OPERATION CODE IDENTIFIER	6
5.8.1 <i>National</i>	6
5.8.2 <i>Private</i>	6
5.9 OPERATION CODE LENGTH	7
5.10 OPERATION CODE	7
5.11 ERROR CODE IDENTIFIER	7
5.11.1 <i>National</i>	7
5.11.2 <i>Private</i>	7
5.12 ERROR CODE LENGTH.....	7
5.13 ERROR CODE	7

Chapter T1.114.2

5.14	PROBLEM CODE IDENTIFIER	7
5.15	PROBLEM CODE LENGTH.....	7
5.16	PROBLEM CODE	8
5.16.1	<i>General</i>	8
5.16.2	<i>Invoke</i>	8
5.16.3	<i>Return Result</i>	8
5.16.4	<i>Return Error</i>	8
5.16.5	<i>Transaction Portion</i>	9
5.17	PARAMETER SET IDENTIFIER	9
5.18	PARAMETER SET LENGTH	9
5.19	PARAMETER SEQUENCE IDENTIFIER.....	9
5.20	PARAMETER SEQUENCE LENGTH.....	9
6	PARAMETERS	9

Definition and Functions of Transaction Capabilities Messages

1 SCOPE, PURPOSE, AND APPLICATION

This chapter describes the elements and functions of the signalling information used by the Transaction Capabilities protocol.

This chapter may contain requirements that reference other American National Standards. If so, when the American National Standards referenced in the requirements are superseded by revisions, the revisions shall apply.

2 PROTOCOL MESSAGE REQUIREMENTS

The Transaction Capabilities protocol format is separated into three portions, namely: 1) Transaction; 2) Dialogue; and 3) Component. The Transaction Portion identifies whether the Transaction Capabilities Application Part (TCAP) transaction is expected to consist of single or multiple messages and provides a means to associate these messages with a specific Application Process transaction.

The Component Portion consists of zero, one, or more Components as specified in Chapter T1.114.3.

The Dialogue Portion is used to pass information related to the version of TC, the application context, and Security as specified in Chapter T1.114.3.

The definitions and encodings for the Operations, Parameters, and Error Code elements associated with the Transaction Capabilities protocol are specified in Chapter T1.114.5.

The first field of every element encoded in accordance with CCITT Recommendation X.409-1984 contains a data type identifier. The general format for naming this field is:

```
element_name "IDENTIFIER"
```

When the element-name itself contains the word "identifier" or "ID" (e.g., Transaction ID) this rule will result in a field name that repeats the word "identifier" (e.g., Transaction ID Identifier).

3 TRANSACTION PORTION

3.1 Package Type Identifier

The messages required for TCAP interactions between two signalling nodes are identified in the Package Type Identifier and are as follows:

- (1) *Unidirectional*. A message with this Package Type sends information in one direction only with no reply expected. No TCAP Transaction is established.
- (2) *Query With Permission*. A message with this Package Type initiates a TCAP transaction and informs the destination node (i.e., the node that receives the message) that it may end the TCAP transaction.
- (3) *Query Without Permission*. A message with this Package Type initiates a TCAP transaction and informs the destination node that it may not end the TCAP transaction.
- (4) *Response*. A message with this Package Type ends the TCAP transaction.

- (5) *Conversation With Permission*. A message with this Package Type is the continuation of a TCAP transaction and informs the destination node that it may end the TCAP transaction.
- (6) *Conversation Without Permission*. A message with this Package Type is the continuation of a TCAP transaction and informs the destination node that it may not end the TCAP transaction.
- (7) *Abort*. A message with this Package Type informs the destination node that the sending node has terminated an established TCAP transaction without sending any pending components that may be expected due to a prior message.

3.2 *Total TCAP Message Length*

This is the total octet length of the overall TCAP message. It does not include itself or the Package Type Identifier.

3.3 *Transaction ID Identifier*

This indicates that the Transaction ID(s) follows.

3.4 *Transaction ID Length*

This is the total octet length which is required by the Transaction IDs in the TCAP message. It is the combined length of the Originating and Responding Transaction IDs when present. It does not include itself or the Transaction ID Identifier.

3.5 *Transaction ID*

A Transaction ID is a reference identifier which is used to associate all TCAP messages within an Application Process transaction. It is of local significance only.

3.5.1 **Originating Transaction ID**

This is the Transaction ID assigned by the originator of the message in which it appears..

3.5.2 **Responding Transaction ID**

This is the Originating Transaction ID that was received in the message for which the response is generated.

3.6 *P-Abort Cause Identifier*

This indicates that the P-Abort Cause follows.

3.7 *P-Abort Cause Length*

This is the total octet length of the P-Abort Cause data. It is always one octet. It does not include itself or the P-Abort Cause Identifier.

3.8 *P-Abort Cause*

This indicates the cause for an Abort message, such as:

- 1) *Unrecognized Package Type*. The Package Type is not one of those defined in clauses 3.1 (1) to 3.1 (7).
- 2) *Incorrect (Mistyped) Transaction Portion*. An unexpected or undefined identifier was received within the Transaction Portion.
- 3) *Badly Structured Transaction Portion*. A fundamental encoding problem (e.g., bad length).
- 4) *Unassigned Responding Transaction ID*. The received Transaction ID does not reflect a transaction currently in progress.
- 5) *Permission to Release Problem*. This problem is not delineated.
- 6) *Resource Unavailable*. Sufficient resources are not available at the Transaction sub-layer to establish a transaction.
- 7) *Unrecognized Dialogue Portion ID*. The identifier following the last field of the transaction portion was neither the dialogue portion identifier (4.1) nor the component sequence identifier (5.1).
- 8) *Badly Structured Dialogue Portion*. A fundamental encoding problem (e.g., multiple application contexts in one dialogue portion).
- 9) *Missing Dialogue Portion*. The dialogue portion is missing where it is mandatory (e.g., in the first backward message after a Query with a proposed application context).
- 10) *Inconsistent Dialogue Portion*. The contents of the Dialogue portion are inconsistent with the state of the transaction (e.g., a Query with an empty Dialogue portion or a message after the first backward Conversation that contains an application context).

3.9 *User Abort Information Identifier*

This indicates that the User Abort Information follows.

3.10 *User Abort Information Length*

This is the total octet length of the User Abort Information. It does not include itself or the User Abort Information Identifier.

3.11 *User Abort Information*

This indicates User Specified Information supplied by the TC-user when it aborts a transaction.

3.12 *Component Sequence Identifier*

Information transferred to clause 5.1.

3.13 *Component Sequence Length*

Information transferred to clause 5.2.

4 DIALOGUE PORTION

4.1 *Dialogue Portion Identifier*

This indicates that a Dialogue Portion follows.

4.2 *Dialogue Portion Length*

This is the total octet length of the Dialogue Portion. It does not include itself or the Dialogue Portion Identifier.

4.3 *Protocol Version Identifier*

This indicates that the Protocol Version follows.

4.4 *Protocol Version Length*

This is the total octet length of the Protocol Version. The protocol version is always one octet. It does not include itself or the Protocol Version Identifier.

4.5 *Protocol Version*

This indicates the version of T1.114 that is being used to encode SS7 messages, such as:

- (1) T1.114-1996.

4.6 *Application Context Identifier*

This indicates that an Application Context Name follows.

4.7 *Application Context Length*

This is the total octet length of the Application Context Name. It does not include itself or the Application Context Identifier.

4.8 *User Information Identifier*

The user information that follows this identifier corresponds to any information exchanged between two TC users. The meaning of user information depends on the Application Context Name that accompanies it or is in place during its use. For example, the user information may be used to carry information that further refines the application context by providing the "versions" of the ASEs that are referenced, "initialization" information on the ASEs etc. Its meaning and use are therefore outside the scope of this standard.

4.9 *User Information Length*

This is the total octet length of the User Information. It does not include itself or the User Information Identifier.

4.10 Security Context Identifier

This indicates that a Security Context follows to determine the appropriate interpretation of security information.

4.11 Security Context Length

This is the total octet length of the Security Context. It does not include itself or the Security Context Identifier.

4.12 Confidentiality Identifier

This indicates that Confidentiality Information follows to be used to perform the necessary confidentiality function(s). The information may contain a confidentiality algorithm ID, which identifies the appropriate confidentiality algorithm. It may also contain additional information needed to complete the specified confidentiality function.

4.13 Confidentiality Length

This is the total octet length of the Confidentiality Information. It does not include itself or the Confidentiality Identifier.

5 COMPONENT PORTION

5.1 Component Sequence Identifier

This indicates that a sequence of one or more Components follows and they are to be processed in the order received.

5.2 Component Sequence Length

This is the total octet length of the Component(s) contained in the Component Portion of the TCAP message. It does not include itself or the Component Sequence Identifier.

5.3 Component Type Identifier

The Component Types that can be used are as follows:

- (1) *Invoke (Last)*. This is used to invoke an operation (such as requesting a database to perform digit translation). When the Invoke Component contains a Correlation ID, "Last" indicates no further responding Components. When the Invoke Component does not contain a Correlation ID, it is always coded "Last".
- (2) *Return Result (Last)*. This is used to return the results of an invoked operation. "Last" indicates that no further responding Components are expected.
- (3) *Return Error*. This reports the unsuccessful completion of an invoked operation.

- (4) *Reject*. This reports the receipt and rejection of an incorrect Package or Component other than another Reject Component.
- (5) *Invoke (Not Last)*. This is similar to the Invoke described in Item (1), except that further responding Components are expected.
- (6) *Return Result (Not Last)*. This is similar to the Return Result described in Item (2), except that further responding Components are expected.

5.4 *Component Length*

This is the total octet length of the Component specified in the Component Type Identifier. It does not include itself or the Component Type Identifier.

5.5 *Component ID Identifier*

This indicates that the Component ID(s) follow.

5.6 *Component ID Length*

This is the total octet length of the Component IDs when present. It does not include itself or the Component ID Identifier.

5.7 *Component ID*

A Component ID is a reference identifier which labels the Component within a TCAP transaction. These allow correlation of Invokes and Responses.

5.7.1 **Invoke ID**

This is the component ID assigned by the originator.

5.7.2 **Correlation ID**

This is the Invoke ID that was received in the Component for which the response is generated.

5.8 *Operation Code Identifier*

This indicates that the Operation Code follows.

5.8.1 **National**

This indicates that the Operation Code is defined in this series of standards.

5.8.2 **Private**

This indicates that the Operation Code is defined within network specific TCAP applications. The structure of the Operation Code is not constrained by National TCAP.

5.9 Operation Code Length

This is the total octet length of the Operation Code. It does not include itself or the Operation Code Identifier.

5.10 Operation Code

The operation codes are application specific information carried unexamined by TCAP. Operation codes used by TC-users are provided in Chapter T1.114.5.

5.11 Error Code Identifier

This indicates that the Error Code follows.

5.11.1 National

This indicates that the Error Code is defined in this series of standards.

5.11.2 Private

This indicates that the Error Code is defined within network specific TCAP applications. The structure of the Error Code is not constrained by National TCAP.

5.12 Error Code Length

This is the total octet length of the Error Code associated with a failed operation. It does not include itself or the Error Code Identifier.

5.13 Error Code

This provides the reason why a specific operation could not be successfully completed. The error codes are application specific information carried unexamined by TCAP. Error codes used by TC-users are provided in Chapter T1.114.5.

5.14 Problem Code Identifier

This indicates the class of problem that caused a Component or Transaction Portion to be rejected.

5.15 Problem Code Length

This is the total octet length of the Problem Code associated with the rejection of a Component or Transaction Portion. It does not include itself or the Problem Code Identifier.

5.16 Problem Code

This indicates the specific reason why a Component or Transaction Portion had to be rejected. The Problem Code is partitioned into a Problem Type followed by a Problem Specific associated with each Problem Type. The Problem Types and Specifiers for which a Component or Transaction Portion is rejected are classified as follows:

5.16.1 General

This describes a problem of a general nature, such as:

- (1) *Unrecognized Component*. The Component Type has not been defined.
- (2) *Incorrect Component Portion*. An unexpected or undefined identifier within the Component Portion.
- (3) *Badly Structured Component Portion*. A fundamental encoding problem (e.g., bad length).
- (4) *Incorrect Component Coding*. General encoding problems not covered under Items (1) to (3).

5.16.2 Invoke

This describes a problem specific to an Invoke Component, such as:

- (1) *Duplicate Invoke ID*. An Invoke ID is received which has already become assigned to another operation in progress.
- (2) *Unrecognized Operation Code*. The operation has not been defined by the Application Process.
- (3) *Incorrect Parameter*. An unexpected or undefined Parameter was received.
- (4) *Unrecognized Correlation ID*. The received Correlation ID does not reflect an operation that is currently in progress.

5.16.3 Return Result

This describes a problem specific to a Return Result Component, such as:

- (1) *Unassigned Correlation ID*. The received Correlation ID does not reflect an operation that is currently in progress.
- (2) *Unexpected Return Result*. The invoked operation does not report success.
- (3) *Incorrect Parameter*. An unexpected or undefined Parameter (result) was received.

5.16.4 Return Error

This describes a problem specific to a Return Error Component, such as:

- (1) *Unassigned Correlation ID*. The received Correlation ID does not reflect an operation that is currently in progress.
- (2) *Unexpected Return Error*. The Return Error Component did not report failure of the invoked operation.
- (3) *Unrecognized Error*. The reported error has not been defined by the Application Process.
- (4) *Unexpected Error*. The reported error is not applicable to the invoked operation.
- (5) *Incorrect Parameter*. An unexpected or undefined Parameter was received.

5.16.5 Transaction Portion¹

This describes a problem specific to the Transaction Portion, such as:

- (1) *Unrecognized Package Type*. The Package Type has not been defined.
- (2) *Incorrect Transaction Portion*. An unexpected or undefined identifier was received within the Transaction Portion.
- (3) *Badly Structured Transaction Portion*. A fundamental encoding problem (e.g., bad length).
- (4) *Unassigned Responding Transaction ID*. The received Transaction ID is derivable but does not reflect a transaction currently in progress.
- (5) *Permission to Release Problem*. This problem is not delineated.
- (6) *Resource Unavailable*. Sufficient resources are not available at the Transaction sub-layer to establish a transaction.

5.17 Parameter Set Identifier

This indicates that a Parameter Set follows -- e.g., parameters may be provided and processed in an unspecified order.

5.18 Parameter Set Length

This is the total octet length of the Parameter Set. It does not include itself or the Parameter Set Identifier.

5.19 Parameter Sequence Identifier

This indicates that a Parameter Sequence follows -- i.e., parameters are provided and processed in a specified order.

5.20 Parameter Sequence Length

This is the total octet length of the Parameter Sequence. It does not include itself or the Parameter Sequence Identifier.

6 PARAMETERS

The Parameter Identifier unambiguously identifies a specific parameter. The Parameter Length provides the total octet length of the parameter but does not include itself or the Parameter Identifier. The contents are the actual value of the parameter. The parameter identifiers and contents are application specific information carried unexamined by TCAP.

¹ Applications using T1.114-1988 report Transaction Portion problems using a Reject component. It is preferred that other applications report these problems using the Abort package type.

Chapter T1.114.3
TC Formats and Codes

TABLE OF CONTENTS

1 SCOPE, PURPOSE, AND APPLICATION	1
2 DATA ELEMENT ENCODING	1
2.1 IDENTIFIER.....	1
2.1.1 <i>Universal, Application-wide and Context-specific Classes</i>	2
2.1.1.1 <i>Low Identifier Codes</i>	2
2.1.1.2 <i>High Identifier Codes</i>	2
2.1.2 <i>Private Use Class</i>	2
2.1.2.1 <i>Low Identifier Codes</i>	2
2.1.2.2 <i>High Identifier Codes</i>	2
2.2 LENGTH OF CONTENTS	4
2.3 TCAP MESSAGE STRUCTURE.....	4
3 TRANSACTION PORTION	12
3.1 PACKAGE TYPE IDENTIFIER	12
3.2 TOTAL TCAP MESSAGE LENGTH.....	12
3.3 TRANSACTION ID IDENTIFIER.....	12
3.4 TRANSACTION ID LENGTH	12
3.5 TRANSACTION IDS.....	12
3.5.1 <i>Originating Transaction ID</i>	13
3.5.2 <i>Responding Transaction ID</i>	13
3.6 P-ABORT CAUSE IDENTIFIER	13
3.7 P-ABORT CAUSE LENGTH	13
3.8 P-ABORT CAUSE.....	13
3.9 USER ABORT INFORMATION IDENTIFIER	14
3.10 USER ABORT INFORMATION LENGTH	14
3.11 USER ABORT INFORMATION	14
4 DIALOGUE PORTION.....	14
4.1 DIALOGUE PORTION IDENTIFIER.....	14
4.2 DIALOGUE PORTION LENGTH	15
4.3 PROTOCOL VERSION IDENTIFIER	15
4.4 PROTOCOL VERSION LENGTH.....	15
4.5 PROTOCOL VERSION	15
4.6 APPLICATION CONTEXT IDENTIFIER	15
4.7 APPLICATION CONTEXT LENGTH.....	15
4.8 USER INFORMATION IDENTIFIER.....	15
4.9 USER INFORMATION LENGTH	16
4.10 USER INFORMATION.....	16
4.10.1 <i>External Identifier</i>	16
4.10.2 <i>External Length</i>	16
4.10.3 <i>Direct Reference Identifier</i>	16
4.10.4 <i>Direct Reference Length</i>	16
4.10.5 <i>Direct Reference Content</i>	16
4.10.6 <i>Indirect Reference Identifier</i>	17
4.10.7 <i>Indirect Reference Length</i>	17
4.10.8 <i>Indirect Reference Contents</i>	17
4.10.9 <i>Data Value Descriptor Identifier</i>	17
4.10.10 <i>Data Value Descriptor Length</i>	17
4.10.11 <i>Data Value Descriptor Contents</i>	17
4.10.12 <i>Encoding Identifier</i>	17
4.10.13 <i>Encoding Length</i>	18
4.10.14 <i>Encoding Content</i>	18
4.11 SECURITY CONTEXT IDENTIFIER.....	18
4.12 SECURITY CONTEXT LENGTH	19
4.13 CONFIDENTIALITY IDENTIFIER.....	19

4.14 CONFIDENTIALITY LENGTH	19
4.15 CONFIDENTIALITY INFORMATION	19
4.15.1 Confidentiality Algorithm ID Identifier.....	19
4.15.2 Confidentiality Algorithm ID Length.....	19
4.15.3 Confidentiality Algorithm ID.....	20
4.15.4 Confidentiality Value Identifier.....	20
4.15.5 Confidentiality Value	20
5 COMPONENT PORTION	20
5.1 COMPONENT SEQUENCE IDENTIFIER.....	20
5.2 COMPONENT SEQUENCE LENGTH	20
5.3 COMPONENT TYPE IDENTIFIER	20
5.4 COMPONENT LENGTH.....	21
5.5 COMPONENT ID IDENTIFIER	21
5.6 COMPONENT ID LENGTH.....	21
5.7 COMPONENT IDS	21
5.7.1 Invoke ID.....	22
5.7.2 Correlation ID.....	22
5.8 OPERATION CODE IDENTIFIER	22
5.9 OPERATION CODE LENGTH	22
5.10 OPERATION CODE	22
5.11 ERROR CODE IDENTIFIER	22
5.12 ERROR CODE LENGTH.....	23
5.13 ERROR CODE	23
5.14 PROBLEM CODE IDENTIFIER	23
5.15 PROBLEM CODE LENGTH.....	23
5.16 PROBLEM CODE	23
5.16.1 Problem Type.....	23
5.16.2 Problem Specifier.....	24
5.17 PARAMETER SET IDENTIFIER.....	24
5.18 PARAMETER SET LENGTH	25
5.19 PARAMETER SEQUENCE IDENTIFIER.....	25
5.20 PARAMETER SEQUENCE LENGTH.....	25
6 PARAMETERS	25
7 SUMMARY OF IDENTIFIERS.....	25
A DESCRIPTION OF TCAP IN ASN.1	28

TABLE OF FIGURES

FIGURE 1/T1.114.3 - BIT ORDER	1
FIGURE 2A/T1.114.3 - UNIVERSAL, APPLICATION-WIDE, CONTEXT-SPECIFIC AND PRIVATE USE CLASS	3
FIGURE 2B/T1.114.3 - PRIVATE USE CLASS	3
FIGURE 3/T1.114.3 - TCAP MESSAGE STRUCTURE	4
FIGURE 4/T1.114.3 - DETAILED TCAP MESSAGE STRUCTURE WITH AN INVOKE COMPONENT	10
FIGURE 5/T1.114.3 - DETAILED TCAP DIALOGUE PORTION STRUCTURE	11
FIGURE 6/T1.114.3 - DETAILED TCAP PARAMETER SET/SEQUENCE STRUCTURE.....	11

TC Formats and Codes

1 SCOPE, PURPOSE, AND APPLICATION

This chapter provides the formats and encoding of the messages used in SS7 Transaction Capabilities. It encompasses the various elements associated with the Transaction Portion and Component Portion of the TCAP message. These include identifiers and lengths. It is based on the encoding rules provided in CCITT (currently ITU-T) Red Book Recommendation X.409-1984¹ with the extension to that encoding of CCITT (currently ITU-T) Blue Book Recommendation X.209. Use of the formal description language to specify this chapter is provided in Informative Annex A to this chapter.

2 DATA ELEMENT ENCODING

Each data element in a TCAP message is encoded using a sequence of octets logically divided into an Identifier, Length of Contents, and Contents. The Identifier distinguishes one type from another and governs the interpretation of the Contents. The length specifies the length of the Contents. The Contents is the substance of the element, containing the primary information the element is intended to convey. The specification of the Contents is an integral part of this standard. Therefore, only those fields and values specified in this chapter may be used as part of the standard message set.

Bits are labeled as follows, with Bit A the least significant and the first transmitted:

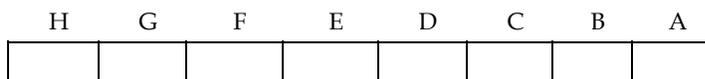


Figure 1/T1.114.3 - Bit Order

2.1 Identifier

All Identifiers use the two most significant bits to indicate the Identifier class. These bits are coded as follows:

Class	Bit Assignment (Bits HG)	Usage in this Specification
Universal	00	Universal
Application-wide	01	International TCAP
Context-specific	10	Context Specific
Private Use	11	National TCAP/Private TCAP

The Universal class is used for Identifiers that are standardized in the CCITT (currently ITU-T) Red Book Recommendation X.409-1984 and are application-independent types. Application-wide identifiers are standardized for a particular entity. For Transaction, Dialogue, and Component Portion Identifiers the bit assignment 01 will refer to the international standardized TCAP. Context-specific Identifiers are defined within an application and are distinct within a limited context (e.g., a particular set or sequence type). The Private Use class is reserved for private use. For Transaction, Dialogue, and

¹ All CCITT (currently ITU-T) Recommendations may be available from the International Telecommunications Union. <<http://www.itu.int/ITU-T/>>.

Component Portion Identifiers in this standard, Private Use Identifiers are shared between this national (inter network) TCAP specification and private TCAP specifications. This chapter specifies the codings for national TCAP use. The national Identifier codes not assigned by this chapter are reserved for future use. Privately assigned Identifier codes must be specified in the context of a private class TCAP data type.

Certain Identifiers for parameters and operation and error codes are standardized in Chapter T1.114.5.

Bit F is used to indicate whether the data element is primitive or constructor as shown below. A primitive element is one whose structure is atomic (i.e., one value only). A constructor element is one whose content is one or more data elements, which may themselves be primitive or constructor data elements.

Form	Bit F
Primitive	0
Constructor	1

2.1.1 Universal, Application-wide and Context-specific Classes

2.1.1.1 Low Identifier Codes

Bits A to E of the Identifier octet represent an Identifier code that distinguishes one data type from another of the same class. Identifier codes in the range 00000 to 11110 are provided in one octet.

2.1.1.2 High Identifier Codes

The extension mechanism is achieved by coding bits A to E as 11111. If bit H of the extension octet is set to 0, then no further octets for this Identifier are used. All preceding Identifier extension octets must have bit H set to 1.

The Identifier code bits of the first and subsequent extension octets consist of bits A to G of each extension octet, where bit G is the most significant bit of each extension octet. The resulting Identifier code is formed by concatenating the constituent Identifier code bits of the extension octets, where the first extension octet is the most significant octet. Higher Identifier code values are coded using the minimum number of extension octets.

2.1.2 Private Use Class

2.1.2.1 Low Identifier Codes

When the private use class is specified, bits A to E of the first Identifier octet are reserved for national TCAP use only. Identifier codes in the range 00000 to 11110 are provided in one octet.

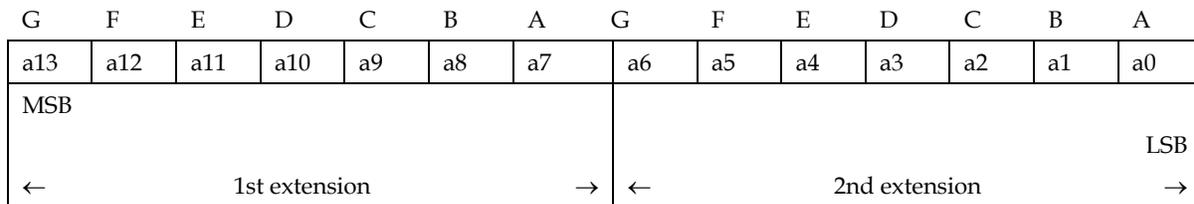
2.1.2.2 High Identifier Codes

The extension mechanism is achieved by coding bits A to E as 11111. If bit H of the extension is set to 0, then no further octets for this Identifier are used. All preceding Identifier extension octets must have bit H set to 1. The extended formats are shared between Private TCAP and National TCAP usage. If bit G of the first extension octet is set to 0, a nationally assigned Identifier is implied. If bit G is set to 1, a privately assigned Identifier is assigned.

The Identifier code bits of the first and subsequent extension octets consist of bits A to G of each extension octet, where bit G is the most significant bit of each extension octet. The resulting Identifier code is formed by concatenating the constituent Identifier code bits of the extension octets, where the first extension octet is the most significant octet. Higher Identifier code values are coded using the minimum number of extension octets.

Figure 2A/T1.114.3 shows an example of the use of the extension mechanism.

H	G	F	E	D	C	B	A								
CLASS								0	1	1	1	1	1	1	First Octet
1	a13	a12	a11	a10	a9	a8	a7	Second Octet (1st extension used octet)							
0	a6	a5	a4	a3	a2	a1	a0	Third Octet (2nd extension used octet)							



NOTE - In the Private Use class, a13 has a special meaning. In the Private Use class, a13 equal to 0 implies a nationally assigned Identifier and a13 equal to 1 implies a privately assigned Identifier.

Figure 2A/T1.114.3 - Universal, Application-wide, Context-specific and Private Use Class

It should be noted the role bit G plays in the second octet of a Private Use class TCAP Identifier is to divide the Identifier codes into privately assigned and nationally assigned values. Figure 2B/T1.114.3 shows an example of this

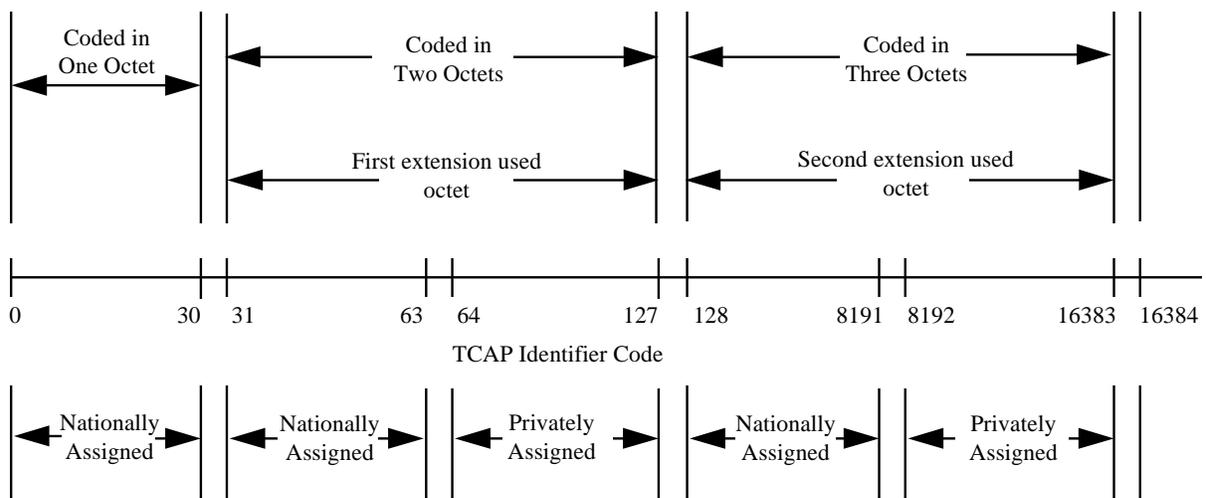


Figure 2B/T1.114.3 - Private Use Class

2.2 Length Of Contents

The Length of Contents field is coded to indicate the number of octets in the contents. The length does not include the Identifier or the Length of Contents.

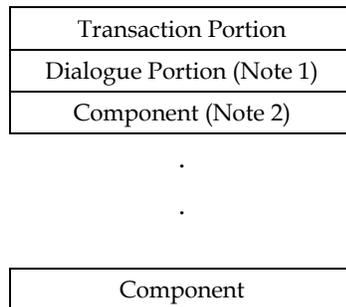
When the contents are less than 128 octets long, lengths are encoded as a binary number using Bits A to G. Bit H is reserved and is coded 0.

When the contents are longer than 127 octets, the long form of the Length of Contents is used. The long form is from 2 to 127 octets long. Bit H of the first octet has the value 1 Bits A to G of the first octet encode a number one less than the size of the Length of Contents field in octets as an unsigned binary number whose most significant (MSB) and least significant (LSB) bit are Bit G and Bit A respectively. The length itself is encoded as an unsigned binary number whose MSB and LSB are Bit H of the second octet and Bit A of the last octet, respectively. This binary number shall be encoded in the fewest possible octets, with no leading octets having the value 0.

The maximum value that may be encoded is constrained by the network (e.g., MTP and SCCP) message size limitations.

2.3 TCAP Message Structure

A TCAP message includes a Transaction Portion. A TCAP message (except an Abort) also includes zero, one, or more Components (see Figure 3/T1.114.3).²



NOTE 1 - The Dialogue Portion is optional. Application Context in the Dialogue Portion may only appear in the first forward and first backward message of a transaction.

NOTE 2 - The Component Portion is optional. However, either the Dialogue Portion or the Component Portion (or both) must be present.

Figure 3/T1.114.3 - TCAP Message Structure

The seven Package Types, specified in clause 3 of this chapter, have the structure shown in the following tables. An M following the indicated data element indicates that its presence is mandatory and may not be null. An element marked by M* must be present, but its content may be empty (i.e., it may have a length equal to zero). An element marked by O is optional. Its content may be empty or it may be absent from the message.

² For a UNIDIRECTIONAL message, it includes one or more Components.

ATIS-1000114.2004 (S2019)

Chapter T1.114.3

UNIDIRECTIONAL	Mandatory Indication
Package Type Identifier Total TCAP Message Length	M
Transaction ID Identifier Transaction ID Length (= 0)	M*
Dialogue Portion	O
Component Sequence Identifier Component Sequence Length Components	M

QUERY WITH PERMISSION QUERY WITHOUT PERMISSION	Mandatory Indication
Package Type Identifier Total TCAP Message Length	M
Transaction ID Identifier Transaction ID Length Originating Transaction ID	M
Dialogue Portion	O
Component Sequence Identifier Component Sequence Length Components	O

RESPONSE	Mandatory Indication
Package Type Identifier Total TCAP Message Length	M
Transaction ID Identifier Transaction ID Length Responding Transaction ID	M
Dialogue Portion	O
Component Sequence Identifier Component Sequence Length Components	O

ATIS-1000114.2004 (S2019)

Chapter T1.114.3

CONVERSATION WITH PERMISSION CONVERSATION WITHOUT PERMISSION	Mandatory Indication
Package Type Identifier Total TCAP Message Length	M
Transaction ID Identifier Transaction ID Length Originating Transaction ID Responding Transaction ID	M
Dialogue Portion	O
Component Sequence Identifier Component Sequence Length Components	O

ABORT (P-Abort)	Mandatory Indication
Package Type Identifier Total TCAP Message Length	M
Transaction ID Identifier Transaction ID Length Responding Transaction ID	M
P-Abort Cause Identifier P-Abort Cause Length P-Abort Cause	M

ABORT (User Abort)	Mandatory Indication
Package Type Identifier Total TCAP Message Length	M
Transaction ID Identifier Transaction ID Length Responding Transaction ID	M
Dialogue Portion	O
User Abort Information Identifier User Abort Information Length User Abort Information	M*

The Dialogue Portion, specified in clause 4 of this chapter, has the following structure:

ATIS-1000114.2004 (S2019)

Chapter T1.114.3

DIALOGUE PORTION	Mandatory Indication
Dialogue Portion Identifier Dialogue Portion Length	M
Protocol Version Identifier Protocol Version Length Protocol Version	O
Application Context Identifier Application Context Length Application Context Name	O (Notes 1,2)
User Information Identifier User Information Length User Information	O
Security Context Identifier Security Context Length Security Context Information	O
Confidentiality Identifier Confidentiality Length Confidentiality Information	O

NOTE 1 - Application Context is allowed only in Unidirectional, Query, Abort (user abort), and first backward Conversation or Response messages.

NOTE 2 - The Application Context Name may be empty, or the same as the previously proposed Application Context Name, in a first backward Conversation or Response message when the previously proposed context is acceptable.

The Components type mentioned above are specified in clause 5.3 of this chapter. Each component has one of the structures shown in the following tables:

INVOKE COMPONENT	Mandatory Indication
Component Type Identifier Component Length	M
Component ID Identifier Component ID Length Component IDs	M*
Operation Code Identifier Operation Code Length Operation Code	M
Parameter Set/Sequence Identifier Parameter Set/Sequence Length Parameter Set/Sequence	M*

ATIS-1000114.2004 (S2019)

Chapter T1.114.3

RETURN RESULT COMPONENT	Mandatory Indication
Component Type Identifier Component Length	M
Component ID Identifier Component ID Length Component IDs	M*
Parameter Set/Sequence Identifier Parameter Set/Sequence Length Parameter Set/Sequence	M*

RETURN ERROR COMPONENT	Mandatory Indication
Component Type Identifier Component Length	M
Component ID Identifier Component ID Length Component IDs	M*
Error Code Identifier Error Code Length Error Code	M
Parameter Set/Sequence Identifier Parameter Set/Sequence Length Parameter Set/Sequence	M*

REJECT COMPONENT	Mandatory Indication
Component Type Identifier Component Length	M
Component ID Identifier Component ID Length Component IDs	M*
Problem Code Identifier Problem Code Length Problem Code	M
Parameter Set/Sequence Identifier Parameter Set/Sequence Length Parameter Set/Sequence	M*

The following table shows the structure of a single EXTERNAL item. The user information, if present, is composed of one or more such items.

ATIS-1000114.2004 (S2019)

Chapter T1.114.3

EXTERNAL	Mandatory Indication
External Identifier External Length	M
Direct Reference Identifier Direct Reference Length Direct Reference Content	M
Indirect Reference Identifier Indirect Reference Length Indirect Reference Content	O (Note 1)
Data Value Descriptor Identifier Data Value Descriptor Length Data Value Descriptor Content	O (Note 2)
Encoding Identifier Encoding Length Encoding Content	M

NOTE 1 - This field is included for completeness only. Since there is no presentation layer negotiation, the indirect reference cannot be used.

NOTE 2 - This field is included for completeness. TC based applications generally will not need to use this human-readable descriptor.

The following shows the structure of a single Parameter. A Parameter Set/Sequence is composed of zero or more Parameters.

PARAMETER	Mandatory Indication
Parameter Identifier Parameter Length Parameter Contents	O

The detailed message structure and Parameter Set/Sequence structure are shown in Figures 4/T1.114.3, 5/T1.114.3, and 6/T1.114.3.

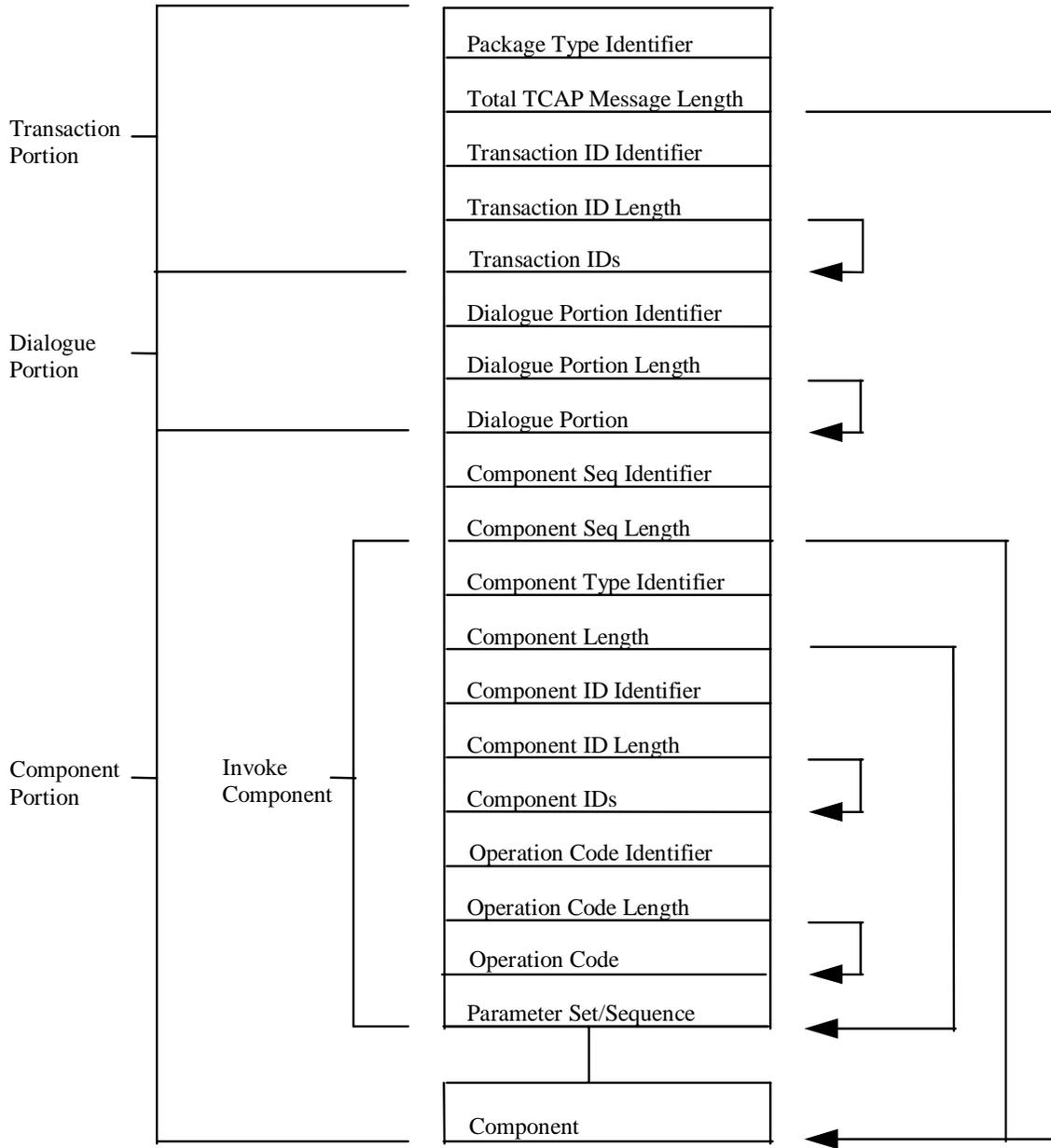


Figure 4/T1.114.3 - Detailed TCAP Message Structure With An Invoke Component

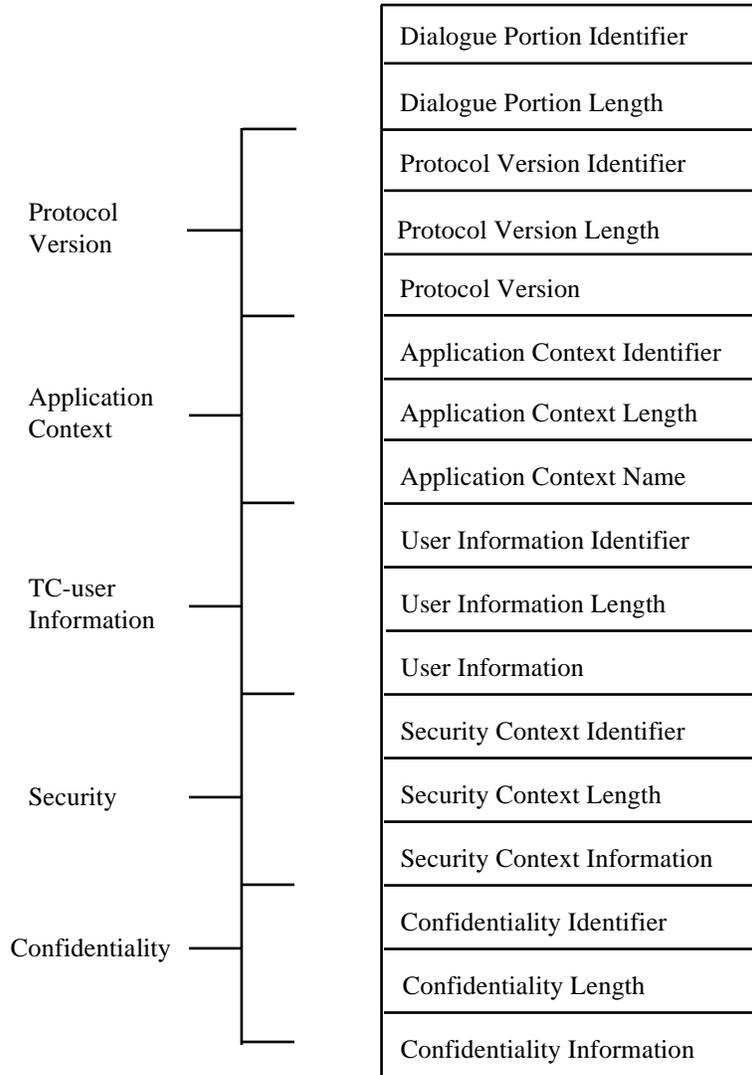


Figure 5/T1.114.3 - Detailed TCAP Dialogue Portion Structure

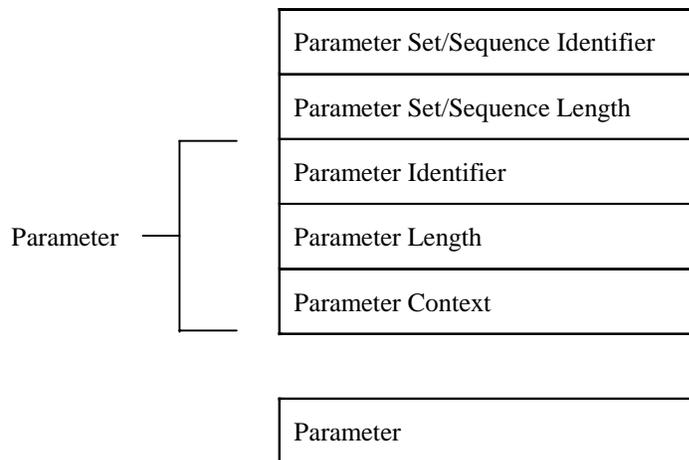


Figure 6/T1.114.3 - Detailed TCAP Parameter Set/Sequence Structure

3 TRANSACTION PORTION

3.1 Package Type Identifier

This field consists of one octet and is mandatory for all TCAP messages. Package Type Identifiers are coded national, constructor as follows:

Package Type Identifiers	H	G	F	E	D	C	B	A
Unidirectional	1	1	1	0	0	0	0	1
Query With Permission	1	1	1	0	0	0	1	0
Query Without Permission	1	1	1	0	0	0	1	1
Response	1	1	1	0	0	1	0	0
Conversation With Permission	1	1	1	0	0	1	0	1
Conversation Without Permission	1	1	1	0	0	1	1	0
Abort	1	1	1	1	0	1	1	0

3.2 Total TCAP Message Length

This length field indicates total message length.

3.3 Transaction ID Identifier

Transaction IDs are assigned to a TCAP message to permit transaction association. The Transaction ID Identifier is coded national, primitive with Identifier code 7, that is:

	H	G	F	E	D	C	B	A
Transaction ID Identifier	1	1	0	0	0	1	1	1

3.4 Transaction ID Length

Transaction ID length is the total length in octets used by the Transaction IDs in a TCAP message.

The value of this length is determined by the length of the Originating and Responding Transaction IDs combined. It may be 0, 4, or 8 octets. Only the Unidirectional package has a 0 octet Transaction ID.

3.5 Transaction IDs

Zero or more IDs may be required depending upon the Package Type Identifier used. The following table depicts this relationship:

Package Type Identifier	Originating ID	Responding ID
Unidirectional	No	No
Query With Permission	Yes	No
Query Without Permission	Yes	No
Response	No	Yes
Conversation With Permission	Yes	Yes
Conversation Without Permission	Yes	Yes
Abort	No	Yes

3.5.1 Originating Transaction ID

This field contains the Transaction ID assigned by the originator. When present, it consists of four octets and is the first of the Transaction IDs (when both an Originating and a Responding Transaction ID are present).

3.5.2 Responding Transaction ID

This field contains the Transaction ID assigned by the responder. The Responding Transaction ID is a reflection of the Originating Transaction ID and has the same length.

3.6 P-Abort Cause Identifier

This field identifies the P-Abort cause values and is coded national, primitive with Identifier code 23, that is:

	H	G	F	E	D	C	B	A
P-Abort Cause Identifier	1	1	0	1	0	1	1	1

3.7 P-Abort Cause Length

The P-Abort Cause length indicates the total length of the P-Abort Cause data. The P-Abort Cause is always one octet long.

3.8 P-Abort Cause

The P-Abort Cause values are coded as shown in the following table:

P-Abort Causes	H	G	F	E	D	C	B	A
Unrecognized Package Type	0	0	0	0	0	0	0	1
Incorrect Transaction Portion	0	0	0	0	0	0	1	0
Badly Structured Transaction Portion	0	0	0	0	0	0	1	1
Unassigned Responding Transaction ID	0	0	0	0	0	1	0	0
Permission To Release Problem	0	0	0	0	0	1	0	1
Resource Unavailable	0	0	0	0	0	1	1	0
Unrecognized Dialogue Portion ID	0	0	0	0	0	1	1	1
Badly Structured Dialogue Portion	0	0	0	0	1	0	0	0
Missing Dialogue Portion	0	0	0	0	1	0	0	1
Inconsistent Dialogue Portion	0	0	0	0	1	0	1	0

3.9 User Abort Information Identifier

This field identifies the User Abort Information and is coded national, with Identifier code 24. This identifier may be coded either primitive or constructor, depending on the format of the information provided by the user. The Dialogue Portion should be used to identify the TC-user providing the information.

	H	G	F	E	D	C	B	A
User Abort Information String	1	1	0/1	1	1	0	0	0

3.10 User Abort Information Length

The User Abort Information length indicates the total length of the User Abort Information data.

3.11 User Abort Information

The TC-user may provide any information desired as the contents of the User Abort information element.

4 DIALOGUE PORTION

4.1 Dialogue Portion Identifier

This field identifies the Dialogue Portion and is coded national, constructor with Identifier code 25, that is:

	H	G	F	E	D	C	B	A
Dialogue Portion Identifier	1	1	1	1	1	0	0	1

4.2 *Dialogue Portion Length*

The Dialogue Portion Length indicates the total length of the Dialogue Portion.

4.3 *Protocol Version Identifier*

This field identifies the version of T1.114 and is coded national, primitive with Identifier code 26, that is:

	H	G	F	E	D	C	B	A
Protocol Version Identifier	1	1	0	1	1	0	1	0

4.4 *Protocol Version Length*

The Protocol Version Length indicates the total length of the Protocol Version.

4.5 *Protocol Version*

Each supported protocol version is indicated by a value of one. The values are coded as follows: Each unsupported protocol version is indicated by a value of zero in its assigned bit.

H	G	F	E	D	C	B	A
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved T1.114-2004	T1.114-2000	T1.114-1996

4.6 *Application Context Identifier*

This field identifies an Application Context Name and annotates one of the following values:

	H	G	F	E	D	C	B	A
Integer Application Context Identifier	1	1	0	1	1	0	1	1
Object Identifier App. Context Identifier	1	1	0	1	1	1	0	0

4.7 *Application Context Length*

The Application Context Length indicates the total length of the Application Context Name.

4.8 *User Information Identifier*

The TC-user may provide any information desired as the contents of the User Information element. It is a sequence of external elements as defined in CCITT (currently ITU-T) X.208-1988. This identifier is coded national, constructor with Identifier code 29, that is:

	H	G	F	E	D	C	B	A
User Information Identifier	1	1	1	1	1	1	0	1

4.9 User Information Length

The User Information Length indicates the total length of the user information in the Dialogue Portion.

4.10 User Information

The user information is a sequence of elements having the EXTERNAL data type, as defined in CCITT (currently ITU-T) X.208-1988. The TC-user defines the syntax and content for each external element. Each external element contains at least a direct reference defining the syntax of the element and an encoding of the element contents.

4.10.1 External Identifier

External Identifier is coded universal, constructor, with identifier code 8, that is:

	H	G	F	E	D	C	B	A
External Identifier	0	0	1	0	1	0	0	0

4.10.2 External Length

This field specifies the number of octets in the Direct Reference, Indirect Reference, Data Value Description, and Encoding Information.

4.10.3 Direct Reference Identifier

The direct reference identifier is coded universal primitive with identifier code 6 -- i.e., an object identifier.

	H	G	F	E	D	C	B	A
Direct Reference Identifier	0	0	0	0	0	1	1	0

4.10.4 Direct Reference Length

This field specifies the number of octets in the object identifier encoding.

4.10.5 Direct Reference Content

The content of the direct reference is an object identifier coded as specified in CCITT (currently ITU-T) X.209-1988.

4.10.6 Indirect Reference Identifier

The indirect reference identifier is coded universal primitive with an identifier of 2 -- i.e., an integer.

	H	G	F	E	D	C	B	A
Indirect Reference Identifier	0	0	0	0	0	0	1	0

4.10.7 Indirect Reference Length

The indirect reference length is the number of octets needed to encode the integer.

4.10.8 Indirect Reference Contents

The indirect reference content is an integer identifier assigned to identify an abstract syntax during presentation layer negotiation. Since there is no presentation layer defined for TC, this element is not applicable for TC-use, but is included for completeness.

4.10.9 Data Value Descriptor Identifier

The data value descriptor identifier is coded universal primitive with an identifier of 7 -- i.e., an object descriptor.

	H	G	F	E	D	C	B	A
Data Value Descriptor Identifier	0	0	0	0	0	1	1	1

4.10.10 Data Value Descriptor Length

The data value descriptor length is the number of octets in the graphic character string making up the descriptor.

4.10.11 Data Value Descriptor Contents

The content of a data value descriptor is a human-readable string of characters identifying the object contained in the external element. For TC-based applications, where there is little or no human interaction, this element will normally be omitted.

4.10.12 Encoding Identifier

If the content of the external element is the value of a single ASN.1 data type and if the encoding rules for this type are as specified in this chapter of T1.114 then the sending implementation may use any of the encoding choices:

- ◆ single-ASN1-type
- ◆ octet-aligned
- ◆ arbitrary

as an implementation option.

If the content of the external element, using the agreed encoding, is an integral number of octets, then the sending implementation may use any of the encoding choices:

- ◆ octet-aligned
- ◆ arbitrary

as an implementation option.

If the content of the external element, using the agreed encoding, is not an integral number of octets, the encoding choice shall be:

- ◆ arbitrary.

The identifiers for the three encoding choices are coded context-specific with identifiers 0, 1, and 2 respectively

	H	G	F	E	D	C	B	A
single-ASN1-type Identifier	1	0	Note	0	0	0	0	0
octet-aligned Identifier	1	0	0	0	0	0	0	1
arbitrary Identifier	1	0	0	0	0	0	1	0

NOTE - The value of this bit will be primitive (0) or constructor (1) depending on whether the ASN.1 type encoded is a primitive or constructor value.

4.10.13 Encoding Length

The encoding length is the number of octets in the encoding contents.

4.10.14 Encoding Content

If the encoding content identifier specifies a single-ASN1-type, then the encoding content is the encoding of the indicated type according to the rules of this chapter of T1.114.

If the encoding content identifier specifies an octet-aligned type, then the encoding content is the sequence of octets derived by applying the agreed upon encoding rules to the user provided data value.

If the encoding contents identifier specifies an arbitrary type, then the encoding content is derived by applying the agreed upon encoding rules to the user provided data value to generate a sequence of bits. This bit sequence is then encoded as an IMPLICIT BITSTRING, according to the rules of CCITT (currently ITU-T) X.209-1988.

4.11 Security Context Identifier

This identifier is coded context specific (in the context of the dialogue portion sequence), primitive, and annotates one of the following values:

	H	G	F	E	D	C	B	A
Integer Security Context	1	0	0	0	0	0	0	0
Object Identifier Context	1	0	0	0	0	0	0	1

The Security Context determines how other security information should be interpreted.

4.12 Security Context Length

The Security Context length is the total length of the Security Context.

4.13 Confidentiality Identifier

The Confidentiality Identifier is coded context specific (in the context of the dialogue portion sequence), constructor, with Identifier value 2, that is:

	H	G	F	E	D	C	B	A
Confidentiality Identifier	1	0	1	0	0	0	1	0

4.14 Confidentiality Length

The Confidentiality length is the total length of the confidentiality information.

4.15 Confidentiality Information

The Confidentiality Information parameter contains information, subject to a specific security context, that shall be used to perform the necessary confidentiality (i.e., encryption/decryption) functions. It is a sequence consisting of an optional identifier of a confidentiality algorithm and an optional confidentiality value to be used by that algorithm. (If the sequence is empty, confidentiality information is omitted entirely).

4.15.1 Confidentiality Algorithm ID Identifier

The Confidentiality Algorithm ID Identifier is coded context specific (in the context of the confidentiality information sequence), primitive, and takes one of the following values:

	H	G	F	E	D	C	B	A
Integer ID	1	0	0	0	0	0	0	0
Object Identifier ID	1	0	0	0	0	0	0	1

4.15.2 Confidentiality Algorithm ID Length

The Confidentiality Algorithm ID Length is the total length of the Confidentiality Algorithm ID.

4.15.3 Confidentiality Algorithm ID

The confidentiality algorithm is used to decipher data that has been encrypted. The Confidentiality Algorithm ID is an integer or an object identifier, which identifies this algorithm.

4.15.4 Confidentiality Value Identifier

The Confidentiality Value is specified as part of the definition of an individual confidentiality algorithm.

4.15.5 Confidentiality Value

The Confidentiality Value is specified as part of the definition of an individual confidentiality algorithm. It may consist of any information that can be encoded using the Basic Encoding Rules of CCITT (currently ITU-T) X.209-1988.

5 COMPONENT PORTION

5.1 Component Sequence Identifier

This field identifies the following Component Sequence and is coded national, constructor with Identifier code 8, that is:

	H	G	F	E	D	C	B	A
Component Sequence Identifier	1	1	1	0	1	0	0	0

5.2 Component Sequence Length

This field encodes the total length in octets of the Component Sequence.

5.3 Component Type Identifier

The Component Type Identifier is coded national, constructor as follows:

Component Type Identifier	H	G	F	E	D	C	B	A
Invoke (Last)	1	1	1	0	1	0	0	1
Return Result (Last)	1	1	1	0	1	0	1	0
Return Error	1	1	1	0	1	0	1	1
Reject	1	1	1	0	1	1	0	0
Invoke (Not Last)	1	1	1	0	1	1	0	1
Return Result (Not Last) ³	1	1	1	0	1	1	1	0

³ Applications using T1.114-1988 or T1.114-1992 may send multiple Return Result Components -- i.e., zero or more Return Result (Not Last) followed by either a Return Result (Last) or an Invoke (Last) Component. Other applications do not send a

5.4 Component Length

This field specifies how many additional octets are in this Component.

5.5 Component ID Identifier

The Component ID Identifier is coded as national, primitive with Identifier code 15, that is:

	H	G	F	E	D	C	B	A
Component ID Identifier	1	1	0	0	1	1	1	1

5.6 Component ID Length

The Component ID Length indicates the total length of the Component Ids. It may be 0, 1, or 2 octets.

5.7 Component Ids

The number of Component IDs is determined by the Component Type and is shown below.

Component Type	Invoke ID	Correlation ID
Invoke	Optional *	Reflected
Return Result	Absent	Reflected
Return Error	Absent	Reflected
Reject	Absent	Reflected
* Mandatory when Correlation ID present or if reply expected		

Where the reflected status is indicated for Correlation IDs, the IDs are mandatory only if an Invoke ID was present in the corresponding Invoke.

If a received Return Result or Return Error is to be rejected, the Correlation ID in the received message is reflected in the Reject component.

When both the invoke ID and the correlation ID are included, the invoke ID will be first followed by the correlation ID.

Return Result (Not Last) Component for the purposes of segmenting a long reply. For future applications with a need to reply to the Component before the complete result of the operation is known, an Invoke (Not Last) is preferred to a Return Result (Not Last).

5.7.1 Invoke ID

The Invoke ID is assigned to a Component initiating an Operation. It is optional and is one octet long, if present.

5.7.2 Correlation ID

The Correlation ID is assigned to Components sent in response to another Component. It is mandatory when the received Component had an Invoke ID. It is one octet long, if present.

5.8 Operation Code Identifier

This Operation Code Identifier identifies the Operation Code that follows as being either National TCAP or Private TCAP. It is coded national, primitive, as follows:

Operation Code Identifier	H	G	F	E	D	C	B	A
National TCAP	1	1	0	1	0	0	0	0
Private TCAP	1	1	0	1	0	0	0	1

5.9 Operation Code Length

This field specifies the length of the Operation Code. It is always 2 octets long when the Operation Code Identifier is coded "National TCAP." There are no restrictions on the length when the Operation Code Identifier is coded "Private TCAP."

5.10 Operation Code

The Operation Code is partitioned into an Operation Family followed by a Specifier associated with each Operation Family member. The lengths of the Operation Family field and Specifier field are one octet each.

The operation codes are application specific information carried unexamined by TCAP. Operation codes used by TC-users are provided in Chapter T1.114.5.

5.11 Error Code Identifier

This Error Code Identifier identifies the Error Code that follows as being either coded national, primitive, as follows:

Error Code Identifier	H	G	F	E	D	C	B	A
National TCAP	1	1	0	1	0	0	1	1
Private TCAP	1	1	0	1	0	1	0	0

5.12 Error Code Length

This field specifies the length of the error code. It is always one octet long when the Error Code Identifier is coded "National TCAP." There are no restrictions on the length when the Error Code Identifier is coded "Private TCAP."

5.13 Error Code

This provides the reason why a specific operation could not be completed successfully. The error codes are application specific information carried unexamined by TCAP. Error codes used by TC-users are provided in Chapter T1.114.5.

5.14 Problem Code Identifier

This field indicates that a Problem Code follows. It is coded national, primitive with Identifier code 21, that is:

	H	G	F	E	D	C	B	A
Problem Code Identifier	1	1	0	1	0	1	0	1

5.15 Problem Code Length

This field specifies the length of the Problem Code. It is always 2 octets.

5.16 Problem Code

This field indicates the reason the Component or Transaction Portion⁴ was rejected. The Problem Code is partitioned into a Problem Type followed by a Problem Specifier associated with each Problem Type. The length of the Problem Type field and Problem Specifier field are one octet each. These two fields follow the Problem Code Length.

The Problem Type values are coded as follows:

5.16.1 Problem Type

Problem Type	H	G	F	E	D	C	B	A
Not used	0	0	0	0	0	0	0	0
General	0	0	0	0	0	0	0	1
Invoke	0	0	0	0	0	0	1	0
Return Result	0	0	0	0	0	0	1	1
Return Error	0	0	0	0	0	1	0	0
Transaction Portion	0	0	0	0	0	1	0	1

⁴ Applications using T1.114-1988 report Transaction Portion problems using a Reject component. It is preferred that other applications report these problems using the Abort package type.

5.16.2 Problem Specifier

The Problem Specifier indicates a specific problem found within the type.

Problem Type	Problem Specifier	H	G	F	E	D	C	B	A
All families	Reserved	1	1	1	1	1	1	1	1
	Not used	0	0	0	0	0	0	0	0
General	Unrecognized Component Type	0	0	0	0	0	0	0	1
	Incorrect Component Portion	0	0	0	0	0	0	1	0
	Badly Structured Component Portion	0	0	0	0	0	0	1	1
	Incorrect Component Coding	0	0	0	0	0	1	0	0
Invoke	Duplicate Invoke ID	0	0	0	0	0	0	0	1
	Unrecognized Operation Code	0	0	0	0	0	0	1	0
	Incorrect Parameter	0	0	0	0	0	0	1	1
	Unrecognized Correlation ID	0	0	0	0	0	1	0	0
Return Result	Unassigned Correlation ID	0	0	0	0	0	0	0	1
	Unexpected Return Result	0	0	0	0	0	0	1	0
	Incorrect Parameter	0	0	0	0	0	0	1	1
Return Error	Unassigned Correlation ID	0	0	0	0	0	0	0	1
	Unexpected Return Error	0	0	0	0	0	0	1	0
	Unrecognized Error	0	0	0	0	0	0	1	1
	Unexpected Error	0	0	0	0	0	1	0	0
	Incorrect Parameter	0	0	0	0	0	1	0	1
Transaction Portion	Unrecognized Package Type	0	0	0	0	0	0	0	1
	Incorrect Transaction Portion	0	0	0	0	0	0	1	0
	Badly Structured Transaction Portion	0	0	0	0	0	0	1	1
	Unassigned Responding Transaction ID	0	0	0	0	0	1	0	0
	Permission to Release	0	0	0	0	0	1	0	1
	Resource Unavailable	0	0	0	0	0	1	1	0

5.17 Parameter Set Identifier

This field indicates that a set of Parameters is to follow. It is coded as national, constructor with Identifier code 18, that is:

	H	G	F	E	D	C	B	A
Parameter Set Identifier	1	1	1	1	0	0	1	0

5.18 *Parameter Set Length*

This length indicates the total length in octets of the Parameter Set.

5.19 *Parameter Sequence Identifier*

This field indicates that a sequence of Parameters is to follow. It is coded as universal, constructor with Identifier code 16, that is:

	H	G	F	E	D	C	B	A
Parameter Sequence Identifier	0	0	1	1	0	0	0	0

5.20 *Parameter Sequence Length*

This length indicates the total length in octets of the Parameter Sequence.

6 PARAMETERS

The Parameter Identifier uniquely identifies a particular parameter. The Parameter Length encodes the length of the particular parameter being described. Finally, the content encodes the actual value of the parameter. The parameter contents may be empty. If the parameter content is empty, then the parameter length would be 0.

The parameter identifiers and contents are application specific information carried unexamined by TCAP. Parameters used by TC-user Application Service Elements (ASEs) are provided in Chapter T1.114.5, and in other ASE specifications.

7 SUMMARY OF IDENTIFIERS

The following tables highlight the currently assigned identifiers. Duplicated context specific identifiers (bits H G = 10) in the following tables occur in different sets/sequences and therefore are unambiguous.

ATIS-1000114.2004 (S2019)

Chapter T1.114.3

TRANSACTION PORTION IDENTIFIERS	H	G	F	E	D	C	B	A
Unidirectional Package Type	1	1	1	0	0	0	0	1
Query With Permission Package Type	1	1	1	0	0	0	1	0
Query Without Permission Package Type	1	1	1	0	0	0	1	1
Response Package Type	1	1	1	0	0	1	0	0
Conversation With Permission Package Type	1	1	1	0	0	1	0	1
Conversation Without Permission Package Type	1	1	1	0	0	1	1	0
Abort Package Type	1	1	1	1	0	1	1	0
P-Abort Cause	1	1	0	1	0	1	1	1
User Abort Information	1	1	0	1	1	0	0	0
Transaction ID	1	1	0	0	0	1	1	1

DIALOGUE PORTION IDENTIFIERS	H	G	F	E	D	C	B	A
Dialogue Portion	1	1	1	1	1	0	0	1
Protocol Version	1	1	0	1	1	0	1	0
Integer Application Context	1	1	0	1	1	0	1	1
Object Identifier Application Context	1	1	0	1	1	1	0	0
User Information Identifier	1	1	1	1	1	1	0	1
Integer Security Context	1	0	0	0	0	0	0	0
Object Identifier Security Context	1	0	0	0	0	0	0	1
Confidentiality Identifier	1	0	1	0	0	0	1	0
Integer Confidentiality Algorithm ID Identifier	1	0	0	0	0	0	0	0
Object Identifier Confidentiality Algorithm ID Identifier	1	0	0	0	0	0	0	1

ATIS-1000114.2004 (S2019)

Chapter T1.114.3

COMPONENT IDENTIFIERS	H	G	F	E	D	C	B	A
Component Sequence	1	1	1	0	1	0	0	0
Invoke Component (Last)	1	1	1	0	1	0	0	1
Return Result Component (Last)	1	1	1	0	1	0	1	0
Return Error Component	1	1	1	0	1	0	1	1
Reject Component	1	1	1	0	1	1	0	0
Invoke Component (Not Last)	1	1	1	0	1	1	0	1
Return Result Component (Not Last) ⁵	1	1	1	0	1	1	1	0
Component ID	1	1	0	0	1	1	1	1
National TCAP Operation Code	1	1	0	1	0	0	0	0
Private TCAP Operation Code	1	1	0	1	0	0	0	1
Parameter Set	1	1	1	1	0	0	1	0
National TCAP Error Code	1	1	0	1	0	0	1	1
Private TCAP Error Code	1	1	0	1	0	1	0	0
Problem Code	1	1	0	1	0	1	0	1
Parameter Sequence	0	0	1	1	0	0	0	0

⁵ Applications using T1.114-1988 or T1.114-1992 may send multiple Return Result Components -- i.e., zero or more Return Result (Not Last) followed by either a Return Result (Last) or an Invoke (Last) Component. Other applications do not send a Return Result (Not Last) Component for the purposes of segmenting a long reply. For future applications with a need to reply to the Component before the complete result of the operation is known, an Invoke (Not Last) is preferred to a Return Result (Not Last).

Annex A
(informative)

A DESCRIPTION OF TCAP IN ASN.1

See ITU-T Recommendations of the X.68x series for a description of ASN.1. See ITU-T Recommendation X.690 for an explanation of the basic encoding rules.

```
TCAP-Remote-Operations-Information-Objects {iso(1) memberbody(2) usa(840) t1-114(10013)
modules(0) informationObjects(1) version4(4) }
```

```
DEFINITIONS ::=
```

```
BEGIN
```

```
--Exports Everything
```

```
IMPORTS emptyBind, emptyUnbind FROM
```

```
{joint-iso-ccitt remote-operations(4) useful-definitions(7) version1(0) };
```

```
OPERATION ::= CLASS {
```

```

    &ArgumentType          OPTIONAL,
    &argumentTypeOptional  BOOLEAN OPTIONAL,
    &returnResult          BOOLEAN DEFAULT TRUE,
    &ResultType            OPTIONAL,
    &resultTypeOptional    BOOLEAN OPTIONAL,
    &Errors                 ERROR OPTIONAL,
    &Linked                 OPERATION OPTIONAL,
    &synchronous            BOOLEAN DEFAULT FALSE,
    &alwaysReturns          BOOLEAN DEFAULT TRUE,
    &InvokePriority         Priority OPTIONAL,
    &ResultPriority         Priority OPTIONAL,
    &invokeLast             BOOLEAN DEFAULT FALSE,
    &operationCode          OperationCode UNIQUE OPTIONAL
}

```

```
WITH SYNTAX
```

```

{
  [ARGUMENT                &ArgumentType [OPTIONAL
                             &argumentTypeOptional]]
  [RESULT                   &ResultType [OPTIONAL
                             &resultTypeOptional]]
  [RETURN RESULT           &returnResult]
  [ERRORS                   &Errors]
  [LINKED                   &Linked]
  [SYNCHRONOUS              &synchronous]
  [ALWAYS RETURNS          &alwaysReturns]
  [INVOKE PRIORITY         &InvokePriority]
  [RESULT PRIORITY         &ResultPriority]
  [LAST                     &invokeLast]
  [CODE                     &operationCode]
}

```

```
ERROR ::= CLASS
```

```

{
  &ParameterType          OPTIONAL,
  &parameterTypeOptional  BOOLEAN OPTIONAL,
  &ErrorPriority           Priority OPTIONAL,
  &errorCode               ErrorCode UNIQUE OPTIONAL
}

```

```
WITH SYNTAX
```

```

{
  [PARAMETER                &ParameterType [OPTIONAL
                             &parameterTypeOptional]]
  [PRIORITY                  &ErrorPriority]
  [CODE                      &errorCode]
}

```

```

}
OPERATION-PACKAGE ::= CLASS                                {
    &Both                                                    OPERATION OPTIONAL,
    &Consumer                                                OPERATION OPTIONAL,
    &Supplier                                                OPERATION OPTIONAL,
    &id                                                       OBJECT IDENTIFIER UNIQUE OPTIONAL
}
WITH SYNTAX
{
    [OPERATIONS                                             &Both]
    [CONSUMER INVOKES                                       &Supplier]
    [SUPPLIER INVOKES                                       &Consumer]
    [ID                                                       &id]
}
CONNECTION-PACKAGE ::= CLASS                              {
    &bind                                                      OPERATION DEFAULT emptyBind,
    &unbind                                                    OPERATION DEFAULT emptyUnbind,
    &responderCanUnbind                                       BOOLEAN DEFAULT FALSE,
    &unbindCanFail                                           BOOLEAN DEFAULT FALSE,
    &id                                                       OBJECT IDENTIFIER UNIQUE OPTIONAL
}
WITH SYNTAX
{
    [BIND                                                    &bind]
    [UNBIND                                                  &unbind]
    [RESPONDER UNBIND                                       &responderCanUnbind]
    [FAILURE TO UNBIND                                       &unbindCanFail]
    [ID                                                       &id]
}
CONTRACT ::= CLASS
{
    &connection                                               CONNECTION-PACKAGE OPTIONAL,
    &OperationsOf                                             OPERATION-PACKAGE OPTIONAL,
    &InitiatorConsumerOf                                       OPERATION-PACKAGE OPTIONAL,
    &InitiatorSupplierOf                                       OPERATION-PACKAGE OPTIONAL,
    &id                                                       OBJECT IDENTIFIER UNIQUE OPTIONAL
}
WITH SYNTAX
{
    [CONNECTION                                             &connection]
    [OPERATIONS OF                                          &OperationsOf]
    [INITIATOR CONSUMER OF                                   &InitiatorConsumerOf]
    [RESPONDER CONSUMER OF                                   &InitiatorSupplierOf]
    [ID                                                       &id]
}
ROS-OBJECT-CLASS ::= CLASS
{
    &Is                                                         ROS-OBJECT-CLASS OPTIONAL,
    &Initiates                                                CONTRACT OPTIONAL,
    &Responds                                                 CONTRACT OPTIONAL,
    &InitiatesAndResponds                                       CONTRACT OPTIONAL,
    &id                                                       OBJECT IDENTIFIER UNIQUE
}
WITH SYNTAX
{
    [IS                                                      &Is]
    [BOTH                                                    &InitiatesAndResponds]
    [INITIATES                                               &Initiates]
    [RESPONDS                                               &Responds]
    ID                                                       &id
}
OperationCode ::= CHOICE
{
    national                                                  [PRIVATE 16]
    private                                                    IMPLICIT INTEGER -32768..32767,
    [PRIVATE 17] IMPLICIT INTEGER
}
ErrorCode ::= CHOICE
{
    national                                                  [PRIVATE 19] INTEGER -128..127,
    private                                                    [PRIVATE 20] INTEGER
}

```

```
Priority      ::= INTEGER (0..MAX)
}
END      --end of Information Object Specifications
```

ATIS-1000114.2004 (S2019)

Chapter T1.114.3

```
TCAPPackage {iso(1) memberbody(2) usa(840) t1-114(10013) modules(0) tcapPackage(0)
version4(4)}
```

```
DEFINITIONS ::=\
- - iso(1) memberbody(2)
- - usa(840) T1.114(10013)
```

```
BEGIN
- - defining a module called TCAPPackage which contains type
- - definitions for the contents of any generic TCAP message
- - exports everything
```

```
IMPORTS OPERATION, ERROR FROM
    {iso(1) memberbody(2) usa(840) t1-114(10013) modules(0) information-objects(1)
version4(4)}:
```

```
PackageType ::= CHOICE { unidirectional [PRIVATE 1] IMPLICIT UniTransactionPDU
                          queryWithPerm [PRIVATE 2] IMPLICIT TransactionPDU
                          queryWithoutPerm [PRIVATE 3] IMPLICIT TransactionPDU
                          response [PRIVATE 4] IMPLICIT TransactionPDU
                          conversationWithPerm [PRIVATE 5] IMPLICIT TransactionPDU
                          conversationWithoutPerm [PRIVATE 6] IMPLICIT TransactionPDU
                          abort [PRIVATE 22] IMPLICIT Abort }
```

```
UniTransactionPDU ::= SEQUENCE {
    identifier TransactionID,
    dialoguePortion DialoguePortion OPTIONAL,
    componentPortion ComponentSequence}
```

```
TransactionPDU ::= SEQUENCE {
    identifier TransactionID,
    dialoguePortion DialoguePortion OPTIONAL,
    componentPortion ComponentSequence OPTIONAL}
- -TransactionPDU should include either a Dialogue Portion, a Component
Sequence or both
```

```
TransactionID ::= [PRIVATE 7] IMPLICIT OCTET STRING
- - 0 octets for the Unidirectional, 4 octets for Query, Response & Abort
- - 8 octets for Conversation in the order Originating then Responding TID
```

```
Abort ::= SEQUENCE {
    identifier TransactionID,
    dialogPortion DialoguePortionOPTIONAL
    causeInformation CHOICE {
        abortCause P-Abort-cause,
        userInformation UserAbortInformation
        OPTIONAL
    }
}
- - When the Abort package is generated by the Transaction sublayer,
- - the P-Abort-cause must be present
```

```
P-Abort-cause ::= [PRIVATE 23] IMPLICIT INTEGER{
    unrecognizedPackageType (1),
    incorrectTransactionPortion (2),
    badlyStructuredTransactionPortion (3),
    unassignedRespondingTransactionID (4),
    permissionToReleaseProblem (5), - - not delineated
    resourceUnavailable (6),
    unrecognizedDialoguePortionID (7),
    badlyStructuredDialoguePortion (8),
    missingDialoguePortion (9),
    inconsistentDialoguePortion (10)}
```

ATIS-1000114.2004 (S2019)

Chapter T1.114.3

```
DialoguePortion ::= [PRIVATE 25] IMPLICIT SEQUENCE {
    version          ProtocolVersion OPTIONAL,
    ApplicationContext CHOICE {
        integerApplicationId IntegerApplicationContext,
        objectApplicationId ObjectIDApplicationContext
    } OPTIONAL,
    userInformation  UserInformation OPTIONAL,
    securityContext  CHOICE {
        integerSecurityId [0] IMPLICIT INTEGER,
        objectSecurityId [1] IMPLICIT OBJECT IDENTIFIER
    } OPTIONAL,
    confidentiality  IMPLICIT Confidentiality OPTIONAL,
}

ProtocolVersion ::= [PRIVATE 26] IMPLICIT OCTET STRING (SIZE (1))
--0000 0000 not used
--0000 0001 T1.114-1996
--0000 0010 T1.114-2000
--0000 0011 T1.114-2004
--other reserved
--These values can be combined using the bit-wise logical or operation
-- to indicate support for more than one version, e.g. the value 0000 0011
-- means that 1996, 2000 and 2004 versions are supported

IntegerApplicationContext ::= [PRIVATE 27] IMPLICIT INTEGER

ObjectIDApplicationContext ::= [PRIVATE 28] IMPLICIT OBJECT IDENTIFIER

UserInformation ::= [PRIVATE 29] IMPLICIT SEQUENCE OF EXTERNAL

Confidentiality ::= SEQUENCE {
    confidentialityId CHOICE {
        integerConfidentialityId [0] IMPLICIT INTEGER,
        objectConfidentialityId [1] IMPLICIT OBJECT IDENTIFIER
    } OPTIONAL,
    -- The extension marker indicates the possible presence of items
    -- in the confidentiality set that are used by the confidentiality
    -- algorithm.
}

UserAbortInformation ::= [PRIVATE 24] EXTERNAL

ComponentSequence ::= [PRIVATE 8] IMPLICIT SEQUENCE OF ComponentPDU

- Component Portion specification starts below

ComponentPDU{ InvokeId: InvokeIdSet, OPERATION: Invocable, OPERATION: Returnable } ::=
CHOICE {
    invokeLast [PRIVATE 9] IMPLICIT Invoke {{InvokeIdSet}, {Invocable}}
        (CONSTRAINED BY { --invocable.&invokeLast must be TRUE -- }
        ! RejectProblem : general-incorrectComponentPortion),
    returnResultLast [PRIVATE 10] IMPLICIT ReturnResult{{Returnable}}
    returnError [PRIVATE 11] IMPLICIT ReturnError{{Errors{{Returnable}}}}
    reject [PRIVATE 12] IMPLICIT Reject
    invokeNotLast [PRIVATE 13] IMPLICIT Invoke{{InvokeIdSet}, {Invocable}}
        (CONSTRAINED BY { --invocable.&invokeLast must be FALSE -- }
        ! RejectProblem : general-incorrectComponentPortion)
    returnResultNotLast [PRIVATE 14] IMPLICIT ReturnResult {{Returnable}}
}
(CONSTRAINED BY { -- must conform to the above definition -- }
! RejectProblem : general-unrecognisedComponentType)

Invoke{ InvokeID: InvokeIdSet, OPERATION: Operations } ::= SEQUENCE {
    componentIDs [PRIVATE 15] IMPLICIT OCTET STRING SIZE(0..2)
```

ATIS-1000114.2004 (S2019)

Chapter T1.114.3

```
- The invoke ID precedes the correlation id. There may be no
-- identifier, only an invoke ID, or both invoke and correlation
-- ID's.
(CONSTRAINED BY { -- must be unambiguous -- }
! RejectProblem : invoke-duplicateInvocation ),
(CONSTRAINED BY { -- correlation ID must identify an
-- outstanding operation -- }
! RejectProblem : invoke-unrecognisedCorrelationId )
OPTIONAL,
operationCode      OPERATION.&operationCode
                  ((Operations)
! RejectProblem : invoke-unrecognisedOperation),
parameter

                  OPERATION.&ParameterType
                  ((Operations)@Opcode)
! RejectProblem : invoke-mistypedArgument ) OPTIONAL
}
(CONSTRAINED BY { -- must conform to the above definition -- }
! RejectProblem : general-incorrectComponentPortion )
(CONSTRAINED BY { -- must have consistent encoding -- }
! RejectProblem : general-badlyStructuredCompPortion )
(CONSTRAINED BY { -- must conform to ANSI T1.114.3 encoding rules -- }
! RejectProblem : general-incorrectComponentCoding )
ReturnResult{ OPERATION: Operations } ::= SEQUENCE {
componentID [PRIVATE 15] IMPLICIT OCTET STRING SIZE(1)
              (CONSTRAINED BY { --must be that of an outstanding
              operation-- }
! RejectProblem : returnResult-unrecognisedCorrelationId)
              (CONSTRAINED BY { -- which returns a result -- }
! RejectProblem : returnResult-unexpectedReturnResult),
parameter    OPERATION.&ResultType
              ({Operations}@opcode})
! RejectProblem : returnResult-incorrectParameter) OPTIONAL
}
(CONSTRAINED BY { -- must conform to the above definition -- }
! RejectProblem : general-incorrectComponentPortion )
(CONSTRAINED BY { -- must have consistent encoding -- }
! RejectProblem : general-badlyStructuredCompPortion )
(CONSTRAINED BY { -- must conform to ANSI T1.114.3 encoding rules -- }
! RejectProblem : general-incorrectComponentCoding )
ReturnError{ ERROR: Errors } ::= SEQUENCE {
componentID [PRIVATE 15] IMPLICIT OCTET STRING SIZE(1)
              (CONSTRAINED BY ( --must be that of an outstanding
              operation-- )
! RejectProblem : returnError-unrecognisedCorrelationId)
              (CONSTRAINED BY { --which returns an error-- }
! RejectProblem : returnError-unexpectedReturnError),
errorCode    ERROR.&errorCode
              ({Errors}
! RejectProblem : returnError-unrecognisedError)
              (CONSTRAINED BY { -- must be in the &Errors field of the
              --associated operation -- }
! RejectProblem : returnError-unexpectedError),
parameter    Error.&ParameterType
              ({Errors}@errorcode)
! RejectProblem : returnError-incorrectParameter) OPTIONAL
}
(CONSTRAINED BY { -- must conform to the above definition -- }
! RejectProblem : general-incorrectComponentPortion )
(CONSTRAINED BY { -- must have consistent encoding -- }
! RejectProblem : general-badlyStructuredCompPortion )
(CONSTRAINED BY { -- must conform to ANSI T1.114.3 encoding rules -- }
! RejectProblem : general-incorrectComponentCoding )
```

ATIS-1000114.2004 (S2019)

Chapter T1.114.3

```
Reject ::= SEQUENCE {
    componentID [PRIVATE 15] IMPLICIT OCTET STRING SIZE(0..1),
    rejectProblem [PRIVATE 21] IMPLICIT Problem,
    parameter CHOICE {
        paramSequence [PRIVATE 16] IMPLICIT SEQUENCE { },
        paramSet [PRIVATE 18] IMPLICIT SET { }
    } --The choice between paramSequence and paramSet is
        -- implementation dependent, however paramSequence
        -- is preferred.
}
(CONSTRAINED BY { -- must conform to the above definition -- }
! RejectProblem : general-incorrectComponentPortion )
(CONSTRAINED BY { -- must have consistent encoding -- }
! RejectProblem : general-badlyStructuredCompPortion )
(CONSTRAINED BY { -- must conform to ANSI T1.114.3 encoding rules -- }
! RejectProblem : general-incorrectComponentCoding )

- - PROBLEMS, the specification of Problems follows

Problem ::= INTEGER {
    general-unrecognisedComponentType (257),
    general-incorrectComponentPortion (258),
    general-badlyStructuredCompPortion (259),
    general-incorrectComponentCoding (260),
    invoke-duplicateInvocation (513),
    invoke-unrecognisedOperation (514),
    invoke-incorrectParameter (515),
    invoke-unrecognisedCorrelationID (516),
    returnResult-unrecognisedCorrelationID (769),
    returnResult-unexpectedReturnResult (770),
    returnResult-incorrectParameter (771),
    returnError-unrecognisedCorrelationID (1025),
    returnError-unexpectedReturnError (1026),
    returnError-unrecognisedError (1027),
    returnError-unexpectedError (1028),
    returnError-incorrectParameter (1029)
    -- Applications using T1.114-1988 report Transaction portion
    -- problems using a Reject component with a problem code in
    -- the range 1281-1286.
    -- It is preferred that other applications report
    -- these problems using the Abort package type
    transaction-unrecognizedPackageType (1281),
    transaction-incorrectTransPortion (1282),
    transaction-badlyStructuredTransPortion (1283),
    transaction-unassignedRespondingTransID (1284),
    transaction-permissionToReleaseProblem (1285),
    transaction-resourceUnavailable (1286)
}
END - - end of the TCAPPackage Module
```

Chapter T1.114.4
Transaction Capabilities Procedures

TABLE OF CONTENTS

1 SCOPE, PURPOSE, AND APPLICATION	1
1.1 BASIC GUIDELINE	1
1.2 OVERVIEW.....	1
2 ADDRESSING	1
3 NORMAL PROCEDURES	1
3.1 FUNCTIONAL GROUPING	2
3.2 TRANSACTION PORTION	2
3.2.1 <i>Connectionless Network Services (No Application Services Part Functions Required)</i>	2
3.2.1.1 <i>Actions at the Initiating End</i>	2
3.2.1.2 <i>Actions at the Receiving End</i>	3
3.2.1.3 <i>Conversation Mode</i>	3
3.2.1.4 <i>Permission or Not to Release in the Conversation Mode</i>	3
3.2.1.5 <i>Termination of TCAP Transaction</i>	4
3.2.1.6 <i>TC-user Transaction versus Logical Connection</i>	4
3.3 DIALOGUE PORTION.....	4
3.3.1 <i>Dialogue Portion Functions</i>	4
3.3.2 <i>Protocol Version</i>	5
3.3.3 <i>Application Context</i>	5
3.4 COMPONENT PORTION.....	5
3.4.1 <i>Assignment of Component Identifiers by a TC-user</i>	6
3.4.2 <i>Assignment of Component States by a TC-user</i>	7
3.4.3 <i>Maintenance of Invoke IDs</i>	7
4 ABNORMAL PROCEDURES	7
4.1 CONNECTIONLESS NETWORK SERVICE	7
4.1.1 <i>General</i>	7
4.1.2 <i>Introduction</i>	7
4.1.3 <i>Abnormal Conditions</i>	7
4.1.3.1 <i>Protocol Errors</i>	7
4.1.3.2 <i>Application Errors</i>	8
4.1.3.3 <i>End User Abnormalities</i>	8
4.1.4 <i>Detection</i>	8
4.1.5 <i>Reporting</i>	9
4.1.5.1 <i>Reject</i>	9
4.1.5.2 <i>Return Error</i>	10
4.1.5.3 <i>Return Result</i>	11
4.1.6 <i>Recovery</i>	11
4.1.7 <i>Abnormal Procedures Relating to Transaction Portion</i>	11
4.1.8 <i>Abnormal Procedures Relating to the Dialogue Portion</i>	12
4.1.9 <i>Abnormal Procedures Relating to Operations</i>	13
4.1.10 <i>Abnormal Procedures Related to Components</i>	14

TABLE OF FIGURES

FIGURE 1/T1.114.4 - A SIMPLE EXAMPLE OF EXCHANGE OF TCAP MESSAGES.....	15
FIGURE 2/T1.114.4 – A COMPLEX EXAMPLE OF EXCHANGE OF TCAP MESSAGES	16
FIGURE 3/T1.114.4 - AN EXAMPLE OF MULTIPLE TCAP TRANSACTIONS	17
FIGURE 4/T1.114.4 - AN EXAMPLE OF AN EXCHANGE OF COMPONENTS IN A SINGLE TRANSACTION	17

TABLE OF TABLES

TABLE 1/T1.114.4 - TRANSACTION PORTION ABNORMAL PROCEDURES.....	12
TABLE 2/T1.114.4 - ABNORMAL PROCEDURES FOR OPERATIONS.....	13
TABLE 3/T1.114.4 - COMPONENT ABNORMAL PROCEDURES.....	15

Transaction Capabilities Procedures

1 SCOPE, PURPOSE, AND APPLICATION

Transaction Capabilities (TC) allows TC-users to exchange Components via Transaction Capabilities Application Part (TCAP) messages. Procedures described in this chapter specify the rules governing the information content and the exchange of TCAP messages between TC-users.

This chapter may contain requirements that reference other American National Standards. If so, when the American National Standards referenced in the requirements are superseded by revisions approved by the American National Standards Institute, Inc, the revisions shall apply.†

1.1 Basic Guideline

To maximize flexibility in service architecture and implementation style, TCAP procedures restrict themselves to supporting the exchange of Components between TC-users. Operation (Application) procedures are not part of TCAP.

1.2 Overview

Clause 2 discusses addressing rules for TCAP messages. Clause 3 describes TC procedure under normal (error-free, smooth running) conditions. Clause 4 details TC procedures under abnormal conditions.

2 ADDRESSING

TCAP messages will use any of the addressing options afforded by the Signalling Connection Control Part (SCCP). These addressing options are provided in T1.112.4. The assignment and use of Global Title values is specified in Annex B of T1.112.3.

Addressing within the TC-user may be accomplished via the use of Transaction IDs.

Addressing options available from the Application Service Part (ASP) are not addressed.

3 NORMAL PROCEDURES

The interface between an Application Process and an Application Part is entirely implementation specific -- i.e., it is not like a standardized primitive interface between two adjacent layers of the OSI Reference Model with specified services assigned to each layer. This section includes some discussions of the role of an Application Process as a user of the TCAP that can be viewed as outside the scope of the specifications of TCAP procedures and properly belonging to Chapters T1.114.1, T1.114.2, or both. Nevertheless, the discussions are included in this section to aid in understanding these specifications.

Also, when the selection of a parameter value required by a lower layer is not discussed in the TCAP procedures (e.g., Quality of Service), it is assumed that the value is specified by the TC-user, and TCAP simply passes the value down through its primitive interface. The same assumption applies to the

† A change bar in the right margin indicates a change from T1.114-2000.

parameters received from a lower layer through the primitive interface (e.g., calling party address), which are not required for TCAP functions.

3.1 *Functional Grouping*

The functions of the TCAP are grouped into three portions: Transaction, Dialogue, and Component.

The Transaction portion of the TCAP procedures provides an application level association over which Components are exchanged -- i.e., it provides the means to identify a group of Components as belonging to a particular transaction. The Dialogue portion of the TCAP procedures provides the mechanism to determine a mutually acceptable Application Context for the exchange of operation parameter and error information. The Dialogue portion also provides the mechanism to exchange security context and confidentiality information. The Component portion of the TCAP procedure provides a TC-user with the capability to invoke an operation at a remote TC-user and receive the responses.

The fields of a TCAP message correspond to this functional grouping of the features of the TCAP procedures.

3.2 *Transaction Portion*

3.2.1 Connectionless Network Services (No Application Services Part Functions Required)

The Transaction portion of the TCAP procedures identifies each TCAP message and, therefore, all the contained Components, as belonging to a particular TC-user transaction. In addition, the procedures allow association of separate TCAP messages in any direction to a TC-user transaction.

Transaction association within a TC-user serves two functions. It allows linking of a query with its response. In that function, Transaction association serves a role like that of Component correlation. In its other function, Transaction association identifies the context to help interpret a broader group of Components contained in one or more TCAP messages.

Transaction IDs identify a TC-user transaction, while the Package Types indicate the sending end's view on establishing and maintaining a TCAP transaction.

3.2.1.1 Actions at the Initiating End

When a TC-user has to send one or more Components in a TCAP message to another TC-user but does not need to enter into a TCAP transaction, the Unidirectional Package type is specified to TCAP. Use of Unidirectional Package Type implies that Component correlation is not applicable. No Transaction ID is included in a TCAP message of the Unidirectional Package Type. In all other cases, a TCAP transaction is established.

To help clarify the discussion of TCAP transaction message exchange, the sending node of the first TCAP message is labeled node "A", and the receiving node is labeled node "B".

When a TC-user at node "A" wishes to initiate a TCAP transaction with a TC-user at node "B", the first message is one of the two Query Package Types (With or Without Permission to release). The TC-user at node "A" selects and assigns to the TCAP message of the Query Package Type a Transaction ID value for the originating Transaction ID field. This Transaction ID value, when included in any future message from node "A" as the originating or in a message to node "A" as the responding Transaction

ID, identifies the TC-user transaction, i.e., that the messages and, therefore, the contained Components belong to the same TC-user transaction from the perspective of node "A".

When the TC-user at node "A" initiating a TCAP transaction anticipates sending more Components that it would like the TC-user at node "B" to treat as part of the same transaction, the TC-user at node "A" specifies a Query Without Permission (to release) Package Type.

When the TCAP transaction initiating TC-user at node "A" anticipates the converse of that described above, it sends a TCAP message of the Query With Permission (to release) Package Type.

3.2.1.2 Actions at the Receiving End

No response is required on the receipt of a message of the Unidirectional Package Type.

In response to a TCAP message of the Query Without Permission (to release) Package Type, the TC-user at node "B" should establish an TC-user transaction from its perspective by responding with a message of one of the two Conversation Package Types (With or Without Permission to Release).

A message of the Conversation Package Type from node "B" includes a responding Transaction ID value that is identical to (i.e., a reflection of) the originating Transaction ID value of the TCAP transaction initiating message from node "A". In addition, a message of the Conversation Package Type is also assigned an originating Transaction ID by node "B". This Transaction ID value, as in the corresponding situation in node "A", when included in any future message, identifies the TC-user transaction from the perspective of node "B".

In responding to a message of the Query With Permission (to release) Package Type, the TC-user at node "B" decides whether or not to establish an TC-user transaction from its perspective. If the TC-user at node "B" does want to establish an TC-user transaction from its perspective, it responds with a message of one of the two Conversation Package Types, and the procedure described in the last paragraph applies. Otherwise, it responds with a message of the Response Package Type. A message of the Response Package Type includes only one Transaction ID -- a responding Transaction ID assigned in the same manner as described in the last paragraph.

3.2.1.3 Conversation Mode

Once the TC-user at node "B" has sent a message of one of the two Conversation Package Types, all future messages in either direction shall remain one of the two Conversation Package Types until the TCAP transaction is to be terminated.

3.2.1.4 Permission or Not to Release in the Conversation Mode

A TC-user sends a message of the Conversation With Permission Package Type, when, for the present TCAP transaction, the TC-user: (1) has completed responding to all received Components; and (2) does not anticipate sending any new Components without further interactions with the peer TC-user. Otherwise, a message of the Conversation Without Permission Package Type is sent.

The permission given by one TC-user to another TC-user to release a TCAP transaction at the conversation mode is not a request to release, it is only a permission to release. It is the TC-user receiving the permission that has to decide whether or not to release the TCAP transaction. It may send several more messages to either complete responding to previously received Components or send new Components over the TCAP transaction before releasing. Also, the permission to release given by a TC-

user can be revoked. Based on future events, the TC-user which gave the permission is allowed to revoke it by sending a message of the Conversation Without Permission (to release) Package Type.

3.2.1.5 Termination of TCAP Transaction

The usual way to terminate a TCAP transaction is for the TC-user that has received the permission to release from the remote TC-user, to send, when it chooses, a message of the Response Package Type.

In addition, by prearranged agreements, a TCAP transaction may be terminated at the discretion of the TC-user at both ends without sending or receiving an explicit Response message. TC-users will inform the respective TCAPs that the transaction has been terminated.

In special situations, independent of whether a TC-user has received the permission to release from the other end or not, the TC-user can terminate the TCAP transaction by sending a message of the Response Package Type.

Figures 1/T1.114.4 and 2/T1.114.4 depict examples of exchanges of TCAP messages between two TC-users. Figure 3/T1.114.4 depicts another example of message exchange to illustrate how a TCAP transaction can be terminated by one end, while the TC-user transaction is continued by the other end.

3.2.1.6 TC-user Transaction versus Logical Connection

For clarification of TC-user transaction and its local significance, a comparison with a logical connection (SCCP type) is useful. A group of Components transferred in one or more TCAP messages with the same Transaction ID assigned by a TC-user at a node "A", for instance, are viewed as belonging to a single TC-user transaction by the TC-user at node "A". At the other end, from the perspective of a TC-user at a node "B", the Components may be viewed as forming multiple groups belonging to separate TC-user transactions each identified by a Transaction ID assigned by "B". In the context of SCCP, this would be like associating multiple destination reference numbers to a single source reference number. Although the TCAP procedures do allow emulating a logical connection by viewing the exchanged components as belonging to a single TC-user transaction from the point of view of either of the ends by assigning only one Transaction ID at each end, the TCAP procedures do not insist on it.

Another example of the flexibility of the TCAP procedure for TC-user transaction is illustrated by point-to-multipoint communications. A TC-user can send n components, one each to n other TC-users. Although each remote TC-user views the Component exchange as an independent TC-user transaction, TCAP procedure allows the sending TC-user the flexibility to view the exchanges as n different TC-user transactions by assigning n different originating Transaction IDs, or as one TC-user transaction by assigning a single originating Transaction ID to all the messages.

3.3 Dialogue Portion

3.3.1 Dialogue Portion Functions

The operation codes, parameters, and error codes in the components of a TCAP message are provided to TCAP by the TC-User at the sending end. The Dialogue portion of the TCAP procedures provides mechanism to determine the interpretation of this information at the receiving end. The Dialogue procedures include establishment of the context which the Transaction portion identifies to help

interpret components contained in one or more TCAP messages. The Dialogue portion also provides the mechanism to exchange security context and confidentiality information.

Protocol version information identifies which versions of T1.114 are supported. Application Context information identifies the context in which component information is to be understood. User information provides an Application-Process-specific mechanism to convey additional information needed to interpret the components.

3.3.2 Protocol Version

A message of the Query or Unidirectional Package Type may (optionally) specify the protocol version(s) that are known to be consistent with the sending TC-user. At the receiving end, if at least one of the protocol versions is consistent with the TC-user, the proposed Protocol Version is acceptable.

If the protocol version is acceptable, the first backward Conversation or Response message may (optionally) specify one of the protocol versions that is known to be consistent with the TC-User. Later messages in the transaction should not include Protocol Version information.

3.3.3 Application Context

A message of the Unidirectional Package Type may (optionally) suggest an application context for the transaction. If the proposed application context is acceptable to the receiving TC-user, the components are processed. Otherwise, the components are discarded.

A message of the Query Package Type may (optionally) suggest an application context for the transaction. If the proposed application context is acceptable to the receiving application, the components -- if there are any present -- are processed and the first backward message (either a Conversation or a Response) includes a Dialogue portion without an application context, or a Dialogue portion with the same application context as proposed in the Query message. Later messages in the transaction should not include application context information.

If the application context proposed in a Query message is not acceptable to the receiving application, the TC-user may still wish to continue the dialogue. In this case, a Conversation message should be returned with a different application context name in the dialogue portion. (The received components should not be processed until a mutually acceptable application context has been established.) If the receiving TC-User does not wish to continue the dialogue, an Abort message should be returned. Since this abort is generated by the TC-user, a P-Abort cause should not be included. A Dialogue portion may (optionally) be sent in the Abort message including an alternate application context name.

The TC-user receiving a Conversation message with an proposed different context application context name (i.e., a proposed different context), accepts the new context by continuing the conversation (optionally with components to be added to any from the initial query). No additional application context information is exchanged after the first backward Conversation message. If the different application context is not acceptable, the TC-user should initiate an abort (user abort).

3.4 Component Portion

A TCAP message can carry multiple Components; each Component corresponds to a single OPDU of X.410 of the CCITT (now ITU-T) Red Book as extended for TCAP. Under normal conditions, TC-users send and/or receive Invoke and Return Result Components only.

A TC-user provides to TCAP all the elements needed to construct a Component. An Invoke Component includes a single operation and the parameters (argument) necessary to perform the operation. Therefore, for an Invoke Component, the TC-user provides to TCAP the name of the Operation, the name of the parameters, and the parameter values. Similarly, for a Return Result Component, the TC-user provides the parameters (results) to TCAP to be contained in the Component.

A TC-user need not wait for one operation to complete before invoking another. At any instant in time, a TC-user may have any number of operations in progress at another TC-user.

The other information provided by a TC-user to TCAP to formulate a component relates to component correlation and component states.

In TCAP, it is permissible to respond to an Invoke with another Invoke, as well as a Return Result. It is also permissible in TCAP to reply to an Invoke with any number of Invokes and Return Results in one or more TCAP messages. TC-users based on T1.114-1988 or T1.114-1992 may transfer information unidirectionally (e.g., to periodically report status information) by sending and receiving Return Result components that are not associated with an operation in progress at the sending TC-user; however, the preferred method of unidirectional information transfer for any TC-user is in an Invoke component with no reply required.¹

A TCAP message can contain any mix of Components: initial Invoke Component, Invoke Component that is responding to a previous Invoke Component, and Return Result Component (Return Error and Reject Components can also be included in TCAP messages and will be discussed under abnormal procedures in clause 4 of this chapter).

Figure 4/T1.114.4 depicts an example of an exchange of Components in a single TCAP transaction. The Component identifiers are shown within parentheses; the Invoke ID is followed by the Correlation ID, if used. Also shown explicitly is whether or not a responding Component is the last of the responses.

3.4.1 Assignment of Component Identifiers by a TC-user

- ◆ *Invoke ID*: When the sending TC-user needs to do so, it selects and assigns an Invoke ID to the Invoke Component. When a TC-user sends an Invoke with a Correlation ID, the Invoke shall carry an Invoke ID.
- ◆ *Correlation ID*: A Correlation ID needs to be included in a Return Result Component, as well as in an Invoke Component if it is responding to a previous Invoke that included an Invoke ID. In either case, the Correlation ID is identical to (i.e., a reflection of) the Invoke ID of the Component being responded to.

To further clarify, a TC-user can specify for TCAP:

- (1) Invoke ID for an Invoke Component (nonresponding).
- (2) Invoke ID and Correlation ID for an Invoke Component (responding).
- (3) Correlation ID for a Return Result Component.

¹ These are extensions to X.410 of the CCITT Red Book, which implies that a single Return Result may respond to an Invoke.

3.4.2 Assignment of Component States by a TC-user

For an Invoke Component, the TC-user specifies to TCAP whether or not a reply is required to the Component.

When a Component (whether an Invoke or a Return Result) is responding to a previous Component, i.e., when a Component includes a Correlation ID, the TC-user specifies to TCAP whether or not the Component is the last response to the invoking Component.

3.4.3 Maintenance of Invoke IDs

The Invoke ID distinguishes the associated operation from any number of other operations the invoking TC-user may have in progress at the invoked TC-user. The invoking TC-user may not reuse an Invoke ID that was previously assigned to an operation for which it expects but has not yet received a complete response. Similarly, the invoked end maintains the received Invoke ID (to be reflected as the Correlation ID) until it has completed responding. Figure 4/T1.114.4 depicts the above concepts.

4 ABNORMAL PROCEDURES

4.1 Connectionless Network Service

4.1.1 General

This clause describes the procedures and messages needed to detect, report, and recover from abnormal conditions associated with TCAP or the supported TC-user. TC-user-dependent detection and recovery procedures are not considered part of TCAP.

4.1.2 Introduction

The TC-user sending a TCAP message is responsible for ensuring that the message is precise and correct. The TC-user that should receive the message is responsible for detecting abnormal conditions; it is also responsible for reporting abnormal conditions to the TC-user causing the abnormal condition. Abnormal conditions may also be reported locally to maintenance. In addition to triggering recovery procedures, reporting abnormal conditions allows program and data errors to be more easily identified. The TC-user is responsible for any reattempt recovery procedure.

4.1.3 Abnormal Conditions

Depending on where the error occurred, abnormal conditions can be divided into three categories: 1) protocol; 2) application; and 3) end user.

4.1.3.1 Protocol Errors

Protocol errors are caused by incorrect TCAP messages. These errors are detected by either TCAP or the TC-user. For example, this category includes, but is not limited to:

- (1) *Unrecognized Package Type or Component Type.* Package Type or Component Type is not defined in Chapter T1.114.3.
- (2) *Unrecognized Operation.* Operation is not defined for this TC-user.

- (3) *Unassigned Responding Transaction ID or Correlation ID.* No such transaction with this responding transaction ID or operation with this correlation ID is in progress.
- (4) *Unrecognized Dialogue or Component Portion Identifier.* The identifier following the last field of the transaction portion was neither the dialogue portion identifier nor the component sequence identifier as defined in Chapter T1.114.3.
- (5) *Badly Structured Dialogue Portion.* A fundamental encoding problem (e.g., multiple application contexts in one dialogue portion).
- (6) *Missing Dialogue Portion.* The dialogue portion is missing where it is mandatory (e.g., in the first backward message after a Query with a proposed application context).
- (7) *Inconsistent Dialogue Portion.* The contents of the Dialogue portion are inconsistent with the state of the transaction (e.g., a Query with empty Dialogue portion or a message after the first backward Conversation that contains an application context).

4.1.3.2 Application Errors

Application errors are caused by violations of TC-user procedures or unavailability of network resources. For example, this category includes, but is not limited to:

- (1) *Unexpected Component Sequence.* Components do not follow application script.
- (2) *Unexpected Data Value.* Data value not defined for this operation and application.
- (3) *Unavailable Resource (e.g., announcement, call register).* A requested resource is not incorporated at the serving entity.
- (4) *Missing Customer Record.* Customer record for this application cannot be found.

4.1.3.3 End User Abnormalities

End user abnormalities are caused by the end user violating the correct TC-user procedure, even though the error is within the definition of the TC-user. For example, this category includes, but is not limited to:

- (1) *Caller Abandonment.* Caller hangs up prematurely.
- (2) *Improper Caller Response.* Improper information input during caller-participation phase.

4.1.4 Detection

Detection of errors is performed in the following sequence:

- (1) Protocol Errors
- (2) Application Errors
- (3) End User Abnormalities

Therefore, detecting an application error means that there are no protocol errors, and detecting an end user abnormality implies that there are no protocol or application errors.

Detection of protocol errors is governed by TCAP abnormal procedures. However, some protocol errors may actually be detected by the TC-user. Detection of application and end user errors are entirely TC-user dependent. Each TC-user defines those application and end user abnormal conditions it will detect and those it will report.

4.1.5 Reporting

Abnormal conditions are reported to the TC-user causing the error using the Reject, Return Error, or Return Result Components, or the Abort package type defined in Chapter T1.114.2.

4.1.5.1 Reject

Protocol errors in the Component portion of a TCAP message are reported using the *Reject Component*. The Reject Component reports the receipt and rejection of an incorrect Package²) or a Component. It is sent in eventual response to an incorrect Package, or a Component whose type is other than Reject.

- ◆ *Reject: [ID, Problem]*: The Invoke ID in a rejected Invoke or the Correlation ID in a rejected Return Result (or Return Error) is reflected in the Reject Component.

Problems are divided into five categories: Transaction Portion, General, Invoke Component, Return Result Component, Return Error Component.

The following *Transaction* portion problems are reported²:

- (1) *Unrecognized Package Type*. The Package Type has not been defined.
- (2) *Incorrect Transaction Portion*. An unexpected or undefined identifier was received within the Transaction portion.
- (3) *Badly Structured Transaction Portion*. A fundamental encoding problem (e.g., bad length).
- (4) *Unassigned Responding Transaction ID*. The received Transaction ID is derivable but does not reflect a transaction currently in progress.
- (5) *Permission to Release Problem*. This problem is not delineated.
- (6) *Resource Unavailable*. Sufficient resources are not available at the Transaction sub-layer to establish a transaction.

The following *General* problems are reported:

- (1) *Unrecognized Component Type*. The component type has not been defined.
- (2) *Incorrect Component Portion*. An unexpected or undefined indicator was received within the Component portion.
- (3) *Badly Structured Component Portion*. A fundamental encoding problem (e.g., bad length).
- (4) *Incorrect Component Coding*. A component encoding problem (e.g., an operation code or parameter length value less than 127 is encoded using the long form).

The following *Invoke-Component-specific* problems are reported:

- (1) *Duplicate Invoke ID (when requested by the TC-user)*. An Invoke ID is received that has already been assigned to another Operation in progress.

² Applications using T1.114-1988 report Transaction Portion problems using a Reject component. It is preferred that other applications report these problems using the Abort package type.

- (2) *Unrecognized Operation Code*. The Operation Code has not been defined by the TC-user.
- (3) *Incorrect Parameter*. An unexpected or undefined Parameter was received.
- (4) *Unrecognized Correlation ID*. The received Correlation ID does not reflect an Operation currently in progress.

The following *Return-Result-Component-specific* problems are reported:

- (1) *Unassigned Correlation ID*. The received Correlation ID does not identify an Operation currently in progress.
- (2) *Unexpected Return Result*. The invoked operation does not report success.
- (3) *Incorrect Parameter*. An unexpected or undefined parameter was received.

The following *Return Error Component-specific* problems are reported:

- (1) *Unassigned Correlation ID*. The received Correlation ID does not reflect an operation currently in progress.
- (2) *Unexpected Return Error*. The Return Error Component does not report failure of the invoked operation.
- (3) *Unrecognized Error*. The reported error has not been defined by the TC-user.
- (4) *Unexpected Error*. The reported error is not applicable to the invoked operation.
- (5) *Incorrect Parameter*. An unexpected or undefined parameter was received.

Problems specific to Return Result and Return Error cannot always be correlated in the Component portion of TCAP because Return Result Component and Return Error Component Correlation IDs are not normally retained, only reflected. Invokes which do not carry an Invoke ID have a similar problem. Correlation in the Transaction portion of TCAP may or may not be possible depending on the number of Transaction IDs used. Even without correlation in the Transaction or Component portion of TCAP, tabulation of reported abnormal conditions can be useful for detecting program and data errors.

4.1.5.2 Return Error

The *Return Error Component* reports the unsuccessful completion of an operation. It is sent in eventual response to an Invoke Component if the latter is correct, the operation is one that reports failure only or both success and failure, and the operation fails. The TC-user defines which application errors and end user abnormalities are considered failures from the perspective of the operation.

- ◆ *Return Error: [ID, Error, Parameter]*: If the Invoke Component includes an Invoke ID or the Return Result Component includes a Correlation ID, this ID is reflected in the Return Error Component. The error element specifies the application error being reported.

The parameter (variable data) specifies the invalid data value, if applicable.

4.1.5.3 Return Result

The *Return Result Component* reports the successful completion of an operation. The TC-user defines which application errors and end user abnormalities are considered abnormal conditions from a TC-user perspective, but successful completions from the perspective of the operation.

- ◆ *Return Result: [ID, Parameters]*: The ID element (as in 4.1.5.2) identifies the operation whose success is being reported.

A parameter specifies the end user error type.

4.1.6 Recovery

Recovery from errors is TC-user dependent. In addition to improving the management of Transaction IDs and other resources associated with a particular transaction, abnormal condition reports can be used to identify errors in stored data.

4.1.7 Abnormal Procedures Relating to Transaction Portion

Transaction portion problems are reported using “ABORT” package type with a “P-Abort Cause” value. The actions to be taken by the local end (the end which detects the protocol error) and the remote end (the end which sent the erroneous package) are described in the Table 1/T1.114.4. If the local end receives a Unidirectional message and a protocol error is detected, the message should be discarded. If the local end receives a Query, Conversation (with or without Permission), or Response message and a protocol error is detected, the reaction will depend on which (if any) Transaction IDs can be derived. For a Query no local action is required. If the Originating Transaction ID is derivable (i.e., can be extracted from the errored message), an Abort should be sent to the remote end to close the transaction. If the Originating Transaction ID cannot be derived, the message should be discarded. If the local end receives a Conversation (with or without Permission) message and a protocol error is detected, and either the Originating Transaction ID or the Responding Transaction ID cannot be derived, the message should be discarded. If the local end receives a Response message and a protocol error is detected, the message should be discarded. If the local end receives an Abort message and a protocol error is detected, the message should be discarded. If the local end receives a message and a protocol error is detected and the message type cannot be determined from the errored message, the local reaction is determined from the derivable Transaction IDs (if any). If the Originating Transaction ID is derivable, the remote end should be advised by Abort to close the transaction. If the Responding Transaction ID is derivable, the local user should be advised to close the transaction.

If an application is using T1.114-1988, a Response package type with the Reject Component (in place of the Abort message) should be used to inform the remote end when a Query or Conversation message type is found to have protocol problems. An Unidirectional package type with the Reject Component (in place of the Abort message) should be used to inform the remote end if an unknown message type is found to have protocol problems.

As with T1.114-1992, protocol errors detected in Response or Unidirectional package types lead to discard of the message.

Table 1/T1.114.4 - Transaction Portion Abnormal Procedures

Local End (detects protocol error)				Remote End	
Message Type Received	Message Disposition	Transaction State Machine	User Advised	Transaction State Machine	User Advised
UNIDIRECTIONAL	Discard ³	N/A	No	N/A	No
QUERY, OTID not derivable ⁴	Discard	N/A	No	N/A	No
QUERY, OTID derivable	Abort	N/A	No	Returns to Idle ⁵	Yes ⁶
CONV., OTID or RTID not derivable	Discard	N/A	No	N/A	No
CONV., RTID not assigned ⁶ OTID derivable	Abort	N/A	No	Returns to Idle ⁶	Yes ⁶
CONV., RTID assigned OTID derivable	Abort	Returns to Idle	Yes	Returns to Idle ⁶	Yes ⁶
RESPONSE, RTID not derivable or RTID not assigned	Discard	N/A	No	N/A	No
RESPONSE, RTID assigned	Discard	Returns to Idle	Yes	N/A	No
ABORT, RTID not derivable or RTID not assigned	Discard	N/A	No	N/A	No
ABORT, RTID assigned	Discard	Returns to Idle	Yes	N/A	No
Unknown, RTID not derivable	Discard	N/A	No	N/A	No
Unknown, RTID derivable and assigned OTID derivable	Abort	Returns to Idle	Yes	Returns to Idle ⁶	Yes ⁶
Unknown, RTID derivable but not assigned OTID derivable	Abort	N/A	No	Returns to Idle ⁶	Yes

The following states are defined in the Transaction sub-layer:

- ◆ *Idle*: There is no transaction (initiated or established) for this Transaction ID.
- ◆ *(insert one of the TCAP Package names here) Package sent*: A package of type (insert TCAP Package name) has been sent to the remote end.
- ◆ *(insert one of the TCAP Package names here) Package received*: A package of type (insert TCAP Package name) has been received from the remote end.

4.1.8 Abnormal Procedures Relating to the Dialogue Portion

Dialogue portion problems are reported in messages of the abort package type. Fundamental protocol errors relating to the dialogue portion are reported using the P-Abort cause values listed in clause

³ There is a possibility of advising the remote end for management purposes.

⁴ Indicates whether or not it can be extracted from the message.

⁵ The state transition is applicable and the user informed only if the Transaction ID has been assigned at this end to an active or initiated transaction.

⁶ Indicates whether or not it is in use at the local end.

4.1.3.1. Problems detected by the TC-user -- e.g., unacceptable application context, are reported using an abort (user abort) as described in clause 3.3.

4.1.9 Abnormal Procedures Relating to Operations

Protocol errors in the Component portion of a TCAP message are reported using the Reject component. The Reject component is sent in response to an incorrect component other than Reject.

When the Invoke ID in a to-be-rejected Invoke component or the Correlation ID in a to-be-rejected Return Result or Return Error component is available, this ID is reflected in the Reject component.

The actions taken by TCAP to report Component portion errors are shown in Table 2/T1.114.4.

This Table applies indirectly to TC-user detected errors. In this case, the TC-user is directly aware of the abnormality and initiates actions resulting in the same outcome.

Table 2/T1.114.4 - Abnormal Procedures for Operations

Local End (detects protocol error)				Remote End	
Component Type Received	Initiate Reject	Component State Machine	Local User Advised by TC-REJECT	Component State Machine	Local User Advised by TC-REJECT
Invoke Last	Yes	Invoke ID - N/A, Correlation ID - No Change	Yes ⁷	Returns to Idle ⁸	Yes
Invoke Not Last	Yes	Invoke ID - N/A, Correlation ID - No Change	Yes ⁸	Returns to Idle ⁹	Yes
Return Result-Last	Yes	No Change	Yes ⁸	N/A	Yes
Return Result-Not Last ⁹	Yes	No Change	Yes ⁸	N/A	Yes
Return Error	Yes	No Change	Yes ⁸	N/A	Yes
Unrecognized. Component or Unassigned ID	Yes	N/A	Yes ⁸	N/A ¹⁰	Yes
Reject	No	No Change	Yes	N/A	N/A

⁷ This is to alert the TC-user so it can issue a dialogue control primitive to send the Reject formulated by the Component Portion.

⁸ The Component state machine applies to the operation being invoked by the far end.

⁹ TC-users using T1.114-1988 or T1.114-1992 may send multiple Return Result components (i.e., zero or more Return Result (Not Last) followed by either a Return Result (Last) or an Invoke (Last) component). Other application do not send a Return Result (Not Last) component for the purpose of segmenting a long reply. For future applications with a need to reply to a component before the complete result of the operation is known, an Invoke(Not Last) is preferred.

¹⁰ If the Component sent by the remote end resulted in an active Component State Machine at the remote end, it should be set to idle.

The following states are defined in the Component sub-layer:

- ◆ *Idle*: No invocation has been sent for this component ID.
- ◆ *Operation pending*: An invocation has been sent for which a response is awaited.
- ◆ *Operation in progress*: An invocation has been received.

In the case of multiple components within a message where the bad component is detected by TCAP, the need to examine further the contents of the message is not addressed.

The message type used to carry the Reject Component is decided by the local TC-user when informed by TCAP about the reception of a bad component. The local TC-user may decide to continue, terminate, or abort the transaction associated with the message containing the bad component.

The procedures for rejecting abnormal multiple Return Results responding to the same Invoke are not addressed.

4.1.10 Abnormal Procedures Related to Components

When a problem is detected in a Component, or the local TC-user decides to abnormally terminate the transaction, the local TC-user will decide the package type to be used to convey this information to the remote user. This clause describes various alternatives a TC-user has in case of an abnormal condition.

If a bad Component is received in a Query or Conversation package type and the TC-user decides to continue the dialogue, a Conversation package type with the Reject Component should be sent. If, for any reason, the TC-user decides to terminate the dialogue and there are no operations in progress or pending, a Response package with a Reject Component, or an Abort package with User Abort information, should be sent. Since there are no operations in progress or pending, the Component state machine is idle and the sending of a Response or an Abort package in this case achieves the same result. However, a Reject Component cannot be sent in an Abort package.

If the TC-user decides to terminate the dialogue when there are operations pending or in progress, an Abort package with User Abort information should be sent to terminate the transaction (dialogue)¹¹. By using the Abort package, the Component and the Transaction state machine will become idle and the transaction will be terminated at the local (receiving) and remote end.

Further consideration of the utility of sending a Reject in an Unidirectional package when the message type received was Response is required. This would see the Component portion error reported outside the transaction in which it occurred.

Table 3/T1.114.4 shows the various user options for handling Component portion problems.

¹¹ If operations are pending but no Transaction ID is available from the remote end (e.g., after an Invoke has been sent in a Query but before an Invoke has been received in a Conversation), no message can be sent to terminate the transaction at the remote end. The TC-user may decide to terminate the transaction, in which case subsequent messages received may stimulate a P-Abort. Alternatively, the TC-user may wait for a message from the remote end before sending a message containing User Abort information.

Table 3/T1.114.4 - Component Abnormal Procedures

Package Type Received	User Option			
	UNIDIRECTIONAL with Reject	Continue Transaction	Terminate Transaction	
		CONVERSATION with Reject	No Op Pending/ In Progress	Op Pending/ In Progress
			RESPONSE with Reject or U-ABORT	U-ABORT
UNIDIRECTIONAL	X ¹²			
QUERY		X	X	X
CONVERSATION		X	X	X
RESPONSE	X ¹²⁾			

For a TC-user using T1.114-1988, the Response message with a Reject Component should be used instead of the Abort message with User Abort information.

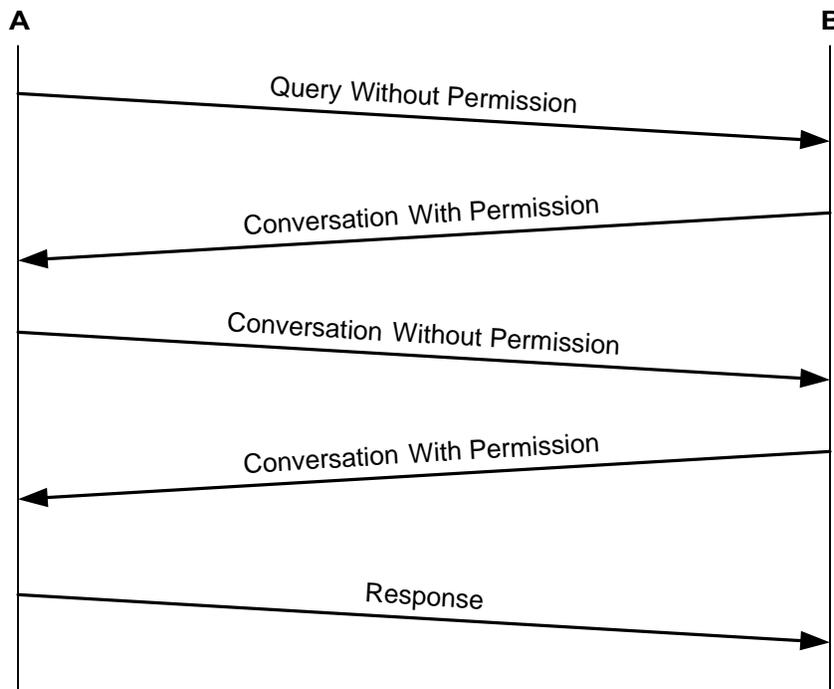


Figure 1/T1.114.4 - A simple example of exchange of TCAP messages

12 A Unidirectional message with the Reject Component could be sent.

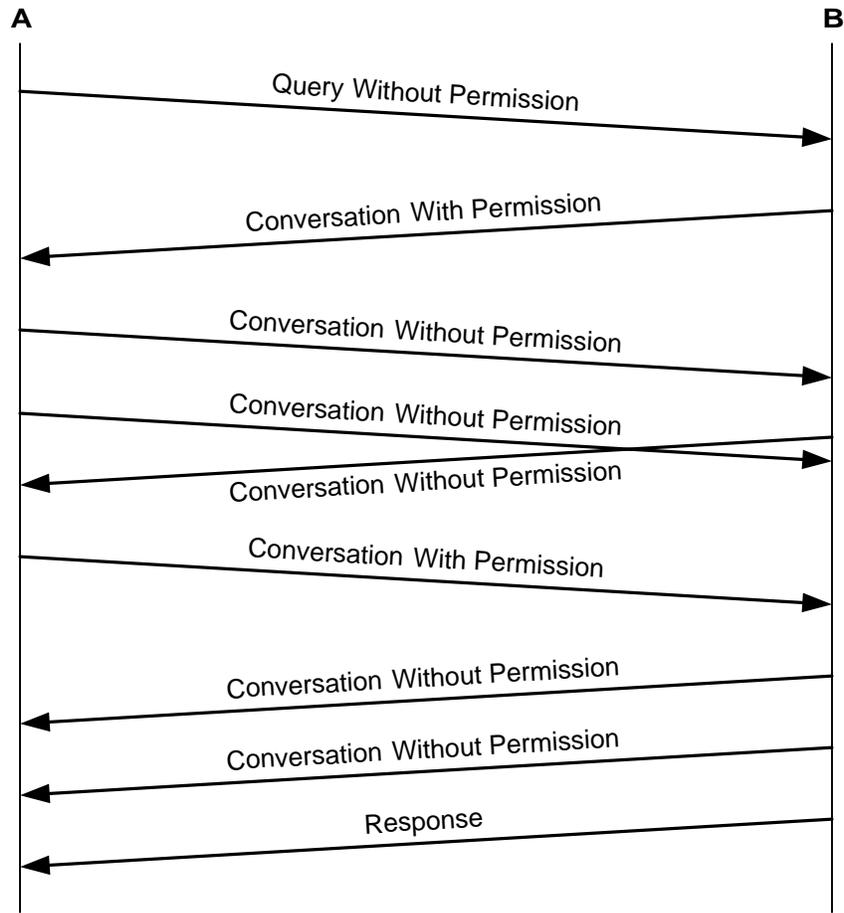


Figure 2/T1.114.4 - A complex example of exchange of TCAP messages

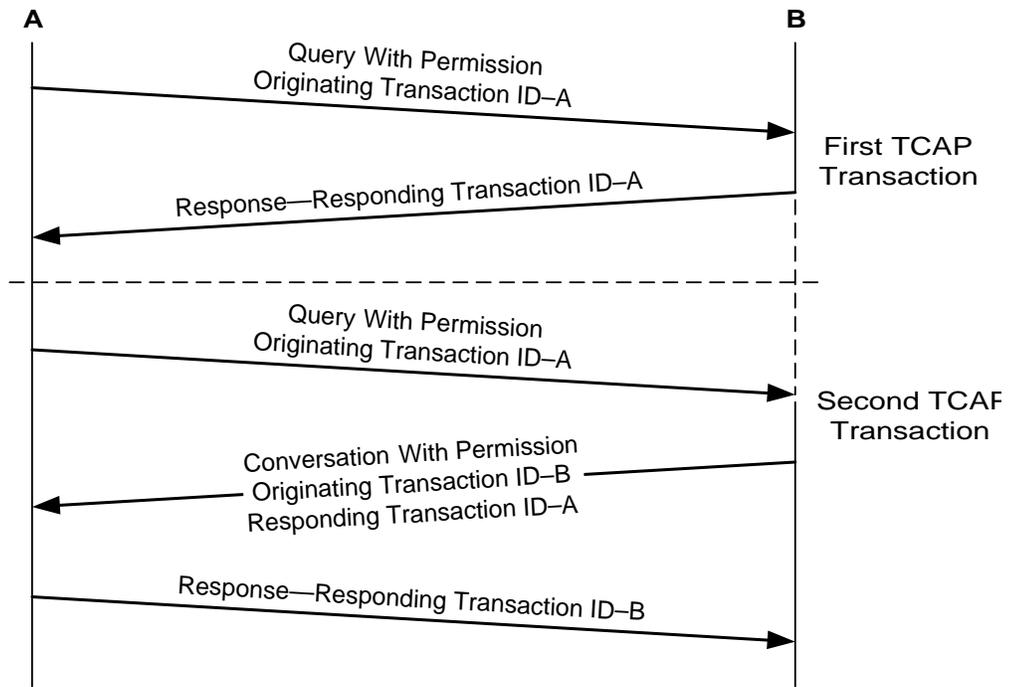


Figure 3/T1.114.4 - An example of multiple TCAP transactions

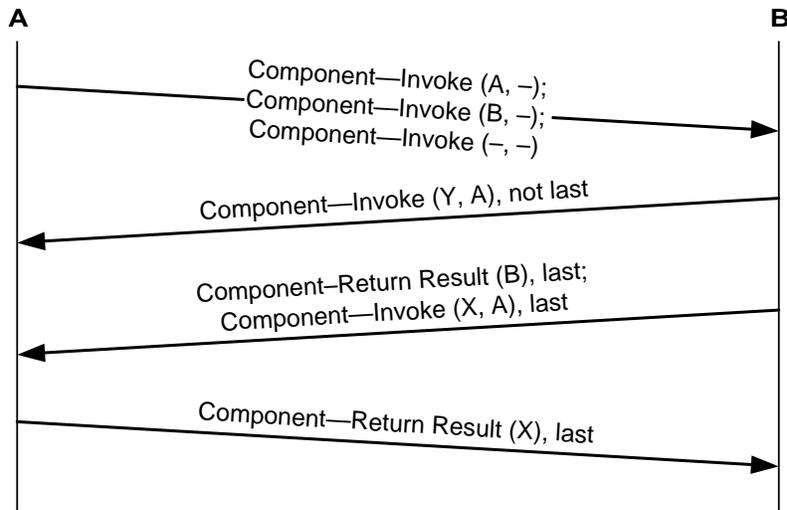


Figure 4/T1.114.4 - An example of an exchange of components in a single transaction

Chapter T1.114.5

Definition and Functions of Transaction Capabilities Operations, Parameters, and Error Codes

EDITORIAL NOTE – In this Chapter, all page numbers are in the format **T1.114.5-#**.

TABLE OF CONTENTS

1 SCOPE, PURPOSE, AND APPLICATION	1
2 OPERATIONS	1
2.1 OPERATION CODE	1
2.1.1 <i>Parameter Family – 0000001</i>	1
2.1.2 <i>Charging Family – 0000010</i>	2
2.1.3 <i>Provide Instructions Family – 0000011</i>	2
2.1.4 <i>Connection Control Family – 0000100</i>	2
2.1.5 <i>Caller Interaction Family – 0000101</i>	2
2.1.6 <i>Send Notification Family – 0000110</i>	3
2.1.7 <i>Network Management Family – 0000111</i>	3
2.1.8 <i>Procedural Family – 0001000</i>	3
2.1.9 <i>Operation Control Family – 0001001</i>	3
2.1.10 <i>Report Event Family – 0001010</i>	4
2.1.11 <i>Miscellaneous Family – 1111110</i>	4
3 ERRORS	4
3.1 ERROR CODE	4
4 PARAMETERS	5
4.1 TIMESTAMP – 0001011	6
4.2 AUTOMATIC CODE GAP (ACG) INDICATORS – 10000001	6
4.2.1 <i>Control Cause Indication</i>	6
4.2.2 <i>Duration</i>	6
4.2.3 <i>Gap</i>	6
4.3 STANDARD ANNOUNCEMENT- 10000010	7
4.4 CUSTOMIZED ANNOUNCEMENT- 10000011	7
4.5 DIGITS – 10000100	7
4.5.1 <i>Type of Digits</i>	7
4.5.2 <i>Nature of Number</i>	8
4.5.3 <i>Encoding</i>	8
4.5.4 <i>Numbering Plan</i>	8
4.5.5 <i>Number of Digits</i>	9
4.5.6 <i>Digit Representation</i>	9
4.6 STANDARD USER ERROR CODE – 10000101	9
4.7 PROBLEM DATA - 10000110	9
4.8 SCCP CALLING PARTY ADDRESS – 10000111	10
4.9 TRANSACTION ID- 10001000	10
4.10 PACKAGE TYPE – 10001001	10
4.11 SERVICE KEY – 10101010	10
4.12 BUSY/IDLE STATUS – 10001011	10
4.13 CALL FORWARDING STATUS – 10001100	10
4.14 ORIGINATING RESTRICTIONS – 10001101	11
4.15 TERMINATING RESTRICTIONS – 10001110	11
4.16 DN (DIRECTORY NUMBER) TO LINE SERVICE TYPE MAPPING – 10001111	12
4.17 DURATION- 10010000	12
4.18 RETURNED DATA- 10110001	12
4.19 BEARER CAPABILITY REQUESTED – 10010010	13
4.20 BEARER CAPABILITY SUPPORTED – 10010011	16
4.21 REFERENCE ID – 10010100	16
4.22 BUSINESS GROUP PARAMETER – 10010101	16
4.22.1 <i>Attendant Status (AttSt)</i>	17
4.22.2 <i>Business Group Identifier Type (BGID)</i>	17

Chapter T1.114.5

4.22.3	Line Privileges Information Indicator (LP II).....	17
4.22.4	Party Selector.....	17
4.22.5	Business Group ID.....	17
4.22.6	Sub-Group ID.....	18
4.22.7	Line Privileges.....	18
4.23	SIGNALLING NETWORKS IDENTIFIER – 10010110.....	18
4.24	GENERIC NAME – 10010111.....	19
4.25	MESSAGE WAITING INDICATOR TYPE – 10011000.....	19
4.26	LOOK AHEAD FOR BUSY RESPONSE – 10011001.....	19
4.26.1	Location.....	20
4.26.2	Spare.....	20
4.26.3	Acknowledgement Type.....	20
4.27	CIRCUIT IDENTIFICATION CODE – 10011010.....	20
4.27.1	Circuit Identification Code (Octet 1).....	20
4.27.2	Circuit Identification Code (Octet 2).....	21
4.27.3	Spare (Octet 2).....	21
4.28	PRECEDENCE IDENTIFIER – 10011011.....	21
4.28.1	Precedence Level (Octet 1).....	21
4.28.2	Spare (Octet 1).....	21
4.28.3	Network Identity (NI) (Octet 2 and 3).....	21
4.28.4	Service Domain (National Identifier) (Octets 4 - 6).....	22
4.29	CALL REFERENCE – 10011100.....	22
4.29.1	Call Identity (Octets 1 - 3).....	22
4.29.2	Point Code (Octets 4 - 6).....	22
4.30	AUTHORIZATION – 11011101.....	22
4.31	INTEGRITY – 11011110.....	22
4.32	SEQUENCE NUMBER - 01011111 00011111.....	22
4.33	NUMBER OF MESSAGES – 01011111 0010000.....	23
4.34	DISPLAY TEXT – 01011111 0010001.....	23
4.35	KEYEXCHANGE - 01111111 00100010.....	23
4.36	SCCP CALLED PARTY ADDRESS – 01011111 00100011.....	23
A	OPERATION FAMILY EXAMPLES.....	47
A.1	SCOPE.....	47
A.1.1	Parameter Family Operation (T1.114.5, clause 2.1.1).....	47
A.1.2	Provide Instructions Family Operation (T1.114.5, clause 2.1.3).....	48
A.1.3	Connection Control Family Operation (T1.114.5, clause 2.1.4).....	48
A.1.4	Send Notification Family Operation (T1.114.5, clause 2.1.6).....	49
A.1.5	Procedural Family Operation (T1.114.5, clause 2.1.8).....	49
A.1.6	Operation Control Family Operation (T1.114.5, clause 2.1.9).....	49
A.1.7	Report Event Family Operation (T1.114.5, clause 2.1.10).....	49
A.1.8	Miscellaneous Family Operation (T1.114.5, clause 2.1.11).....	50

TABLE OF FIGURES

FIGURE 1/T1.114.5 - NATIONAL OPERATION CODE FORMAT.....	28
FIGURE 2/T1.114.5 - NATIONAL ERROR CODE FORMAT.....	28
FIGURE 3/T1.114.5 - PARAMETER GENERAL FORMAT.....	28
FIGURE 4/T1.114.5 - TIMESTAMP PARAMETER FORMAT (YEAR, MONTH, DATE AND TIME ENCODING).....	29
FIGURE 5/T1.114.5 - ACG INDICATORS FORMAT.....	30
FIGURE 6/T1.114.5 - STANDARD ANNOUNCEMENT FORMAT.....	31
FIGURE 7/T1.114.5 - CUSTOMIZED ANNOUNCEMENT FORMAT.....	32
FIGURE 8/T1.114.5 -DIGITS FORMAT.....	32
FIGURE 9/T1.114.5 - STANDARD USER ERROR CODE FORMAT.....	35
FIGURE 10/T1.114.5 - PACKAGE TYPE FORMAT.....	35
FIGURE 11/T1.114.5 - BUSY/IDLE STATUS FORMAT.....	36
FIGURE 12/T1.114.5 - CALL FORWARDING STATUS FORMAT.....	36
FIGURE 13/T1.114.5 - ORIGINATING RESTRICTIONS FORMAT.....	37
FIGURE 14/T1.114.5 - TERMINATING RESTRICTIONS FORMAT.....	37

Chapter T1.114.5

FIGURE 15/T1.114.5 - DN TO LINE SERVICE MAPPING TYPE FORMAT 38

FIGURE 16/T1.114.5 - DURATION PARAMETER FORMAT 39

FIGURE 17/T1.114.5 - BEARER CAPABILITY REQUESTED PARAMETER FORMAT 39

FIGURE 18/T1.114.5 - BEARER CAPABILITY SUPPORTED PARAMETER FORMAT 40

FIGURE 19/T1.114.5 - BUSINESS GROUP PARAMETER FORMAT 40

FIGURE 20/T1.114.5 - SIGNALLING NETWORKS IDENTIFIER FORMAT 40

FIGURE 21/T1.114.5 - GENERIC NAME FORMAT 41

FIGURE 22/T1.114.5 - MESSAGE WAITING INDICATOR TYPE FORMAT 42

FIGURE 23/T1.114.5 - LOOK AHEAD FOR BUSY RESPONSE FORMAT 42

FIGURE 24/T1.114.5 - CIRCUIT IDENTIFICATION CODE FORMAT 43

FIGURE 25/T1.114.5 - PRECEDENCE FORMAT 43

FIGURE 26/T1.114.5 - CALL REFERENCE IDENTIFIER FORMAT 44

FIGURE 27/T1.114.5 - INTEGRITY PARAMETER IDENTIFIER FORMAT 44

FIGURE 28/T1.114.5 - SEQUENCE NUMBER IDENTIFIER FORMAT 44

FIGURE 29/T1.114.5 - NUMBER OF MESSAGES FORMAT 45

FIGURE 30/T1.114.5 - DISPLAY TEXT FORMAT 45

FIGURE 31/T1.114.5 - KEYEXCHANGE PARAMETER IDENTIFIER FORMAT 45

FIGURE 32/T1.114.5 - SCCP CALLING PARTY ADDRESS FORMAT 46

TABLE OF TABLES

TABLE 1/T1.114.5 - NATIONAL OPERATIONS 24

TABLE 2/T1.114.5 NATIONAL ERRORS 26

TABLE 3/T1.114.5 NATIONAL PARAMETERS 27

Definition and Functions of Transaction Capabilities Operations, Parameters, and Error Codes

1 SCOPE, PURPOSE, AND APPLICATION*

The Transaction Capabilities protocol format is separated into two portions, namely *Transaction* and *Component*. The *Transaction Portion* identifies whether the Transaction Capabilities Application Part (TCAP) transaction is expected to consist of single or multiple messages and provides a means to associate these messages with a specific Application Process transaction.

The *Component Portion* consists of zero, one, or more Components.

The functions and encodings for the Transaction and Component Portions are specified in Chapters T1.114.2 and T1.114.3 respectively.

This Chapter describes the functions and encodings for the Operation, Parameter, and Error Code elements used by the Transaction Capabilities protocol.

2 OPERATIONS

2.1 Operation Code

This code consists of two octets: 1) an Operation Family (bits A through G), followed by 2) an Operation Specifier (bits A through H) which indicates a specific operation within the family. Bit H of the Operation Family octet specifies whether or not a reply is expected. The operation may be defined for either national or private use (see Chapter T1.114.3, clause 4.6). When the operation is defined for national use, the Operation Code format as illustrated in Figure 1/T1.114.5 is used. When the operation is defined for private use, there is no restriction on either the length of the Operation Code or the format of the Operation Code itself. The Families and associated Specifiers are identified in Table 1/T1.114.5 and are defined and coded as follows.

2.1.1 Parameter Family - 0000001

Indicates that the following operation on Parameters is to be performed.

- (1) *00000001 - Parameter - Provide Value Specifier*. This operation is used to indicate that the values of the Parameters identified in the Parameter Set are to be provided. For an example, see Annex A, clause A.1.1.1.
- (2) *00000010 - Parameter - Set Value Specifier*. This operation indicates that the values of the Parameters identified in the Parameter Set are to be set.

* This chapter defines the TCAP data elements which are considered to be application specific. These include the applicable data elements which were originally contained in the T1.114-1988, Chapters 2 and 3.

A change bar on the right-hand margin indicates a change from T1.114.5-2000.

2.1.2 Charging Family - 0000010

This indicates that the following charging operation is to be performed.

- (1) *0000001 - Charging - Bill Call Specifier.* This operation indicates that a billing record should be made.

2.1.3 Provide Instructions Family - 0000011

This requests instructions according to the service script.

- (1) *00000001 - Provide Instructions - Start Specifier.* This operation initiates the interpretation of the service script.
- (2) *00000010 - Provide Instructions - Assist Specifier.* This operation is used to request instructions upon detection of an Assist situation. For an example, see Annex A, clause A.1.2.

2.1.4 Connection Control Family - 0000100

This indicates that the Connection Control Operation identified in the Specifier is to be performed.

- (1) *00000001 - Connection Control - Connect Specifier.* This operation indicates that the connection is to be established using the given called address and any other needed information.
- (2) *00000010 - Connection Control - Temporary Connect Specifier.* This operation is identical to "Connect" except that a "Forward Disconnect" clause 2.1.4 (4) will follow. For an example, see Annex A, clause A.1.3.
- (3) *00000011 - Connection Control - Disconnect Specifier.* This operation indicates that the connection is to be terminated.
- (4) *00000100 - Connection Control - Forward Disconnect Specifier.* This operation informs a node that it may discontinue its "Temporary Connect" to another node. For an example, see Annex A, clause A.1.3.

2.1.5 Caller Interaction Family - 0000101

This instructs the exchange to interact with the caller as indicated in the Specifier.

- (1) *00000001 - Caller Interaction - Play Announcement Specifier.* This operation identifies which announcement is to be played to the caller. Different announcements may be provided by specifying the relevant announcement identifier (e.g., including announcements in other languages).
- (2) *00000010 - Caller Interaction - Play Announcement and Collect Digits Specifier.*
This operation is identical to "Play Announcement" with the addition of digit collection from the caller.

Chapter T1.114.5

- (3) 00000011 - *Caller Interaction - Indicate Information Waiting Specifier*. This operation is intended to allow an Application Process to request another Application Process to indicate that information is waiting.
- (4) 00000100 - *Caller Interaction - Indicate Information Provided Specifier*. This operation is intended to allow an Application Process to request another Application Process to indicate that all information has been provided.

2.1.6 Send Notification Family - 0000110

This indicates that a notification of an event (such as call completion confirmation or party becoming idle) is to be sent. The event being searched for is indicated by the Specifier of the operation.

- (1) 00000001 - *Send Notification - When Party Free Specifier*. This operation indicates that the sender should be informed when party is idle. For an example, see Annex A, clause A.1.4.

2.1.7 Network Management Family - 0000111

This Family caters to Network Management operations.

- (1) 00000001 - *Network Management - Automatic Code Gap Specifier*. This operation initiates selective inhibiting of codes for a given period of time.

2.1.8 Procedural Family - 0001000

This indicates that the procedure identified in the Specifier is to be performed.

- (1) 00000001 - *Procedural - Temporary Handover Specifier*. This operation was formerly used in a Temporary Handover. It is not supported by this version of T1.114.
- (2) 00000010 - *Procedural - Report Assist Termination Specifier*. This operation indicates the end of an Assist. For an example, see Annex A, clause A.1.5.
- (3) 00000011 - *Procedural - Security*. This operation passes the Security Authorization, Integrity, Sequence Number, and Time Stamp (parameters) -- e.g., for identification and authentication, resource access control, and system access control. The specific functions performed are governed by the Security Context.

2.1.9 Operation Control Family - 0001001

This Family allows the subsequent control of operations that have been invoked. (If only one operation is outstanding within a transaction, then correlation between an operation control instruction and the invoked operation that it is to affect is not necessary). The Specifier is:

- (1) 00000001 - *Operation Control - Cancel Specifier*. This operation is sent to order or indicate the cancellation of an operation. (The only operation that may currently be cancelled is the Send Notification operation.) For an example, see Annex A, clause A.1.6.

2.1.10 Report Event Family - 0001010

This Family indicates that there has been an event occurrence at a remote location.

- (1) *00000001 - Report Event - Voice Message Available Specifier.* This operation is used to report the availability of a voice message from the exchange at which a Voice Message Storage and Retrieval (VMSR) System is connected to an exchange where the subscriber of a voice message and retrieval service is located. For an example, see Annex A, clause A.1.7.
- (2) *00000010 - Report Event - Voice Message Retrieved Specifier.* This operation is used to inform an exchange to remove the message available indicator for a VMSR subscriber. For an example, see Annex A, clause A.1.7.

2.1.11 Miscellaneous Family - 1111110

A general Operation Family that does not fit under the preceding headings.

- (1) *00000001 - Miscellaneous - Queue Call Specifier.* This operation requests that a call be placed in a queue of calls. For an example, see Annex A, clause A.1.8.
- (2) *00000010 - Miscellaneous - Dequeue Call Specifier.* This operation requests that a call be removed from a queue of calls. For an example, see Annex A, clause A.1.8.

3 ERRORS

3.1 Error Code

This provides the reason why a specific operation could not be completed successfully. An error that resulted in the unsuccessful completion of an operation may be identified for either national or private use (see Chapter T1.114.3, clause 4.9). When the errors are defined for national use, the Error Code format as illustrated in Figure 2/T1.114.5 is used. When the errors are defined for private use, there is no restriction on either the length of the Error Code or the format of the Error Code itself. The national errors are identified in Table 2/T1.114.5 and are defined and coded as follows:

- (1) *00000000 - Not Used.* This error value is not used.
- (2) *00000001 - Unexpected Component Sequence.* An incorrect sequence of Components was received (e.g., "Disconnect" followed by "Play Announcement").
- (3) *00000010 - Unexpected Data Value.* The data value was not as expected (e.g., routing number expected but billing number received).
- (4) *00000011 - Unavailable Resource.* A requested resource is not incorporated at the serving entity.
- (5) *00000100 - Missing Customer Record.* A customer record could not be located.
- (6) *00000101 - Spare code.*
- (7) *00000110 - Data Unavailable.* The data identified in the requested operation was unavailable.
- (8) *00000111 - Task Refused.* An entity normally capable of the task requested cannot or chooses not to do the task at this time.
- (9) *00001000 - Queue Full.* Indicates that the queue is currently full.

- (10) *00001001 - No Queue.* This is in response to a Miscellaneous - Queue Call operation and indicates the lack of a Queue Structure.
- (11) *00001010 - Timer Expired.* A timer associated with a specific service feature has expired.
- (12) *00001011 - Data Already Exists.* A Parameter value already exists and in order to change it a Parameter Change operation should be invoked.
- (13) *00001100 - Unauthorized Request.* The user is unauthorized to access the data base.
- (14) *00001101 - Not Queued.* This is in response to Miscellaneous - Queue Call operation and indicates that queuing has not taken place due to the choice not to queue by the application.
- (15) *00001110 - Unassigned DN.* The destination DN is not currently assigned to an active interface.
- (16) *00001111 - Spare.*
- (17) *00010000 - Notification Unavailable to Destination DN.* Notification cannot be provided to the destination for some short term reason (e.g., the line is temporarily out of service).
- (18) *00010001 - VMSR System Identification did not Match User Profile.* The destination number DN is not a customer of the identified VMSR system.
- (19) *00010010 - Security Error.* Security failure occurred (e.g., authorization failure). Security information was either missing or inaccurate. Therefore, the message was not processed.
- (20) *00010011 - Missing Parameter.* A missing parameter caused the operation to fail (e.g., a required Integrity Algorithm ID was not included in the Security operation).
- (21) *00010100 - Unexpected Parameter Sequence.* The parameters were received out of order (e.g., Time Stamp was not received before Integrity). An unexpected parameter was included in the message (e.g., an Integrity Algorithm ID was included with the Security operation but an implicitly agreed Integrity Algorithm is already in use).
- (22) *00010101 - Unexpected Message.* A message was received out of order for the application script (e.g., a request for access to a resource before a required Security operation has been invoked).
- (23) *00010110 - Unexpected Package Type.* An operation was received in a different package type than was expected (e.g., A Security operation in a Response message with no other component).

4 PARAMETERS

The Parameters associated with the Component Portion of a TCAP message are illustrated in Figure 3/T1.114.5 and defined as follows:

- (1) *Parameter Identifier* which uniquely identifies the specific Parameter that follows.
- (2) *Parameter Length* which provides the total octet length of the specified Parameter but does not include itself or the Parameter Identifier.
- (3) *Parameter Contents* which are the actual values of the Parameter described by the Parameter Identifier.

The national parameters are identified in Table 3/T1.114.5.

The Parameters and their respective formats are defined and coded as follows.

4.1 *Timestamp – 00010111*

This parameter identifies the time that an event occurred. It uses local time and gives the difference between the local time and universal coordinated time (Greenwich Mean Time, or GMT). The Timestamp Identifier is coded universal and has a primitive form. It is 15 octets long and of type VISIBLE STRING.

Its contents give the year, month, date, and time of the event as shown in Figure 4/T1.114.5.

4.2 *Automatic Code Gap (ACG) Indicators – 10000001*

These indicate the cause for applying an ACG control, also the time duration for the ACG control to be in effect and the time interval (gap) between ACG application. The one octet ACG Indicator Identifier is coded contextual (in the context of the Parameter Set) and has a primitive form.

The format of the ACG Indicator Identifier and its three octet contents are shown in Figure 5/T1.114.5 and defined and coded as below.

4.2.1 **Control Cause Indication**

This octet indicates the reason that ACG control is being initiated.

- (1) *00000001 - Vacant Code.* Calls are being received for an unassigned code.
- (2) *00000010 - Out-of-Band.* Calls are being received for a band that a customer has not subscribed.
- (3) *00000011 - Database Overload.* The database is overloaded.
- (4) *00000100 - Destination Mass Calling.* An excessive number of calls are being received for a destination.
- (5) *00000101 - Operation Support System (OSS) Initiated.* The ACG control has been externally initiated by an Operation Support System.

4.2.2 **Duration**

This octet indicates (see Figure 5/T1.114.5) the time duration in seconds that an ACG control should be applied.

4.2.3 **Gap**

This octet indicates (see Figure 5/T1.114.5) the time interval in seconds between applications of the ACG control. It also allows the ACG control to be cancelled or maintained indefinitely.

4.3 Standard Announcement- 10000010

This indicates one of the Standard announcements. The one octet Standard Announcement Identifier is coded contextual (in the context of the Parameter Set) and has a primitive form.

The format of the Standard Announcement Identifier and its one octet contents are shown in Figure 6/T1.114.5 and defined and coded as follows:

- (1) 00000000 - *Not Used*. This code value is not used.
- (2) 00000001 - *Out-of-Band*. Customer has not subscribed to this band.
- (3) 00000010 - *Vacant Code*. Unassigned code.
- (4) 00000011 - *Disconnected Number*. The called number has been disconnected.
- (5) 00000100 - *Reorder (120 Impulses per Second (IPM))*. All trunks are busy.
- (6) 00000101 - *Busy (60 IPM)*. The called number is busy.
- (7) 00000110 - *No Circuit Available*. There is no circuit available to the called number.
- (8) 00000111 - *Reorder*. An announcement such as "your call could not be completed, please try again later".
- (9) 00001000 - *Audible Ring*. The called party is being alerted.

4.4 Customized Announcement- 10000011

This indicates and provides information on an announcement that is not standard. The one octet Customized Announcement Identifier is coded contextual (in the context of the Parameter Set) and has a primitive form.

The format of the Customized Announcement Identifier and its variable length contents are shown in Figure 7/T1.114.5 and defined as follows.

- (1) *Announcement Set (User defined code)*. Identifies the specific Announcement Set from which the announcement is to be selected.
- (2) *Individual Announcement (User defined code)*. Indicates the individual announcement within the set.

4.5 Digits – 10000100

This provides information on the number of digits that follow, type (e.g., called party), nature (e.g., National), Numbering Plan (e.g., telephony), and encoding method (e.g., BCD). The Digits Identifier is coded contextual (in the context of the Parameter Set) and has a primitive form.

The format of the Digits Identifier and its variable length contents are shown in Figure 8/T1.114.5 and defined and coded as follows.

4.5.1 Type of Digits

This identifies the informational content of the digits, as follows:

ATIS-1000114.2004 (S2019)

Chapter T1.114.5

- (1) 00000000 - *Not Used*. This code value is not used.
- (2) 00000001 - *Called Party Number*. The customer dialed digits.
- (3) 00000010 - *Calling Party Number*. The Automatic Number Identification (ANI) information of the calling number.
- (4) 00000011 - *Caller Interaction*. The digits dialed by the caller after request for interaction.
- (5) 00000100 - *Routing Number*. The digits are associated with network routing.
- (6) 00000101 - *Billing Number*. The digits contain billing information.
- (7) 00000110 - *Destination Number*. The digits used to identify a particular line (i.e., I want information on the line that corresponds to this destination number).
- (8) 00000111 - *Local Access and Transport Area (LATA)*. The digits identify the LATA.
- (9) 00001000 - *Carrier*. The digits identify the carrier.
- (10) 00001001 - *Last Calling Party*. The number of the party that last called. That is the DN Information of the last calling number.
- (11) 00001010 - *Last Party Called*. The number of the party that was most recently called.
- (12) 00001011 - *Calling Directory Number*. These digits identify the number that would have to be dialed to cause a call to terminate to the calling party.
- (13) 00001100 - *VMSR Identifier*. These digits identify the VMSR system.
- (14) 00001101 - *Original Called Number*. These digits identify the number of the party who initiated the first redirection of the call.
- (15) 00001110 - *Redirecting Number*. These digits identify the number of the party who initiated the last redirection of the call.
- (16) 00001111 - *Connected Number*. These digits identify the number of the party who is connected to the calling party on the call.

4.5.2 Nature of Number

This provides additional information on the "Type of Digits" (e.g., International, presentation restriction). It is coded as a bit map as illustrated in Figure 8/T1.114.5.

4.5.3 Encoding

This indicates the digits encoding scheme.

- (1) DCBA/0000 - *Not Used*.
- (2) DCBA/0001 - *Binary Coded Decimal (BCD)*.
- (3) DCBA/0010 - *IA5*.

4.5.4 Numbering Plan

This indicates the numbering plan associated with the digits.

- (1) *HGFE/0000* - Unknown or Not Applicable.
- (2) *HGFE/0001* - ISDN Numbering Plan (CCITT Recommendation E.164).
- (3) *HGFE/0010* - Telephony Numbering Plan (CCITT Recommendation E.164).
- (4) *HGFE/0011* - Data Numbering Plan (CCITT Recommendation X.121).
- (5) *HGFE/0100* - Telex Numbering Plan (CCITT Recommendation F.69).
- (6) *HGFE/0101* - Maritime Mobile Numbering Plan (CCITT Recommendation E.210, 211).
- (7) *HGFE/0110* - Land Mobile Numbering Plan (CCITT Recommendation E.212, 213).
- (8) *HGFE/0111* - Private Numbering Plan (Service Provider Defined).

4.5.5 Number of Digits

This indicates in binary the number of digits that follow.

4.5.6 Digit Representation

Provision is made to represent digits encoded as specified in Section 4.5.3 and illustrated in Figure 8/T1.114.5.

4.6 Standard User Error Code – 10000101

The unsuccessful completion of a TCAP Transaction may be due to user error. The Standard User Error Code Identifier is coded contextual (in the context of the Parameter Set) and has a primitive form.

The format of the Standard User Error Code Identifier and its one octet contents are shown in Figure 9/T1.114.5.

The Standard User Errors are as follows.

- (1) *00000001* - *Caller Abandon*. The caller hangs up prior to completion of the TCAP Transaction.
- (2) *00000010* - *Improper Caller Response*. The caller responded incorrectly during a Caller Interaction operation – see clause 2.1, (5).

4.7 Problem Data - 10000110

This indicates the specific data that caused the problem in a TCAP Transaction. The Problem Data Identifier is coded contextual (in the context of the Parameter Set) and has a primitive form.

The format of the Problem Data Identifier and its variable length contents which contain the incorrect data element are illustrated in Figure 3/T1.114.5.

4.8 SCCP Calling Party Address – 10000111

This parameter was formerly used in a Temporary Handover. It is not supported by this version of T1.114.

4.9 Transaction ID- 10001000

This parameter was formerly used in a Temporary Handover. It is not supported by this version of T1.114.

4.10 Package Type – 10001001

This parameter was formerly used in a Temporary Handover. It is not supported by this version of T1.114.

4.11 Service Key – 10101010

This specifies the Parameters that should be used to access the record. The Service Key Identifier is coded contextual (in the context of the Parameter Set) and has a constructor form.

The format of the Service Key Identifier and its variable length contents which contain one or more parameters are illustrated in Figure 3/T1.114.5.

4.12 Busy/Idle Status – 10001011

Indicates whether the status of a line is busy or idle. The Busy/Idle Status Identifier is coded contextual (in the context of the Parameter Set) and has a primitive form.

The Busy/Idle Status Identifier and its one octet length contents are shown in Figure 11/T1.114.5. They are defined and coded as follows.

- (1) 00000001 - *Busy*.
- (2) 00000010 - *Idle*.

4.13 Call Forwarding Status – 10001100

Indicates if a call forwarding service option is active, inactive, or is not supported on a line. The Call Forwarding Status Identifier is coded contextual (in the context of the Parameter Set) and has a primitive form.

The format of the Call Forwarding Status Identifier and its one octet length contents are shown in Figure 12/T1.114.5. The options are as follows:

- (1) *Call Forwarding Variable*. A call is forwarded immediately -- i.e., prior to ringing.
- (2) *Call Forwarding on Busy*. A call is forwarded only if the line is busy.

Chapter T1.114.5

- (3) *Call Forwarding Don't Answer*. A call is forwarded only if the called line does not answer after a predetermined number of rings.
- (4) *Selective Forwarding*. This code point indicates that a call would be selectively forwarded.

4.14 Originating Restrictions – 10001101

Indicates the status of any originating restrictions on a line. The Originating Restrictions Identifier is coded contextual (in the context of the Parameter Set) and has a primitive form.

The format of the Originating Restrictions Identifier and its one octet length contents are shown in Figure 13/T1.114.5. They are defined and coded as follows:

- (1) 00000000 - *Denied Originating (DO)*. No calls are permitted to be originated.
- (2) 00000001 - *Fully Restricted Originating (FRO)*. The effect of the FRO restriction is to block direct and indirect access to parties outside the business group. A FRO line may originate intragroup calls, except to the attendant.
- (3) 00000010 - *Semi Restricted Originating (SRO)*. The effect of the SRO restriction is to block direct outgoing access to parties outside the business group, but to allow indirect outgoing access to parties outside the business group via an attendant, call forwarding, three-way calling, call transfer, and conference calling. An SRO line may originate any call, except a direct call to outside the business group.
- (4) 00000011 - *Unrestricted Originating (UO)*. There are no restrictions on the calls that may normally be originated.

4.15 Terminating Restrictions – 10001110

Indicates the status of any terminating restrictions on a line. The Terminating Restrictions Identifier is coded contextual (in the context of the Parameter Set) and has a primitive form.

The format of the Terminating Restrictions Identifier and its one octet length contents are shown in Figure 14/T1.114.5. They are defined and coded as follows.

- (1) 00000000 - *Denied Terminating (DT)*. No calls are permitted to be terminated.
- (2) 00000001 - *Fully Restricted Terminating (FRT)*. The effect of the FRT restriction is to block direct and indirect incoming access from parties outside the business group. A FRT line may receive intragroup calls (direct and indirect), except from the attendant.
- (3) 00000010 - *Semi-Restricted Terminating (SRT)*. The effect of the SRT restriction is to block direct incoming access from parties outside the business group, but to allow incoming access from parties outside the business group via an attendant, call forwarding, call pick-up, three-way calling, call transfer, and conference calling. An SRT line may receive all calls, except direct calls that originated from outside the business group.
- (4) 00000011 - *Unrestricted Terminating (UT)*. There are no restrictions on the calls that may normally be terminated.
- (5) 00000100 - *Call Rejection Applies*. The called party has requested that a call be rejected.

4.16 DN (Directory Number) to Line Service Type Mapping – 10001111

This indicates which line service type is associated with a DN and also whether a DN is applicable to both originating and terminating service. The DN (Directory Number) to Line Service Type Mapping Identifier is coded contextual (in the context of the Parameter Set) and has a primitive form.

The format of this Identifier and its one octet length contents are shown in Figure 15/T1.114.5.

The line service types are as follows.

- (1) *Individual*. Single Party Service.
- (2) *Coin*. A Pay station line.
- (3) *Multiline Hunt*. Calls to a busy line can be routed to other specified lines without assigning a DN to each line.
- (4) *PBX*. A PBX line.
- (5) *Choke*. Traffic to this DN is currently subject to network management imposed constraints.
- (6) *Series Completion*. Calls to a busy line are routed to another specified DN in the same office.
- (7) *Unassigned DN*. The DN is valid, but is not currently assigned to a customer.
- (8) *Multi-Party*. Two or more parties share the same line.
- (9) *Non-Specific*. Does not fit into any of the above categories.
- (10) *Temporarily Out-of-Service*. The DN is out of service at this time.

4.17 Duration- 10010000

This provides timing information (hours, minutes, and seconds) which permits a service to indicate its specific duration requirements for an Operation (e.g., Send Notification - When Party Free). The duration parameter identifies the time period during which the status of the Called Party is monitored. The Duration Identifier is coded contextual (in the context of the Parameter Set) and has a primitive form.

The format of the Duration Identifier and its three octet length contents are shown in Figure 16/T1.114.5.

4.18 Returned Data- 10110001

This provides the actual data that caused a problem and may be useful for diagnostic purposes. The Returned Data Identifier is coded contextual (in the context of the Parameter Set) and has a constructor form.

The format of the Returned Data Identifier and its variable length contents which contain the data element(s) with incorrect information are illustrated in Figure 3/T1.114.5.

4.19 *Bearer Capability Requested – 10010010*

This indicates which of the bearer capabilities is being requested. The Bearer Capability Requested Identifier is coded contextual (in the context of the Parameter Set) and has a primitive form.

The format of the Bearer Capability Requested parameter is shown in Figure 17/T1.114.5. The contents are defined and coded as follows:

NOTE - Not all of the capabilities coded below are supported.

(1) *Extension Indicator (ext).*

- (a) 0 - octet continues through the next octet (for example, octet 2 to 2a, 2a to 2b, 3 to 3a)
- (b) 1 - last octet

(2) *Coding Standard.*

- (a) 00 - CCITT standardized in this standard
- (b) 01 - reserved for other international standards
- (c) 10 - national standard
- (d) 11 - reserved

(3) *Information Transfer Capability.*

- (a) 00000 - speech
- (b) 01000 - unrestricted digital information
- (c) 01001 - restricted digital information (note)
- (d) 10000 - 3.1 kHz audio
- (e) 10001 - 7 kHz audio
- (f) 10010 - 15 kHz audio
- (g) 11000 - Video

All other values are reserved

NOTE - Only permitted in conjunction with 64 kbit/s information transfer rate. See CCITT Recommendation I.340, Appendix 1, for details.

(4) *Transfer Mode.*

- (a) 00 - circuit mode
- (b) 10 - packet mode

All other values are reserved.

(5) *Information Transfer Rate (octets 2 and 2b).* (see Note 1)

- (a) 00000 - channel size
- (b) 10000 - 64 kbit/s (note 2)
- (c) 10011 - 384 kbit/s
- (d) 10101 - 1536 kbit/s
- (e) 10111 - 1920 kbit/s

All other values are reserved.

00000 shall be used for packet-mode calls.

NOTES:

1. When octet 2b is omitted, the bearer capability is bi-directional symmetric at the information transfer rate specified in octet 2. When octet 2b is included, the information rate in octet 2 refers to the origination to destination direction.
2. This information transfer rate is also used for voiceband analog circuits.

(6) *Structure.*

- (a) 000 - default (note)
 - (b) 001 - 8 kHz integrity
 - (c) 100 - service data unit integrity
 - (d) 111 - unstructured
- All other values are reserved

NOTE - If octet 2a is omitted, or the structure field is coded 000, then the value of the structure attribute is according to the following:

transfer mode	transfer capability	structure
circuit	speech	8 kHz integrity
circuit	unrestricted digital	8 kHz integrity
circuit	restricted digital	8 kHz integrity
circuit	audio	8 kHz integrity
circuit	video	8 kHz integrity
packet	unrestricted digital	service data unit integrity

(7) *Configuration.*

- (a) 00 - point-to-point
- (b) 10 - multipoint

All other values are reserved. If omitted, the configuration is assumed to be point-to-point.

(8) *Establishment.*

- (a) 00 - demand

All other values are reserved. If omitted, the establishment is assumed to be on demand.

(9) *Symmetry.*

- (a) 00 - bi-directional symmetric
- (b) 01 - bi-directional asymmetric
- (c) 10 - unidirectional (origination>destination)
- (d) 11 - unidirectional (destination>origination)

If omitted, the symmetry is assumed to be bi-directional symmetric.

(10) *Multiplier or layer identification.*

- (a) 00 - bearer capability multiplier: bits 5-1 represent the number (binary encoding) of instances of bearer capability-requested; for example, 00010 means two instances of the described bearer capability are requested.
- (b) 01 - user information layer 1 protocol
- (c) 10 - user information layer 2 protocol
- (d) 11 - user information layer 3 protocol

Chapter T1.114.5

NOTE - When this subfield indicates a user information protocol, bits 5-1 of the same octet represent the corresponding identification as per items (11), (12), and (13). If octet 3 is omitted, the user information low layer protocols are assumed to be undefined. Octet 3 may be omitted if the transfer mode is "circuit-mode" and the information transfer capability is "unrestricted digital information" or "restricted digital information"; otherwise, octet 3 must be provided.

(11) *User Information layer 1 protocol identification.*

- (a) 00000 - CCITT Recommendation I.412; no additional layer 1 protocol specified for this bearer capability.
- (b) 00001 - Rate adaption: the extension bit in this octet is set to 0 and octet 3a rate subfield is coded:

Code	synchronous rate	CCITT reference
00000		
00001	0.6 kbit/s	Recommendation X.1 and I.461
00010	1.2 kbit/s	Recommendation X.1 and I.461
00011	2.4 kbit/s	Recommendation X.1 and I.461
00100	3.6 kbit/s	Recommendation V.5 and I.463
00101	4.8 kbit/s	Recommendation X.1 and I.461
00110	7.2 kbit/s	Recommendation V.5 and I.463
00111	8.0 kbit/s	Recommendation I.460
01000	9.6 kbit/s	Recommendation X.1 and I.461
01001	14.4 kbit/s	Recommendation V.5 and I.463
01010	16.0 kbit/s	Recommendation I.460
01011	19.2 kbit/s	Recommendation V.5 and I.463
01100	32.0 kbit/s	Recommendation I.460
01110	48.0 kbit/s	Recommendation X.1 and I.461
01111	56.0 kbit/s	Recommendation I.463

- (c) 00010 - CCITT Recommendation G.711 u-law
- (d) 00011 - CCITT Recommendation G.711 A-law
- (e) 00100 - CCITT Recommendation G.721 32 kbit/s ADPCM and Recommendation I.460
- (f) 00101 - CCITT Recommendation G.722, G.725, 7 kHz audio

All other values are reserved.

(12) *User Information layer 2 protocol identification.*

- (a) 00000 - undefined
- (b) 00010 - CCITT Recommendation Q.921
- (c) 00100 - CCITT Recommendation Q.710
- (d) 00110 - CCITT Recommendation X.25 link level

All other values are reserved.

(13) *User information layer 3 protocol identification.*

- (a) 00000 - undefined

(b) 00010 - CCITT Recommendation Q.931

(c) 00110 - CCITT recommendation X.25 packet level

All other values are reserved.

4.20 Bearer Capability Supported – 10010011

This indicates whether or not a requested bearer capability is supported. The Bearer Capability Supported Identifier is coded contextual (in the context of the Parameter Set) and has a primitive form.

The format of the Bearer Capability Supported Identifier and its one octet length contents are shown in Figure 18/T1.114.5.

The contents are defined and coded as follows.

- (1) 00000001 - Bearer Capability is not supported.
- (2) 00000010 - Bearer Capability is supported.
- (3) 00000011 - Not Authorized.
- (4) 00000100 - Not presently available.
- (5) 00000101 - Not implemented.

4.21 Reference ID – 10010100

This identifies the transaction between the Database and Exchange A during the Assist service. The Reference ID Identifier is coded contextual (in the context of the Parameter Set) and has a primitive form.

The format of this Identifier and its four octet length contents (number assigned by the database) are illustrated in Figure 3/T1.114.5.

4.22 Business Group Parameter – 10010101

This indicates, as required, the Multi-Location Business Group (MBG) information associated with each type of number in the message.

The numbers to which the Business Group Parameter can apply are as follows:

- (1) Calling Party Number.
- (2) Called Party Number.
- (3) Connected Party Number.
- (4) Redirecting Number.
- (5) Original Called Party Number.

The Business Group Parameter Identifier is coded contextual (in the context of the Parameter Set) and has a primitive form. The parameter is seven octets long. Its format is shown in Figure 19/T1.114.5. The contents are defined and coded as detailed in the following subclauses.

4.22.1 Attendant Status (AttSt)

Indicates (bit G) whether the party identified by the Party Selector is an Attendant.

- (1) *0 - No Indication.*
- (2) *1 - Attendant Line.*

4.22.2 Business Group Identifier Type (BGID)

Indicates (bit F) the service associated with the Business Group Identifier.

- (1) *0 - MBG Identifier.*
- (2) *1 - IWPN Identifier.*

4.22.3 Line Privileges Information Indicator (LP II)

Indicates (bit E) whether the restrictions are fixed line privileges or customer defined line privileges.

- (1) *0 - Fixed Line Privileges.*
- (2) *1 - Customer-Defined Line Privileges.*

4.22.4 Party Selector

Indicates (bits ABCD) the number to which the Business Group information applies.

- (1) *0000 - No Indication.*
- (2) *0001 - Calling Party Number.*
- (3) *0010 - Called Party Number.*
- (4) *0011 - Connected Party Number.*
- (5) *0100 - Redirecting Number.*
- (6) *0101 - Original Called Number.*

4.22.5 Business Group ID

A three octet field indicates which Business Group the party identified by the Party Selector belongs.

- (1) *0000 0000 - No Indication.*
- (2) *000 0001 - Public Network.*
- (3) *0000 0010*

to - Assigned Business Group Codes
1111 1111

4.22.6 Sub-Group ID

A two octet field defined by the customer to indicate the Sub-Group membership of the party, identified by the Party Selector, within the customer's organization.

- (1) 0000 0000 - *No Indication.*
- (2) 0000 0001
to - Customer Assigned Sub-Group codes
1111 1111.

4.22.7 Line Privileges

Defined by the customer to indicate the line privileges of the party identified by the Party Selector. If the LP II field is coded "0" Fixed Line Privileges, the LP field is divided into two subfields to represent the terminating (bits ABCD) and originating (bits EFGH) restrictions respectively.

- (1) 0000 - *Unrestricted.*
- (2) 0001 - *Semi-Restricted.*
- (3) 0010 - *Fully-Restricted.*
- (4) 0011 - *Fully-Restricted Intraswitch.*
- (5) 0100 - *Denied.*

If the "LP II" field is coded Customer-Defined Line Privileges, the Line Privileges field is coded as follows:

- (1) 0000 0000 - *No Indication*
- (2) 0000 0001
to - Customer-Defined Line Privileges codes
1111 1111

4.23 Signalling Networks Identifier – 10010110

This provides the information for identifying at least one CCS network. The network ID and Cluster Id are used to identify the CCS network. The Signalling Networks Identifier is coded contextual (in the context of the Parameter Set) and has a primitive form.

The format of the Signalling Networks identifier and its variable contents are shown in Figure 20/T1.114.5.

4.24 *Generic Name – 10010111*

This provides specific name related information. It includes indications of name availability and presentation, and may also include the characters of the name based on a network option. The Generic Name Parameter is coded contextual (in the context of the Parameter Set), and has a primitive form.

The format of the Generic Name Parameter and its variable contents are shown in Figure 21/T1.114.5 and defined as follows:

- (1) *Type of Name.*
 - (a) 000 - Spare
 - (b) 001 - Calling name
 - (c) 010 - Original called name
 - (d) 011 - Redirecting name
 - (e) 100 - Connected name
 - (f) 101-111 - Spare
- (2) *Availability.*
 - (a) 0 - Name available/unknown
 - (b) 1 - Name not available
- (3) *Presentation.*
 - (a) 00 - Presentation allowed
 - (b) 01 - Presentation restricted
 - (c) 10 - Blocking toggle
 - (d) 11 - No indication

4.25 *Message Waiting Indicator Type – 10011000*

This provides additional information to the customer about a waiting message. This information is prearranged between the service provider and the customer. The Message Waiting Indicator is coded contextual (in the context of the Parameter Set) and has a primitive form.

The format of the Message Waiting Indicator parameter and its one octet contents are shown in Figure 22/T1.114.5 and defined as follows:

- (1) *DCBA - 1st Digit.*
- (2) *HGFE - 2nd Digit.*

4.26 *Look Ahead for Busy Response – 10011001*

This indicates whether preemptable resources were found. The Look Ahead for Busy Response parameter is coded contextual (in the context of the Parameter Set) and has a primitive form.

The format of the Look Ahead for Busy parameter and its one octet length contents are shown in Figure 23/T1.114.5. It is coded and defined as follows in the following subclauses.

4.26.1 Location

This indicates the location which initiated the response.

- (1) *DCBA/0000 - User.*
- (2) *DCBA/0001 - Private Network serving the local user.*
- (3) *DCBA/0010 - Public Network serving the local user.*
- (4) *DCBA/0011 - Transit Network.*
- (5) *DCBA/0100 - Public Network serving the remote user.*
- (6) *DCBA/0101 - Private Network serving the remote user.*
- (7) *DCBA/0110 - Reserved.*
- (8) *DCBA/0111 - International Network.*
- (9) *DCBA/1000 - Network beyond interworking point.*
- (10) *DCBA/XXXX - All other values are reserved.*

4.26.2 Spare

This indicates that bits FE are spare.

4.26.3 Acknowledgement Type

This indicates whether the request for search and reservation of circuits was accepted.

- (1) *HG/00 - Path reservation denied.*
- (2) *HG/01 - Negative Acknowledgement.*
- (3) *HG/10 - Positive Acknowledgement.*
- (4) *HG/11 - Spare.*

4.27 Circuit Identification Code - 10011010

This is used to identify the physical path between two exchanges. The Circuit Identification Code parameter is coded contextual (in the context of the Parameter Set) and has a primitive form.

The format of the Circuit Identification Code parameter and its two octet length contents are shown in Figure 24/T1.114.5.

The contents are defined and coded as follows.

4.27.1 Circuit Identification Code (Octet 1)

This octet indicates the least significant bits of the Circuit Identification Code.

- (1) *HGFEDCBA/XXXXXXXX - Least Significant Bits.*

4.27.2 Circuit Identification Code (Octet 2)

This octet indicates the most significant bits of the Circuit Identification Code.

- (1) *FEDCBA/XXXXXX - Most Significant Bits.*

4.27.3 Spare (Octet 2)

This indicates that bits H and G of the second octet are spare.

4.28 Precedence Identifier – 10011011

This is used to identify the MLPP call in terms of priority treatment and MLPP Service Domain. The Precedence parameter is coded contextual (in the context of the Parameter Set) and has a primitive form.

The format of the Precedence parameter and its six octet length contents are shown in Figure 25/T1.114.5.

The contents are defined and coded as follows.

4.28.1 Precedence Level (Octet 1)

This indicates the Precedence Level of the call.

- (1) *DCBA/0000 - Flash Override.*
- (2) *DCBA/0001 - Flash.*
- (3) *DCBA/0010 - Immediate.*
- (4) *DCBA/0011 - Priority.*
- (5) *DCBA/0100 - Routine.*

4.28.2 Spare (Octet 1)

This indicates that the following bits of octet 1 are spare.

- (1) *HGFE/0000 - Spare bits.*

4.28.3 Network Identity (NI) (Octet 2 and 3)

Each digit is coded in binary coded decimal representation from 0 to 9. The first digit of this field is coded 0. The TCC (Telephony Country Code) follows in the second to fourth NI digits (the most significant TCC digit is the second NI digit). If the TCC is one or two digits long, the excess digit(s) is inserted with the code for RPOA or network identification, if necessary. If octet 2 is not required, it is coded all zeros.

4.28.4 Service Domain (National Identifier) (Octets 4 - 6)

This is a code expressing in pure binary representation the number allocated to a national MLPP Service domain. These numbers are allocated in accordance with the procedures in T1.113.3-2000, Annex B.

4.29 Call Reference – 10011100

This is used to identify a particular MLPP call independent of the physical circuit. The Call Reference parameter is coded contextual (in the context of the Parameter Set) and has a primitive form. The format of the Call Reference parameter and its six octet length contents are shown in Figure 26/T1.114.5.

The contents are defined and coded as follows.

4.29.1 Call Identity (Octets 1 - 3)

These octets indicate the identification number allocated to the call, in pure binary representation.

4.29.2 Point Code (Octets 4 - 6)

These octets indicate the code of the signalling point in which the call identity is relevant.

4.30 Authorization – 11011101

The Authorization contains context specific information for identification and authentication of the sending entity (e.g., login ID, password). The duration of an agreed identification or authentication is controlled by the application granting the identification or authentication.

4.31 Integrity – 11011110

The Integrity parameter contains the information within a specific context that should be used in the destination CCS node to check whether or not the received message has been modified.

The format of the Integrity Identifier and its variable length contents are shown in Figure 27/T1.114.5.

4.32 Sequence Number - 01011111 00011111

The Sequence Number uniquely identifies a message in a sequence of messages. This information may be used to detect message insertions, deletions or replays in a transaction where messages are exchanged.

The format of the Sequence Number is shown in Figure 28/T1.114.5. The context is an integer.

4.33 *Number of Messages – 01011111 0010000*

The Number of Messages parameter contains information about the number of messages that are waiting in a Message Storage and Retrieval System for a client user.

The format of the Number of Messages parameter is shown in Figure 29/T1.114.5. The context is an integer.

4.34 *Display Text – 01011111 0010001*

The Display Text parameter contains text information about messages that are waiting in a Message Storage and Retrieval System for a client user. This text information may be presented to the client user as part of Message Waiting Notification (MWN).

The format of the Display Text parameter and its variable length contents are shown in Figure 30/T1.114.5.

4.35 *KeyExchange - 01111111 00100010*

The KeyExchange parameter contains the information within a specific context that can be used for the exchange of cryptographic keys.

The format of the KeyExchange Identifier and its variable length contents are shown in Figure 31/T1.114.5.

4.36 *SCCP Called Party Address – 01011111 00100011*

The SCCP Called Party Address parameter limits control to only those calls that send queries to the specified SCCP address. It is a variable length parameter of type OCTET STRING. The contents of this parameter use the standard SCCP address encoding given in 3.4 of Chapter T1.112.3 of T1.112-2001. See Figure 32/T1.114.5.

Table 1/T1.114.5 - National Operations

Operation Name		Operation Code														
		* Family							Specifier							
Family	Specifier	G	F	E	D	C	B	A	H	G	F	E	D	C	B	A
All Families	Reserved	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
All Families	Not Used	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Parameter	Provide Value	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
Parameter	Set Value	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0
Charging	Bill Call	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1
Provide Instructions	Start	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1
Provide Instructions	Assist	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0
Connection Control	Connect	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1
Connection Control	Temporary Connect	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0
Connection Control	Disconnect	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1
Connection Control	Forward Disconnect	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0
Caller Interaction	Play Announcement (PA)	0	0	0	0	1	0	1	0	0	0	0	0	0	0	1
Caller Interaction	PA and Collect Digits	0	0	0	0	1	0	1	0	0	0	0	0	0	1	0
Caller Interaction	Indicate Information Waiting	0	0	0	0	1	0	1	0	0	0	0	0	0	1	1
Caller Interaction	Indicate Information Provided	0	0	0	0	1	0	1	0	0	0	0	0	1	0	0
Send Notification	When Party Free	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1
Network Management	Automatic Code Gap	0	0	0	0	1	1	1	0	0	0	0	0	0	0	1
Procedural	Temporary Handover **	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0
Procedural	Report Assist Termination	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
Procedural	Security	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1

* Bit H is used to distinguish between Operations that require a reply and those that do not. A value of 1 indicates that a reply is required; a value of 0 indicates that a reply is not required.

** This operation was formerly used in a Temporary Handover. It is not supported by this version of T1.114.

Table 1/T1.114.5 - (continued) National Operations

Operation Name		Operation Code														
		* Family							Specifier							
Family	Specifier	G	F	E	D	C	B	A	H	G	F	E	D	C	B	A
Operation Control	Cancel	0	0	0	1	0	0	1	0	0	0	0	0	0	0	1
Report Event	Voice Message Available	0	0	0	1	0	1	0	0	0	0	0	0	0	0	1
Report Event	Voice Message Retrieved	0	0	0	1	0	1	0	0	0	0	0	0	0	1	0
Spare																
Miscellaneous	Queue Call	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1
Miscellaneous	Dequeue Call	1	1	1	1	1	1	0	0	0	0	0	0	0	1	0
Reserved		1	1	1	1	1	1	1								

* Bit H is used to distinguish between Operations that require a reply and those that do not. A value of 1 indicates that a reply is required; a value of 0 indicates that a reply is not required.

ATIS-1000114.2004 (S2019)

Chapter T1.114.5

Table 2/T1.114.5 National Errors

Error Name	Error Code							
	H	G	F	E	D	C	B	A
Not used	0	0	0	0	0	0	0	0
Unexpected Component Sequence	0	0	0	0	0	0	0	1
Unexpected Data Value	0	0	0	0	0	0	1	0
Unavailable Resource	0	0	0	0	0	0	1	1
Missing Customer Record	0	0	0	0	0	1	0	0
Spare	0	0	0	0	0	1	0	1
Data Unavailable	0	0	0	0	0	1	1	0
Task Refused	0	0	0	0	0	1	1	1
Queue Full	0	0	0	0	1	0	0	0
No Queue	0	0	0	0	1	0	0	1
Timer Expired	0	0	0	0	1	0	1	0
Data Already Exists	0	0	0	0	1	0	1	1
Unauthorized Request	0	0	0	0	1	1	0	0
Not Queued	0	0	0	0	1	1	0	1
Unassigned DN	0	0	0	0	1	1	1	0
Spare	0	0	0	0	1	1	1	1
Notification Unavailable to Destination DN	0	0	0	1	0	0	0	0
VMSR System Identification did not Match User Profile	0	0	0	1	0	0	0	1
Security Error	0	0	0	1	0	0	1	0
Missing Parameter	0	0	0	1	0	0	1	1
Unexpected Parameter Sequence	0	0	0	1	0	1	0	0
Unexpected Message	0	0	0	1	0	1	0	1
Unexpected Package Type	0	0	0	1	0	1	1	0

Table 3/T1.114.5 National Parameters

Parameter Name	Identifier Code							
	H	G	F	E	D	C	B	A
Timestamp	0	0	0	1	0	1	1	1
ACG Indicators	1	0	0	0	0	0	0	1
Standard Announcement	1	0	0	0	0	0	1	0
Customized Announcement	1	0	0	0	0	0	1	1
Digits	1	0	0	0	0	1	0	0
Standard User Error Code	1	0	0	0	0	1	0	1
Problem Data	1	0	0	0	0	1	1	0
SCCP Calling Party Address *	1	0	0	0	0	1	1	1
Transaction ID *	1	0	0	0	1	0	0	0
Package Type *	1	0	0	0	1	0	0	1
Service Key	1	0	0	0	1	0	1	0
Busy/Idle Status	1	0	0	0	1	0	1	1
Call Forwarding Status	1	0	0	0	1	1	0	0
Originating Restrictions	1	0	0	0	1	1	0	1
Terminating Restrictions	1	0	0	0	1	1	1	0
DN to Line Service Type Mapping	1	0	0	0	1	1	1	1
Duration	1	0	0	1	0	0	0	0
Returned Data	1	0	0	1	0	0	0	1
Bearer Capability Requested	1	0	0	1	0	0	1	0
Bearer Capability Supported	1	0	0	1	0	0	1	1
Reference ID	1	0	0	1	0	1	0	0
Business Group	1	0	0	1	0	1	0	1
Signalling Networks Identifier	1	0	0	1	0	1	1	0
Generic Name	1	0	0	1	0	1	1	1
Message Waiting Indicator Type	1	0	0	1	1	0	0	0
Look Ahead for Busy	1	0	0	1	1	0	0	1
Circuit Identification Code	1	0	0	1	1	0	1	0
Precedence Identifier	1	0	0	1	1	0	1	1
Call Reference Identifier	1	0	0	1	1	1	0	0
Authorization	1	0	1	1	1	1	0	1
Integrity	1	0	1	1	1	1	1	0

* This parameter was formerly used in a Temporary Handover. It is not supported in this version of T1.114.

Table 3/T1.114.5 - (continued) National Parameters

Parameter Name	Identifier Code							
	H	G	F	E	D	C	B	A
Sequence Number	1	0	0	1	1	1	1	1
	0	0	0	1	1	1	1	1
Number of Messages	1	0	0	1	1	1	1	1
	0	0	1	0	0	0	0	0
Display Text	1	0	0	1	1	1	1	1
	0	0	1	0	0	0	0	1
Key Exchange	1	0	1	1	1	1	1	1
	0	0	1	0	0	0	1	0
SCCP Called Party Address	1	0	0	1	1	1	1	1
	0	0	1	0	0	0	1	1

H	G	F	E	D	C	B	A
* OPERATION FAMILY							
OPERATION SPECIFIER							

* Bit H is used to distinguish between Operations that require a reply and those that do not.

Bit H	Reply Required
1	Yes
0	No

Figure 1/T1.114.5 - National Operation Code Format

When a reply is required, a reply must be sent. When a reply is not required, a reply may be sent based on the outcome of the Operation or other criteria.

H	G	F	E	D	C	B	A
ERROR CODE							

Figure 2/T1.114.5 - National Error Code Format

H	G	F	E	D	C	B	A
PARAMETER TYPE IDENTIFIER							
PARAMETER TYPE LENGTH							
PARAMETER TYPE CONTENTS							

Figure 3/T1.114.5 - Parameter General Format

ATIS-1000114.2004 (S2019)

Chapter T1.114.5

	H	G	F	E	D	C	B	A
Timestamp	0	0	0	1	0	1	1	1

The Timestamp length is 15 octets. The contents fields are as follows:

Contents	H	G	F	E	D	C	B	A
Octets 1-2	YY							
Octets 3-4	MM							
Octets 5-6	DD							
Octets 7-8	hh							
Octets 9-10	mm							
Octet 11	- or +							
Octets 12-13	hh							
Octets 14-15	mm							

Where YY = Year (e.g., 88), MM = Month (e.g., July = 07), DD = Date, first hh and mm = hours and minutes in local time (e.g., 5:30 PM = 1730) and second hh and mm gives the difference between local time and GMT (e.g, Atlanta GA is five hours behind = -0500). Each character is coded in one octet, therefore each item above requires 2 octets.

For CCITT Recommendation T.61 encoding, the digits are coded as shown below.

Digits	H	G	F	E	D	C	B	A
Digit 0	0	0	1	1	0	0	0	0
Digit 1	0	0	1	1	0	0	0	1
Digit 2	0	0	1	1	0	0	1	0
Digit 3	0	0	1	1	0	0	1	1
Digit 4	0	0	1	1	0	1	0	0
Digit 5	0	0	1	1	0	1	0	1
Digit 6	0	0	1	1	0	1	1	0
Digit 7	0	0	1	1	0	1	1	1
Digit 8	0	0	1	1	1	0	0	0
Digit 9	0	0	1	1	1	0	0	1
+	0	0	1	0	1	0	1	1
-	0	0	1	0	1	1	0	1

Figure 4/T1.114.5 - Timestamp Parameter Format (Year, Month, Date and Time Encoding)

ATIS-1000114.2004 (S2019)

Chapter T1.114.5

	H	G	F	E	D	C	B	A
ACG Indicators Identifier	1	0	0	0	0	0	0	1

The ACG Indicators length is 3 octets. The contents fields are as follows:

Control Cause Indication
Duration
Gap

The Control Cause Indication field is coded as follows:

Control Cause Indication	H	G	F	E	D	C	B	A
Vacant Code	0	0	0	0	0	0	0	1
Out-of-Band	0	0	0	0	0	0	1	0
Database Overload	0	0	0	0	0	0	1	1
Destination Mass Calling	0	0	0	0	0	1	0	0
Operation Support System Initiated	0	0	0	0	0	1	0	1

The Duration field is coded as follows:

Duration	H	G	F	E	D	C	B	A
Not Used	0	0	0	0	0	0	0	0
1 Second	0	0	0	0	0	0	0	1
2 Seconds	0	0	0	0	0	0	1	0
4 Seconds	0	0	0	0	0	0	1	1
8 Seconds	0	0	0	0	0	1	0	0
16 Seconds	0	0	0	0	0	1	0	1
32 Seconds	0	0	0	0	0	1	1	0
64 Seconds	0	0	0	0	0	1	1	1
128 Seconds	0	0	0	0	1	0	0	0
256 Seconds	0	0	0	0	1	0	0	1
512 Seconds	0	0	0	0	1	0	1	0
1024 Seconds	0	0	0	0	1	0	1	1
2048 Seconds	0	0	0	0	1	1	0	0

Figure 5/T1.114.5 - ACG Indicators Format

ATIS-1000114.2004 (S2019)

Chapter T1.114.5

The Gap field is coded as follows:

Gap	H	G	F	.E	D	C	B	A
Remove Gap Control	0	0	0	0	0	0	0	0
0.00 Seconds	0	0	0	0	0	0	0	1
0.10 Seconds	0	0	0	0	0	0	1	0
0.25 Seconds	0	0	0	0	0	0	1	1
0.50 Seconds	0	0	0	0	0	1	0	0
1.00 Seconds	0	0	0	0	0	1	0	1
2.00 Seconds	0	0	0	0	0	1	1	0
5.00 Seconds	0	0	0	0	0	1	1	1
10.00 Seconds	0	0	0	0	1	0	0	0
15.00 Seconds	0	0	0	0	1	0	0	1
30.00 Seconds	0	0	0	0	1	0	1	0
60.00 Seconds	0	0	0	0	1	0	1	1
120.00 Seconds	0	0	0	0	1	1	0	0
300.00 Seconds	0	0	0	0	1	1	0	1
600.00 Seconds	0	0	0	0	1	1	1	0
Stop All Calls	0	0	0	0	1	1	1	1

Figure 5/T1.114.5 (continued) - ACG Indicators Format

	H	G	F	E	D	C	B	A
Standard Announcement Identifier	1	0	0	0	0	0	1	0

The Standard Announcement length is 1 octet. The contents are coded as follows:

Standard Announcement Identifier	H	G	F	E	D	C	B	A
Not Used	0	0	0	0	0	0	0	0
Out-of-Band	0	0	0	0	0	0	0	1
Vacant Code	0	0	0	0	0	0	1	0
Disconnected Number	0	0	0	0	0	0	1	1
Reorder (120 IPM)	0	0	0	0	0	1	0	0
Busy (60 IPM)	0	0	0	0	0	1	0	1
No Circuit Available	0	0	0	0	0	1	1	0
Reorder	0	0	0	0	0	1	1	1
Audible Ring	0	0	0	0	1	0	0	0

Figure 6/T1.114.5 - Standard Announcement Format

	H	G	F	E	D	C	B	A
Customized Announcement Identifier	1	0	0	0	0	0	1	1

The Customized Announcement length is variable. The contents are coded as follows:

	H	G	F	E	D	C	B	A
Announcement Set	Defined by the user							
Announcement ID (1)	Defined by the user							
Announcement ID (2)	Defined by the user							

Figure 7/T1.114.5 - Customized Announcement Format

	H	G	F	E	D	C	B	A
Digits Identifier	1	0	0	0	0	1	0	0

The Digits length is variable. The contents are as follows:

Type of Digits	
Nature of Number	
Numbering Plan	Encoding
Number of Digits	
Digits	

Figure 8/T1.114.5 -Digits Format

ATIS-1000114.2004 (S2019)

Chapter T1.114.5

The Type of Digits field is coded as follows:

Type of Digits	H	G	F	E	D	C	B	A
Not Used	0	0	0	0	0	0	0	0
Called Party Number	0	0	0	0	0	0	0	1
Calling Party Number	0	0	0	0	0	0	1	0
Caller Interaction	0	0	0	0	0	0	1	1
Routing Number	0	0	0	0	0	1	0	0
Billing Number	0	0	0	0	0	1	0	1
Destination Number	0	0	0	0	0	1	1	0
LATA	0	0	0	0	0	1	1	1
Carrier	0	0	0	0	1	0	0	0
Last Calling Party	0	0	0	0	1	0	0	1
Last Party Called	0	0	0	0	1	0	1	0
Calling Directory Number	0	0	0	0	1	0	1	1
VMSR Identifier	0	0	0	0	1	1	0	0
Original Called Number	0	0	0	0	1	1	0	1
Redirecting Number	0	0	0	0	1	1	1	0
Connected Number	0	0	0	0	1	1	1	1

The Nature of Number field is coded as follows:

Values	Bit	Value
National	A	0
International	A	1
No Presentation Restricted	B	0
Presentation Restricted	B	1

The Encoding field is coded as follows:

Encoding	D	C	B	A
Not Used	0	0	0	0
BCD	0	0	0	1
IA5	0	0	1	0

Figure 8/T1.114.5 (continued) - Digits Format

ATIS-1000114.2004 (S2019)

Chapter T1.114.5

The Numbering Plan field is coded as follows:

Numbering Plan	H	G	F	E
Unknown or Not applicable	0	0	0	0
ISDN Numbering	0	0	0	1
Telephony Numbering	0	0	1	0
Data Numbering	0	0	1	1
Telex Numbering	0	1	0	0
Maritime Mobile Numbering	0	1	0	1
Land Mobile Numbering	0	1	1	0
Private Numbering Plan	0	1	1	1

The Number of Digits field is binary coded to indicate the number of digits in the Digits field.

If the encoding is set to BCD, the digits are coded as follows:

2nd Digit	1st Digit
nth Digit	(n-1)th Digit

The (BCD) digits are coded as follows:

Digit	H/D	G/C	F/B	E/A
Digit 0 or filler	0	0	0	0
Digit 1	0	0	0	1
Digit 2	0	0	1	0
Digit 3	0	0	1	1
Digit 4	0	1	0	0
Digit 5	0	1	0	1
Digit 6	0	1	1	0
Digit 7	0	1	1	1
Digit 8	1	0	0	0
Digit 9	1	0	0	1
Spare	1	0	1	0
Code 11	1	0	1	1
Code 12	1	1	0	0
*	1	1	0	1
#	1	1	1	0
ST	1	1	1	1

Figure 8/T1.114.5 (continued) - Digits Format

	H	G	F	E	D	C	B	A
Standard User Code Identifier	1	0	0	0	0	1	0	1

The Standard User Error Code length is 1 octet. The contents are coded as follows:

Standard User Error Code Identifier	H	G	F	E	D	C	B	A
Caller Abandon	0	0	0	0	0	0	0	1
Improper Caller Response	0	0	0	0	0	0	1	0

Figure 9/T1.114.5 - Standard User Error Code Format

Package Type identifier	H	G	F	E	D	C	B	A
Unidirectional	1	1	1	0	0	0	0	1
Query with Permission	1	1	1	0	0	0	1	0
Query without Permission	1	1	1	0	0	0	1	1
Response	1	1	1	0	0	1	0	0
Conversation with Permission	1	1	1	0	0	1	0	1
Conversation without Permission	1	1	1	0	0	1	1	0
Abort	1	1	1	0	0	1	1	1

Figure 10/T1.114.5 - Package Type Format

This figure was formerly used in a Temporary Handover. It is not supported in this version of T1.114.

ATIS-1000114.2004 (S2019)

Chapter T1.114.5

	H	G	F	E	D	C	B	A
Busy/Idle Status Identifier	1	0	0	0	1	0	1	1

The Busy/Idle Status length is 1 octet. The contents are coded as follows:

Busy/Idle Status Identifier	H	G	F	E	D	C	B	A
Busy	0	0	0	0	0	0	0	1
Idle	0	0	0	0	0	0	1	0

Figure 11/T1.114.5 - Busy/Idle Status Format

	H	G	F	E	D	C	B	A
Call Forwarding Status Identifier	1	0	0	0	1	1	0	0

The Call Forwarding Status length is 1 octet. The contents are coded as follows:

Bits	Service
HG	Call Forwarding Variable
FE	Call Forwarding on Busy
DC	Call Forwarding Don't Answer
BA	Selective Forwarding

Bit Values indicate status as follows:

Bit Value	Status
00	Service not supported
01	Active
01	Not Active
11	Spare

Figure 12/T1.114.5 - Call Forwarding Status Format

	H	G	F	E	D	C	B	A
Originating Restrictions Identifier	1	0	0	0	1	1	0	1

The Originating Restrictions length is 1 octet. The contents are coded as follows:

Originating Restrictions Identifier	H	G	F	E	D	C	B	A
Denied Origination	0	0	0	0	0	0	0	0
Fully Restricted Origination	0	0	0	0	0	0	0	1
Semi-Restricted Origination	0	0	0	0	0	0	1	0
Unrestricted Origination	0	0	0	0	0	0	1	1

Figure 13/T1.114.5 - Originating Restrictions Format

	H	G	F	E	D	C	B	A
Terminating Restrictions Identifier	1	0	0	0	1	1	1	0

The Terminating Restrictions length is 1 octet. The contents are coded as follows:

Terminating Restrictions Identifier	H	G	F	E	D	C	B	A
Denied Termination	0	0	0	0	0	0	0	0
Fully Restricted Termination	0	0	0	0	0	0	0	1
Semi-Restricted Termination	0	0	0	0	0	0	1	0
Unrestricted Termination	0	0	0	0	0	0	1	1
Call Rejection Applies	0	0	0	0	0	1	0	0

Figure 14/T1.114.5 - Terminating Restrictions Format

	H	G	F	E	D	C	B	A
DN to Line Service Type Mapping Identifier	1	0	0	0	1	1	1	1

ATIS-1000114.2004 (S2019)

Chapter T1.114.5

The DN to Line Service Type Mapping length is 1 octet. The contents are coded as follows:

Match Status	H	G
Spare	0	0
No Match	0	1
Match	0	0
Spare	0	1

Line Service Type	F	E	D	C	B	A
Individual	0	0	0	0	0	0
Coin	0	0	0	0	0	1
Multiline Hunt	0	0	0	0	1	0
PBX	0	0	0	0	1	1
Choke	0	0	0	1	0	0
Series Completion	0	0	0	1	0	1
Unassigned DN	0	0	0	1	1	0
Multi-Party	0	0	0	1	1	1
Non-specific	0	0	1	0	0	0
Temporarily out of Service	0	0	1	0	0	1

Figure 15/T1.114.5 - DN to Line Service Mapping Type Format

	H	G	F	E	D	C	B	A
Duration Identifier	1	0	0	1	0	0	0	0

The Duration parameter length is 3 octets. The contents are coded as follows:

H	G	F	E	D	C	B	A
Hours				Hours			
Minutes				Minutes			
Seconds				Seconds			

Figure 16/T1.114.5 - Duration Parameter Format

	8	7	6	5	4	3	2	1
1	1	Coding Standard			Information transfer capability			
2	ext	Transfer mode			Information transfer rate			
2a	ext	Structure			Configuration		Establishment	
2b	1	Symmetry			Information transfer rate (destination to origination)			
3	ext	Multiplier or layer identification			Bearer capability multiplier/ Protocol identification			
3a	1	Spare			Rate			

Figure 17/T1.114.5 - Bearer Capability Requested Parameter Format

NOTES:

- (1) Octet 2a may be omitted, except when 2b is present
- (2) Octet 2b may be omitted.
- (3) Octet 3 may be omitted. It may also be repeated; for example to identify protocols at one or more layers.
- (4) Octet 3a is only present if octet 3 indicates rate adaptation.

	H	G	F	E	D	C	B	A
Bearer Capability Supported Identifier	1	0	0	1	0	0	1	1

The Bearer Capability Supported length is 1 octet. The contents are coded as follows:

Bearer Capability Supported Identifier	H	G	F	E	D	C	B	A
Bearer Capability is not Supported	0	0	0	0	0	0	0	1
Bearer Capability is Supported	0	0	0	0	0	0	1	0
Bearer Capability Not Authorized	0	0	0	0	0	0	1	1
Bearer Capability Not Presently Available	0	0	0	0	0	1	0	0
Bearer Capability Not Implemented	0	0	0	0	0	1	0	1

Figure 18/T1.114.5 - Bearer Capability Supported Parameter Format

H	G	F	E	D	C	B	A
Business Group Parameter Identifier							
Parameter Length							
Spare	AttSt	BGID	LP11	Party Selector			
Business Group ID							
Sub-Group ID							
Line Privileges							

Figure 19/T1.114.5 - Business Group Parameter Format

	H	G	F	E	D	C	B	A
Signalling Networks Identifier	1	0	0	1	0	1	1	0

The Signalling Networks length field is variable. It is 2 octets per network in the list that follows:

Signalling Networks Identifier	H	G	F	E	D	C	B	A
Network ID 1								
Network ID 2								
"								
"								
Network ID n								

Figure 20/T1.114.5 - Signalling Networks Identifier Format

ATIS-1000114.2004 (S2019)

Chapter T1.114.5

The Networks Identifier octets notify the receiving application about the networks the sending application anticipates will be traversed.

	H	G	F	E	D	C	B	A
Generic Name Identifier	1	0	0	1	0	1	1	1

The Generic Name length is variable. The contents are coded as follows:

H	G	F	E	D	C	B	A
Type of Name			Availability	Spare		Presentation	
character 1							
character 2							
character 3							

The characters are IA5 encoded and can be up to 15 octets.

The Type of Name field is coded as follows:

Type of Name	Value
Spare	000
Calling name	001
Original called name	010
Redirecting name	011
Connected name	100
Spare	101 - 111

The Availability field is coded as follows:

Availability	Value
Name available/unknown	0
Name not available	1

The Presentation field is coded as follows:

Presentation	Value
Presentation Allowed	00
Presentation Restricted	01
Blocking Toggle	10
No Indication	11

Figure 21/T1.114.5 - Generic Name Format.

	H	G	F	E	D	C	B	A
Message Waiting Indicator Type Identifier	1	0	0	1	1	0	0	0

The Message Waiting Indicator Type length is 1 octet. The contents are coded as follows:

Message Waiting Indicator Type	H	G	F	E	D	C	B	A
Contents	2nd Digit				1st Digit			

Figure 22/T1.114.5 - Message Waiting Indicator Type Format

	H	G	F	E	D	C	B	A
Look Ahead for Busy Response Identifier	1	0	0	1	1	0	0	1

The Look Ahead for Busy Response length is 1 octet. The contents are coded as follows:

Look Ahead for Busy Response	H	G	F	E	D	C	B	A
Contents	Ack. Type		Spare		Location			

The Location field is coded as follows:

Location	D	C	B	A
User	0	0	0	0
Private Network Serving the local user	0	0	0	1
Public Network Serving the local user	0	0	1	0
Transit Network	0	0	1	1
Public Network serving the remote user	0	1	0	0
Private Network serving the remote user	0	1	0	1
Reserved	0	1	1	0
International Network	0	1	1	1
Network beyond interworking point	1	0	0	0
All other values are reserved				

Bits FE are spare.

The Acknowledgment Type field is coded as follows:

Acknowledgment Type	H	G
Path Reservation Denied	0	0
Negative Acknowledgment	0	1
Positive Acknowledgment	1	0
Spare	1	1

Figure 23/T1.114.5 - Look Ahead for Busy Response Format

ATIS-1000114.2004 (S2019)

Chapter T1.114.5

	H	G	F	E	D	C	B	A
Circuit Identification Code Identifier	1	0	0	1	1	0	1	0

The Circuit Identification Code length is 2 octets. The contents are coded as follows:

Circuit Identification Code (CIC)	H	G	F	E	D	C	B	A
Octet 1	CIC Least Significant Bits							
Octet 2	Spare		CIC Most Significant Bits					

Figure 24/T1.114.5 - Circuit Identification Code Format

	H	G	F	E	D	C	B	A
Precedence Identifier	1	0	0	1	1	0	1	1

The Precedence length is 6 octets. The contents are coded as follows:

	H	G	F	E	D	C	B	A
Octet 1	Spare				Precedence Level			
Octet 2	1st NI Digit				2nd NI Digit			
Octet 3	3rd NI Digit				4th NI Digit			
Octet 4	Most Significant Bits (MLPP Service Domain) - Least Significant Bits							
Octet 5								
Octet 6								

The Precedence Level field is coded as follows:

Precedence Level	D	C	B	A
Flash Override	0	0	0	0
Flash	0	0	0	1
Immediate	0	0	1	0
Priority	0	0	1	1
Routine	0	1	0	0

Figure 25/T1.114.5 - Precedence Format

Bits HGFE are spare.

ATIS-1000114.2004 (S2019)

Chapter T1.114.5

	H	G	F	E	D	C	B	A
Call Reference Identifier	1	0	0	1	1	1	0	0

The Call Reference length is 6 octets. The contents are coded as follows:

Call Reference Identifier	H	G	F	E	D	C	B	A
Octet 1	Call Identify							
Octet 2								
Octet 3								
Octet 4	Point Code							
Octet 5								
Octet 6								

Figure 26/T1.114.5 - Call Reference Identifier Format

	H	G	F	E	D	C	B	A
Integrity Identifier	1	1	0	1	1	1	1	0

The Integrity Parameter length is variable. The contents are coded as follows:

Integrity Parameter Identifier
Integrity Parameter Length
Integrity AlgID Parameter Identifier
Integrity AlgID Parameter Length
Integrity AlgID Parameter Value
Integrity Value Parameter Identifier
Integrity Value Parameter Length
Integrity Value Parameter Value

Figure 27/T1.114.5 - Integrity Parameter Identifier Format

	H	G	F	E	D	C	B	A
Sequence Number	0	1	0	1	1	1	1	1
	0	0	0	1	1	1	1	1

The Sequence Number parameter length is variable. The contents are coded as follows:

Sequence Number Parameter Identifier
Sequence Number Parameter Length
Sequence Number Parameter Value

Figure 28/T1.114.5 - Sequence Number Identifier Format

	H	G	F	E	D	C	B	A
Number of Messages	0	1	0	1	1	1	1	1
	0	0	1	0	0	0	0	0

The Number of Messages parameter length is variable. The contents are coded as follows:

Number of Messages Identifier
Number of Messages Parameter Length
Number of Messages Parameter Value

Figure 29/T1.114.5 - Number of Messages Format

	H	G	F	E	D	C	B	A
Display Text	0	1	0	1	1	1	1	1
	0	0	1	0	0	0	0	1

The Display Text parameter length is variable. The contents are coded as follows:

H	G	F	E	D	C	B	A
character 1							
character 2							
character 3							

Figure 30/T1.114.5 - Display Text Format

	H	G	F	E	D	C	B	A
KeyExchange	0	1	1	1	1	1	1	1
	0	0	1	0	0	0	1	0

The KeyExchange parameter length is variable. The contents are coded as follows:

KeyExchange Parameter Identifier
KeyExchange Parameter Length
KeyExchange AlgID Parameter Identifier
KeyExchange AlgID Parameter Length
KeyExchange AlgID Parameter Value
KeyExchange Value Parameter Identifier
KeyExchange Value Parameter Length
KeyExchange Value Parameter Value

Figure 31/T1.114.5 - KeyExchange ParameterIdentifier Format

	H	G	F	E	D	C	B	A
SCCP Calling Party Address	0	1	0	1	1	1	1	1
	0	0	1	0	0	0	1	1

Figure 32/T1.114.5 - SCCP Calling Party Address Format

The SCCP Calling Party Address parameter length is variable. The contents of this parameter use the standard SCCP address encoding given in 3.4 of Chapter T1.112.3 of T1.112-2001 and as shown in Figures 3/T1.112.3, 4/T1.112.3, 4a/T1.112.3, 5/T1.112.3, 6/T1.112.3, and 6a/T1.112.3.

Annex A
(informative)

A OPERATION FAMILY EXAMPLES

A.1 Scope

This Informative Annex illustrates how the operations defined in T1.114.5, clause 2, may be used to support the offering of non-ISDN services to customers. The services and associated operations which were selected as examples in the Informative Annex are based on those standardized in the ATIS T1 Committee. Suitable references to relevant ANSI documents will be included when the information becomes available.

A.1.1. Parameter Family Operation (T1.114.5, clause 2.1.1)

The Provide Value Specifier of this operation is used to indicate that the values of the Parameters identified in the Parameter Set are to be provided. In the case of the Multi-Location Business group Automatic Recall/Automatic Callback services, the following mandatory parameters are specified:

- (1) The Busy-Idle status of the called MBG customer.
- (2) The Call Forwarded status of the called MBG customer.
- (3) The DN to LINE Service Type mapping of the called MBG customer.
- (4) The Business Group information of the called MBG customer.
- (5) The Service Key which encompasses the following:
 - ◆ The number of the called MBG customer.
 - ◆ The number of the calling MBG customer
 - ◆ The calling MBG customer Business Group information.

When the operation is performed successfully, a Return Result with the following parameters is returned:

- (1) The Busy-Idle status of the called MBG customer.
- (2) The Call Forwarded status of the called MBG customer.
- (3) The DN to Line Service Type mapping of the called MBG customer.
- (4) The Business Group information pertaining to the called MBG customer.

If the operation cannot be performed, the Return Error cause may be one of the following:

- (1) Unexpected data value.
- (2) Data unavailable.
- (3) Task refused.

A.1.2. Provide Instructions Family Operation (T1.114.5, clause 2.1.3)

The Assist Specifier of this operation is used when a node that currently has control of a call requires the use of a resource (e.g., Announcement System) that it does not have. The call is passed to another node that does have the resource. The second node detects (in a network specific manner) that the call is one on which it is to provide an "Assist", and requests instruction from the database with this operation.

The following mandatory parameters are specified:

- (1) A reference identification to enable correlation of the transaction between the database and Exchange A.

It is an operation where only errors are reported. These may be one of the following:

- (1) Unexpected data value.
- (2) Task refused.

A.1.3. Connection Control Family Operation (T1.114.5, clause 2.1.4)

The Temporary Connect Specifier of this operation is used to indicate that the connection is to be established using the given called address and any other needed information, and that a "Forward Disconnect" request will subsequently be sent.

The following mandatory parameters are specified:

- (1) The routing number of the Requested Exchange (Exchange B).
- (2) The SCCP address of the database.
- (3) A reference identification to enable correlation of the transaction between the database and the Serving Exchange (Exchange A).

It is an operation where only errors are reported. These may be one of the following:

- (1) Unexpected data value.
- (2) Task refused.

No parameters are specified as being required.

No response is expected.

The Forward Disconnect Specifier of this operation informs a node that it may discontinue its "Temporary Connect" to another node.

No parameters are specified as being required.

No response is expected.

A.1.4. Send Notification Family Operation (T1.114.5, clause 2.1.6)

The When Party Free Specifier of this operation indicates that the sender should be informed when party is idle. In the case of the Multi-Location Business Group Automatic Recall/Automatic Callback services, the following mandatory parameters are specified:

- (1) The Service Key which encompasses:
 - ◆ The number of the called MBG customer.
 - ◆ The number of the calling MBG customer.
 - ◆ The duration of time that scanning should be in effect.

When the operation is successfully performed, a Return Result with the following parameter is returned:

- (1) The Busy-Idle status of the called MBG customer.

If the operation cannot be performed, the Return Error cause may be one of the following:

- (1) Unexpected data value.
- (2) Unavailable resource.
- (3) Task refused.

A.1.5. Procedural Family Operation (T1.114.5, clause 2.1.8)

The Report Assist Termination Specifier of this operation indicates the end of an Assist (see A.1.2.).

No parameters are specified as being required.

No response is expected.

A.1.6. Operation Control Family Operation (T1.114.5, clause 2.1.9)

The Cancel Specifier of this operation is sent to order or indicate the cancellation of an operation. (The only operation that may currently be cancelled is the Send Notification [T1.114.5, Section 2.1.6] operation.) It specifies the Service Key parameter which encompasses the following:

- (1) The destination number of the called party.

No response is expected.

A.1.7 Report Event Family Operation (T1.114.5, clause 2.1.10)

The Voice Message Available Specifier of this operation is used to report the availability of a voice message from an exchange at which a Voice Message Storage and Retrieval (VMSR) system is connected to an exchange where the subscriber of a voice message and retrieval service is located.

The following mandatory and optional parameters are specified:

(1) Mandatory Parameters

- ◆ Destination number of the subscriber.
- ◆ Identification number of the VMSR system reporting availability of the message(s).

(2) Optional Parameters

- ◆ Calling Number of the party that left a message with the VMSR system.
- ◆ The time the message was left at the VMSR system.
- ◆ The number of messages waiting at the VMSR system for the client user.
- ◆ VMSR system-provided text providing information about the waiting messages, e.g., "urgent" messages.

This operation reports success via a Return Result (Last) with no parameters.

If the operation cannot be performed, the Return Error cause may be one of the following:

- (1) Task refused.
- (2) Unassigned DN.
- (3) Notification unavailable to destination.
- (4) VMSR system ID does not match subscriber profile.

The Voice Message Retrieved Specifier of this operation is used to inform an exchange to remove the message available indicator for a VMSR subscriber.

The following mandatory parameters are specified:

- (1) Subscriber number for which the message available indication is to be removed.
- (2) The identification of the VMSR system requesting the removal of the message.

When the operation is performed successfully, a Return Result with an empty parameter list is returned.

If the operation cannot be performed, the Return Error cause may be one of the following:

- (1) Task refused.
- (2) Unassigned DN.
- (3) Notification unavailable to destination.
- (4) VMSR system ID does not match the subscriber profile.

A.1.8. Miscellaneous Family Operation (T1.114.5, clause 2.1.11)

The Queue Call Specifier of this operation requests that a place be reserved in a queue of calls. In the case of the Multi-Location Business Group Automatic Recall/Automatic Callback services, it requests that a place be reserved in a queue of calls to a Directory Number.

It specifies the Service Key parameter which encompasses the following:

- (1) The number of the called customer.
- (2) The number of the calling customer.

When the operation is performed successfully a Return Result with no parameters is returned.

If the operation cannot be performed, the Return Error cause may be one of the following:

- (1) Unexpected data value.
- (2) Task refused.
- (3) Queue full.
- (4) No queue.
- (5) Not Queued at node's discretion.

The Dequeue Call Specifier of this operation requests that a call be removed from a queue of calls. In the case of the Multi-Location Business Group Automatic Recall/Automatic Callback services, this operation requests that the call be removed from the queue of calls to a Directory Number.

It specifies the Service Key parameter which encompasses the following:

- (1) The number of the called customer.
- (2) The number of the calling customer.

It is an operation where no response is expected.